

WebMedia 2026

32º simpósio brasileiro de sistemas multimídia e web

MINICURSOS DO 32º SIMPÓSIO BRASILEIRO DE SISTEMAS MULTIMÍDIA E WEB

27 A 30 DE OUTUBRO DE 2026
LAVRAS, MG, BRASIL

REALIZAÇÃO

SBC Sociedade Brasileira de Computação

ORGANIZAÇÃO

UFLA Universidade Federal de Lavras

APOIO

OGEDAI Laboratório de Geoprocessamento, Ciência de Dados e Inteligência Artificial

PATROCÍNIO

CNPq Conselho Nacional de Desenvolvimento Científico e Tecnológico

CAPES

FAPEMIG

nic.br cgi.br



XXXII Simpósio Brasileiro de Sistemas Multimídia e Web

De 27 a 30 de Outubro de 2026

Lavras, Brasil

LIVRO DE MINICURSOS

Organizadores

Vinicius Vitor dos Santos Dias (UFLA)
Carlos Henrique Gomes Ferreira (UFOP)
Roberto Willrich (UFSC)
Eduardo Barrére (UFJF)
Sandra Souza Rodrigues (UFLA)
André de Lima Salgado (UFLA)
André Pimenta Freire (UFLA)
Marluce Rodrigues Pereira (UFLA)

Realização

Sociedade Brasileira de Computação – SBC

Organização

Universidade Federal de Lavras – UFLA

Esta obra está sob a licença Creative Commons Atribuição 4.0 (CC-BY). Você pode redistribuir este livro em qualquer suporte ou formato e copiar, remixar, transformar e criar a partir do conteúdo deste livro para qualquer fim, desde que cite a fonte.

Dados Internacionais de Catalogação na Publicação (CIP)

S612 Simpósio Brasileiro de Sistemas Multimídia e Web (32. : 27 – 30 outubro 2026 : Lavras)

Minicursos do WebMedia 2026 [recurso eletrônico] / org: Vinicius Vitor dos Santos Dias e Carlos Henrique Gomes Ferreira – Dados eletrônicos. – Porto Alegre: Sociedade Brasileira de Computação, 2026.

187 p. : il. : PDF ; 30 MB

Modo de acesso: World Wide Web.
ISBN 978-85-7669-685-8 (e-book)

1. Computação – Brasil – Evento. 2. Sistemas de informação. 3. Inteligência artificial. I. Dias, Vinicius Vitor dos Santos. II. Ferreira, Carlos Henrique Gomes. III. Sociedade Brasileira de Computação. IV. Título.

CDU 004(063)

Ficha catalográfica elaborada por Annie Casali – CRB-10/2339

Biblioteca Digital da SBC – SBC OpenLib

Sociedade Brasileira de Computação
Av. Bento Gonçalves, 9500
Setor 4 | Prédio 43.412 | Sala 219 | Bairro
Agronomia Caixa Postal 15012 | CEP 91501-970
Porto Alegre - RS
(51) 99252-6018
sbc@sbc.org.br

Prefácio

Em sua 32ª edição, realizada na cidade de Lavras (MG), entre 27 e 30 de outubro de 2026, o evento contou com 3 tutoriais e 5 minicursos, escolhidos em um processo de seleção das propostas enviadas em resposta a uma chamada pública de alcance nacional.

Assim como nas edições anteriores do WebMedia, os ministrantes dos minicursos foram convidados a produzir um texto correspondente ao conteúdo apresentado, para publicação na forma de capítulo de livro. Destaca-se que o evento contou com 3 tutoriais e 5 minicursos, sendo que este volume reúne, a posteriori, o material referente aos cinco minicursos.

O Capítulo 1, intitulado *Avaliando Artefatos Digitais: Acessibilidade, Legibilidade e Comunicação Visual*, aborda a análise de elementos e propriedades recorrentes em diferentes artefatos digitais sob a ótica conjugada de acessibilidade, legibilidade e comunicação visual. Sob abordagem eminentemente prática, baseada na observação, comparação e reavaliação de diferentes exemplos, o capítulo discute como princípios e evidências podem ser utilizados para fundamentar decisões de análise e de projeto.

O Capítulo 2, intitulado *Computação Física Inclusiva com micro:bit*, apresenta fundamentos e práticas de computação física inclusiva, delimitando o uso do micro:bit e da plataforma MakeCode para o desenvolvimento de soluções no ecossistema de Tecnologia Assistiva. Aliando conceitos de IHC voltada à acessibilidade e o Design Universal para a Aprendizagem, o capítulo guia a construção de protótipos com feedbacks multimodais (visuais, sonoros e táteis) e acionadores físicos adaptados.

O Capítulo 3, intitulado *Gêmeos Digitais: Uma Ponte entre os Mundos Físico e Virtual*, fornece uma visão abrangente sobre gêmeos digitais, representações virtuais dinâmicas de entidades, processos e sistemas do mundo real, cobrindo conceitos fundamentais, paradigmas habilitadores, padrões e plataformas. O capítulo discute o papel das tecnologias multimídia e Web, explora domínios de aplicação e desafios para a construção de ecossistemas interoperáveis, confiáveis e seguros.

O Capítulo 4, intitulado *Personas e Acessibilidade Web: Desenvolvendo para Diferentes Usuários*, apresenta fundamentos teóricos e práticos de acessibilidade web, incluindo as diretrizes WCAG, HTML semântico, atributos ARIA e tecnologias assistivas. O capítulo promove uma visão crítica: a conformidade técnica, por si só, não garante uma experiência verdadeiramente acessível e propõe a criação de personas para fomentar empatia e uma perspectiva mais inclusiva no desenvolvimento.

O Capítulo 5, intitulado *Wikidata in the Era of Generative AI: Interfaces, Techniques, and Applications*, introduz o Wikidata, a versão em grafo de conhecimento da Wikipédia, com ênfase em interfaces e técnicas para construir aplicações de IA generativa. O capítulo apresenta iniciativas recentes como o Wikidata Embedding Project e o Wikidata Model Context Protocol (MCP), ilustrando conceitos como vinculação de entidades (grounding) e RAG por meio de um agente de checagem de fatos e um sistema de perguntas e respostas.

Agradecemos aos organizadores do evento pela oportunidade e aos revisores pelo trabalho na avaliação e na seleção das propostas aceitas. Temos certeza de que este livro será útil para todas as pessoas interessadas nos temas abordados.

Lavras, outubro de 2026.

Vinicius Vitor dos Santos Dias (UFLA)
Carlos Henrique Gomes Ferreira (UFOP)
Coordenadores de Minicursos e Tutorais

XXXII Simpósio Brasileiro de Sistemas Multimídia e Web

27 a 30 de Outubro de 2026
Lavras, Brasil

Coordenação Geral

André de Lima Salgado (UFLA) – *Coordenador Geral*
André Pimenta Freire (UFLA) – *Coordenador Geral*
Marluce Rodrigues Pereira (UFLA) – *Coordenador Geral*
Leonardo Chaves Dutra da Rocha (UFSJ) – *Coordenador do Comitê de Programa*
Kamila Rios da Hora Rodrigues (ICMC/USP) – *Coordenadora do Comitê de Programa*
Rudinei Goularte (ICMC/USP) – *Coordenador do Comitê de Programa*

Coordenação de Minicursos e Tutoriais

Carlos Henrique Gomes Ferreira (UFOP)
Vinicius Vitor dos Santos Dias (UFLA)

Coordenação de Palestras e Painéis

Gustavo do Vale (UFLA)
Thiago Henrique Silva (UTFPR)

Coordenadores do VIII Concurso de Teses e Dissertações (CTD)

Celso Alberto Saibel Santos (UFES)
Heitor Augustus Xavier Costa (UFLA)
Windson Viana de Carvalho (UFC)

Coordenadores do VI Concurso de Trabalhos de Iniciação Científica (CTIC)

Fabrício Aguiar Silva (UFV)
Frederico Araujo Durao (UFBA)

Coordenadores do XXV Workshop de Ferramentas e Aplicações (WFA)

Carlos André G. Ferraz (UFPE)
Luciano Volcan Agostini (UFPEL)
Luiz Henrique Andrade Correia (UFLA)

Coordenação do V WebMedia for Everyone

Alessandreia Marta de Oliveira (UFJF)
Kamila Rios da Hora Rodrigues (ICMC/USP)

Workshop Brasileiro de Sistemas Web3 (BrWeb3)

Antonio José Grandson Busson (BTG Pactual)
Julio Cesar Duarte (IME)
Sergio Colcher (PUC-Rio)

Coordenação do Workshop Futuro da TV Digital Interativa

Alessandra Alaniz Macedo (USP)
Darlinton Barbosa Feres Carvalho (UFSJ)

Coordenadores de Publicação

Roberto Willrich (UFSC)
Eduardo Barrére (UFJF)
Sandra Souza Rodrigues (UFLA)

Coordenação de Patrocínios e Contatos Institucionais

Adriano César Machado Pereira (UFMG)
Luiz Henrique de Campos Merschmann (UFLA)
Maurício Ronny de Almeida Souza (UFLA)

Coordenação do Prêmio LF

César Augusto Camillo Teixeira (UFSCAR)
Paulo Henrique Lopes Rettore (UFLA)

Coordenação da Comissão Especial de Sistemas Multimídia e Web

Carlos André G. Ferraz (UFPE) – *Coordenador*
Débora C. Muchaluat Saade (UFF) – *Vice-Coordenadora*

Comitê Gestor

Alessandra Alaniz Macedo (USP)
André de Lima Salgado (UFLA)
André Pimenta Freire (UFLA)
Carlos de Salles Soares Neto (UFMA)
Celso Alberto Saibel Santos (UFES)
Cesar Augusto Camillo Teixeira (UFSCar)
Joel André Ferreira dos Santos (Cefet/RJ)
Jonice Oliveira (UFRJ)
José Valdeni De Lima (UFRGS)
Kamila Rios Da Hora Rodrigues (USP)
Marluce Rodrigues Pereira (UFLA)
Paulo Victor Borges Oliveira Lima (PUC-Rio)
Sérgio Colcher (PUC-Rio)
Windson Viana De Carvalho (UFC)

Sociedade Brasileira de Computação (SBC)

Presidência

Thais Vasconcelos Batista (UFRN) – *Presidente*
Cristiano Maciel (UFMT) – *Vice-Presidente*

Diretoria

Alírio Santos Sá (UFBA)
André Luís de Medeiros Santos (UFPE)
Carlos Eduardo Ferreira (USP)
Denis Lima do Rosário (UFPA)
Eunice Pereira dos Santos Nunes (UFMT)
Flávia Maria Santoro (INTELLI)
Francisco Dantas Medeiros Neto (UERN)
José Viterbo Filho (UFF)
Leila Ribeiro (UFRGS)
Marcelo Antonio Marotta (UNB)
Michelle Silva Wangham (UNIVALI)
Renata de Matos Galante (UFGRS)
Rodrigo Silva Duran (IFB)
Ronaldo Alves Ferreira (UFMS)

Contato

Av. Bento Gonçalves, 9500
Setor 4 - Prédio 43.412 - Sala 219
Bairro Agronomia
91.509-900 – Porto Alegre RS
CNPJ: 29.532.264/0001-78
`http://www.sbc.org.br`



Sumário

Capítulo 1. Avaliando Artefatos Digitais: Acessibilidade, Legibilidade e Comunicação Visual 1

Maria da Graça Campos Pimentel (USP), Kamila Rios da Hora Rodrigues (USP)

Capítulo 2. Computação Física Inclusiva com Micro:bit 41

Mariana Arfux Pereira Cavalcante de Castro (USP), Bruna C.R. Cunha (USP)

Capítulo 3. Gêmeos Digitais: Uma Ponte entre os Mundos Físico e Virtual 64

Everton Cavalcante (UFRN), Thais Batista (UFRN), Flavia C. Delicato (UFF), André Almeida (IFRN)

Capítulo 4. Personas e Acessibilidade Web: Desenvolvendo para Diferentes Usuários 102

Matheus A.S. Santos (USP), Beatriz L. Silva (USP), Marina F. Souza (USP), Bruna C. R. Cunha (USP)

Capítulo 5. Wikidata in the Era of Generative AI: Interfaces, Techniques, and Applications 133

Marcelo Machado (UFJF), João Pedro Porto Campos (PUC-Rio), Alexya Silva (UFJF), Gabriel Banaggia (PUC-Rio), Guilherme Lima (PUC-Rio)

Capítulo 1

Avaliando Artefatos Digitais: Acessibilidade, Legibilidade e Comunicação Visual

Maria da Graça Campos Pimentel e Kamila Rios da Hora Rodrigues

Resumo

Artefatos digitais comunicam por meio de seu conteúdo e das escolhas de apresentação, organização e representação da informação. Este capítulo apresenta fundamentos para avaliar essa comunicação, articulando princípios de design e organização perceptiva, legibilidade e compreensão, hierarquia visual e acessibilidade. Esses princípios são relacionados a problemas observáveis em exemplos didáticos, fornecendo um repertório para identificar e justificar problemas, analisar suas consequências e orientar a avaliação e o reprojeto. Ao final, os diferentes aspectos são reunidos em uma avaliação integrada e aplicados ao reprojeto progressivo de um artefato digital.

Abstract

Digital artifacts communicate through their content and choices concerning information presentation, organization, and representation. This chapter presents foundations for evaluating this communication, bringing together principles of design and perceptual organization, legibility and comprehension, visual hierarchy, and accessibility. These principles are related to observable problems in educational examples, providing a repertoire for identifying and justifying problems, analyzing their consequences, and guiding evaluation and redesign. Finally, these aspects are brought together in an integrated evaluation and applied to the progressive redesign of a digital artifact.

1.1 Introdução

A comunicação pode ser descrita como um processo de geração, transmissão e recepção de informação, no qual o sinal pode ser degradado ou sofrer interferência de ruído, segundo o princípio *Relação Sinal-Ruído* (Lidwell et al. 2023). Em artefatos digitais, a comunicação depende da informação que se pretende transmitir e das escolhas realizadas para representá-la, organizá-la e apresentá-la. Em *cards*, *slides*, páginas Web, interfaces interativas, tabelas, visualizações de dados e figuras incorporadas a textos, decisões sobre

tipografia, cor, contraste, agrupamento, alinhamento, imagens e distribuição espacial podem favorecer ou interferir nessa comunicação.

Este capítulo se apoia em um conjunto pequeno e consolidado de referências sobre design, comunicação visual, usabilidade e acessibilidade. O objetivo é reunir fundamentos para identificar e justificar problemas observáveis em artefatos digitais, em lugar de estabelecer um conjunto exaustivo de regras de design. A partir dessa literatura, são discutidos princípios relevantes para examinar diferentes aspectos da apresentação da informação e fundamentar sua avaliação. Os nomes desses princípios são apresentados em itálico e, quando provenientes de Lidwell et al. (2023), são também identificados pela sigla LBW seguida do número atribuído ao princípio na obra, por exemplo, [LBW170].

A ampla disponibilidade de ferramentas de Inteligência Artificial Generativa modificou a produção de artefatos visuais. Imagens, *cards*, apresentações e outras composições podem ser produzidos rapidamente, inclusive por pessoas sem formação em design. Essa facilidade não garante comunicação adequada: os artefatos podem apresentar excesso de elementos, hierarquia pouco clara, problemas de legibilidade e contraste, dependência de características exclusivamente visuais ou barreiras de acessibilidade. A redução do esforço de produção torna mais relevante dispor de critérios de avaliação. A aparência oferece uma base insuficiente para essa avaliação. O princípio *Efeito Estética-Usabilidade* descreve a tendência de designs esteticamente agradáveis serem percebidos como mais fáceis de usar, ainda que não o sejam [LBW 004].

O princípio *Hierarquia de Necessidades* organiza as necessidades de um projeto em cinco níveis: Funcionalidade, Confiabilidade, Usabilidade, Proficiência e Criatividade [LBW 090]. Nos artefatos tratados neste capítulo, a função fundamental é comunicar a informação necessária; qualidades de níveis superiores, como Criatividade, não substituem necessidades mais fundamentais. O princípio *Efeito IKEA* descreve a atribuição de maior valor a objetos em cuja criação houve investimento de esforço [LBW 096]. Esses princípios evidenciam limitações de avaliações baseadas na aparência do artefato ou na apreciação de quem participou de sua produção.

A conformidade com requisitos de acessibilidade constitui outra dimensão necessária, mas não suficiente, para avaliar a comunicação de um artefato digital. As Diretrizes de Acessibilidade para Conteúdo Web (WCAG) organizam seus requisitos segundo quatro princípios: perceptível (*Perceivable*, P), operável (*Operable*, O), compreensível (*Understandable*, U) e robusto (*Robust*, R), referidos pelo acrônimo POUR [W3C 2024, Seção 3.2].

O foco deste capítulo está na avaliação de *cards*, *slides*, páginas Web, interfaces interativas, tabelas, pequenas visualizações de dados e figuras utilizadas em textos. A questão que orienta a discussão é:

> O que, em um artefato digital, favorece ou prejudica a comunicação da informação e com que fundamentos essa avaliação pode ser justificada?

Os princípios discutidos constituem um repertório para apoiar essa análise, cuja pertinência depende do conteúdo, do público, do contexto, da tarefa e, quando aplicável, das formas de interação envolvidas. Busca-se, assim, fundamentar a identificação de



Figura 1.1: *Card* fictício utilizado ao longo do capítulo, que reúne fotografia, identificação do evento, título e palestrante, data e local, condições de participação, informações para inscrição e elementos institucionais, distribuídos em diferentes grupos, alinhamentos, níveis de proeminência e espaços em branco.

problemas, a discussão de suas consequências e as decisões de avaliação e reprojeto, em vez de estabelecer uma receita para produzir um "bom design".

A Figura 1.1 apresenta um *card* fictício utilizado ao longo do capítulo. Em uma área compacta, ele reúne decisões sobre conteúdo, tipografia, imagens, contraste, alinhamento, agrupamento, hierarquia e distribuição espacial. Seu uso como exemplo recorrente tem por objetivo explicitar problemas, suas possíveis consequências para a comunicação e os fundamentos que justificam sua avaliação.

As Figuras 1.2 e 1.3 apresentam, respectivamente, um exemplo de alternativa textual e as propriedades tipográficas consideradas nas análises deste capítulo.

Todas as figuras são autorais e elaboradas para fins didáticos, com exemplos artificiais e controlados para variar aspectos específicos, mantendo os demais constantes quando possível. Assim, integram a análise dos princípios discutidos, além de ilustrá-los. Todas incluem alternativas textuais que descrevem o conteúdo e, quando relevantes, características da composição, como posição, proporção, alinhamento, agrupamento, distribuição espacial, espaço em branco e tratamento tipográfico [W3C 2026]. A Figura 1.2 reproduz a alternativa textual da Figura 1.1, incluindo sua organização espacial

Card fictício de divulgação de uma palestra. Uma fotografia ocupa cerca de um quarto da altura do card, na região superior, mostrando uma pequena piscina natural à esquerda, junto a formações rochosas ao fundo, com vegetação e uma árvore sobre as rochas e céu azul com nuvens; a piscina natural se abre, à direita, para o mar azul-claro, que ocupa o restante da imagem à direita e se estende até o horizonte. Sobre a fotografia, na metade direita, aparece em destaque o logotipo E, B, C, D 2026, ocupando cerca de metade da largura e mais da metade da altura da foto. Todos os textos utilizam fonte sem serifa, com variações de tamanho, peso e uso de maiúsculas. Logo abaixo da fotografia aparecem Encontro Brasileiro de Comunicação Digital 2026 e, mais à direita, PALESTRA, ambos em letras maiúsculas e em tamanho pequeno. A região central é ocupada pelo título Design de Informação para Artefatos Digitais, em tamanho maior e negrito, seguido por duas linhas em tamanho menor contendo Maria Silva e Universidade Brasileira de Tecnologia. Após um pequeno espaço em branco, aparecem três linhas em tamanho menor: 18 de novembro de 2026, 14h; Centro de Convenções Horizonte; e Itatingaba, S,P. Após outro espaço em branco, aparecem, espaçados, os textos ENTRADA GRATUITA e VAGAS LIMITADAS, em letras maiúsculas e negrito. Após espaço em branco, aparecem Inscrições e, abaixo, o endereço eletrônico www.e b c d 2026.org.br/palestras, com um Q R Code à direita. Após outro espaço em branco, uma linha horizontal separa o rodapé, no qual aparecem, em tamanho menor, à esquerda, Realização, E, B, C, D, e, U, B, T; e à direita Patrocínio, representado por dois espaços reservados para logotipos.

Figura 1.2: Alternativa textual completa fornecida para o *card* da Figura 1.1.

Família tipográfica	com serifa	Design Digital
	sem serifa	Design Digital
Tamanho	10 pt	Design Digital
	14 pt	Design Digital
Peso	regular	Design Digital
	negrito	**Design Digital**
Estilo	romano	Design Digital
	itálico	*Design Digital*
Caixa	caixa baixa	design digital
	iniciais maiúsculas	Design Digital
	caixa alta	DESIGN DIGITAL
Entrelinha	12 pt	Design Digital Comunicação Visual
	18 pt	Design Digital Comunicação Visual

Figura 1.3: Propriedades tipográficas aplicadas ao mesmo conteúdo, com 10 pontos como tamanho-base exceto na comparação de tamanho: família com e sem serifa; tamanhos de 10 e 14 pontos; peso regular e negrito; estilo romano e itálico; caixa baixa, iniciais maiúsculas e caixa alta; entrelinhas de 12 e 18 pontos.

Tabela 1.1: Organização do capítulo.

Seção	**Conteúdo**
1.1	Introdução
1.2	Comunicação e apresentação da informação
1.2.1	Sinal, ruído e comunicação
1.2.2	Simplificação e redução
1.2.3	Interferência e elementos sem função informacional
1.3	Percepção e organização visual
1.3.1	Organização perceptiva
1.3.2	Agrupamento
1.3.3	Percepção de formas e relações
1.3.4	O processo de composição
1.4	Legibilidade, leitura e compreensão
1.4.1	Legibilidade
1.4.2	Leitura e compreensão
1.4.3	Atenção, destaque e hierarquia
1.5	Acessibilidade como requisito da comunicação
1.5.1	Cor, contraste e representação
1.5.2	Informação visual e representação acessível
1.5.3	Conteúdo visual em mídias sociais
1.5.4	Siglas, abreviações e sua representação
1.5.5	Quando princípios entram em tensão
1.6	Da análise de aspectos isolados à avaliação integrada
1.7	Da avaliação ao reprojeto
1.8	Verificação da acessibilidade deste capítulo
1.9	Considerações finais
1.10	Cuidados éticos
1.11	Agradecimentos
Apêndice A	Princípios para avaliação de artefatos digitais

e seu tratamento tipográfico. Entre as propriedades tipográficas consideradas estão família, tamanho, peso, estilo, uso de caixa e entrelinha. *Caixa alta* corresponde a letras maiúsculas e *caixa baixa*, a minúsculas; entrelinha corresponde à distância entre as linhas de base de linhas sucessivas. A Figura 1.3 exemplifica essas propriedades, observadas ao longo do capítulo.

A acessibilidade também orientou a produção deste capítulo. Algumas decisões podem ser observadas diretamente durante a leitura, como a ausência de hifenização e o uso das formas por extenso em lugar de abreviaturas. Outras, como a estrutura semântica do documento e as alternativas textuais das figuras, destinam-se à interpretação por tecnologias assistivas. A Seção 1.8 descreve essas decisões e a verificação realizada.

A Tabela 1.1 apresenta a organização do capítulo, desde os fundamentos da comunicação, da percepção e da organização visual até a acessibilidade, a avaliação integrada, o reprojeto e as seções finais. O Apêndice A sintetiza os princípios utilizados.

1.2 Comunicação e apresentação da informação

Esta seção discute sinal, ruído, simplificação, redundância e interferência na comunicação da informação.

1.2.1 Sinal, ruído e comunicação

Uma primeira forma de analisar um artefato digital é perguntar quais elementos contribuem para comunicar a informação relevante e quais competem com ela. No *card* da Figura 1.1, texto, tipografia, imagens, cores, destaques e distribuição espacial podem favorecer ou interferir nessa comunicação.

O princípio *Relação Sinal-Ruído* fornece fundamento para essa análise [LBW 170]. Escrita pouco clara, representações gráficas inadequadas e ícones ou rótulos ambíguos podem degradar o sinal; elementos sem função informacional podem competir com a informação relevante.

Melhorar a relação entre sinal e ruído, entretanto, não significa remover elementos. Como ilustra a Figura 1.4, a retirada do que interfere na comunicação pode tornar a informação mais evidente e modificar sua distribuição espacial, enquanto a remoção excessiva pode eliminar informação necessária. Nas três versões, os textos permanecem centralizados, mas a redução do conteúdo aumenta o espaço em branco, sobretudo na dimensão vertical. A análise deve considerar *quanto* há no artefato, *o que o elemento faz para a comunicação* e como sua presença ou ausência afeta a composição.



Figura 1.4: Efeitos da remoção de elementos sobre o conteúdo e sua distribuição espacial: (a) seis linhas ocupam quase toda a altura; (b) a remoção de dois elementos aumenta o espaço em branco acima e abaixo; (c) a remoção também do local concentra o conteúdo na metade superior e elimina informação necessária.

1.2.2 Simplificação e redução

Reduzir o ruído não significa tornar um artefato tão simples quanto possível. O princípio *Navalha de Ockham* estabelece a preferência pela solução mais simples entre alternativas funcionalmente equivalentes [LBW 128]. A condição é essencial: uma alternativa mais simples deixa de ser preferível se comprometer a função do artefato. Para o conteúdo textual, o princípio *Elimine Palavras Desnecessárias* recomenda eliminar palavras que não contribuem para a comunicação, sem eliminar informação necessária [Krug 2014, 48–53].

O princípio *Subtração Progressiva* trata a simplificação como um processo iterativo de remoção de elementos [LBW 148]. A cada remoção, deve-se verificar se o artefato continua cumprindo sua função e interromper o processo quando uma



Figura 1.5: Subtração progressiva da precisão na representação de um horário em quatro retângulos de mesmas dimensões. O horário mantém fonte sem serifa, tamanho, peso e centralização, enquanto sua redução progressiva aumenta o espaço em branco lateral: (a) horas, minutos, segundos e centésimos de segundo; (b) remoção dos centésimos; (c) remoção dos segundos; (d) remoção também dos minutos. A simplificação deve ser interrompida quando a precisão restante deixa de atender à finalidade da informação.

nova retirada passar a prejudicá-la. A Figura 1.5 ilustra esse processo pela redução sucessiva da precisão de um horário. As quatro representações mantêm as mesmas dimensões e o horário centralizado, enquanto são removidos, sucessivamente, centésimos de segundo, segundos e minutos; com isso, o conteúdo ocupa progressivamente menos espaço horizontal. O ponto em que a simplificação deve ser interrompida depende da precisão necessária à finalidade da informação.

Nem toda informação, entretanto, precisa estar disponível simultaneamente. O princípio *Divulgação Progressiva* propõe apresentar inicialmente as informações e os controles necessários e tornar os demais disponíveis quando solicitados ou necessários [LBW 147]. Essa estratégia permite administrar a complexidade sem eliminar conteúdo.

O princípio *Horror ao Vazio* descreve a tendência de preencher o espaço disponível com elementos ou objetos [LBW 092]. Em artefatos informacionais, entretanto, espaço não ocupado não é necessariamente espaço desperdiçado. O espaço em branco participa da composição: pode separar grupos, evidenciar relações entre elementos e reduzir a competição visual. Preencher uma área apenas porque ela está disponível pode aumentar a densidade visual sem acrescentar informação relevante. Por essa razão, nas figuras, legendas e alternativas textuais deste capítulo, o espaço em branco, quando relevante para a discussão apresentada, é tratado explicitamente como um aspecto a ser observado na avaliação da composição. Da mesma forma, buscamos tornar observáveis características como alinhamento, agrupamento, distribuição espacial e tratamento tipográfico, incluindo família, tamanho, peso, estilo, uso de caixa e entrelinha.

Simplificação também não implica eliminar redundância. O princípio *Redundância* mostra que múltiplas formas de transmitir a mesma informação podem aumentar a confiabilidade da comunicação [LBW 156]. A Figura 1.6 mantém constantes a disposição e as cores de três elementos, alterando a forma de dois deles. Na primeira representação, três categorias são codificadas por círculos e distinguidas apenas pelas cores vermelha, azul e verde. Na segunda, as cores são preservadas, mas as categorias passam a ser diferenciadas também pela forma, utilizando círculo, quadrado e triângulo. A informação deixa, assim, de depender exclusivamente da percepção cromática. Essa segunda codificação também atende ao critério de sucesso 1.4.1 da WCAG2.2, segundo o qual a cor não deve ser o único meio visual de transmitir informação [W3C 2024, SC 1.4.1]. Redução e redundância são compatíveis: a redução elimina elementos

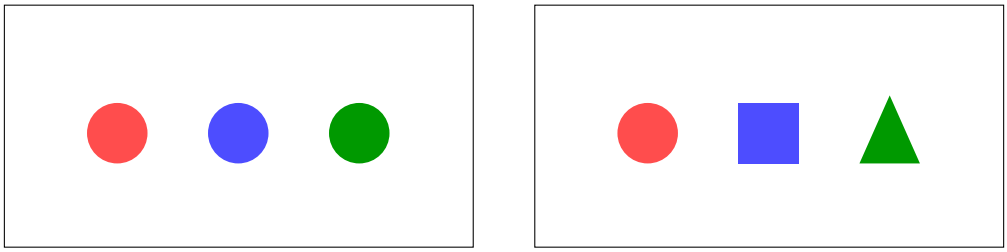


Figura 1.6: Codificação de três categorias abstratas em dois retângulos de mesmas dimensões. Em ambos, três formas de tamanho semelhante ocupam a região central, distribuídas horizontalmente e com espaço em branco, mantendo posições e cores: (a) círculos vermelho, azul e verde distinguem as categorias apenas pela cor; (b) círculo vermelho, quadrado azul e triângulo verde combinam cor e forma, criando redundância.

desnecessários, enquanto a redundância pode preservar formas complementares de perceber ou compreender a informação.

1.2.3 Interferência e elementos sem função informacional

Elementos que interferem na comunicação podem exercer diferentes funções no artefato. O princípio *Efeitos de Interferência* descreve situações em que processos mentais concorrentes tornam mais lento ou menos preciso o processamento da informação [LBW 098]. Isso pode ocorrer quando características de uma representação apresentam informações incongruentes ou quando elementos concorrentes disputam a atenção necessária à realização de uma tarefa.

Embora relacionado a *Relação Sinal-Ruído*, esse princípio enfatiza outro aspecto. Enquanto este considera a relação entre a informação relevante e aquilo que interfere em sua comunicação, *Efeitos de Interferência* trata dos efeitos da concorrência entre processos ou informações durante sua percepção e interpretação. Um elemento pode, portanto, transmitir informação e ainda produzir interferência.

A função informacional das imagens constitui um caso particular dessa análise. O papel de uma imagem na comunicação depende da informação que ela acrescenta e de como sua presença afeta os demais elementos do artefato. Elementos predominantemente decorativos podem ocupar espaço sem acrescentar informação relevante, enquanto imagens relacionadas ao conteúdo podem desempenhar diferentes funções comunicacionais.

A Figura 1.7 compara três usos da região superior de um mesmo *card*. Na primeira representação, estrelas desempenham função predominantemente decorativa. Na segunda, o desenho de um tsuru ilustra o tema da oficina. Na terceira, uma sequência simplificada da dobradura acrescenta informação sobre a atividade a ser realizada.

1.3 Percepção e organização visual

A organização visual de um artefato influencia as relações percebidas entre seus elementos e, consequentemente, a interpretação da informação. Esta seção parte



Figura 1.7: Três versões do mesmo *card*, comparando funções da imagem: (a) estrelas desempenham função decorativa; (b) um desenho de tsuru ilustra o tema da oficina; (c) uma sequência simplificada da dobradura acrescenta informação sobre a atividade.

de princípios de organização perceptiva associados à Gestalt, examinando como configurações visuais são percebidas como totalidades organizadas. Em seguida, aborda princípios de agrupamento e relações envolvidas na percepção de formas e estruturas visuais. Por fim, passa da percepção à composição, discutindo princípios e referências que podem orientar deliberadamente a organização dos elementos de um artefato.

1.3.1 Organização perceptiva

O termo alemão *Gestalt* refere-se a uma estrutura ou experiência que, considerada como um todo, apresenta qualidades que vão além da soma de suas partes [CUP 2026]. Na psicologia da Gestalt, as configurações são consideradas como totalidades cuja organização resulta das relações entre seus elementos [Ellis 1938, 2]. Essa perspectiva é particularmente relevante para artefatos digitais: elementos visuais são percebidos também pelas relações que estabelecem entre si e pela configuração que formam.

Um princípio associado a essa organização perceptiva é o de *Prägnanz*, ou *Boa Forma*. Quando uma configuração admite diferentes organizações perceptivas, há uma tendência à percepção de formas simples, regulares e estáveis [Ellis 1938, 148, 352]. Nesse contexto, *good* caracteriza propriedades de sua organização perceptiva, em vez de indicar que a configuração seja bela, correta ou adequada como solução de design. A Figura 1.8 contrasta um quadrado, como configuração simples e regular, com uma representação tridimensional formada por dois quadrados deslocados e ligados por seus vértices, que admite diferentes orientações espaciais, ilustrando como a percepção resulta também das relações entre os elementos da configuração.

1.3.2 Agrupamento

A percepção de agrupamentos permite que elementos distintos sejam interpretados como partes de uma mesma unidade ou estrutura. Entre os princípios da organização perceptiva da Gestalt que contribuem para esse agrupamento estão *Proximidade* (Figura 1.9), *Similaridade* (Figura 1.10), *Região Comum* e *Conectividade Uniforme* (Figura 1.11) e

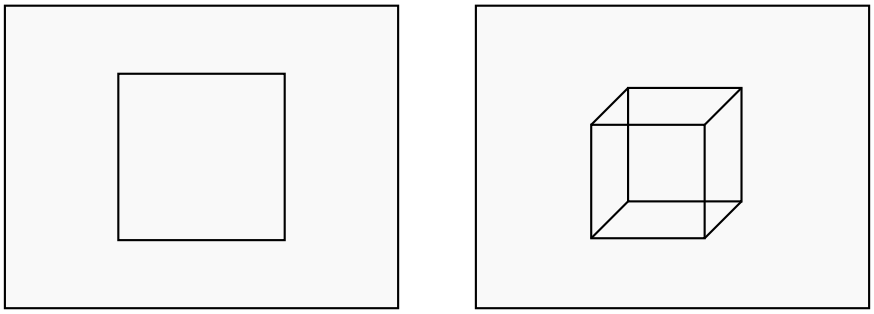


Figura 1.8: Duas configurações: (a) um quadrado, como configuração simples e regular; (b) um cubo, formado por dois quadrados deslocados com vértices correspondentes ligados, cuja orientação espacial pode ser percebida de diferentes maneiras.

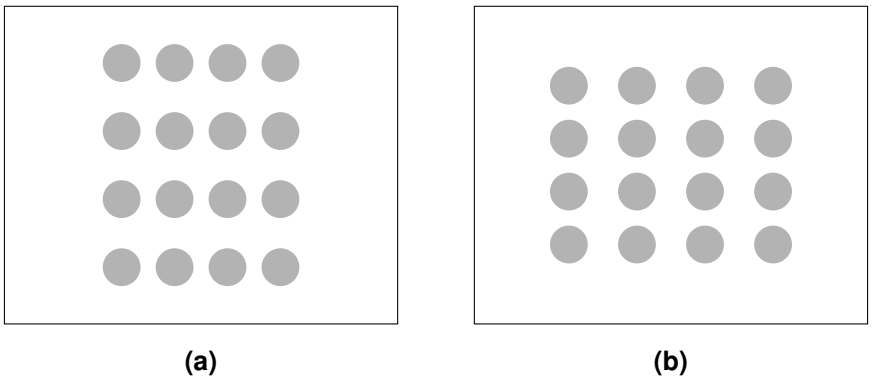


Figura 1.9: Efeito da proximidade sobre o agrupamento perceptivo de 16 círculos de mesmo tamanho e tonalidade, distribuídos em quatro linhas e quatro colunas: (a) menor espaçamento horizontal favorece o agrupamento em linhas; (b) menor espaçamento vertical favorece o agrupamento em colunas.

Destino Comum (Figura 1.12), que descrevem diferentes relações entre os elementos e podem atuar simultaneamente ou sugerir organizações concorrentes.

O princípio *Proximidade* descreve a tendência de elementos próximos entre si serem percebidos como pertencentes a um mesmo grupo [LBW 152]. Na Figura 1.9, a alteração apenas do espaçamento modifica o agrupamento percebido: menor distância horizontal favorece a percepção de linhas, enquanto menor distância vertical favorece a percepção de colunas.

O princípio *Similaridade* descreve o agrupamento de elementos que compartilham características visuais [LBW 171]. Cor, forma, tamanho, preenchimento e outras características podem estabelecer relações entre elementos mesmo quando sua posição permanece inalterada. Na Figura 1.10, posições e espaçamentos são mantidos: na primeira representação, varia o preenchimento dos círculos; na segunda, varia a forma dos elementos. Em ambos, a característica compartilhada favorece o agrupamento em colunas. Embora a cor também possa produzir esse efeito, utiliza-se uma representação

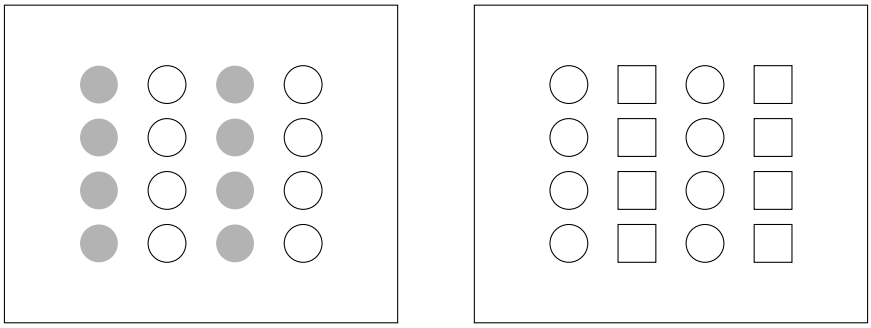


Figura 1.10: Agrupamento por similaridade de 16 elementos nas mesmas posições: (a) círculos de mesmo tamanho, em colunas alternadas de elementos preenchidos e vazados, favorecem o agrupamento por preenchimento; (b) círculos e quadrados vazados, em colunas alternadas, favorecem o agrupamento por forma.

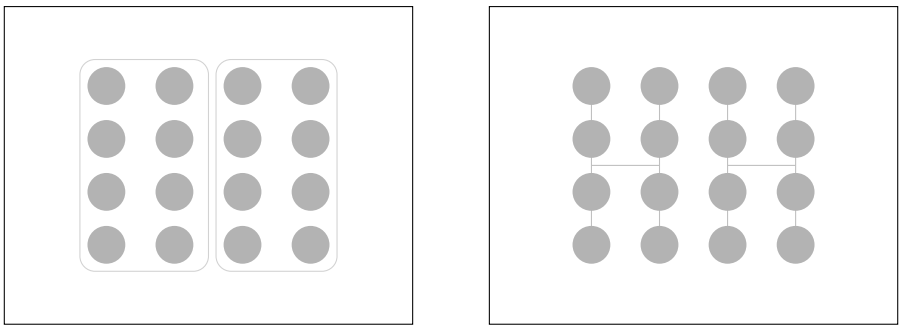


Figura 1.11: Agrupamento dos mesmos 16 círculos, mantidos nas mesmas posições: (a) dois contornos delimitam grupos de duas colunas por região comum; (b) linhas conectam as colunas aos pares, formando os mesmos grupos por conexão.

em escala de cinza para que a distinção entre os grupos independa da percepção cromática, em consonância com o critério de sucesso 1.4.1 da WCAG2.2, segundo o qual a cor não deve constituir o único meio visual de transmitir informação [W3C 2024, SC 1.4.1].

O agrupamento também pode ser estabelecido pela delimitação de uma região ou por conexões entre elementos. Lidwell et al. discutem essas estratégias no princípio *Conectividade Uniforme* [LBW 189]: elementos em uma mesma região (*Região Comum*) ou conectados tendem a ser percebidos como relacionados. Na Figura 1.11, os grupos são estabelecidos mantendo posições e características dos círculos: na primeira representação, contornos delimitam pares de colunas; na segunda, linhas conectam esses pares.

O princípio *Destino Comum* descreve a tendência de elementos que se movem na mesma direção serem percebidos como pertencentes a um mesmo grupo [LBW 031]. Na Figura 1.12, os círculos mantêm tamanho, tonalidade e distribuição espacial, enquanto as setas indicam duas direções de movimento. O agrupamento decorre da direção compartilhada indicada pelas setas.

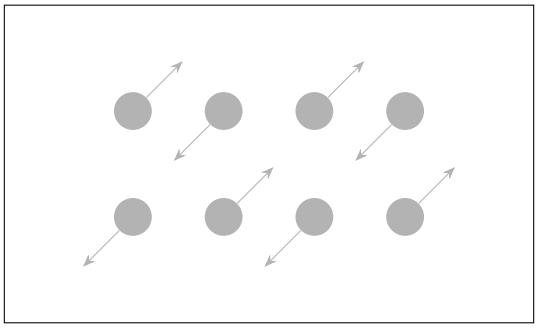


Figura 1.12: Representação estática de destino comum: oito círculos estão dispostos em duas linhas de quatro, com espaçamento uniforme, enquanto setas associadas a cada círculo alternam entre duas direções diagonais; círculos associados a setas na mesma direção tendem a ser percebidos como pertencentes ao mesmo grupo.

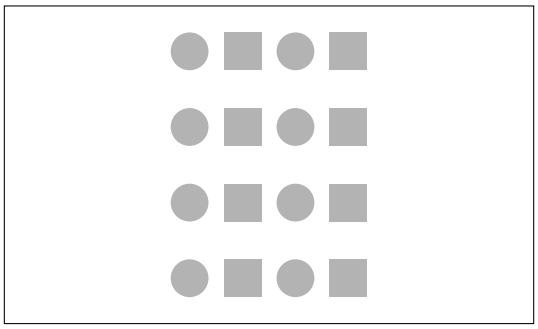


Figura 1.13: Competição entre princípios de agrupamento: a menor distância horizontal favorece a percepção de quatro linhas por proximidade, enquanto a alternância entre colunas de círculos e quadrados favorece outro agrupamento por similaridade de forma.

Esses princípios podem atuar simultaneamente e sugerir organizações distintas. Na Figura 1.13, a menor distância horizontal favorece a percepção de linhas, enquanto círculos e quadrados em colunas alternadas favorecem outro agrupamento por similaridade de forma. A análise deve considerar quais relações visuais são estabelecidas e se reforçam ou competem com a organização da informação que se pretende comunicar.

1.3.3 Percepção de formas e relações

Além dos princípios de agrupamento, outros princípios associados à organização perceptiva da Gestalt descrevem relações entre os elementos que influenciam a percepção de formas e estruturas visuais. Entre eles estão as relações entre figura e fundo (*Figura e Fundo*), o fechamento de formas incompletas (*Fechamento*) e a percepção de continuidade entre elementos (*Continuidade*).

O princípio *Figura e Fundo* descreve a percepção de elementos como figura ou fundo [LBW 067]. Para direcionar a atenção a determinada informação, essa distinção deve permitir que os elementos relevantes sejam percebidos claramente como figura em relação ao contexto. A relação também pode ser deliberadamente ambígua: certas configurações admitem mais de uma organização, e a percepção pode alternar entre

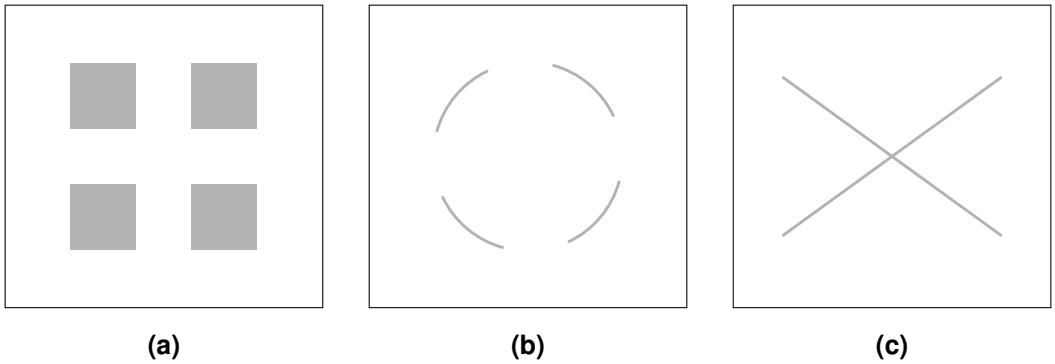


Figura 1.14: Exemplos de organização perceptiva: (a) quatro quadrados cinza separados por uma região branca em forma de cruz ilustram relações entre figura e fundo; (b) quatro arcos separados favorecem a percepção de um círculo apesar das interrupções no contorno, ilustrando fechamento; (c) duas linhas diagonais que se cruzam formando um X favorecem a percepção de duas trajetórias contínuas, ilustrando boa continuidade.

elas. Na Figura 1.14(a), o espaço branco entre quatro quadrados cinza forma uma cruz, mostrando que o espaço entre elementos também pode constituir uma forma percebida.

O princípio *Fechamento* descreve a tendência de perceber como completas formas com partes ausentes, desde que haja informação visual suficiente para sugerir a configuração [LBW 027]. Na Figura 1.14(b), quatro arcos separados favorecem a percepção de um círculo apesar das interrupções em seu contorno.

O princípio *Continuidade*, ou *Boa Continuação*, descreve a tendência de perceber elementos alinhados como parte de uma linha ou percurso contínuo, em vez de mudanças abruptas de direção [LBW 084]. Na Figura 1.14(c), as duas diagonais que se cruzam favorecem a percepção de duas trajetórias que continuam através do ponto de interseção.

1.3.4 O processo de composição

Os princípios anteriores explicam como relações entre elementos são percebidas. A composição trata de questão complementar: como organizá-los para melhorar a comunicação. Isso envolve decidir sua posição, proporção, destaque e as relações visuais e espaciais que estabelecem entre si.

Contraste, *Repetição*, *Alinhamento* e *Proximidade*

Williams apresenta quatro princípios básicos de design: *Contraste*, *Repetição*, *Alinhamento* e *Proximidade*, para os quais alude, bem-humorada, a um acrônimo "bastante inapropriado" [Williams 2015, 13], explicitado no índice remissivo como CRAP [Williams 2015, 237], palavra que significa "lixo" em português.

Esses princípios orientam decisões de composição e diferem dos princípios de organização perceptiva discutidos anteriormente: os princípios perceptivos ajudam a explicar como relações visuais são percebidas, enquanto os princípios de Williams orientam a organização deliberada dos elementos.

A Figura 1.15 aplica os quatro princípios a uma mesma composição, formada pelos textos “Palestra”, “Design Digital”, “18 nov” e “Auditório Azul”. A Figura 1.15(a) apresenta a composição inicial, com tratamento tipográfico uniforme e os textos distribuídos pelos quatro cantos. As quatro versões seguintes aplicam um princípio por vez, exemplificando *Contraste*, *Repetição*, *Alinhamento* e *Proximidade*; a última combina as quatro intervenções.

O *Contraste* é produzido quando elementos são deliberadamente diferenciados, permitindo distinguir funções, níveis de importância ou partes da composição [Williams 2015, 69–84]. Segundo Williams, para estabelecer contraste a diferença deve ser suficientemente clara para parecer intencional. Na Figura 1.15(b), o aumento do tamanho e do peso de “Design Digital” estabelece contraste em relação aos demais textos. O contraste pode ser produzido por características como tamanho, peso tipográfico, forma, posição ou cor, mas não deve ser confundido com os requisitos específicos de contraste para acessibilidade, discutidos posteriormente.

A *Repetição* consiste em repetir características visuais ao longo da composição, como tipografia, formas, espessuras, espaçamentos ou outros elementos gráficos [Williams 2015, 55–68]. Essa recorrência contribui para estabelecer unidade e tornar perceptível que diferentes partes pertencem ao mesmo artefato ou sistema visual. Na Figura 1.15(c), “18 nov” e “Auditório Azul” recebem o mesmo tratamento em itálico, estabelecendo uma característica visual recorrente entre os dois elementos.

O *Alinhamento* estabelece conexões visuais entre os elementos por meio de seu posicionamento. Williams recomenda evitar posicionamentos arbitrários: cada elemento deve manter alguma relação visual com outro elemento da composição [Williams 2015, 33–54]. Na Figura 1.15(d), os quatro textos são posicionados segundo o mesmo eixo à esquerda, em contraste com a distribuição pelos quatro cantos da composição inicial. O alinhamento pode organizar elementos espacialmente separados, criando linhas visuais que estruturam a composição e tornam perceptíveis as relações entre suas partes.

A *Proximidade* estabelece relações pela distância entre os elementos: itens relacionados devem ser aproximados para serem percebidos como unidade, e não como elementos independentes [Williams 2015, 15–32]. Na Figura 1.15(e), “Palestra” e “Design Digital” são aproximados na região superior, enquanto “18 nov” e “Auditório Azul” são aproximados na região inferior, formando dois grupos. Assim, a proximidade contribui para explicitar a estrutura da informação e reduzir a necessidade de outros recursos para indicar relações. Embora relacionado ao agrupamento perceptivo por proximidade discutido anteriormente, aqui o princípio é empregado como decisão de composição.

A Figura 1.16 permite aplicar conjuntamente os quatro princípios à composição inicial. A repetição da mesma família tipográfica contribui para a unidade visual, enquanto variações de tamanho, peso e caixa estabelecem contraste e diferentes níveis de proeminência. Entretanto, a fotografia ocupa parcela expressiva da área e, com o logotipo em grandes dimensões e o título em tamanho e peso elevados, estabelece múltiplos focos de destaque. As cinco linhas tracejadas evidenciam diferentes eixos verticais de alinhamento, inclusive entre blocos próximos. Os contornos pontilhados mostram que a proximidade estabelece grupos reconhecíveis, mas algumas relações podem ser

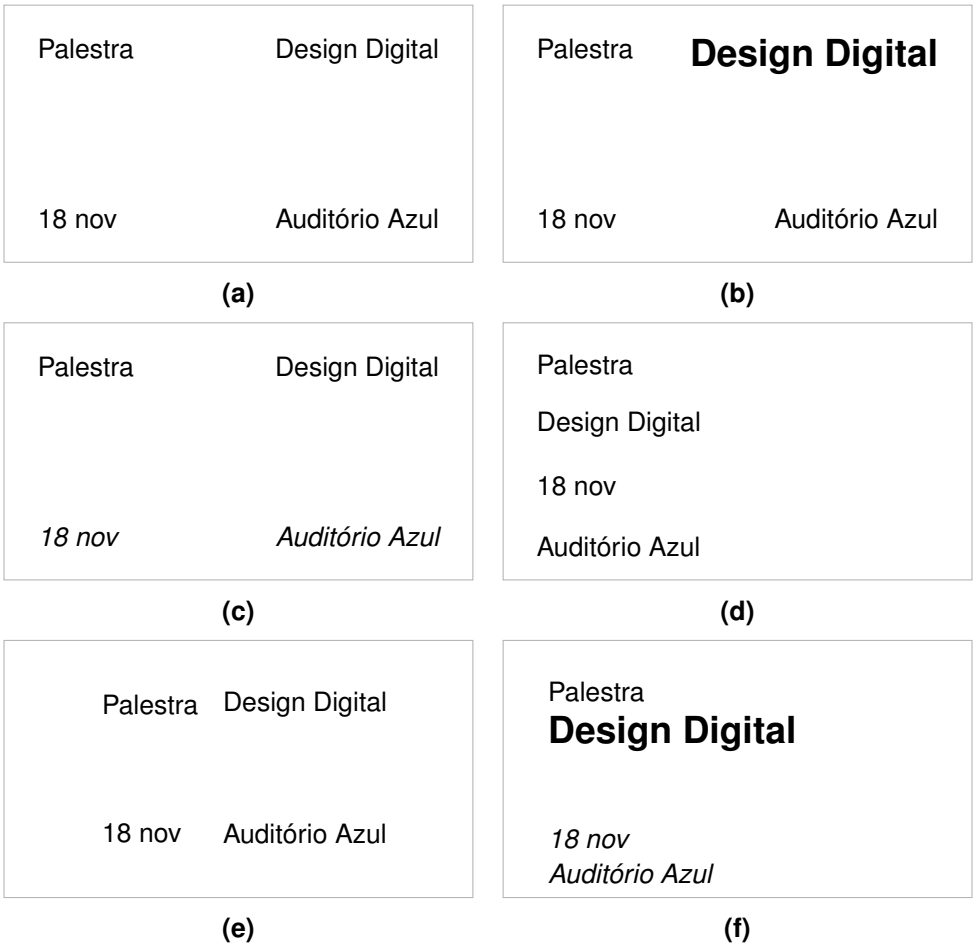


Figura 1.15: Aplicação isolada e combinada de quatro princípios à mesma composição: (a) composição inicial, com tratamento tipográfico uniforme e textos nos quatro cantos; (b) contraste por tamanho e peso; (c) repetição do tratamento em itálico; (d) alinhamento dos quatro textos à esquerda; (e) proximidade entre informações relacionadas, formando dois grupos; (f) combinação de contraste, repetição, alinhamento e proximidade.

reforçadas, como entre as informações de inscrição, o endereço eletrônico e o QR Code. Assim, a análise de contraste, repetição, alinhamento e proximidade identifica tanto relações que estruturam o artefato quanto aspectos que podem ser revistos no reprojeto.

Consistência, *Regra dos Terços*, *Proporção Áurea*, *Associações Número-Espaço* e *Mapeamento*

A repetição de características e padrões pode contribuir para a *Consistência*, princípio relacionado à manutenção de elementos semelhantes em situações semelhantes [LBW 035]. Embora repetição e consistência possam se reforçar, os princípios têm escopos distintos: a repetição orienta a recorrência de características visuais na composição, enquanto a consistência envolve a manutenção de padrões e expectativas ao longo de um artefato ou conjunto de artefatos.



Figura 1.16: Análise da organização visual da Figura 1.1 segundo contraste, repetição, alinhamento e proximidade. A fotografia ocupa parcela expressiva da composição, enquanto o logotipo em grandes dimensões, o título destacado e as variações de tamanho, peso e caixa estabelecem diferentes níveis de proeminência. A repetição da mesma família tipográfica contribui para a unidade visual. Cinco linhas verticais tracejadas evidenciam diferentes eixos de alinhamento, e quatro contornos pontilhados tornam explícitos agrupamentos estabelecidos por proximidade entre informações relacionadas.

Além da consistência entre elementos e padrões, algumas referências podem orientar sua distribuição no espaço. A *Regra dos Terços* [LBW 160] e a *Proporção Áurea* [LBW 083] oferecem estruturas que podem ser utilizadas como referências para a composição, sem determinar uma composição adequada ou garantir a qualidade do resultado. A proporção áurea corresponde aproximadamente à razão 1:1,618; ao dividir uma dimensão segundo essa proporção, as partes correspondem a 38,2% e 61,8% do total. A Figura 1.17 apresenta o mesmo *card* sobre duas grades: na primeira, linhas horizontais e verticais dividem a área em terços; na segunda, as divisões são posicionadas a 38,2% e 61,8% da largura e da altura. A posição dos elementos varia entre as duas versões, ilustrando o uso das grades como referências para a organização espacial.

Essas estruturas podem orientar o enquadramento de imagens. Na proporção áurea, a grade fornece uma região central de referência para o principal ponto de interesse; na regra dos terços, as interseções entre as linhas fornecem referências para o posicionamento de elementos de interesse [LBW 160].

A disposição espacial também pode estabelecer relações com a informação representada. O princípio *Associações Número-Espaço* descreve associações entre números e posições espaciais [LBW 127]. Em culturas com direção predominante de



Figura 1.17: Organização do mesmo card segundo duas grades, mantendo conteúdo e tratamento tipográfico e variando a posição dos elementos: (a) linhas horizontais e verticais dividem a área em terços; (b) linhas horizontais e verticais dividem a área segundo a proporção áurea.

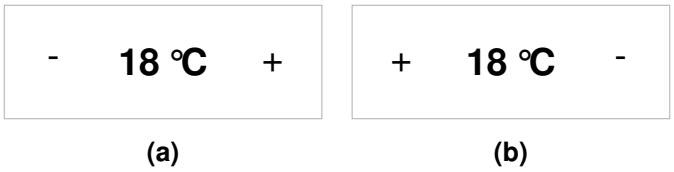


Figura 1.18: Duas disposições para o mesmo controle de temperatura, mantendo dimensões, fonte sem serifa em 14 pontos, posições e espaçamentos: (a) sinal de menos à esquerda e sinal de mais à direita; (b) sinal de mais à esquerda e sinal de menos à direita.

leitura da esquerda para a direita, números menores tendem a ser associados à esquerda e números maiores, à direita. Um exemplo é o efeito SNARC (*Spatial-Numerical Association of Response Codes*), no qual respostas a números menores são mais rápidas à esquerda e respostas a números maiores, à direita [Göbel 2018]. Essas associações são flexíveis e podem ser influenciadas pela direção de leitura [Göbel 2018]. Na Figura 1.18, a primeira versão segue a associação esquerda-menor e direita-maior, posicionando o sinal de menos à esquerda e o de mais à direita; a segunda inverte essas posições.

De forma mais geral, o princípio *Mapeamento* trata da correspondência entre elementos e relações de um sistema [LBW 113]. Em um artefato digital, a disposição espacial pode contribuir para tornar perceptíveis relações entre informações, controles e seus efeitos. Na Figura 1.18, a posição dos controles estabelece uma correspondência espacial com a direção da alteração do valor. Assim, a composição pode expressar relações presentes na estrutura da informação e na interação, além de organizar visualmente os elementos.

1.4 *Legibilidade*, leitura e compreensão

A organização visual discutida na seção anterior não garante, por si só, que a informação textual possa ser percebida e compreendida. A avaliação de textos em artefatos digitais envolve aspectos distintos, embora relacionados: a possibilidade de distinguir visualmente caracteres, palavras e blocos de texto; a facilidade de ler e compreender o conteúdo; e o uso de recursos visuais para estabelecer destaque e hierarquia. Esta seção



Figura 1.19: Forma visual das mesmas palavras em duas áreas de mesmas dimensões, mantendo família sem serifa, tamanho de 12 pontos, peso, disposição em uma linha, linha de base e espaço em branco: (a) iniciais maiúsculas e demais letras minúsculas produzem contornos com variações de altura; (b) palavras inteiramente em maiúsculas apresentam contornos de altura mais uniforme e aproximadamente retangulares.

examina esses aspectos, distinguindo inicialmente *Legibilidade* e *Leiturabilidade* e, em seguida, discutindo mecanismos para dirigir a atenção e estabelecer hierarquia visual.

1.4.1 *Legibilidade*

A *Legibilidade* diz respeito à facilidade com que elementos textuais podem ser percebidos e distinguidos visualmente [LBW 107]. Ela difere da *Leiturabilidade*, discutida na próxima subseção: caracteres e palavras podem ser facilmente distinguíveis e, ainda assim, o texto ser difícil de ler ou compreender em razão de sua linguagem, estrutura ou organização.

A legibilidade depende de características que atuam em conjunto. No princípio *Legibilidade*, Lidwell et al. discutem tamanho e tipo de fonte, uso de maiúsculas e minúsculas, contraste entre texto e fundo, organização dos blocos, comprimento das linhas e entrelinha [LBW 107]. A escolha entre fontes com ou sem serifa não deve ser tratada como uma regra geral segundo a qual uma categoria seria intrinsecamente mais legível: os autores observam que não há diferença geral de desempenho entre elas [LBW 107]. A composição da página também deve evitar linhas viúvas, que prejudicam a continuidade visual da leitura. Esse cuidado foi adotado sistematicamente na composição deste capítulo, com ajustes de texto e paginação.

O uso extensivo de letras maiúsculas merece atenção particular. Williams observa que as palavras são reconhecidas não apenas pelas letras que as compõem, mas também por suas formas; quando inteiramente em maiúsculas, apresentam contornos mais semelhantes e predominantemente retangulares, reduzindo diferenças visuais que auxiliam seu reconhecimento [Williams 2015, 201]. O problema não se restringe a textos corridos: em *cards* e *slides*, títulos, chamadas, datas, locais e outras informações podem ser apresentados simultaneamente em maiúsculas, reduzindo tanto a legibilidade quanto as possibilidades de diferenciação entre os elementos.

A Figura 1.19 compara as mesmas palavras mantendo família tipográfica, tamanho, peso, posição e espaçamento. Com inicial maiúscula e demais letras minúsculas, as diferentes alturas dos caracteres produzem contornos variados; inteiramente em maiúsculas, as palavras apresentam altura mais uniforme e contornos aproximadamente retangulares. A comparação torna visível a redução das diferenças de forma entre as palavras provocada pelo uso de caixa alta.

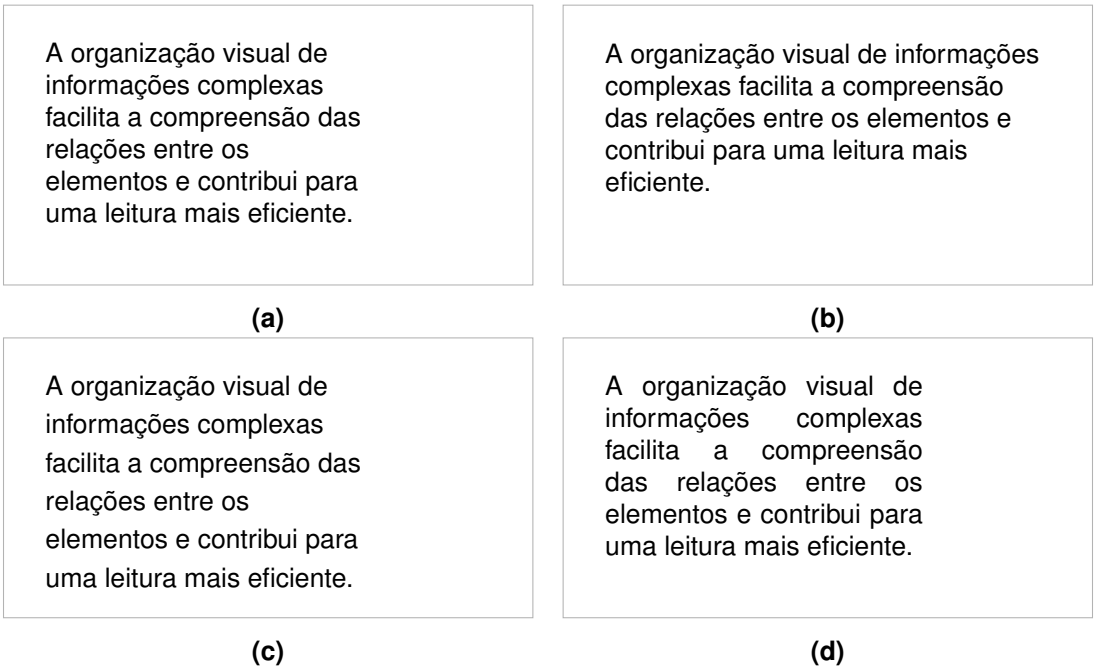


Figura 1.20: Configurações do mesmo bloco textual em áreas de mesmas dimensões, mantendo fonte sem serifa de 10 pontos, peso e posição inicial: (a) texto alinhado à esquerda em um bloco de largura intermediária; (b) maior largura do bloco, produzindo linhas mais longas e alterando a distribuição do espaço em branco; (c) entrelinha de 16,2 pontos, em comparação com 12 pontos nas demais configurações, aumentando o espaçamento vertical e a altura ocupada pelo texto; (d) texto justificado, com margens esquerda e direita alinhadas e espaçamentos variáveis entre as palavras.

As maiúsculas podem, entretanto, ser empregadas como recurso de composição. Williams admite seu emprego deliberado quando o efeito visual o justifica, desde que se considere a redução de legibilidade e leiturabilidade envolvida [Williams 2015, 201]. Maiúsculas e minúsculas podem produzir contraste de forma: seu emprego seletivo pode diferenciar elementos, enquanto seu uso generalizado elimina essa possibilidade.

Comprimento de linha e entrelinha afetam a apresentação do texto. Linhas muito longas ou muito curtas e uma entrelinha inadequada podem dificultar a leitura. Essas características devem ser consideradas em conjunto com o tamanho da fonte e a organização do bloco textual, pois alterações em uma delas modificam as condições em que as demais são percebidas [LBW 107]. O alinhamento interfere nessa organização: o alinhamento à esquerda mantém um ponto de início consistente para as linhas, enquanto a justificação pode produzir espaçamentos irregulares entre palavras [LBW 107]. Na Figura 1.20, a mesma frase é apresentada em áreas de mesmas dimensões, mantendo fonte, tamanho e peso. Um bloco mais largo produz linhas mais longas e redistribui o espaço em branco; uma entrelinha maior aumenta o espaço vertical ocupado pelo texto; e a justificação alinha as duas margens, mas introduz espaçamentos variáveis entre palavras.

Família e tamanho da fonte e contraste entre texto e fundo constituem outras dimensões da legibilidade [LBW 107]. Essas características também interagem: uma



Figura 1.21: Variação de propriedades tipográficas da expressão “Boa Legibilidade?” em áreas de mesmas dimensões e fundo branco, com texto centralizado e amplo espaço em branco: (a) fonte sem serifa, preta, em 14 pontos; (b) fonte serifada, mantendo 14 pontos, peso e contraste; (c) fonte sem serifa em 10 pontos; (d) fonte sem serifa em 14 pontos e cinza claro, com menor contraste.

mesma escolha tipográfica pode produzir resultados diferentes em tamanhos ou condições de contraste distintos. A Figura 1.21 mantém a expressão “Boa Legibilidade?” centralizada em áreas de mesmas dimensões e com amplo espaço em branco, variando uma característica por vez. A comparação entre fontes com e sem serifa evidencia a mudança de forma sem pressupor superioridade de uma categoria; a redução do tamanho diminui a área ocupada pelo texto; e o uso de cinza claro sobre fundo branco reduz seu contraste.

A orientação dos elementos constitui outro aspecto relevante. O princípio *Sensibilidade à Orientação* descreve diferenças na facilidade de detectar e discriminar elementos em função de sua orientação [LBW 130]. Isso favorece orientações convencionais quando orientações incomuns dificultam a percepção, embora estas possam ser adequadas quando desempenham uma função específica na composição.

1.4.2 Leitura e compreensão

Enquanto a *Legibilidade* está relacionada à possibilidade de perceber e distinguir visualmente o texto, a *Leiturabilidade* diz respeito à facilidade com que seu conteúdo pode ser lido e compreendido [LBW 153]. Assim, um texto pode ser visualmente legível e, ainda assim, apresentar dificuldades decorrentes da linguagem, da quantidade de informação, de sua estrutura ou da organização do conteúdo.

A quantidade de texto constitui uma dessas dimensões. Em *Elimine Palavras Desnecessárias*, Krug recomenda eliminar palavras que não contribuem para a comunicação, reduzindo o volume textual sem eliminar informação necessária [Krug 2014, 48–53]. O objetivo não é produzir o menor texto possível, mas retirar o que não contribui para a mensagem, preservando o necessário à compreensão. Siglas, jargão e termos não explicados podem dificultá-la, mesmo em textos curtos e legíveis.

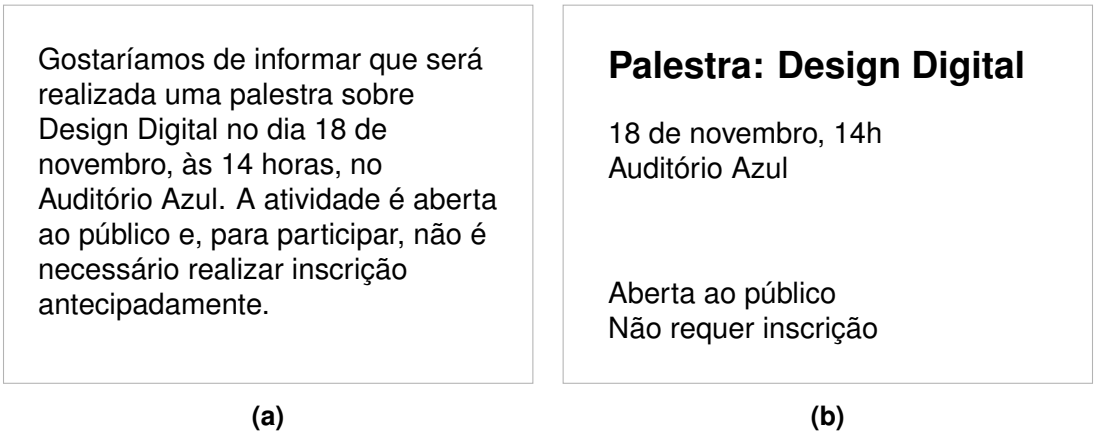


Figura 1.22: Duas apresentações das mesmas informações sobre a palestra "Design Digital" em áreas de mesmas dimensões, mantendo a família tipográfica sem serifa: (a) informações em 11 pontos e peso regular, redigidas como um único parágrafo corrido ocupando grande parte da área; (b) conteúdo reduzido e organizado em três grupos separados por espaço em branco, com título em 14 pontos e negrito e demais informações em 11 pontos agrupadas por proximidade, deixando amplo espaço em branco à direita.

Na Figura 1.22, as mesmas informações são apresentadas inicialmente em um único parágrafo em texto corrido, em fonte sem serifa de 11 pontos e peso regular. Na segunda versão, o conteúdo é reduzido às informações necessárias e distribuído em três grupos separados por espaço em branco: título, data e local e condições de participação. O título recebe maior tamanho, 14 pontos, e peso negrito, enquanto as demais informações permanecem em 11 pontos. O alinhamento à esquerda e a proximidade estabelecem relações entre as informações. A redução textual atua, assim, em conjunto com hierarquia tipográfica, agrupamento e espaço em branco.

A compreensão depende também do conhecimento exigido de quem utiliza o artefato. Dix et al. distinguem duas formas de recuperação de informação da memória: em recordação, a informação é recuperada da memória; em reconhecimento, sua apresentação fornece uma pista para que seja reconhecida [Dix et al. 2004, 38]. Os autores observam que o reconhecimento constitui uma atividade cognitiva menos complexa e relacionam essa diferença ao projeto de sistemas interativos, por exemplo, tornando comandos disponíveis para seleção em vez de exigir que sejam recordados e introduzidos diretamente [Dix et al. 2004, 53].

Essa distinção fundamenta o princípio *Reconhecimento em Vez de Recordação*, que recomenda favorecer o reconhecimento em vez da recuperação não auxiliada de informação da memória [LBW 155]. Uma aplicação aparece no princípio de visibilidade das operações discutido por Dix et al.: tornar perceptíveis as operações disponíveis favorece o reconhecimento; quando essa informação não está na interface, é preciso recordar quando determinada operação pode ou não ser realizada [Dix et al. 2004, 262].

Essa questão aparece claramente no uso de ícones. O princípio *Representação Icônica* trata do emprego de imagens para representar objetos, ações e conceitos [LBW 094]. Reconhecer a forma gráfica de um ícone e compreender o significado atribuído

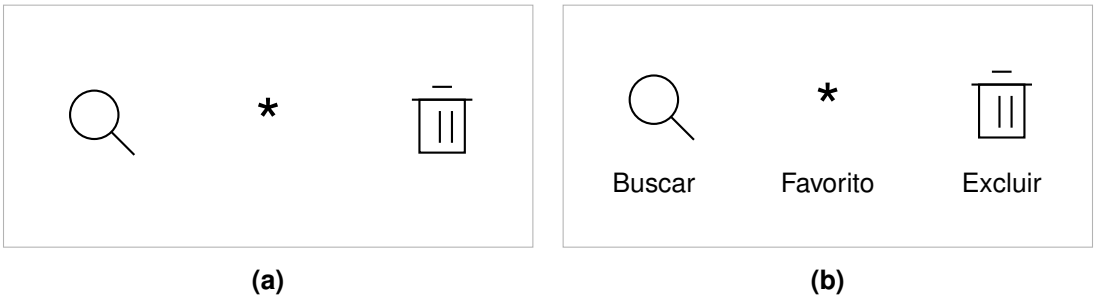


Figura 1.23: Representação das mesmas três ações em áreas de mesmas dimensões: (a) lupa, estrela e lixeira distribuídas horizontalmente e apresentadas apenas como ícones; (b) os mesmos ícones acompanhados pelos rótulos “Buscar”, “Favorito” e “Excluir”, em fonte sem serifa de 10 pontos e peso regular, formando três grupos de ícone e texto.

a ele em uma interface são processos distintos. Quando seu significado é pouco conhecido ou pouco evidente, um rótulo textual pode explicitar a ação ou informação representada. A escolha dos rótulos também requer atenção: Dix et al. observam que é difícil escolher rótulos com o mesmo significado para todas as pessoas e que explicações adicionais podem contribuir para que encontrem a informação procurada [Dix et al. 2004, 206].

Na Figura 1.23, as três ações são inicialmente representadas apenas por ícones distribuídos horizontalmente. Na segunda apresentação, ordem e distribuição são mantidas, mas cada ícone é associado por proximidade a um rótulo textual em 10 pontos: “Buscar”, “Favorito” e “Excluir”. Os rótulos acrescentam uma representação verbal ao símbolo gráfico, reduzindo a dependência da interpretação do ícone isolado.

A quantidade de esforço necessária para compreender ou utilizar um artefato pode ser examinada pelo princípio *Carga de Desempenho*, que trata da carga associada à realização de uma tarefa, incluindo esforço cognitivo e, quando aplicável, físico [LBW 135]. Uma interface pode aumentar a carga cognitiva, por exemplo, quando exige recuperar e introduzir informações que poderiam ser disponibilizadas durante a tarefa.

A Figura 1.24 compara duas versões de uma interface com um código Pix. Na primeira, a interface oferece o campo “Código Pix” e a ação “Continuar”. Na segunda, acrescenta-se a ação “Copiar da área de transferência”, posicionada junto aos controles da tarefa. Quando o código está disponível na área de transferência, essa ação reduz a necessidade de recuperá-lo de outra fonte, memorizá-lo e digitá-lo no campo.

A estrutura das informações também pode contribuir para sua compreensão quando o conteúdo envolve uma sequência de eventos ou relações temporais. O princípio *Narrativa* descreve o emprego de estruturas narrativas na comunicação de informações [LBW 176]. A organização sequencial é pertinente quando ordem, evolução ou contexto fazem parte da informação a comunicar.

1.4.3 Atenção, destaque e hierarquia

A organização da atenção e da hierarquia depende também das relações estabelecidas entre os elementos da composição.

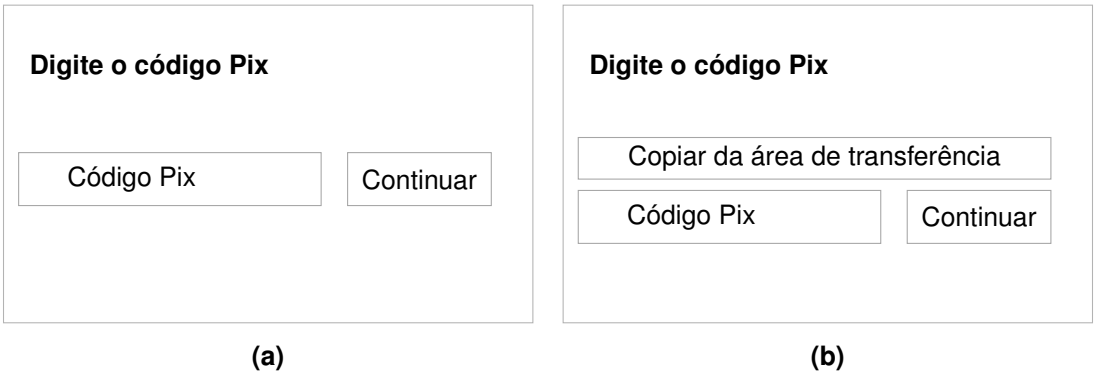


Figura 1.24: Duas versões da interface para inserir um código Pix, mantendo as dimensões da área e a fonte sem serifa de 10 pontos: (a) campo "Código Pix" e botão "Continuar" alinhados horizontalmente; (b) acréscimo da ação "Copiar da área de transferência" acima do campo e do botão, preservando a disposição horizontal desses dois controles.

Os princípios *Escala* e *Equilíbrio* tratam de outras relações entre os elementos de uma composição. A *Escala* utiliza diferenças de tamanho relativo para sinalizar importância e posição, enquanto o *Equilíbrio* considera a distribuição do sinal visual em relação a um eixo, sem exigir uma disposição simétrica [Gordon 2020]. Diferenças de escala aparecem, por exemplo, quando títulos são apresentados em tamanho maior que informações subordinadas. O equilíbrio pode ser observado na própria organização de várias figuras deste capítulo, nas quais duas ou mais representações são distribuídas em áreas equivalentes e separadas por espaçamentos regulares.

A organização visual pode estabelecer diferenças de importância entre elementos de uma composição. Krug denomina *Hierarquia Visual* o emprego de relações visuais para tornar perceptíveis essas diferenças: elementos mais importantes recebem maior proeminência, elementos relacionados são apresentados de maneira visualmente relacionada e elementos subordinados são organizados de modo a evidenciar essa relação [Krug 2014, 33–36]. A hierarquia visual permite, assim, perceber a estrutura da informação antes da leitura detalhada de todo o conteúdo.

Contraste e *Proximidade* constituem recursos distintos para tornar perceptível essa hierarquia. Como discutido anteriormente, Williams trata o *Contraste* como a diferenciação deliberada entre elementos [Williams 2015, 69–84]: diferenças de tamanho, peso tipográfico, forma, posição ou cor podem tornar alguns mais proeminentes. A *Proximidade*, por sua vez, indica relações: aproximar itens relacionados favorece sua percepção como uma unidade e pode evidenciar diferentes níveis de organização da informação [Williams 2015, 15–32][LBW 152]. *Contraste*, *Proximidade* e *Hierarquia Visual* não são equivalentes: *Contraste* diferencia elementos, *Proximidade* estabelece relações entre eles e a *Hierarquia Visual* resulta da organização dessas e de outras relações visuais.

O princípio *Destaque* trata de técnicas para dirigir a atenção a determinadas informações [LBW 091]. Entre os recursos discutidos por Lidwell et al. estão negrito, itálico, sublinhado, mudanças de tipo de fonte, cor e inversão. Os autores observam que



Figura 1.25: Diferentes relações entre destaque e diferenciação visual no mesmo *card*, mantendo conteúdo, família tipográfica, posições e espaçamentos: (a) textos em 10 pontos recebem diferentes tratamentos tipográficos e gráficos, enquanto "Palestra" permanece em peso regular e estilo normal; (b) o destaque é concentrado em "Design Digital", apresentado em letras maiúsculas, 14 pontos, negrito e delimitado por uma caixa, enquanto os demais textos permanecem em 10 pontos; (c) "Design Digital" mantém 14 pontos e negrito, mas retorna à caixa normal e a caixa retangular é removida.

destacar mais de 10% do conteúdo visível dilui o efeito do destaque [LBW 091]. Quando muitos elementos recebem tratamento visual diferenciado simultaneamente, o destaque perde a capacidade de distinguir o que deveria receber atenção.

Essa limitação relaciona-se ao princípio *Habituação*: a exposição repetida a um mesmo estímulo tende a reduzir a resposta a ele [LBW 087]. Recursos utilizados continuamente para chamar a atenção podem, portanto, deixar de funcionar como sinais de excepcionalidade. Destacar tudo, ou usar repetidamente o mesmo destaque para informações de importância muito diferente, enfraquece sua função informacional.

Uma relação diferente é descrita pelo *Efeito von Restorff*: em um conjunto de elementos relativamente semelhantes, aquele que se distingue dos demais tende a receber maior atenção e ser mais facilmente lembrado [LBW 194]. A distinção depende também do contexto: o que se destaca em um conjunto homogêneo pode deixar de fazê-lo quando vários elementos apresentam características excepcionais.

A Figura 1.25 contrasta essas situações mantendo a organização espacial. Nas três representações, os cinco elementos ocupam a região central do *card*, alinhados no mesmo eixo vertical, com espaçamento regular e amplo espaço em branco. Na primeira, diferentes recursos de destaque são distribuídos entre os elementos: "Design Digital" aparece em negrito, "18 nov" e "Auditório Azul" são delimitados por caixas e "14h" aparece em itálico. Na segunda, o destaque é concentrado em "Design Digital", em letras maiúsculas, 14 pontos, negrito e delimitado por uma caixa, enquanto os demais textos permanecem em 10 pontos. Na terceira, "Design Digital" mantém 14 pontos e negrito, mas retorna à caixa normal e perde a caixa retangular, reduzindo os recursos empregados para destacá-lo.

A hierarquia resultante dessas decisões não deve ser confundida com um percurso ocular fixo. Tornar uma informação mais proeminente pode favorecer sua percepção,



Figura 1.26: Hierarquia visual no card da Figura 1.1. A fotografia ocupa cerca de um quarto da altura, na região superior, e as informações são distribuídas abaixo em blocos separados por espaços em branco e organizados predominantemente por alinhamentos à esquerda. As marcações identificam como elementos de maior proeminência a fotografia, o logotipo e o título; de proeminência intermediária, a identificação da palestrante, a data e o local e as condições de participação; e de menor proeminência, a identificação do evento, as informações para inscrição e os elementos institucionais. As marcações não representam um percurso de leitura.

mas não estabelece uma sequência universal segundo a qual todas as pessoas observarão os elementos. A *Hierarquia Visual* deve ser analisada pelas relações de proeminência e subordinação estabelecidas entre eles.

A Figura 1.26 retoma o *card* fictício apresentado no início do capítulo, preservando sua composição. A fotografia ocupa cerca de um quarto da altura do *card*, na região superior, enquanto as informações são distribuídas abaixo em blocos predominantemente alinhados à esquerda e separados por espaços em branco. Sobre essa composição, as marcações permitem observar três níveis de proeminência. A fotografia, o logotipo e o título constituem os elementos de maior proeminência; a identificação da palestrante, a data e o local e as condições de participação apresentam proeminência intermediária; e a identificação do evento, as informações para inscrição e os elementos institucionais apresentam menor proeminência. Essas relações caracterizam a hierarquia visual da composição, mas não determinam um percurso de leitura.

1.5 Acessibilidade como requisito da comunicação

Os princípios discutidos nas seções anteriores tratam de como a informação é selecionada, organizada, apresentada e compreendida. Para que a comunicação seja efetiva, entretanto,

essas decisões precisam contemplar diferentes formas de percepção e interação. A informação e as funcionalidades do artefato precisam estar disponíveis também para pessoas que utilizam diferentes formas de perceber, compreender e interagir com o conteúdo.

A acessibilidade não deve, portanto, ser tratada apenas como uma verificação posterior à produção do artefato. O *World Wide Web Consortium* (W3C) recomenda integrá-la ao longo de todo o processo de produção Web, por meio de atividades não necessariamente sequenciais e retomadas ao longo do tempo para ampliar continuamente a capacidade de acessibilidade [White et al. 2026, Planning and Managing Web Accessibility]. O Modelo de Acessibilidade em Governo Eletrônico (eMAG) também a trata como parte do processo de desenvolvimento, articulando padrões Web, recomendações de acessibilidade e avaliação, e caracteriza sua manutenção como um processo contínuo [Brasil, Departamento de Governo Eletrônico 2014, Seções 2 e 2.4].

No conteúdo Web, a WCAG 2.2 organiza seus requisitos segundo os quatro princípios POUR, apresentados na Seção 1.1: perceptível (*Perceivable*), operável (*Operable*), compreensível (*Understandable*) e robusto (*Robust*) [W3C 2024, Seção 3.2]. Esses requisitos não substituem os princípios de comunicação e organização visual discutidos neste capítulo, mas introduzem condições a serem consideradas conjuntamente para que a informação não dependa de uma única forma de percepção ou interação.

As subseções seguintes examinam algumas dessas relações, concentrando-se em situações nas quais decisões de comunicação visual precisam ser examinadas também quanto às condições de acesso à informação.

1.5.1 Cor, contraste e representação

O *Contraste* discutido anteriormente como princípio de composição não deve ser confundido com os requisitos quantitativos estabelecidos pelas diretrizes de acessibilidade. No primeiro caso, constitui um recurso para diferenciar elementos e estabelecer relações visuais [Williams 2015, 69–84]. Na WCAG2.2, critérios de sucesso estabelecem relações mínimas de contraste para determinados elementos, como texto e imagens de texto. Para texto em tamanho normal, o critério 1.4.3 estabelece relação mínima de 4,5:1 no nível de conformidade AA; para texto grande, 3:1 [W3C 2024, SC 1.4.3]. Na WCAG, considera-se texto grande aquele com pelo menos 18 pt (24 pixels CSS, de *Cascading Style Sheets*) em peso normal ou 14 pt (aproximadamente 18,66 CSS px) em negrito. Assim, uma composição pode apresentar contraste suficiente para desempenhar uma função de diferenciação visual e ainda não satisfazer os requisitos quantitativos de contraste aplicáveis.

A cor também pode desempenhar diferentes funções na comunicação visual. Os princípios *Efeitos da Cor* e *Teoria da Cor* tratam, respectivamente, de efeitos associados à percepção das cores e das relações estabelecidas entre elas [LBW 029] [LBW 030]. Williams discute relações entre cores, incluindo esquemas complementares, tríades e cores análogas, além de variações de tonalidade e relações entre cores quentes e frias [Williams 2015, 97–108]. A cor pode participar da diferenciação, do agrupamento e do destaque de elementos, mas, quando comunica informação, esta não deve depender exclusivamente da percepção cromática.

Esse requisito é explicitado pelo critério de sucesso 1.4.1 da WCAG 2.2: a cor não deve constituir o único meio visual para transmitir informação, indicar uma ação, solicitar uma resposta ou distinguir um elemento visual [W3C 2024, SC 1.4.1]. O eMAG apresenta orientação correspondente em sua Recomendação 4.2, estendendo-a a outras características sensoriais utilizadas isoladamente [Brasil, Departamento de Governo Eletrônico 2014, Recomendação 4.2].

Nesse contexto, a *Redundância*, discutida anteriormente, pode desempenhar uma função de acessibilidade. Uma distinção representada por cor pode ser comunicada também por forma, padrão ou rótulo, para que a informação não dependa de um único recurso perceptivo [LBW 156]. A Figura 1.6 retoma a comparação anterior: em uma versão, a distinção depende apenas da cor; na outra, a mesma informação é disponibilizada também por outra característica visual. Nesse caso, a redundância não constitui ruído a eliminar, mas preserva a informação por meio de representações complementares.

1.5.2 Informação visual e representação acessível

As relações visuais discutidas nas seções anteriores podem comunicar estrutura e significado. *Proximidade*, *Alinhamento*, *Região Comum*, *Conectividade Uniforme* e outras características permitem, por exemplo, perceber que elementos pertencem ao mesmo grupo ou que uma informação funciona como título de outra. A percepção visual dessas relações, entretanto, não garante que a estrutura representada esteja disponível a pessoas que acessam o conteúdo de outras formas.

Essa distinção aparece no critério de sucesso 1.3.1 da WCAG 2.2, Informações e Relações: informações, estruturas e relações comunicadas pela apresentação devem poder ser determinadas programaticamente ou estar disponíveis em texto [W3C 2024, SC 1.3.1]. Uma relação estabelecida visualmente por posição, proximidade ou aparência pode, portanto, precisar também ser representada adequadamente pela estrutura ou semântica do próprio conteúdo. Não se deve presumir que uma estrutura visualmente evidente seja, por si só, acessível.

A Figura 1.27 mantém a mesma apresentação visual nas duas versões: “Design Digital” aparece em 14 pontos e negrito, seguido de “Palestra”, “18 de novembro, 14h” e “Auditório Azul”, em 10 pontos e peso regular. Na primeira, o conteúdo é representado apenas como imagem. Na segunda, uma estrutura disponibilizada à ferramenta de apresentação identifica “Design Digital” como título e explicita sua relação com as demais informações. A comparação evidencia que a mesma apresentação visual pode corresponder a diferentes representações estruturais da informação.

1.5.3 Conteúdo visual em mídias sociais

A publicação de conteúdo em mídias sociais apresenta uma situação particularmente relevante para a acessibilidade de artefatos visuais. Um *card*, por exemplo, pode ser produzido e publicado diretamente como imagem. Nesse caso, textos, imagens, agrupamentos, relações espaciais e outros elementos utilizados para comunicar a informação podem estar disponíveis apenas visualmente, sem uma estrutura semântica subjacente equivalente à que pode ser construída em uma página Web.

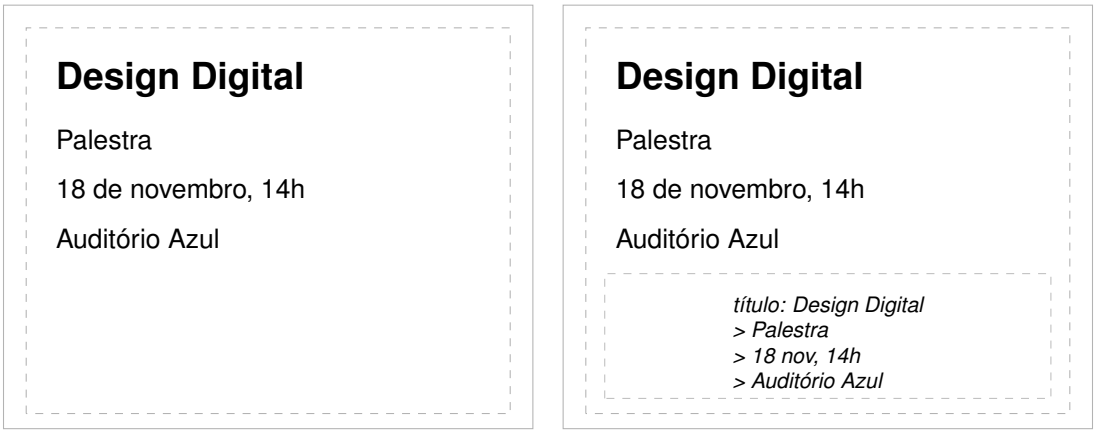


Figura 1.27: Mesma apresentação visual da informação, com título em 14 pontos e demais informações em 10 pontos, acompanhada de duas formas de representação: (a) apenas imagem; (b) imagem com estrutura disponibilizada à ferramenta de apresentação, indicada no exemplo pelo conteúdo delimitado pela linha tracejada interna.

Produzir uma alternativa acessível para esse conteúdo requer considerar, além do texto presente na imagem, seu propósito comunicacional. Nardi (2021) define esse propósito como aquilo que se deseja comunicar por meio da imagem [Nardi 2021, 266]. A autora inclui os *cards* de divulgação de eventos entre as imagens compostas e propõe identificar seus componentes, determinar quais devem ser priorizados para transmitir adequadamente o conteúdo e descrevê-los segundo orientações apropriadas aos diferentes tipos de componentes [Nardi 2021, 275–278].

A Figura 1.28 ilustra essa diferença no mesmo *card* fictício cuja alternativa textual detalhada foi apresentada na Figura 1.2. Aqui, o foco está na seleção das informações em função do propósito comunicacional. Na primeira representação, o *card* é publicado somente como imagem, sem alternativa textual. Na segunda, uma alternativa breve comunica parte das informações presentes no *card*. Na terceira, a alternativa explicita o propósito de divulgação e reúne informações relevantes para comunicar a atividade, incluindo título, palestrante e vínculo institucional, evento, data, horário, local, condições de participação e endereço para inscrição.

Essa distinção entre o conteúdo presente na imagem e aquilo que precisa ser comunicado também aparece em Gleason (2020). O autor distingue *content knowledge*, que envolve o conhecimento do conteúdo visual e de seu contexto, incluindo a intenção de quem publica, de *accessibility knowledge*, que envolve reconhecer quais elementos são relevantes para pessoas com deficiência visual e como organizar essa informação em uma representação não visual [Gleason 2020, 7–8]. Produzir uma representação acessível exige, portanto, considerar conjuntamente o conteúdo e sua finalidade, suas relações e significados e as diferentes formas pelas quais essas informações serão disponibilizadas para diferentes formas de acesso.

Essa distinção também ocorreu na produção deste capítulo: ambientes visualmente equivalentes para apresentação tabular produziram estruturas diferentes no



(a) Somente imagem

Card publicado sem alternativa textual.

(b) Alternativa textual breve

Palestra EBCD, 18/11/2026: *Design de Informação para Artefatos Digitais*, com Maria Silva.

(c) Propósito, conteúdo e relações

Divulgação da palestra *Design de Informação para Artefatos Digitais*, com Maria Silva, da Universidade Brasileira de Tecnologia, no Encontro Brasileiro de Comunicação Digital 2026. A palestra será realizada em 18 de novembro de 2026, às 14h, no Centro de Convenções Horizonte, em Itatingaba – SP. A entrada é gratuita e as vagas são limitadas. Inscrições em www.ebcd2026.org.br/palestras.

Figura 1.28: Três formas de disponibilizar o conteúdo visual da Figura 1.1 em mídia social: (a) somente a imagem, sem alternativa textual; (b) alternativa textual breve, com parte das informações do *card*; (c) alternativa orientada pelo propósito de divulgação, reunindo informações sobre a atividade e suas condições de participação.

PDF, o que levou à substituição do ambiente personalizado de quadros pelo ambiente padrão de tabelas. Essa decisão é retomada na Seção 1.8.

Nardi (2021) organiza suas diretrizes em duas dimensões complementares. O princípio de conteúdo reúne orientações para produzir alternativas que comuniquem adequadamente o propósito do conteúdo visual, enquanto o princípio de tecnologia reúne orientações para interfaces e soluções automatizadas que oferecem essas alternativas [Nardi 2021, 263–296]. A separação evidencia que a acessibilidade do conteúdo visual envolve tanto decisões de quem produz e publica o conteúdo quanto os recursos oferecidos pela plataforma.

A Tabela 1.2 sintetiza as diretrizes propostas por Nardi (2021). A primeira trata diretamente da produção de alternativas para imagens; as demais, dos recursos tecnológicos necessários para viabilizar e apoiar sua produção e acesso.

1.5.4 Siglas, abreviações e sua representação

Siglas e abreviações podem reduzir a quantidade de texto, mas sua compreensão depende de que seu significado seja conhecido ou possa ser obtido. O critério de sucesso 3.1.4 da WCAG 2.2, de nível AAA, estabelece que deve existir um mecanismo para identificar a forma expandida ou o significado de abreviações [W3C 2024, SC 3.1.4]. O eMAG apresenta orientação correspondente ao recomendar uma explicação para siglas, abreviaturas e palavras incomuns, ao menos em sua primeira ocorrência [Brasil, Departamento de Governo Eletrônico 2014, Recomendação 3.12].

A forma visual de uma sigla e o mecanismo para disponibilizar seu significado são aspectos distintos. Inserir espaços entre as letras, por exemplo, modifica sua apresentação visual sem explicitar sua forma expandida. Em conteúdo Web, o significado pode ser

Tabela 1.2: Diretrizes propostas por Nardi (2021) para alternativas ao conteúdo visual em mídias sociais online.

Número	Diretriz
1	Fornecer alternativas para imagem.
2	Viabilizar descrição de conteúdo visual em todas as funcionalidades.
3	Proporcionar descrição de conteúdo visual para todos os tipos de conteúdo visual.
4	Permitir diferentes formatos de descrição.
5	Fornecer orientações sobre como descrever os diferentes tipos de conteúdo visual.
6	Notificar ausência de descrição.
7	Permitir produção colaborativa de descrições.
8	Facilitar o acesso a recursos de descrição.
9	Prover identificação da autoria da descrição.
10	Permitir descrições em diferentes níveis de detalhamento.
11	Prover acesso ao significado dos emojis.
12	Promover melhorias constantes nos recursos para produção e acesso a alternativas ao conteúdo visual.

Fonte: Adaptado de Nardi [Nardi 2021, 263–296].

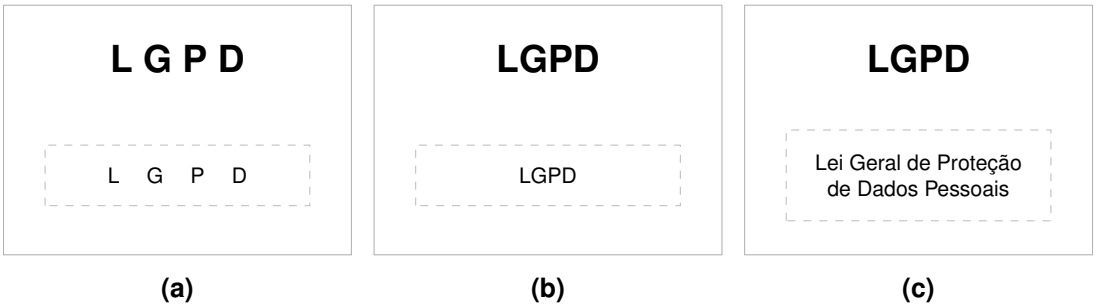


Figura 1.29: Relação entre a apresentação visual de uma sigla e a informação disponibilizada à ferramenta de apresentação: (a) letras visualmente separadas e disponibilizadas com a mesma separação; (b) “LGPD” na apresentação visual e na informação disponibilizada; (c) “LGPD” na apresentação visual e sua forma expandida, “Lei Geral de Proteção de Dados Pessoais”, disponibilizada à ferramenta de apresentação.

disponibilizado por diferentes mecanismos, incluindo sua apresentação no próprio texto ou associação programática à abreviação [W3C 2024, SC 3.1.4]. A escolha depende do formato do artefato e dos recursos disponíveis no meio.

A Figura 1.29 ilustra essa distinção utilizando como exemplo a Lei Geral de Proteção de Dados Pessoais (LGPD). Na primeira representação, “LGPD” é apresentado visualmente com espaços entre as letras, e a informação disponibilizada à ferramenta de apresentação reproduz essa separação. Na segunda, a sigla aparece sem separação tanto visualmente quanto na informação disponibilizada à ferramenta. Na terceira, a apresentação visual permanece “LGPD”, enquanto a informação disponibilizada à ferramenta fornece sua forma expandida, “Lei Geral de Proteção de Dados Pessoais”.

1.5.5 Quando princípios entram em tensão

Os princípios discutidos neste capítulo não constituem objetivos independentes que possam ser maximizados simultaneamente. Uma decisão que favorece determinado

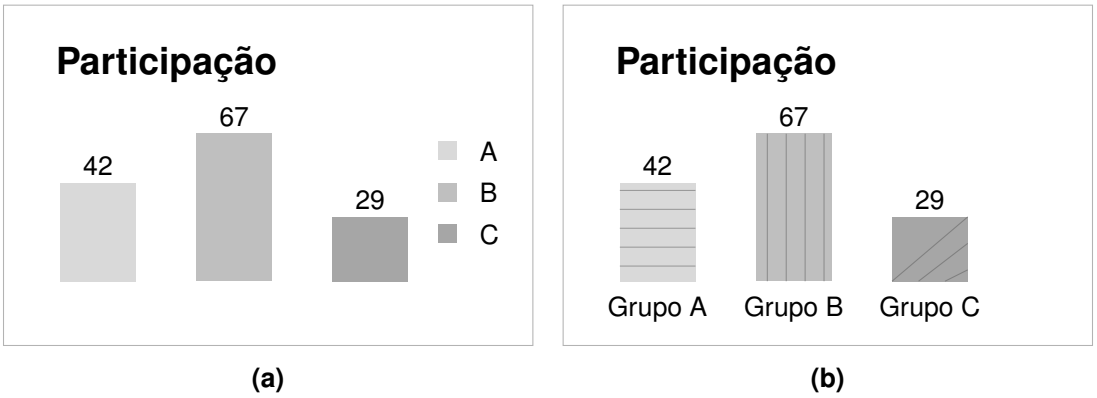


Figura 1.30: Duas representações dos mesmos valores de participação, 42, 67 e 29, mantendo dimensões das barras e tamanhos de fonte: (a) grupos identificados pelas letras “A”, “B” e “C” e diferenciados por tons de cinza; (b) rótulos “Grupo A”, “Grupo B” e “Grupo C” diretamente associados às barras e diferenciação adicional por padrões de linhas horizontais, verticais e diagonais.

aspecto da comunicação pode prejudicar outro, e sua adequação depende da informação a comunicar, das características do artefato e das condições de uso.

A simplificação exemplifica esse problema. Remover elementos desnecessários pode melhorar a relação entre sinal e ruído, mas remover informação necessária compromete a comunicação. De modo semelhante, reduzir o espaço ocupado por elementos pode produzir texto pequeno ou espaçamentos que dificultam a percepção das informações. O problema não está em simplificar ou reduzir, mas em determinar o que pode ser removido ou modificado sem comprometer a informação necessária.

Tensões semelhantes aparecem entre formas de representação. A cor pode estabelecer agrupamentos e distinções, mas não deve constituir o único recurso para comunicar determinada informação [W3C 2024, SC 1.4.1]. A redundância pode parecer incompatível com a redução de elementos, mas pode ser necessária para disponibilizar a mesma informação por formas complementares. A avaliação não consiste, portanto, em aplicar isoladamente cada princípio, mas em examinar as consequências das decisões no conjunto do artefato.

A Figura 1.30 apresenta um exemplo artificial desse compromisso. As duas representações apresentam os mesmos valores de participação, 42, 67 e 29, com o mesmo tamanho de fonte e barras de mesmas dimensões. Na primeira, os grupos são identificados como “A”, “B” e “C” em uma legenda separada, e diferentes tons de cinza distinguem as barras. Na segunda, os rótulos “Grupo A”, “Grupo B” e “Grupo C” são posicionados diretamente abaixo das barras, e padrões de linhas horizontais, verticais e diagonais acrescentam uma forma de diferenciação aos diferentes tons de cinza.

1.6 Da análise de aspectos isolados à avaliação integrada

As seções anteriores examinaram separadamente comunicação, organização perceptiva, composição, legibilidade, compreensão e acessibilidade. Essa separação permite

identificar problemas e fundamentos, mas a avaliação exige considerar esses aspectos conjuntamente e selecionar os princípios pertinentes ao artefato e seu contexto de uso.

Os princípios discutidos não constituem um *checklist* nem se aplicam todos a qualquer artefato. Um *card* estático e um formulário interativo, por exemplo, apresentam problemas distintos. Em artefatos interativos, podem tornar-se relevantes princípios como *Modelo Mental* [LBW 116], *Restrição* [LBW 037], *Confirmação* [LBW 033], *Tolerância a Erros* [LBW 074] e *Lei de Fitts* [LBW 069]. *Mapeamento*, *Consistência* e *Carga de Desempenho*, discutidos anteriormente, também podem assumir maior importância quando o artefato envolve ações e seus efeitos.

A avaliação deve considerar ainda a relação entre o projeto e as pessoas que utilizam o artefato. *Convenções* estabelecidas podem reduzir a necessidade de descobrir como elementos funcionam ou como a informação está organizada [Krug 2014, 29–32], enquanto o *Modelo Mental* influencia a maneira como um sistema é compreendido [LBW 116]. O princípio *Linha de Desejo* [LBW 048] chama atenção para situações em que o comportamento efetivo diverge do percurso previsto pelo projeto, e *Design Centrado no Usuário vs. Design Dirigido pelo Usuário* [LBW 190] distingue considerar necessidades e comportamentos dos usuários de simplesmente reproduzir suas solicitações.

Outros princípios tornam-se pertinentes apenas em determinadas condições. *Regra do Pico-Fim* [LBW 134] diz respeito a experiências ao longo do tempo, enquanto *O Que Você Vê É o Que Você Obtém* (WYSIWYG) [LBW 199] é relevante quando há relação entre um ambiente de autoria e o resultado produzido. O *Efeito Poste de Luz* [LBW 177], por sua vez, alerta para o risco de concentrar a análise naquilo que é mais fácil observar ou medir, em vez de naquilo que é relevante para a avaliação.

A pertinência dos princípios depende, portanto, do artefato, do conteúdo, do público, do contexto, da tarefa e das formas de interação. Como indica o princípio *Hierarquia de Necessidades* [LBW 090], qualidades associadas a níveis superiores não compensam necessidades fundamentais. Além disso, os princípios podem se reforçar ou entrar em tensão, como discutido na Seção 1.5.5.

A Figura 1.31 retoma o *card* da Figura 1.1 e apresenta quatro versões sucessivas de seu reprojeto. As intervenções são cumulativas: cada versão preserva as principais modificações da anterior e introduz novas alterações de conteúdo, tratamento tipográfico, alinhamento, agrupamento, hierarquia e distribuição espacial. A sequência permite observar que uma modificação pode favorecer determinados aspectos sem resolver outros e que o próprio reprojeto exige nova avaliação do conjunto.

O processo pode ser orientado por questões como: o que precisa ser comunicado? A informação necessária está presente? O que recebe atenção? As relações são perceptíveis e coerentes? O conteúdo é legível e compreensível? A informação está disponível a diferentes usuários e modos de acesso? O que interfere na comunicação? Que princípios e evidências justificam cada decisão? Essas questões sintetizam o percurso do capítulo e apoiam a seleção dos princípios pertinentes, sem constituir um novo *checklist*.



Figura 1.31: Quatro versões sucessivas do reprojeto do *card* inicial da Figura 1.1. As intervenções cumulativas modificam progressivamente conteúdo, tipografia, alinhamento, agrupamento, hierarquia e distribuição espacial. Para compor a figura, as quatro versões são reproduzidas com 79% das dimensões dos cards originais, reduzindo em 21% todos os seus elementos, inclusive a tipografia.

1.7 Da avaliação ao reprojeto

A avaliação permite identificar e fundamentar problemas de um artefato, mas não determina uma única solução. Os princípios discutidos neste capítulo oferecem fundamentos para examinar as consequências das decisões de projeto e comparar alternativas, admitindo diferentes soluções para os problemas identificados.

O reprojeto pode, assim, ser entendido como um processo iterativo: aspectos do artefato são modificados em resposta aos problemas identificados e o resultado é novamente avaliado. Uma alteração destinada a melhorar a hierarquia visual pode introduzir problemas de legibilidade; a remoção de elementos para reduzir ruído pode eliminar informação necessária; e uma solução visualmente adequada pode requerer modificações para tornar a informação acessível. Cada modificação deve, portanto, ser examinada considerando os diferentes aspectos envolvidos. Na Figura 1.31, por exemplo, o agrupamento por proximidade introduzido em (c) aproxima excessivamente as condições de participação do bloco de inscrição, exigindo novo ajuste em (d). Esse processo de avaliação e reprojeto é o mesmo, seja o artefato produzido manualmente, com ferramentas convencionais ou com apoio de Inteligência Artificial.

A Tabela 1.3 detalha as intervenções de cada etapa da Figura 1.31 e os princípios relacionados. As modificações são cumulativas, mas não constituem uma sequência aplicável a qualquer artefato. Elas exemplificam como problemas identificados na avaliação podem orientar decisões de conteúdo, tipografia, alinhamento, agrupamento e hierarquia, que precisam ser novamente avaliadas. Outras decisões poderiam responder aos mesmos problemas, desde que igualmente avaliadas e fundamentadas.

1.8 Verificação da acessibilidade deste capítulo

A acessibilidade foi considerada na produção deste capítulo, mas sua implementação exigiu revisar escolhas inicialmente adotadas e verificar a estrutura efetivamente produzida no PDF. Esse processo resultou em um documento marcado, com estrutura semântica e ordem de leitura, e incluiu:

- incorporação de alternativas textuais às figuras, elaboradas para comunicar seu conteúdo e, quando relevante, características da composição necessárias à compreensão dos exemplos;
- substituição do ambiente personalizado de quadros pelo ambiente padrão de tabelas, após a verificação indicar que o primeiro introduzia conteúdo não incorporado à estrutura marcada do PDF, e identificação semântica das células de cabeçalho;
- tratamento de elementos gráficos exclusivamente visuais como artefatos, para que não interfiram na estrutura destinada às tecnologias assistivas;
- tratamento das siglas para preservar sua apresentação visual e fornecer uma representação adequada à leitura por tecnologias assistivas;
- prevenção de hifenizações de palavras que prejudicassem a leitura pelo leitor de tela, e uso das formas por extenso em lugar de abreviaturas comuns.

O PDF foi verificado com o PAC e o veraPDF [axes4 2026, veraPDF 2026]. No veraPDF 1.30.2, o documento passou na validação do perfil PDF/UA-2 sem falhas. No PAC 26.1.0.0, as 31 alternativas textuais das figuras foram consideradas válidas.

Tabela 1.3: Intervenções realizadas nas etapas de reprojeto da Figura 1.31.

Etapa	**Foco**	**Exemplos de modificações**	**Princípios relacionados**
(a)	Conteúdo, legibilidade e contraste	Remoção de “PALESTRA”; substituição da caixa alta por maiúsculas e minúsculas no nome do evento e nas chamadas “Entrada gratuita” e “Vagas limitadas”; aumento do nome do evento de *tiny* para *scriptsize*; redução do logotipo do evento.	*Legibilidade*; *Contraste*; *Relação Sinal-Ruído*.
(b)	Repetição e consistência	Aplicação de negrito a “Entrada gratuita” e “Vagas limitadas”, mantendo o mesmo tratamento tipográfico para as duas condições de participação; incorporação de “2026” à mesma linha do nome do evento, liberando espaço vertical entre o nome do evento e o título da palestra.	*Repetição*; *Consistência*; *Similaridade*.
(c)	Alinhamento e agrupamento	Reunião de “Entrada gratuita” e “Vagas limitadas” em uma única linha alinhada à esquerda, separadas pelo símbolo “•”; aproximação vertical entre as condições de participação e o bloco de inscrição; aproximação entre “Inscrições”, o endereço eletrônico e o *QR Code*, formando um grupo relacionado à inscrição.	*Alinhamento*; *Proximidade*; *Similaridade*.
(d)	Hierarquia e composição integrada	Redução da altura da fotografia e do logotipo; posicionamento do logotipo junto à margem direita; aumento do título; aplicação de negrito ao nome da palestrante; redistribuição vertical dos blocos; aumento do espaço entre as condições de participação e o bloco de inscrição, corrigindo o agrupamento produzido em (c).	*Hierarquia Visual*; *Destaque*; *Proximidade*; *Relação Sinal-Ruído*.

A verificação de contraste apontou duas ocorrências insuficientes na Figura 1.21, que apresenta deliberadamente essa característica para fins didáticos. O PAC apontou ainda oito ocorrências relacionadas à associação de células de cabeçalho das tabelas, compatíveis com um erro espúrio conhecido do PAC [Fischer 2026], e nove na verificação *Completeness of “Link” elements*. Esta última integra a categoria *Quality*: não constitui um critério técnico de conformidade e sinaliza, por heurística, textos potencialmente vinculáveis para inspeção manual. As nove ocorrências, presentes em figuras e referências bibliográficas, foram inspecionadas e não correspondem a links sem a estrutura adequada.

A verificação funcional foi realizada com o leitor de tela NVDA no Adobe Acrobat Reader. Foram examinadas a ordem de leitura e a navegação estrutural do documento, incluindo a navegação entre figuras e tabelas pelos comandos específicos do NVDA, o acesso às alternativas textuais das figuras e a leitura das siglas [POEU 2026, PDFA 2023].

1.9 Considerações finais

A avaliação de artefatos digitais envolve dimensões distintas e relacionadas. Organização perceptiva, composição visual, legibilidade, compreensão e acessibilidade contribuem para a comunicação da informação, mas sua relevância depende do artefato, do conteúdo, do público, das tarefas e do contexto de uso. Os princípios discutidos oferecem fundamentos para identificar características observáveis, justificar sua avaliação e

examinar requisitos que podem se reforçar ou entrar em tensão. Nesse sentido, a avaliação não se reduz à aplicação de uma lista fixa de princípios: requer selecionar aqueles pertinentes ao artefato e considerar conjuntamente seus efeitos sobre a comunicação.

O *card* fictício apresentado no início do capítulo e retomado na Figura 1.31 ilustra esse processo. As intervenções da Tabela 1.3 mostram que o reprojeto não consiste apenas em corrigir problemas isolados: cada modificação altera o artefato e pode produzir novos efeitos. O processo é, portanto, iterativo: identificar um problema, modificar o artefato e avaliar o resultado pode revelar novas necessidades ou relações que não eram evidentes na versão anterior.

Mais do que classificar um *card*, *slide*, página Web ou outro artefato digital como "bom" ou "ruim", interessa identificar o que favorece ou prejudica a comunicação, para quem, em que condições e com que fundamento.

1.10 Cuidados éticos

Os exemplos foram elaborados para fins didáticos sem uso de material nem dados reais. O capítulo não envolveu dados pessoais nem pesquisa com seres humanos.

1.11 Agradecimentos

Ferramentas de Inteligência Artificial Generativa apoiaram a redação. As autoras são responsáveis pela seleção das fontes, análise e decisões conceituais e redação final.

A Princípios para avaliação de artefatos digitais

Este apêndice sintetiza os princípios e conceitos utilizados no capítulo. Sua pertinência depende do artefato, do público, da tarefa e do contexto de uso.

Alinhamento (*Alignment*) O alinhamento cria relações visuais entre elementos e contribui para a organização e a coesão da composição [Lidwell et al. 2023, 006] [Williams 2015, 33–54].

Associações Número-Espaço (*Number-Space Associations*) Números e sequências podem ser associados mentalmente a organizações espaciais relacionadas a magnitude, ordem, orientação e convenções culturais [Lidwell et al. 2023, 127].

Carga de Desempenho (*Performance Load*) Quanto maior o esforço cognitivo ou físico exigido para realizar uma tarefa, maior tende a ser o impacto sobre seu desempenho [Lidwell et al. 2023, 135].

Compreensível (*Understandable*) As informações e a operação da interface devem ser compreensíveis [W3C 2024, Seção 3].

Conectividade Uniforme (*Uniform Connectedness*) Elementos visualmente conectados, por exemplo por linhas ou por uma região comum, tendem a ser percebidos como pertencentes ao mesmo grupo [Lidwell et al. 2023, 189].

Confirmação (*Confirmation*) Ações importantes ou potencialmente prejudiciais devem ser verificadas antes de sua execução [Lidwell et al. 2023, 033].

Consistência (*Consistency*) Elementos semelhantes devem apresentar aparência e comportamento coerentes, favorecendo previsibilidade e aprendizagem [Lidwell et al. 2023, 035].

Continuidade / Boa Continuação (*Continuity / Good Continuation*) A percepção tende a organizar elementos de modo que linhas, curvas e trajetórias sejam percebidas como contínuas [Lidwell et al. 2023, 084].

Contraste (*Contrast*) Diferenças visuais suficientemente fortes permitem distinguir, organizar e enfatizar elementos de uma composição [Williams 2015, 69–84].

Convenções (*Conventions*) Convenções estabelecidas criam expectativas sobre o significado, a localização ou o comportamento de elementos e podem tornar a interação mais previsível [Krug 2014, 29–33].

Design Centrado no Usuário vs. Design Dirigido pelo Usuário (*User-Centered vs. User-Driven Design*) Projetar com foco no usuário requer compreender suas necessidades e problemas, e não simplesmente implementar literalmente as soluções que ele solicita [Lidwell et al. 2023, 190].

Destaque (*Highlighting*) Recursos de destaque direcionam a atenção para informações importantes, mas perdem eficácia quando utilizados em excesso [Lidwell et al. 2023, 091].

Destino Comum (*Common Fate*) Elementos que se movem ou mudam de maneira semelhante tendem a ser percebidos como pertencentes ao mesmo grupo [Lidwell et al. 2023, 031].

Divulgação Progressiva (*Progressive Disclosure*) Apenas as informações e os controles necessários em determinado momento são apresentados, mantendo os demais disponíveis para revelação progressiva [Lidwell et al. 2023, 147].

Efeito Estética-Usabilidade (*Aesthetic-Usability Effect*) Projetos esteticamente agradáveis tendem a ser percebidos como mais fáceis de usar, mesmo quando essa percepção não corresponde à usabilidade objetiva [Lidwell et al. 2023, 004].

Efeito IKEA (*IKEA Effect*) Pessoas que participam da criação de um artefato tendem a atribuir a ele valor maior, o que pode interferir em sua própria avaliação [Lidwell et al. 2023, 096].

Efeito Poste de Luz (*Streetlight Effect*) Há tendência a procurar ou analisar informação onde ela é mais fácil de obter, mesmo quando essa informação não é a mais relevante para o problema [Lidwell et al. 2023, 177].

Efeito von Restorff (*von Restorff Effect*) Um elemento distinto dos demais tende a receber mais atenção e a ser mais facilmente lembrado [Lidwell et al. 2023, 194].

Efeitos da Cor (*Color Effects*) A cor pode influenciar a percepção, a atenção, a interpretação e a resposta aos elementos de um artefato [Lidwell et al. 2023, 029].

Efeitos de Interferência (*Interference Effects*) Processos mentais concorrentes ou informações incongruentes podem interferir na execução de uma tarefa, aumentando esforço e possibilidade de erro [Lidwell et al. 2023, 098].

Elimine Palavras Desnecessárias (*Omit Needless Words*) Palavras que não contribuem para a mensagem podem ser eliminadas, desde que sua remoção não prejudique clareza ou compreensão [Krug 2014, 48–53].

Equilíbrio (*Balance*) O equilíbrio considera a distribuição do sinal visual em relação a um eixo, sem exigir uma disposição simétrica [Gordon 2020].

Escala (*Scale*) Diferenças de tamanho relativo podem sinalizar importância e posição dos elementos em uma composição [Gordon 2020].

Fechamento (*Closure*) A percepção tende a completar formas familiares mesmo quando parte da informação visual está ausente [Lidwell et al. 2023, 027].

Figura e Fundo (*Figure-Ground*) A percepção organiza o campo visual distinguindo elementos percebidos como figura daqueles percebidos como fundo [Lidwell et al. 2023, 067].

Habituação (*Habituation*) A resposta a um estímulo tende a diminuir com sua exposição repetida, reduzindo a eficácia de recursos de destaque usados continuamente [Lidwell et al. 2023, 087].

Hierarquia de Necessidades (*Hierarchy of Needs*) Necessidades fundamentais de um projeto, como funcionalidade, confiabilidade e usabilidade, devem ser atendidas antes que aspectos de níveis superiores sejam priorizados [Lidwell et al. 2023, 090].

Hierarquia Visual (*Visual Hierarchy*) Diferenças visuais estabelecem níveis de importância e tornam perceptíveis relações de proeminência e subordinação entre os elementos de uma composição [Gordon 2020] [Krug 2014, 33–36].

Horror ao Vazio (*Horror Vacui*) A tendência a preencher espaços vazios pode produzir composições excessivamente densas e dificultar a percepção da organização da informação [Lidwell et al. 2023, 092].

Legibilidade (*Legibility*) Características visuais do texto determinam a facilidade com que caracteres, palavras e blocos de texto são percebidos e distinguidos [Lidwell et al. 2023, 107].

Lei de Fitts (*Fitts's Law*) O tempo necessário para atingir um alvo depende principalmente da distância até ele e de seu tamanho [Lidwell et al. 2023, 069].

Leiturabilidade (*Readability*) Características linguísticas e estruturais do conteúdo influenciam a facilidade de compreensão de um texto [Lidwell et al. 2023, 153].

Linha de Desejo (*Desire Line*) Os caminhos efetivamente escolhidos pelas pessoas podem revelar formas de uso ou interação preferidas, inclusive quando diferem daquelas previstas no projeto [Lidwell et al. 2023, 048].

Mapeamento (*Mapping*) Relações coerentes entre controles, informações, movimentos e seus efeitos facilitam a compreensão de como um artefato funciona [Lidwell et al. 2023, 113].

Modelo Mental (*Mental Model*) As pessoas constroem representações internas sobre como sistemas, objetos e processos funcionam e utilizam essas representações para interpretar e prever seu comportamento [Lidwell et al. 2023, 116].

Narrativa (*Storytelling*) A organização de informações em uma sequência significativa de eventos pode favorecer contexto, compreensão e envolvimento [Lidwell et al. 2023, 176].

Navalha de Ockham (*Ockham's Razor*) Entre alternativas funcionalmente equivalentes, deve-se preferir a solução mais simples [Lidwell et al. 2023, 128].

O Que Você Vê É o Que Você Obtém (*WYSIWYG*) A representação utilizada durante a criação deve corresponder, tanto quanto possível, ao resultado que será apresentado ao usuário [Lidwell et al. 2023, 199].

Operável (*Operable*) Os componentes da interface e a navegação devem ser operáveis [W3C 2024, Seção 2].

Perceptível (*Perceivable*) As informações e os componentes da interface devem ser apresentados de formas que possam ser percebidas [W3C 2024, Seção 1].

Prägnanz / Boa Forma (*Good Form*) Quando um estímulo admite diferentes organizações perceptivas, há tendência a percebê-lo segundo uma configuração simples, regular e estável [Ellis 1938, 148, 352].

Proporção Áurea (*Golden Ratio*) A proporção de aproximadamente 1:1,618 pode ser utilizada como referência para estabelecer relações dimensionais em uma composição [Lidwell et al. 2023, 083].

Proximidade (*Proximity*) Elementos espacialmente próximos tendem a ser percebidos como relacionados ou pertencentes ao mesmo grupo [Lidwell et al. 2023, 152] [Williams 2015, 15–32].

Reconhecimento em Vez de Recordação (*Recognition Over Recall*) Reconhecer informações apresentadas por meio de pistas geralmente exige menos esforço de memória do que recuperá-las sem auxílio [Lidwell et al. 2023, 155].

Redundância (*Redundancy*) O uso de mecanismos ou formas de representação complementares pode aumentar a confiabilidade da comunicação e reduzir a dependência de um único meio de transmitir informação [Lidwell et al. 2023, 156].

Região Comum (*Common Region*) Elementos delimitados por uma mesma região tendem a ser percebidos como pertencentes ao mesmo grupo. Essa forma de agrupamento é discutida por Lidwell et al. no princípio *Uniform Connectedness* [Lidwell et al. 2023, 189].

Regra do Pico-Fim (*Peak-End Rule*) A avaliação retrospectiva de uma experiência tende a ser influenciada de maneira desproporcional por seus momentos mais intensos e por seu final [Lidwell et al. 2023, 134].

Regra dos Terços (*Rule of Thirds*) A divisão da área visual por uma grade de três por três pode orientar o posicionamento de elementos relevantes ao longo das linhas e de suas interseções [Lidwell et al. 2023, 160].

Relação Sinal-Ruído (*Signal-to-Noise Ratio*) A comunicação é favorecida quando a informação relevante é maximizada em relação aos elementos irrelevantes ou que interferem em sua percepção [Lidwell et al. 2023, 170].

Repetição (*Repetition*) A repetição sistemática de características visuais contribui para estabelecer consistência, identidade e relações entre elementos [Williams 2015, 55–68].

Representação Icônica (*Iconic Representation*) Imagens e símbolos podem representar objetos, ações ou conceitos por semelhança, exemplo, associação ou convenção [Lidwell et al. 2023, 094].

Restrição (*Constraint*) Restrições adequadas reduzem o conjunto de ações possíveis e podem prevenir erros [Lidwell et al. 2023, 037].

Robusto (*Robust*) O conteúdo deve ser suficientemente robusto para ser interpretado de forma confiável por diferentes agentes de usuário, incluindo tecnologias assistivas [W3C 2024, Seção 4].

Sensibilidade à Orientação (*Orientation Sensitivity*) A orientação de linhas e outros elementos visuais influencia a facilidade com que podem ser detectados e discriminados [Lidwell et al. 2023, 130].

Similaridade (*Similarity*) Elementos visualmente semelhantes tendem a ser percebidos como relacionados ou pertencentes ao mesmo grupo [Lidwell et al. 2023, 171].

Subtração Progressiva (*Progressive Subtraction*) Elementos podem ser removidos iterativamente até o ponto em que uma nova remoção prejudicaria a função do projeto [Lidwell et al. 2023, 148].

Teoria da Cor (*Color Theory*) Relações entre cores podem ser utilizadas para produzir contraste, harmonia, agrupamento e significado [Lidwell et al. 2023, 030].

Tolerância a Erros (*Forgiveness*) Um projeto deve prevenir erros quando possível e permitir recuperação com consequências mínimas quando eles ocorrerem [Lidwell et al. 2023, 074].

Referências

[axes4 2026] axes4 (2026). PAC: PDF Accessibility Checker. `https://pac.pdf-accessibility.org/`. Acesso em: 10 set. 2026.

[Brasil, Departamento de Governo Eletrônico 2014] Brasil, Departamento de Governo Eletrônico (2014). eMAG: Modelo de Acessibilidade em Governo Eletrônico. `https://emag.governoeletronico.gov.br/`. Acesso em: 19 set. 2026.

[CUP 2026] CUP (2026). Gestalt. Cambridge Dictionary, Cambridge University Press

& Assessment. Acesso em: 20 set. 2026.

[Dix et al. 2004] Dix, A., Finlay, J., Abowd, G. D., and Beale, R. (2004). *Human-Computer Interaction*. Prentice Hall, 3 ed.

[Ellis 1938] Ellis, W. D., editor (1938). *A Source Book of Gestalt Psychology*. Routledge & Kegan Paul, London. Disponível no Internet Archive. Acesso em: 19 set. 2026.

[Fischer 2026] Fischer, U. (2026). [PAC] table header scope is not recognized. LaTeX Project, GitHub Discussions, 24 abr. 2026.

[Gleason 2020] Gleason, C. (2020). *Accessible User-Generated Social Media for People with Vision Impairments*. PhD thesis, Carnegie Mellon University, Pittsburgh.

[Göbel 2018] Göbel, S. (2018). How Culturally Predominant Reading Direction and Number Word Construction Influence Numerical Cognition and Mathematics Performance. In *Language and Culture in Mathematical Cognition*, pp 229–256. AP.

[Gordon 2020] Gordon, K. (2020). 5 principles of visual design in ux. https://www.nngroup.com/articles/principles-visual-design/. Nielsen Norman Group.

[Krug 2014] Krug, S. (2014). *Don't Make Me Think, Revisited: A Common Sense Approach to Web and Mobile Usability*. New Riders, 3 ed.

[Lidwell et al. 2023] Lidwell, W., Holden, K., and Butler, J. (2023). *Universal Principles of Design*. Rockport Publishers, 3 ed.

[Nardi 2021] Nardi, C. C. d. S. (2021). *Diretrizes para Produção de Alternativas ao Conteúdo Visual em Mídias Sociais Online sob a Perspectiva de Pessoas com Deficiência Visual*. PhD thesis, UFRJ, Rio de Janeiro.

[PDFA 2023] PDFA (2023). Tagged PDF Best Practice Guide: Syntax. https://pdfa.org/resource/tagged-pdf-best-practice-guide-syntax/. Originally published in 2019; updated July 26, 2023.

[POEU 2026] POEU (acesso em setembro de 2026). Testing Publications with NVDA: PDFs. https://op.europa.eu/en/web/accessibility/transcript-testing-publications-with-nvda-pdfs.

[veraPDF 2026] veraPDF (2026). veraPDF REST Client, veraPDF Consortium. `https://demo.verapdf.org/`. Acesso em: 29 set. 2026.

[W3C 2024] W3C (2024). World Wide Web Consortium, Web Content Accessibility Guidelines (WCAG) 2.2. `https://www.w3.org/TR/WCAG22/`. W3C Recommendation, 12 December 2024. Acesso em: 19 set. 2026.

[W3C 2026] W3C (2026). Complex Images, Web Accessibility Initiative (WAI), World Wide Web Consortium. https://www.w3.org/WAI/tutorials/images/complex/.

[White et al. 2026] White, K., Abou-Zahra, S., and Henry, S. L. (2026). Planning and Managing Web Accessibility. Atualizado 12/08/2026. Acesso em: 19 set. 2026.

[Williams 2015] Williams, R. (2015). *The Non-Designer's Design Book*. Peachpit, 4 ed.

Chapter 2

Computação Física Inclusiva com Micro:bit

Mariana Arfux Pereira Cavalcante de Castro, Bruna C. R. Cunha

Abstract

Assistive Technology (AT) is fundamental to inclusion, necessitating the development of new solutions. While physical AT systems offer benefits, options such as Arduino present barriers for beginners. In this context, the Micro:bit and the MakeCode platform serve as accessible tools for inclusive methodologies. This chapter presents a theoretical-practical approach grounded in Human-Computer Interaction (HCI) focused on accessibility, Universal Design for Learning (UDL), and the board's ecosystem. Furthermore, it provides a guide for creating prototypes with multimodal feedback, demonstrating ways to overcome accessibility barriers in educational settings and interface development. Notably, projects designed for autistic students are presented, fostering their active role in creating their own AT resources.

Resumo

A Tecnologia Assistiva (TA) é fundamental na inclusão, exigindo o desenvolvimento de novas soluções. Embora sistemas físicos de TA tragam benefícios, opções como o Arduino impõem barreiras a iniciantes. Nesse contexto, o Micro:bit e a plataforma MakeCode são ferramentas acessíveis para metodologias inclusivas. Este capítulo apresenta uma abordagem teórico-prática fundamentada em Interação Humano-Computador (IHC) voltada à acessibilidade, Design Universal para a Aprendizagem (DUA) e no ecossistema da placa. Ademais, traz um guia de criação de protótipos com feedbacks multimodais, demonstrando opções para superar barreiras de acessibilidade no contexto educacional e no desenvolvimento de interfaces. Em especial, apresentam-se projetos para estudantes autistas, promovendo o protagonismo na criação de seus próprios recursos de TA.

2.1. Introdução à Computação Física e Acessibilidade Digital

A democratização do acesso às tecnologias digitais tem sido pauta de políticas públicas, produções científicas e estratégias de inclusão social [1, 2, 3]. Nesse cenário, o Pensamento Computacional (PC) e o letramento digital consolidam-se como competências

fundamentais para o pleno exercício da cidadania na sociedade contemporânea [2]. Essa relevância é impulsionada por marcos regulatórios e diretrizes nacionais, como as atualizações da Base Nacional Comum Curricular - Computação (BNCC - Computação) (2022) e a instituição da Política Nacional de Educação Digital (PNED, Lei n°14.533, que estabelecem a inclusão tecnológica como um direito estruturante) [1].

Entretanto, para auxiliar na compreensão dos conceitos abstratos sobre desenvolvimento de sistemas, tornam-se indispensáveis abordagens que medeiem a interação entre o humano e o computador. A Computação Física surge nesse cenário como um paradigma que conecta o mundo digital (*software*) ao mundo físico (*hardware*) [7]. Quando aliada aos princípios da Acessibilidade Digital e às diretrizes da Lei Brasileira de Inclusão da Pessoa com Deficiência (LBI) (Estatuto da Pessoa com Deficiência, Lei n°13.146/2015) [10], essa abordagem é uma possibilidade de transformar objetos concretos em interfaces tangíveis, permitindo a criação de tecnologias e produtos acessíveis.

Essa integração ampara-se no arcabouço normativo de acessibilidade, a exemplo do Cap. III, Art. 8°das Leis n°10.048/2000 e n°10.098/2000 [11], que fundamenta as barreiras na comunicação e na informação. Ao incorporar conceitos como desenho universal e o desenvolvimento de ajudas técnicas (Tecnologias Assistivas), garante-se a autonomia e a participação equitativa de pessoas com diferentes perfis sensoriais, motores ou cognitivos no manuseio de sistemas computacionais.

No âmbito de projeto e prototipagem, o ambiente do Micro:bit pode ser um exemplo de como ferramentas de prototipagem rápidas podem incorporar esses preceitos de *Design* Inclusivo. A plataforma disponibiliza recursos projetados para garantir a acessibilidade de usuários cegos ou com baixa visão, tais como diagramas táteis detalhados do hardware e descrições técnicas completas em texto para leitura de tela, além de possibilidades de alto contraste, leitura de tela e mudança de idiomas na plataforma de programação *MakeCode*. Tais estratégias demonstram como a integração entre hardware acessível, multimodal e boas práticas de Interação Humano-Computador (IHC) viabilizam a superação de barreiras comunicacionais, expandindo o desenvolvimento de interfaces e ecossistemas tecnológicos inclusivos.

2.1.1. Interfaces Tangíveis e Computação Física na Inclusão

As Interfaces Tangíveis de Usuário (TUI - *Tangible User Interfaces*) permitem que a manipulação de dados digitais ocorram através da interação direta com objetos físicos. Para estudantes com deficiência visual, neurodivergências ou limitações motoras, o suporte tátil e multissensorial proporcionado pelas TUIs oferecem um canal intuitivo de *feedback*, reduzindo a carga cognitiva e tornando conceitos abstratos de programação mais concretos e palpáveis [5].

A Computação Física atua, portanto, como uma ponte para a criação de Recursos de Tecnologia Assistiva (RTA). Segundo a legislação brasileira (Lei n°13.146/2015) [10], a TA compreende todo o arsenal de recursos, equipamentos, dispositivos e serviços que objetivam promover a funcionalidade, a autonomia e a inclusão de pessoas com deficiência. Nesse sentido, a prototipagem eletrônica acessível possibilita que os próprios alunos, seus educadores e pesquisadores, construam soluções customizadas para superar barreiras de acessibilidade do cotidiano escolar [4].

2.1.2. O Panorama do Ensino de Tecnologia

No âmbito do ensino do Pensamento Computacional na educação básica, a plataforma Scratch é um ambiente muito utilizado devido a sua gratuidade e experiência visual. Considerando soluções de hardware, o Arduino é outra ferramenta muito adotada, neste caso, em diferentes níveis de ensino.

- **Scratch:** Plataforma de programação baseada em blocos visuais desenvolvida pelo MIT [1]. Suas principais contribuições incluem a eliminação do erro sintático típico das linguagens baseadas em texto e uma interface altamente lúdica. No entanto, em sua aplicação pura, o Scratch permanece restrito à tela do computador. Ademais, o formato exclusivo em blocos, ao mesmo tempo que simplifica conceitos de forma visual, inclui barreiras de interação para pessoas cegas ou com baixa visão.
- **Arduino:** Placa de prototipagem eletrônica de código aberto (*open-source*) focada na criação de circuitos customizáveis [2]. O Arduino democratizou a robótica educacional ao permitir o uso de uma vasta gama de sensores e atuadores externos. Contudo, seu uso apresenta desafios significativos de acessibilidade: a necessidade de montagem de circuitos complexos em matrizes de contato (*protoboards*), o manuseio de componentes minúsculos e o suporte nativo predominante a linguagens baseadas em texto (C/C++) impõem barreiras elevadas para iniciantes, estudantes com limitações motoras finas ou deficiência visual.

2.1.3. Justificativa e Vantagens do Ecossistema Micro:bit

Diante das limitações apresentadas pelo uso isolado do Scratch ou da complexidade do Arduino, o ecossistema Micro:bit posiciona-se como uma opção de Computação Física com sensores acoplados que facilitam a interação e introdução dos conceitos abstratos computacionais [12]. O Micro:bit integra o melhor de ambos os mundos: a simplicidade e acessibilidade das linguagens visuais com a tangibilidade dos dispositivos físicos, apresentando vantagens estratégicas:

- **Hardware:** Diferente do Arduino, que exige a fiação de periféricos externos para tarefas básicas, o Micro:bit V2 possui atuadores (matriz de LEDs, alto-falante) e sensores (acelerômetro, bússola, luz, temperatura, microfone e toque) integrados na própria placa. Isso elimina a dependência inicial de coordenação motora fina para montagem de circuitos em *protoboard*.
- **Conectividade e *Feedback* Multimodal Nativo:** A presença simultânea de saídas visuais (LEDs) e auditivas (alto-falante nativo) possibilita a entrega imediata de *feedbacks* multissensoriais, contornando limitações sensoriais específicas de cada estudante.
- **Ecossistema de Programação Híbrido:** A plataforma *MakeCode* oferece suporte nativo à transição entre blocos visuais, *JavaScript* e *Python*, incorporando desde o início diretrizes globais de acessibilidade para leitores de tela e navegação via teclado. Além de oferecer um aumento gradual sobre linguagens de programação.

[1] https://scratch.mit.edu

[2] https://www.arduino.cc

2.1.4. Mapeamento de Barreiras e o *Design* Universal para a Aprendizagem (DUA)

Os ambientes tradicionais de desenvolvimento de software frequentemente falham ao desconsiderar a diversidade humana, impondo barreiras sensoriais (falta de leitores de tela ou alto contraste), motoras (dependência exclusiva do mouse) e cognitivas (sintaxe complexa e *feedback* abstrato). Para superar essas limitações, este trabalho fundamenta-se nos princípios do *Design* Universal para a Aprendizagem (DUA), uma abordagem pedagógica voltada ao desenvolvimento de ambientes de ensino acessíveis a todos os estudantes [13]. O DUA estrutura-se em três pilares fundamentais, os quais encontram no Micro:bit uma aplicação direta:

1. **Múltiplos Meios de Engajamento (O "Porquê" da Aprendizagem):** O Micro:bit permite a criação de projetos contextualizados e significativos, mantendo o interesse dos alunos através da manipulação de um objeto físico e reativo.
2. **Múltiplos Meios de Representação (O "O quê" da Aprendizagem):** A informação pode ser apresentada de formas variadas texto na tela, ícones na matriz de LEDs, tons sonoros ou respostas táticas , garantindo que estudantes com limitações sensoriais processem o conteúdo de forma equivalente.
3. **Múltiplos Meios de Ação e Expressão (O "Como" da Aprendizagem):** Os estudantes podem demonstrar o aprendizado programando via blocos, texto ou interagindo fisicamente com a placa (chacoalhando, tocando no logotipo, emitindo sons), respeitando os diferentes estilos e capacidades motoras.

Dessa forma, a integração da computação física pautada pelo DUA e suportada pelo ecossistema Micro:bit não apenas mitiga as barreiras dos ambientes tradicionais, mas consolida a tecnologia como um meio de emancipação e inclusão social.

2.2. O Ambiente do Micro:bit sob a Ótica da Inclusão

Nessa seção será apresentada a análise técnica da placa Micro:bit V2, destacando seus sensores e atuadores nativos como canais de *feedback* multimodal.

O desenvolvimento de práticas pedagógicas inclusivas no ensino de computação e robótica educacional exige ferramentas que contemplem a diversidade dos estudantes. Nesse cenário, a placa Micro:bit V2 é um recurso, que promove a acessibilidade física, sensorial e cognitiva por meio da integração entre hardware e o ambiente de programação.

2.2.1. Análise Técnica da Placa

A arquitetura do Micro:bit foi projetada para atuar como uma plataforma de interação multissensorial. A combinação de seus sensores nativos e atuadores integrados permite que a aprendizagem ocorra por diferentes canais perceptivos (visual, auditivo e tátil/proprioceptivo), o que é fundamental para estudantes com deficiências sensoriais ou neurodivergências.

Além disso, o Micro:bit em sua segunda versão (V2), possui adicionais, em relação à sua versão um, de sensores possibilitando mais interações com a placa[3], como o

[3]https://microbit.org/accessibility/full-text-tour/

Logotipo Sensível ao Toque, LED do microfone, o próprio microfone, alto-falante, LED vermelho de energia, LED amarelo do USB e o botão de economia de bateria, todos esses elementos, foram adicionados na placa V2 que é o modelo utilizado e explicitado no capítulo. A Figura 2.1 ilustra a interface frontal do Micro:bit V2, seus componentes e respectivas descrições.

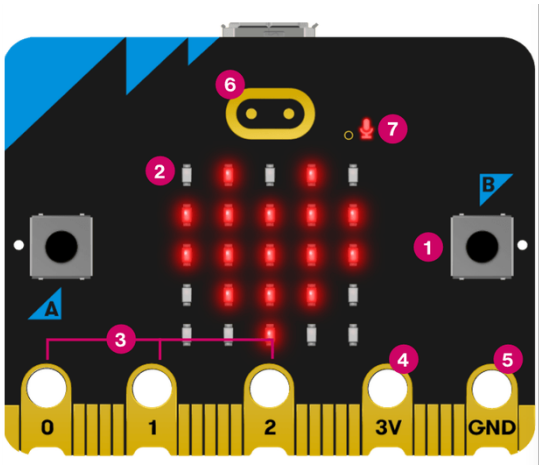


Figure 2.1. Frente da placa Micro:bit V2.

Componentes da parte frontal:

1. **Botões de Entrada (A e B):** Botões físicos de fácil acionamento que oferecem *feedback* tátil mecânico.
2. **Matriz de LEDs (5x5) e Sensor de Luz:** Atua como um *display* visual básico para matrizes, ícones, letras e animações, servindo simultaneamente para mensurar níveis de luminosidade como um sensor de luz.
3. **Pinos GPIO (*General Purpose Input/Output*):** Pinos de entrada e saída de fácil acesso para expansão de circuitos.
4. **Pino de Alimentação (3V):** Fornece tensão para componentes externos.
5. **Pino de Terra (GND):** Ponto de fechamento de circuito.
6. **Logotipo Sensível ao Toque:** Botão sensível ao toque, expandindo as opções de interação tátil.
7. **Microfone Integrado e LED Indicador:** Canal de entrada acústica com LED indicatório para *feedback* visual sobre a captação de som.

Complementarmente, os componentes localizados na parte de trás da placa (Figura 2.2) estão descritos abaixo.

1. **Antena de Rádio e Bluetooth:** Comunicação sem fio para redes de placas e conectividade móvel.

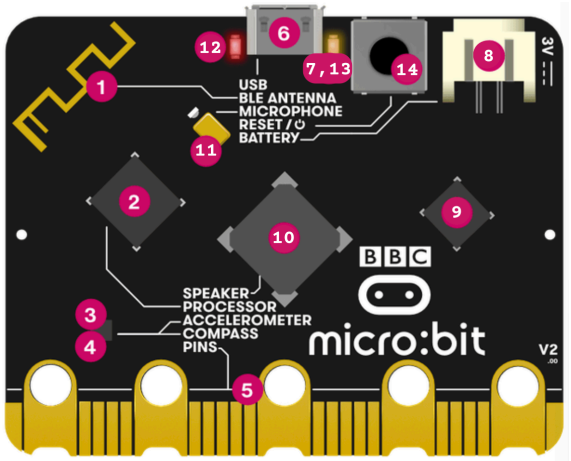


Figure 2.2. Trás da placa Micro:bit V2.

2. **Processador e Sensor de Temperatura:** Unidade central de processamento com termômetro integrado.

3. **Bússola:** Detecção de campos magnéticos e orientação espacial.

4. **Acelerômetro:** Medição de movimento, inclinação e aceleração em três eixos.

5. **Pinos de Conexão Traseiros:** Acesso simplificado à interface de expansão.

6. **Porta Micro USB:** Interface para programação e alimentação de energia.

7. **LED Amarelo de Transferência:** *Feedback* visual durante o carregamento do código.

8. **Conector da Bateria:** Permite o uso autônomo e portátil da placa.

9. **Interface USB:** Gerencia a comunicação entre o computador e o processador principal.

10. **Alto-falante Integrado:** Emissor de áudio nativo para *feedback* sonoro (músicas, tons e alarmes).

11. **Microfone (Face Posterior):** Captador de áudio e pressão sonora ambiental.

12. **LED Vermelho de Alimentação:** Indicação visual de placa energizada.

13. **LED Amarelo da Interface USB:** Indicador do status da conexão USB.

14. **Botão Multifunção de Energia e *Reset*:** Permite colocar a placa em modo de baixo consumo ou reiniciar o programa.

2.2.2. A Plataforma *MakeCode*

A plataforma de programação utilizada é o *MakeCode*, que possui algumas características que fornecem suporte e acessibilidade ao usuário. Como por exemplo:

- **Suporte a Leitores de Tela:** Estruturação de código e blocos compatíveis com softwares de leitura de tela, permitindo que estudantes com deficiência visual naveguem pelos blocos de código e instruções.
- **Modo de Alto Contraste:** Opção de interface gráfica com paletas ajustadas para usuários com baixa visão ou daltonismo.
- **Flexibilidade na Programação:** Permite a transição gradual e adaptável do nível de abstração. O ambiente possibilita transitar sem perdas entre a linguagem baseada em blocos visuais, *JavaScript* e *Python*, adequando-se ao nível de desenvolvimento e às preferências de interação de cada aluno.
- **Materiais Complementares:** o *MakeCode* possui guias de aplicação, diagramas táteis [4] e tutoriais que possuem uma linguagem de fácil compreensão para qualquer público-alvo.

2.2.3. Introdução ao *MakeCode*

Para viabilizar a criação de protótipos inclusivos, a plataforma Web *MakeCode* [5] oferece um ambiente amigável e acessível de programação baseada em blocos e texto. A seguir, detalham-se os passos fundamentais para utilização do ambiente:

2.2.3.1. Criando e Nomeando um Novo Projeto

A plataforma MakeCode disponibiliza uma interface para a construção, simulação e transferência de códigos para a placa Micro:bit. Para fundamentar a aplicação das atividades propostas neste capítulo, esta subseção apresenta o fluxo inicial de navegação no ambiente, detalhando os procedimentos para criação de novos projetos.

Na página inicial do MakeCode (Figura 2.3), deve-se selecionar a opção "Novo Projeto". Em seguida, uma janela de configuração é exibida (Figura 2.4), solicitando a inserção do nome da atividade (por exemplo, *Teste*). Após clicar em "Criar", a área de trabalho do projeto é carregada. O nome atribuído ao código permanece visível e pode ser editado a qualquer momento no campo de texto localizado na barra inferior da tela, conforme ilustrado na Figura 2.5.

2.2.3.2. Visão Geral da Interface

Após a criação do projeto, o MakeCode carrega a sua área de trabalho principal, conforme ilustrado na Figura 2.6. O ambiente é estruturado de forma intuitiva e dividida em três

[4] https://microbit.org/teach/classroom-resources/microbit-tactile-diagram/
[5] https://makecode.microbit.org/

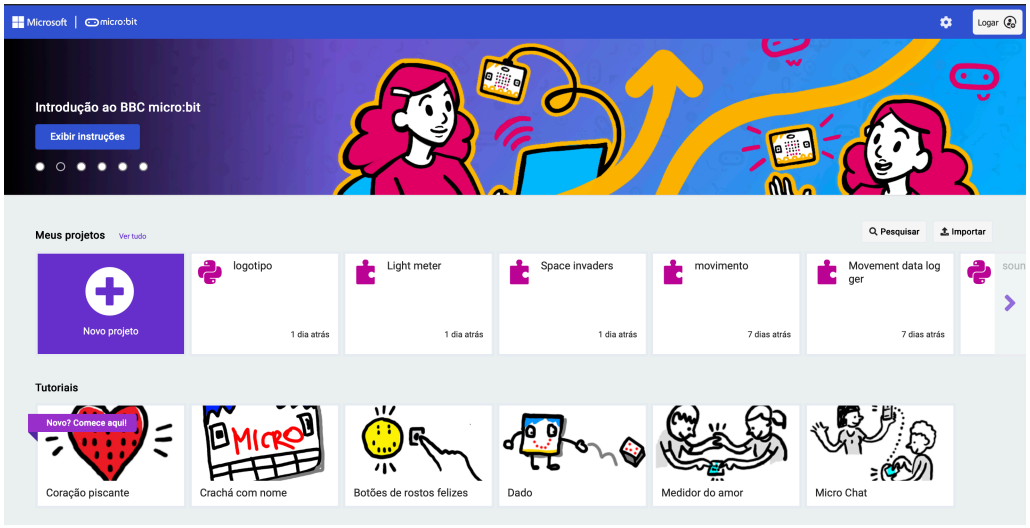


Figure 2.3. Imagem da Página Inicial da Plataforma

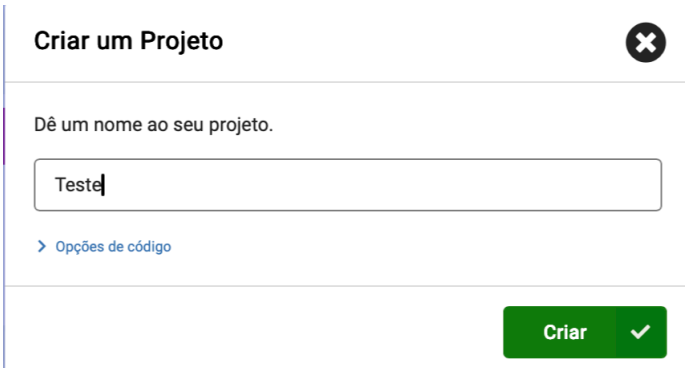


Figure 2.4. Imagem da Janela de Criação da Programação

regiões funcionais suplementares, projetadas para facilitar a visualização e o desenvolvimento dos programas:

- **Simulador (1):** Permite testar o comportamento do código em tempo real diretamente na tela, oferecendo retornos visuais e sonoros (como a execução de sons, matriz de LEDs e interações com os botões) antes de gravar a lógica na placa física.
- **Caixa de Ferramentas / Categorias (2):** Organiza os blocos de programação por funções essenciais (como *Básico*, *Entrada*, *Música*, *LEDs*, *Lógica* e *Variáveis*). Cada categoria conta com codificação por cores distintas, o que facilita a identificação rápida e o reconhecimento visual dos comandos.

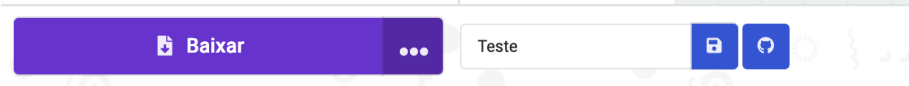


Figure 2.5. Local para Alterar o Nome do Projeto.

- **Área de Trabalho (3):** Espaço central de edição onde os blocos são arrastados, encaixados e estruturados sequencialmente para construir a lógica do programa.

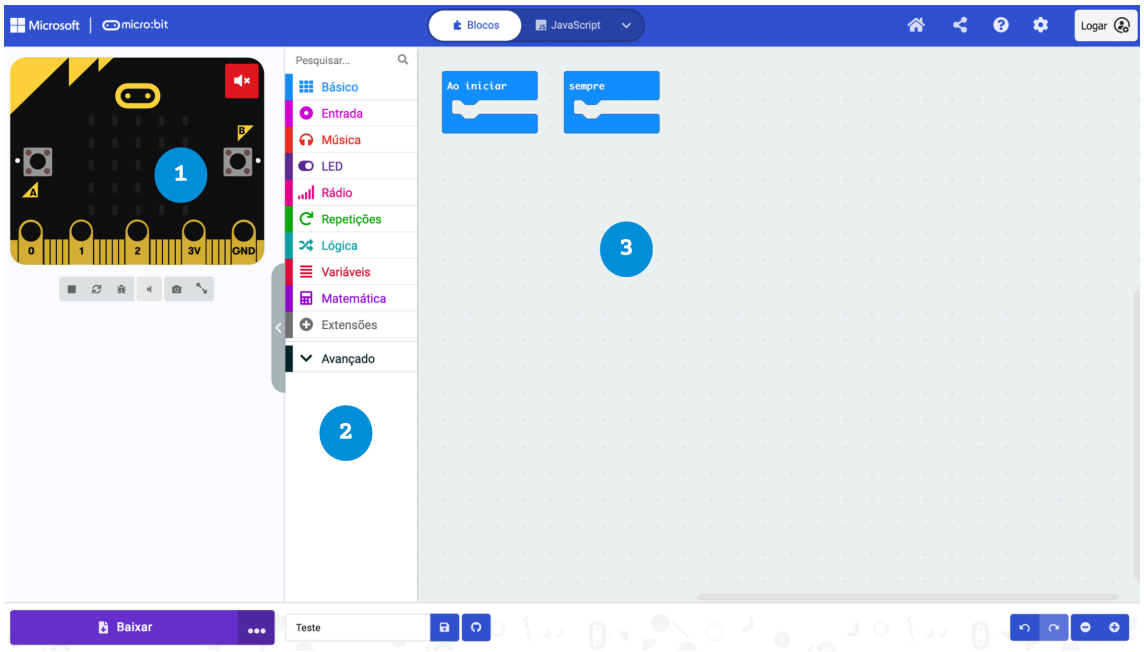


Figure 2.6. Divisão da Interface do *MakeCode*.

2.2.3.3. Transferindo o Código para a Placa Micro:bit (Via Cabo USB):

Após a lógica do programa ser construída e testada no simulador, a etapa seguinte consiste em embarcar o algoritmo no *hardware* físico. A gravação na placa Micro:bit V2 é realizada de forma simplificada via conexão USB, podendo ocorrer por comunicação direta via navegador ou por transferência manual de arquivo, conforme os passos descritos a seguir:

- **Conexão Física:** Conecte a placa Micro:bit ao computador utilizando um cabo Micro-USB adequado para dados.
- **Pareamento Direto (WebUSB):** Clique no botão "Baixar" ou acesse o ícone de três pontos ao lado dele para parear o dispositivo diretamente com o navegador (via suporte WebUSB). Esse procedimento permite que o código seja enviado à placa de forma automática a cada atualização.
- **Transferência Manual (Arquivo *.hex*):** Caso o pareamento direto não esteja ativado, a plataforma gera um arquivo compilado com extensão *.hex*, salvo automaticamente no computador. Para gravar o programa, basta mover ou arrastar o arquivo *.hex* para dentro da unidade de disco nomeada `MICROBIT`, exibida no gerenciador de arquivos do sistema operacional.

- **Confirmação de Gravação:** Durante o processo de transferência, o LED amarelo indicador de interface (representado pelo número 7 da Figura 2.2) piscará rapidamente. Após a conclusão do carregamento, a aplicação iniciará a execução automaticamente no *hardware*.

2.2.3.4. Exportando e Salvando o Arquivo de Programação:

A preservação e o compartilhamento dos artefatos desenvolvidos são etapas essenciais para promover a reusabilidade de projetos e o trabalho colaborativo entre estudantes, educadores e pesquisadores. A plataforma *MakeCode* permite exportar e reimportar arquivos de código de maneira rápida e sem perda de dados, conforme o procedimento descrito a seguir:

- **Exportação do Projeto (*Backup*):** Para salvar uma cópia de segurança no armazenamento local ou compartilhar a aplicação com outros usuários, clique no ícone "Salvar" (representado pelo símbolo de um disco no rodapé da página). Esse comando realizará o download do arquivo compilado em formato *.hex*.
- **Reimportação e Continuidade:** O arquivo *.hex* gerado contém tanto o código executável do *hardware* quanto a estrutura editável de blocos. Para dar continuidade ao desenvolvimento em outro momento ou computador, basta selecionar a opção "Importar" na tela inicial do *MakeCode* e carregar o arquivo previamente salvo.

2.2.4. Propostas de Aplicações Práticas e Sensoriais

A existência de sensores embarcados na placa do Micro:bit V2 permite a construção de artefatos acessíveis e projetos contextualizados de forma simples. Abaixo são destacadas cinco explorações pedagógicas fundamentadas nos sensores da placa tendo o público-alvo, estudantes autistas:

1. **Termômetro Digital (Sensor de Temperatura):** Utiliza o sensor embutido no processador para medir a temperatura ambiente. Permite criar alarmes térmicos sonoros ou visuais para sinalizar variações térmicas de forma tátil e/ou auditiva.
2. **Monitor de Luminosidade (Sensor de Luz):** A matriz de LEDs inverte sua função para mensurar a luz incidente. Permite desenvolver sistemas que podem auxiliar nos avisos sobre sobrecarga sensorial por muita quantidade de luz.
3. **Sonômetro (Microfone Integrado):** Mede a amplitude do som ambiente. Pode ser aplicado na criação de monitores visuais de ruído em sala de aula, convertendo volume sonoro em padrões visuais na matriz de LEDs.
4. **Rastreador de Atividades (Acelerômetro):** Identifica gestos, agitações e inclinação. Utilizado na construção de pedômetros acessíveis, alertas de postura corporal ou interfaces controladas por pequenos movimentos corporais (para alunos com mobilidade reduzida).

5. **Botão Sensível ao Toque (Logotipo Dourado):** Funciona como um atuador capacitivo adicional. Permite acionamentos suaves por proximidade ou toque leve, sendo ideal para criar comutadores acessíveis para estudantes com limitações de força motora fina nos dedos.

2.3. Práticas de Programação Acessível e Design de Atividades Inclusivas

No contexto do Transtorno do Espectro Autista (TEA), as alterações no processamento sensorial, caracterizadas por hiper ou hiporresponsividade a estímulos do ambiente, representam um dos principais fatores desencadeadores de estresse, sobrecarga cognitiva e crises comportamentais no ambiente escolar [15]. Salas de aula tradicionais frequentemente apresentam oscilações imprevisíveis de ruído, luminosidade e temperatura, criando barreiras invisíveis para a permanência e a aprendizagem desses estudantes.

A Computação Física, operacionalizada pelo Micro:bit V2 junto ao MakeCode, atuam como uma ferramenta mediadora e preditiva. Ao converter variáveis ambientais intangíveis em indicadores visuais, sonoros ou táteis claros e padronizados, a placa oferece previsibilidade e concretude ao aluno autista. Dessa forma, a tecnologia deixa de ser apenas um objeto de estudo técnico e passa a funcionar como uma ponte de comunicação não verbal e autorregulação, permitindo que o estudante compreenda o ambiente, comunique seu desconforto de forma autônoma e antecipe intervenções antes do esgotamento sensorial [14].

Nas subseções a seguir, apresenta-se um guia prático "mão na massa" com cinco protótipos desenvolvidos no ambiente MakeCode, detalhando a construção, a lógica de programação e a fundamentação pedagógico-sensorial de cada atividade. Destaca-se que as atividades propostas visam exemplificar como o aluno pode ser colocado como protagonista na implementação de soluções de TA que possam apoiar suas necessidades e desafios causados pelo ambiente. Ideias similares podem ser propostas em outros cenários, por exemplo, ao criar jogos ou recursos de TA para pessoas com outras deficiências e incentivar que alunos proponham e implementem suas próprias soluções.

2.3.1. Protótipo 1: Termômetro Digital Mediador de Conforto Térmico

Fundamentação Sensorial e Pedagógica: Estudantes autistas com hiposensibilidade ou alteração na propriocepção térmica podem apresentar dificuldades em reconhecer e nomear precocemente o próprio desconforto associado ao calor ou ao frio excessivo. Essa dificuldade de autorrelato frequentemente se manifesta como irritabilidade ou desatenção repentina. O Termômetro Digital estabelece um indicador externo e objetivo: ao transformar a temperatura do ambiente em um alerta visual imediato, o protótipo valida a percepção do estudante perante a turma e o professor, justificando ações mitigadoras (como ligar um ventilador ou abrir a janela) antes que o estresse térmico gere uma crise.

Recursos Utilizados: Sensor de Temperatura do Processador e Matriz de LEDs (5x5).

Lógica de Programação:

1. No bloco de repetição contínua (*forever*), realiza-se a leitura do sensor de temperatura interno (em graus Celsius).

2. Condicional I: Se a temperatura for superior a um limiar confortável (ex.: $> 28°C$), a matriz exibe o ícone de alerta ou um símbolo de desconforto, sinalizando a necessidade de readequação do espaço.

3. Condicional II: Caso a temperatura esteja adequada, exibe-se uma carinha feliz ou ícone suave, sinalizando estabilidade térmica.

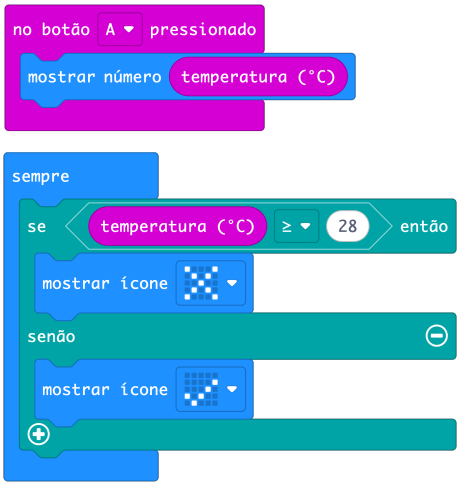


Figure 2.7. Programação em Blocos do Protótipo 1

```
input.onButtonPressed(Button.A, function () {
    basic.showNumber(input.temperature())
})
basic.forever(function () {
    if (input.temperature() >= 28) {
        basic.showIcon(IconNames.No)
    } else {
        basic.showIcon(IconNames.Yes)
    }
})
```

Figure 2.8. Programação em JavaScript do Protótipo 1

```
def on_button_pressed_a():
    basic.show_number(input.temperature())
input.on_button_pressed(Button.A, on_button_pressed_a)

def on_forever():
    if input.temperature() >= 28:
        basic.show_icon(IconNames.NO)
    else:
        basic.show_icon(IconNames.YES)
basic.forever(on_forever)
```

Figure 2.9. Programação em Python do Protótipo 1

2.3.2. Protótipo 2: Monitor de Luminosidade e Suporte Visual à Rotina

Fundamentação Sensorial e Pedagógica: A hipersensibilidade visual a luzes fluorescentes ou variações bruscas de iluminação natural é uma queixa recorrente em indivíduos

autistas, podendo causar fadiga ocular, dor de cabeça e ansiedade. O Monitor de Luminosidade atua como um recurso não verbal de apoio à rotina: a placa monitora continuamente a luz da sala e alerta a turma quando os níveis de iluminação ultrapassam limites confortáveis, servindo também como um sinalizador visual suave para momentos de transição de atividades e relaxamento.

Recursos Utilizados: Sensor de Luz da Matriz de LEDs e Matriz de LEDs para exibição de padrões.

Lógica de Programação:

1. A matriz de LEDs inverte periodicamente sua função para ler o nível de luminosidade ambiental (escala de 0 a 255).

2. Se a luz medida exceder o limite confortável estabelecido (ex.: > 180), a placa exibe um padrão visual de aviso suave para que as luzes ou cortinas sejam ajustadas.

3. Para níveis ideais de iluminação, exibe-se um padrão de menor intensidade gráfica para não sobrecarregar visualmente o aluno.

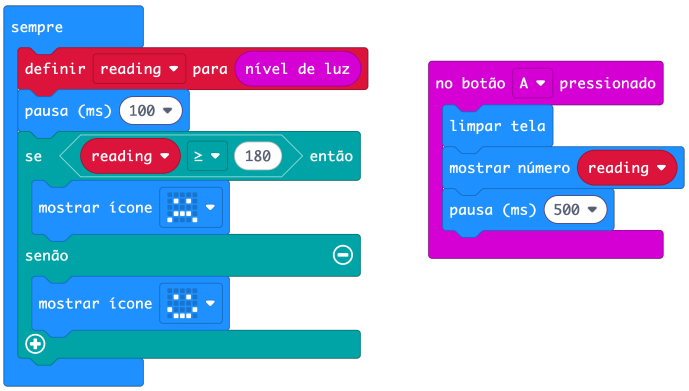


Figure 2.10. Programação em Blocos do Protótipo 2

```
let reading = 0
input.onButtonPressed(Button.A, function () {
    basic.clearScreen()
    basic.showNumber(reading)
    basic.pause(500)
})
basic.forever(function () {
    reading = input.lightLevel()
    basic.pause(100)
    if (reading >= 180) {
        basic.showIcon(IconNames.Sad)
    } else {
        basic.showIcon(IconNames.Happy)
    }
})
```

Figure 2.11. Programação em JavaScript do Protótipo 2

```
reading = 0

def on_button_pressed_a():
    basic.clear_screen()
    basic.show_number(reading)
    basic.pause(500)
input.on_button_pressed(Button.A, on_button_pressed_a)

def on_forever():
    global reading
    reading = input.light_level()
    basic.pause(100)
    if reading >= 180:
        basic.show_icon(IconNames.SAD)
    else:
        basic.show_icon(IconNames.HAPPY)
basic.forever(on_forever)
```

Figure 2.12. Programação em Python do Protótipo 2

2.3.3. Protótipo 3: Sonômetro de Sinalização Não Verbal de Ruído

Fundamentação Sensorial e Pedagógica: A sobrecarga auditiva é uma das causas mais frequentes de crises comportamentais no TEA. Sons agudos ou sobreposição de vozes no laboratório de informática podem desencadear respostas de fuga ou defesa. O Sonômetro transforma o nível sonoro do ambiente em um barramento gráfico visual na matriz de LEDs. Esse recurso transfere a responsabilidade da autorregulação para o grupo: em vez de o professor utilizar comandos verbais impositivos, que adicionam mais estímulo auditivo à sala, a proposta é a própria matriz indicar se o ruído está adequado, permitindo a autorregulação coletiva e protetiva.

Recursos Utilizados: Microfone Integrado e Matriz de LEDs.

Lógica de Programação:

1. O microfone capta continuamente o nível de som ambiente (escala de 0 a 255).
2. Utiliza-se a função nativa *plot bar graph* para mapear proporcionalmente o som na matriz de 5x5 colunas.
3. Se o som ultrapassar um valor crítico de sobrecarga (ex.: > 150), o Micro:bit exibe um alerta visual imediato.

2.3.4. Protótipo 4: Rastreador de Atividades

Fundamentação Sensorial e Pedagógica: Comportamentos motores repetitivos ou estereotipados (como o *flapping* de mãos ou o balanço do corpo) desempenham um papel legítimo de autorregulação emocional e sensorial no autismo. Em vez de reprimir ou punir tais movimentos, o Rastreador de Atividades utiliza o acelerômetro para validar e reorientar essa necessidade motora. O movimento físico do aluno é canalizado para uma interação computacional intencional e gamificada, fornecendo *feedback* visual e reforço positivo imediato à medida que o aluno atinge metas de movimento.

Recursos Utilizados: Acelerômetro Integrado (Detecção de Gestos e Aceleração) e Matriz de LEDs.

Lógica de Programação:



Figure 2.13. Programação em Blocos do Protótipo 3

```
basic.forever(function on_forever() {
    led.plotBarGraph(input.soundLevel(), 255)
    basic.pause(100)
    if (input.soundLevel() >= 150) {
        basic.showIcon(IconNames.Sad)
    }

    basic.pause(500)
})
```

Figure 2.14. Programação em JavaScript do Protótipo 3

1. Declara-se uma variável global de contagem de movimentos (*contador*).
2. Ao detectar o evento de movimento vigoroso do dispositivo (*on shake*), a variável é incrementada em 1 unidade.
3. A matriz exibe o número atualizado de repetições e celebra marcas atingidas com ícones visuais expansivos.

2.3.5. Protótipo 5: Botão de Autorregulação e Descompressão Tátil

Fundamentação Sensorial e Pedagógica: Em momentos de início de sobrecarga sensorial ou aumento de ansiedade, o estudante necessita de uma estratégia rápida de ancoragem (*grounding*). O Botão Sensível ao Toque (logotipo capacitivo dourado) atua como um comutador de acessibilidade: com um toque suave e sem a necessidade de aplicação de força mecânica, o aluno ativa um estímulo sonoro auditivo (como uma melodia suave)

```
def on_forever():
    led.plot_bar_graph(input.sound_level(), 255)
    basic.pause(100)
    if input.sound_level() >= 150:
        basic.show_icon(IconNames.SAD)
    basic.pause(500)
basic.forever(on_forever)
```

Figure 2.15. Programação em Python do Protótipo 3

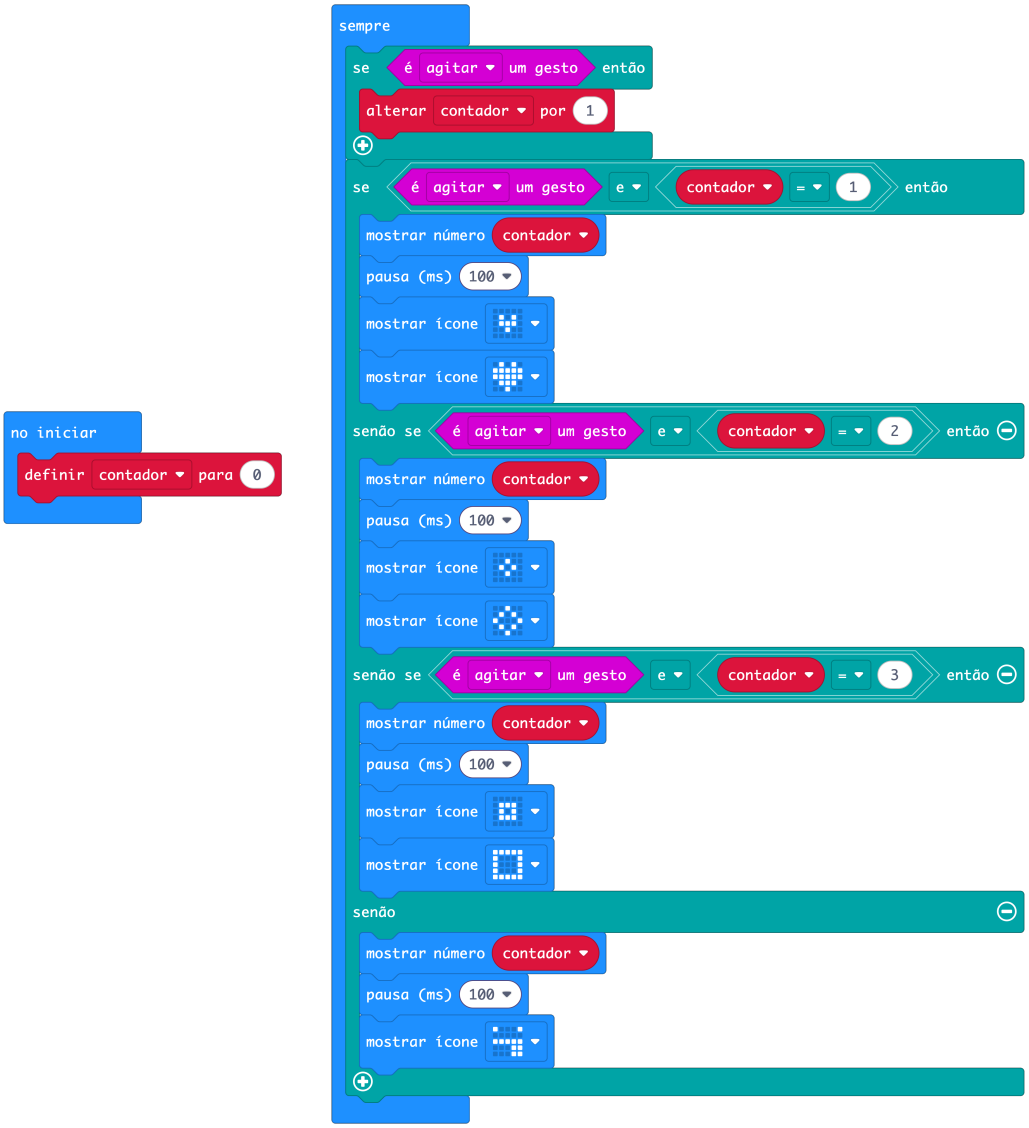


Figure 2.16. Programação em Blocos do Protótipo 4

ou um padrão de animação visual rítmica e previsível na placa. Esse recurso oferece ao próprio estudante o controle ativo sobre um estímulo aconchegante para retomada do foco.

Recursos Utilizados: Logotipo Sensível ao Toque Capacitivo e Alto-falante Integrado.

Lógica de Programação:

1. Configura-se o bloco de evento para o toque no logotipo capacitivo (*on logo pressed*).

2. Ao ser tocado, o Micro:bit reproduz um efeito sonoro suave e previsível via alto-falante nativo, acompanhado por um padrão gráfico desacelerado na matriz de LEDs.

```
let contador = 0
basic.forever(function () {
    if (input.isGesture(Gesture.Shake)) {
        contador += 1
    }
    if (input.isGesture(Gesture.Shake) && contador == 1) {
        basic.showNumber(contador)
        basic.pause(100)
        basic.showIcon(IconNames.SmallHeart)
        basic.showIcon(IconNames.Heart)
    } else if (input.isGesture(Gesture.Shake) && contador == 2) {
        basic.showNumber(contador)
        basic.pause(100)
        basic.showIcon(IconNames.SmallDiamond)
        basic.showIcon(IconNames.Diamond)
    } else if (input.isGesture(Gesture.Shake) && contador == 3) {
        basic.showNumber(contador)
        basic.pause(100)
        basic.showIcon(IconNames.SmallSquare)
        basic.showIcon(IconNames.Square)
    } else {
        basic.showNumber(contador)
        basic.pause(100)
        basic.showIcon(IconNames.Silly)
    }
})
```

Figure 2.17. Programação em JavaScript do Protótipo 4

2.3.6. Orientações Metodológicas para Aplicação e Adaptabilidade em Sala de Aula

Para que a implementação dessas práticas de computação física com o Micro:bit seja efetiva em turmas inclusivas com estudantes autistas, recomendam-se as seguintes diretrizes metodológicas para professores e pesquisadores:

- **Previsibilidade e Antecipação:** Apresentar previamente o Micro:bit, os sons que ele pode emitir e as luzes que serão acesas, permitindo que o estudante explore o dispositivo em um ambiente calmo antes da atividade coletiva.

- **Ajuste Fino de Parâmetros de Sensibilidade:** Os limiares de acionamento (níveis de som, luz e sensibilidade do toque) devem ser calibrados individualmente em conjunto com o estudante, garantindo que o alerta reflita o limite real de conforto de cada aluno.

- **Uso do Suporte Físico e Cases de Proteção:** Utilizar capas de silicone ou suportes de proteção tátil na placa, que além de proteger o circuito contra quedas, suavizam o contato térmico e tátil para alunos com hipersensibilidade.

- **Gamificação Flexível e Sem Pressão Temporal:** Estruturar os desafios de programação sem impor prazos rígidos ou dinâmicas competitivas ruidosas, priorizando o progresso individual e o aprendizado cooperativo.

- **Construção de Espaços de Descompressão:** Permitir que os protótipos de autorregulação (como o botão de toque ou o monitor de ruído) fiquem disponíveis no ambiente para uso espontâneo do aluno sempre que sentir necessidade.

```python
contador = 0

def on_forever():
    global contador
    if input.is_gesture(Gesture.SHAKE):
        contador += 1
    if input.is_gesture(Gesture.SHAKE) and contador == 1:
        basic.show_number(contador)
        basic.pause(100)
        basic.show_icon(IconNames.SMALL_HEART)
        basic.show_icon(IconNames.HEART)
    elif input.is_gesture(Gesture.SHAKE) and contador == 2:
        basic.show_number(contador)
        basic.pause(100)
        basic.show_icon(IconNames.SMALL_DIAMOND)
        basic.show_icon(IconNames.DIAMOND)
    elif input.is_gesture(Gesture.SHAKE) and contador == 3:
        basic.show_number(contador)
        basic.pause(100)
        basic.show_icon(IconNames.SMALL_SQUARE)
        basic.show_icon(IconNames.SQUARE)
    else:
        basic.show_number(contador)
        basic.pause(100)
        basic.show_icon(IconNames.SILLY)
basic.forever(on_forever)
```

Figure 2.18. Programação em Python do Protótipo 4

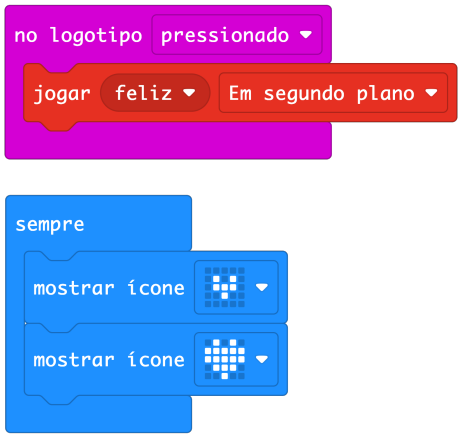


Figure 2.19. Programação em Blocos do Protótipo 5

2.4. Desafios de Avaliação

A avaliação do impacto pedagógico e da usabilidade de Recursos de Tecnologia Assistiva (RTA) desenvolvidos por meio da computação física apresenta complexidades particulares. Diferente do software educacional tradicional, cujas métricas de sucesso baseiam-se predominantemente no tempo de execução de tarefas ou no acerto de respostas em tela, a eficácia de um protótipo físico e acessível como o Micro:bit exige uma abordagem avaliativa holística. É necessário mensurar não apenas o ganho conceitual em computação, mas também o grau de autonomia, o conforto sensorial, a redução de barreiras de interação e a aceitação social da solução no ambiente escolar.

```
input.onLogoEvent(TouchButtonEvent.Pressed, function () {
    music.play(music.builtinPlayableSoundEffect(soundExpression.happy), music.PlaybackMode.InBackground)
})
basic.forever(function () {
    basic.showIcon(IconNames.SmallHeart)
    basic.showIcon(IconNames.Heart)
})
```

Figure 2.20. Programação em JavaScript do Protótipo 5

```
def on_logo_pressed():
    music.play(music.builtin_playable_sound_effect(soundExpression.happy),
        music.PlaybackMode.IN_BACKGROUND)
input.on_logo_event(TouchButtonEvent.PRESSED, on_logo_pressed)

def on_forever():
    basic.show_icon(IconNames.SMALL_HEART)
    basic.show_icon(IconNames.HEART)
basic.forever(on_forever)
```

Figure 2.21. Programação em Python do Protótipo 5

2.4.1. Métricas e Métodos de Avaliação de Usabilidade e Acessibilidade

No campo da Interação Humano-Computador (IHC), a usabilidade de um dispositivo assistivo é definida pela extensão em que o recurso permite que usuários com diferentes perfis funcionais alcancem objetivos específicos com eficácia, eficiência e satisfação em um contexto de uso real (ISO 9241-11). Quando aplicada a estudantes neurodivergentes ou com deficiência, a avaliação de usabilidade deve adaptar métodos consagrados para evitar sobrecarga estressante:

- **Adaptação da Escala de Usabilidade do Sistema (SUS - *System Usability Scale*):** A escala SUS [9] de 10 itens pode ser simplificada e traduzida para formatos visuais ou pictográficos (usando cartões com expressões faciais ou escalas de Likert visuais), permitindo que estudantes autistas ou com dificuldades de comunicação verbal expressem diretamente o grau de facilidade ou frustração ao manipular o protótipo.

- **Protocolo de Pense em Voz Alta Adaptado (*Think-Aloud*):** Em vez de exigir o relato verbal contínuo durante o uso da placa, prioriza-se a observação estruturada de comportamento não verbal, registrando hesitações, buscas por ajuda, sorrisos de engajamento ou sinais de desconforto tátil ao interagir com os sensores e botões do Micro:bit.

- **Análise de Carga Mental e Estresse Sensorial:** Utilização de métricas qualitativas e observacionais para identificar se a resposta do hardware (sons emitidos, brilho dos LEDs) reduz ou eleva a carga cognitiva e a ansiedade do aluno durante a atividade.

- **Manequim de Autoavaliação (*Self-Assessment Manikin* - SAM):** Utilização da ferramenta não verbal e pictográfica SAM [8] como uma alternativa acessível para avaliar as dimensões emocionais, como valência, alerta e dominância, na interação

com os protótipos. Por meio de figuras ilustrativas simplificadas, o protocolo permite que os participantes expressem suas respostas afetivas e o nível de conforto sem a necessidade de relatos verbais complexos.

2.4.2. Métricas de Impacto Pedagógico e Aprendizagem Inclusiva

A avaliação do impacto pedagógico fundamenta-se nos princípios do *Design* Universal para a Aprendizagem (DUA), mensurando em que medida a introdução da computação física tangível expandiu o acesso do estudante ao conhecimento de programação. As principais dimensões avaliativas incluem:

1. **Ganho de Autonomia e Participação Ativa:** Verificação da diminuição da necessidade de suporte verbal direto ou mediação constante do professor à medida que o aluno domina o manuseio e a lógica do protótipo.
2. **Apropriação do Pensamento Computacional:** Avaliação do domínio de conceitos chave (como algoritmos, variáveis, condicionais e eventos de entrada/saída) por meio da demonstração prática no objeto físico, e não estritamente por provas escritas ou testes formais.
3. **Eficácia na Mediação Sensorial e Autorregulação:** Nos protótipos voltados ao ambiente escolar (como o sonômetro e o monitor térmico), mede-se a capacidade da ferramenta em antecipar e mitigar desorganizações comportamentais, reduzindo a frequência de crises sensoriais durante as aulas de informática.

2.4.3. Desafios Metodológicos na Coleta de Dados em Ambientes Inclusivos

Pesquisadores e educadores enfrentam desafios éticos e práticos significativos ao avaliar soluções assistivas em sala de aula:

- **Heterogeneidade dos Perfis de Aprendizagem:** A extrema variabilidade de habilidades e necessidades dentro do espectro autista e de outras neurodivergências inviabiliza amostragens estatísticas padronizadas rigorosas. Métodos baseados em Estudo de Caso Único (*Single-Subject Research Design*) ou Pesquisa-Ação mostram-se mais adequados por permitirem comparar o desempenho e o comportamento do próprio aluno antes, durante e após a introdução da tecnologia assistiva.
- **Efeito Novidade versus Uso Continuado:** Um desafio recorrente é diferenciar o entusiasmo passageiro provocado pela luz e som da placa Micro:bit (efeito novidade) da real eficácia da ferramenta como recurso permanente de aprendizagem e autorregulação. Avaliações longitudinais são recomendadas para verificar a sustentabilidade do uso.
- **Triangulação de Fontes de Dados:** Para obter dados robustos e confiáveis sem expor o estudante a testes invasivos, recomenda-se a triangulação metodológica, combinando: (a) registros em diários de campo do pesquisador/professor; (b) observação videográfica estruturada do comportamento de interação; e (c) percepção dos mediadores (professores, tutores e familiares) sobre mudanças no engajamento e bem-estar do aluno.

2.5. Considerações Finais

A busca pela democratização do acesso às tecnologias digitais e pelo ensino inclusivo de programação exige o desenvolvimento constante de novas abordagens pedagógicas e metodológicas. Este capítulo apresentou a Computação Física Inclusiva apoiada pelo ecossistema Micro:bit V2 e pela plataforma *MakeCode* como uma alternativa viável, de baixo custo e integrativa para mitigar as barreiras sensoriais, motoras e cognitivas presentes nos ambientes tradicionais de ensino de computação.

Ao fundamentar as práticas nos preceitos da IHC voltada à acessibilidade, do *Design* Universal para a Aprendizagem (DUA) e no arcabouço normativo nacional (a exemplo da PNED, BNCC-Computação e LBI), o trabalho demonstrou como a tangibilidade e o *feedback* multimodal (visual, sonoro e tátil) podem transformar o aprendizado de conceitos abstratos em uma experiência concreta. Em particular, a proposição de protótipos funcionais voltados ao contexto do TEA traz possibilidades do uso dessas ferramentas como RTA não apenas para o aprendizado técnico, mas também para a mediação sensorial, previsibilidade e autorregulação dos estudantes.

Além do desenvolvimento de projetos, o capítulo discutiu as particularidades e os desafios associados à avaliação de usabilidade e impacto pedagógico em ambientes inclusivos. A adaptação de protocolos consagrados (como a simplificação pictográfica do SUS, a aplicação da escala não verbal SAM e a condução do *Think-Aloud* adaptado) tornam-se opções para capturar respostas emocionais e atitudinais de forma ética, respeitando a heterogeneidade dos perfis dos participantes e superando abordagens avaliativas estritamente quantitativas ou formais.

Apesar das vantagens mapeadas, reconhece-se que a adoção efetiva da computação física inclusiva nas escolas enfrenta desafios operacionais, como a necessidade de formação continuada de educadores, o investimento contínuo na infraestrutura dos laboratórios e a superação do "efeito novidade" por meio de práticas pedagógicas longitudinais e sustentáveis.

Como trabalhos futuros, incentiva-se a realização de estudos com pesquisa-ação para avaliar a usabilidade e a eficácia pedagógica dos protótipos propostos diretamente em salas de aula inclusivas. Ademais, sugere-se a ampliação do escopo de projetos para contemplar demandas de estudantes com outras especificidades funcionais (como deficiências motoras severas ou deficiência visual), estimulando o protagonismo dos próprios alunos na concepção, prototipagem e validação de suas próprias soluções de Tecnologia Assistiva.

References

[1] BRASIL. (2023) Lei n°14.533, de 11 de janeiro de 2023. Brasília, DF: Presidência da República.

[2] BRASIL. (2022) Ministério da Educação. Computação na Educação Básica: Complemento à BNCC. Brasília, DF: MEC.

[3] BRASIL. (2021) Ministério da Educação. Secretaria de Educação Básica. *Programa Nacional do Livro e do Material Didático (PNLD)*. Brasília, DF: FNDE/MEC.

Disponível em: <https://www.gov.br/fnde/>.

[4] Humberto A. P. Zanetti e Marcos A. F. Borges. (2020) Ensino de Programação utilizando Computação Física: uma Revisão Sistemática da Literatura. *Comunicações em Informática*, v. 4, n. 1, p. 19–22. ISSN 2595-0622. DOI: 10.22478/ufpb.2595-0622.2020v4n1.52065.

[5] Molly Follette Story, James L. Mueller e Ronald L. Mace. (1998) *The Universal Design File: Designing for People of All Ages and Abilities*. Edição revista. Center for Universal Design, North Carolina State University, Raleigh, NC. ERIC ED 460 554.

[6] Angelaki, S., Triantafyllidis, G. A., Besenecker, U. (2022). Lighting in Kindergartens: Towards Innovative Design Concepts for Lighting Design in Kindergartens Based on Childrens Perception of Space. Sustainability, 14(4), 2370. https://doi.org/10.3390/su14042370.

[7] Freitas, M. L. P. de, Pires, F., Pessoa, M. (2025). Uso de Computação Física para desenvolvimento de Pensamento Computacional: um estudo de caso. In Anais do XXXI Workshop de Informática na Escola (pp. 475487). Sociedade Brasileira de Computação (SBC). https://doi.org/10.5753/wie.2025.13490.

[8] Bynion, T. M., Feldner, M. T. (2017). Self-assessment manikin. In V. Zeigler-Hill, T. K. Shackelford (Eds.), *Encyclopedia of Personality and Individual Differences* (pp. 1–3). Springer. https://doi.org/10.1007/978-3-319-28099-8_77-1

[9] Brooke, J. SUS: A quick and dirty usability scale. In: Jordan, P. W.; Thomas, B.; Weerdmeester, B. A.; Mcclelland, I. L. (eds.). Usability Evaluation in Industry. London: Taylor and Francis, 1996. p. 189194.

[10] BRASIL. (2021) Decreto nž 10.645, de 11 de março de 2021. Regulamenta o art. 75 da Lei nž 13.146, de 6 de julho de 2015, para instituir o Plano Nacional de Tecnologia Assistiva. *Diário Oficial da União*, Brasília, DF.

[11] BRASIL. (2004) Decreto nž 5.296, de 2 de dezembro de 2004. Regulamenta as Leis nž 10.048, de 8 de novembro de 2000, que dá prioridade de atendimento às pessoas que especifica, e nž 10.098, de 19 de dezembro de 2000, que estabelece normas gerais e critérios básicos para a promoção da acessibilidade. *Diário Oficial da União*, Brasília, DF.

[12] Micro:bit Educational Foundation, "Micro:bit impact report and educational resources," 2020. [Online]. Available: https://microbit.org

[13] Zerbato, A. P., Mendes, E. G. (2018). Desenho universal para a aprendizagem como estratégia de inclusão escolar. *Educação Unisinos*, 22(2), 147–155.

[14] Koegel, R., Kim, S., Koegel, L., & Schwartz, B. (2013). Improving Socialization for High School Students with ASD by Using Their Preferred Interests. *Journal of Autism and Developmental Disorders*, 43(9), 2121–2134. https://doi.org/10.1007/s10803-013-1765-3

[15] AMERICAN PSYCHIATRIC ASSOCIATION. (2013) *Diagnostic and Statistical Manual of Mental Disorders (DSM-5)*. 5th ed. Washington, DC: American Psychiatric Association. https://doi.org/10.1176/appi.books.9780890425596

Capítulo 3

Gêmeos Digitais: Uma Ponte entre os Mundos Físico e Virtual

Everton Cavalcante (UFRN), Thais Batista (UFRN), Flavia C. Delicato (UFF), André Almeida (IFRN)

Resumo

Gêmeos digitais consistem em representações virtuais dinâmicas de entidades físicas, processos e sistemas do mundo real, permitindo integrar dados, modelos computacionais e mecanismos de análise para apoiar atividades de monitoramento, simulação, previsão, otimização e tomada de decisão. Este capítulo fornece uma visão abrangente sobre gêmeos digitais, apresentando seus conceitos fundamentais, os principais paradigmas habilitadores, padrões e plataformas voltados ao seu desenvolvimento e operação. Além disso, discute o papel das tecnologias multimídia e Web, focando na integração de dados multimodais, na interoperabilidade de sistemas distribuídos e no desenvolvimento de interfaces visuais interativas. O capítulo explora ainda domínios nos quais gêmeos digitais têm sido empregados em diversas aplicações e discute desafios de pesquisa relacionados à construção de ecossistemas de gêmeos digitais interoperáveis, confiáveis e seguros. Como contribuição, este capítulo busca oferecer a pesquisadores, estudantes e profissionais em Computação uma base conceitual sólida e estruturada sobre gêmeos digitais, estimulando o desenvolvimento de novas pesquisas e aplicações na interseção entre esse paradigma e as tecnologias multimídia e da Web.

Abstract

Digital twins consist of dynamic virtual representations of real-world physical entities, processes, and systems, enabling the integration of data, computational models, and analysis mechanisms to support monitoring, simulation, prediction, optimization, and decision-making. This chapter provides a comprehensive overview of digital twins, presenting their fundamental concepts, the main enabling paradigms, standards, and platforms supporting their development and operation. Furthermore, it discusses the role of multimedia and Web technologies in this context, with emphasis on multimodal data integration, communication and interoperability among distributed systems, and interactive

visualization interfaces. The chapter further explores domains where digital twins have been used across diverse applications and discusses research challenges in building interoperable, trustworthy, and secure digital twin ecosystems. As a contribution, this chapter aims to provide researchers, students, and practitioners in computing with a solid, structured conceptual foundation on digital twins, fostering the development of new research and applications at the intersection of this paradigm with multimedia and Web technologies.

3.1. Introdução

A crescente capacidade de observar, conectar, modelar e analisar sistemas do mundo real tem transformado a forma como os sistemas computacionais são utilizados para compreender e intervir em fenômenos físicos. Sensores conectados, redes de comunicação, computação em nuvem e na borda, técnicas de simulação, Inteligência Artificial (IA) e mecanismos de visualização interativa permitem construir representações digitais cada vez mais detalhadas de equipamentos, ambientes, processos e sistemas complexos. Nesse contexto, os **gêmeos digitais** (*digital twins*, em inglês) surgem como um paradigma para integrar representações computacionais a entidades ou processos do mundo real, estabelecendo uma relação dinâmica entre os mundos físico e virtual.

Embora o interesse por gêmeos digitais tenha crescido significativamente nos últimos anos, o uso de representações computacionais para monitorar e analisar sistemas físicos não é recente. O termo "gêmeo digital" foi introduzido em 2002, no contexto do gerenciamento do ciclo de vida de produtos, para designar a criação de uma contraparte digital de um sistema físico voltada ao seu monitoramento em tempo real e à sua otimização [Grieves and Vickers 2017]. Conceitos como modelagem, simulação, monitoramento remoto, sistemas ciberfísicos e engenharia baseada em modelos constituem antecedentes importantes desse paradigma. Contudo, os gêmeos digitais alcançaram maior maturidade técnica e escala de adoção apenas com a convergência de avanços em sensoriamento de baixo custo, infraestruturas de conectividade e processamento e métodos avançados de modelagem e de análise de grandes volumes de dados, o que permitiu que modelos antes relativamente independentes do sistema físico passassem a ser continuamente alimentados por dados de sua contraparte no mundo real.

Apesar desses avanços, a própria definição de gêmeo digital permanece objeto de discussão. Diferentes comunidades científicas e da indústria enfatizam propriedades distintas, como representação virtual, conexão com entidades físicas, atualização contínua, sincronização, simulação, previsão e suporte à tomada de decisão, diversidade essa também refletida em trabalhos que buscam caracterizar o conceito e mapear o estado da arte [Barricelli et al. 2019, Tao et al. 2019, Fuller et al. 2020, Jones et al. 2020, Eramo et al. 2022, Bokhtiar Al Zami et al. 2025]. Em uma tentativa de padronização, a norma internacional ISO/IEC 30173:2023 define um gêmeo digital como a "representação digital de uma entidade-alvo com conexões de dados que permitem a convergência entre estados físicos e digitais a uma taxa de sincronização apropriada" [ISO 2023]. Já o Digital Twin Consortium (DTC), consórcio internacional voltado à proposição de definições, taxonomias e capacidades de referência para gêmeos digitais em múltiplos domínios, define o conceito como uma representação virtual integrada e orientada a dados de entidades e processos do mundo real, com interação sincronizada em determinada frequência e fide-

lidade [Digital Twin Consortium 2022]. A despeito das diferenças entre essas definições, todas convergem para uma característica fundamental: um gêmeo digital não deve ser reduzido a uma representação estática ou a uma visualização de um sistema físico, mas deve estabelecer uma relação estruturada e dinâmica entre a representação digital e o que ela representa.

De maneira geral, um gêmeo digital consiste em uma réplica digital dinâmica, baseada em *software*, de uma entidade do mundo real, continuamente atualizada com dados provenientes de sensores e dispositivos computacionais ao longo de seu ciclo de vida. Essa integração permite que o modelo virtual reflita, em tempo quase real, o estado e o comportamento de seu correspondente físico, oferecendo funcionalidades de monitoramento, análise, visualização, controle e otimização da estrutura e do comportamento de sistemas complexos [Fitzgerald et al. 2024]. O que distingue o modelo de gêmeo digital dos modelos tradicionais de simulação e dos modelos digitais construídos apenas a partir de dados coletados de entidades físicas é a sincronização bidirecional automatizada entre os mundos real e digital: mudanças na entidade física são refletidas na representação virtual e vice-versa, possibilitando não apenas o monitoramento, mas também o controle e a otimização diretos da entidade-alvo.

Um gêmeo digital pode receber e produzir não apenas dados e eventos estruturados, mas também imagens, vídeos, áudio, dados espaciais, mapas, modelos tridimensionais e outras formas de informação multimodal. Por exemplo, câmeras podem fornecer fluxos de vídeo para identificar ocupação de ambientes, movimentação ou situações anômalas; microfones podem fornecer informações acústicas; modelos 3D podem representar a geometria de ambientes e equipamentos; mapas podem associar eventos a localizações; e interfaces gráficas interativas podem permitir que pessoas explorem o estado atual e possíveis estados futuros do sistema. Dessa forma, dados multimodais constituem parte fundamental da representação de um gêmeo digital e de seus mecanismos de observação e análise, característica ilustrada a seguir por meio do exemplo de um estádio inteligente, que servirá de referência ao longo deste capítulo.

Grandes estádios, que acomodam dezenas de milhares de espectadores, enfrentam desafios complexos de logística e segurança, como aglomerações que comprometem a experiência e a segurança durante os eventos, a dificuldade de rastrear a movimentação das pessoas e a tomada de decisão em tempo real [Adolphsen 2025]. Esses cenários têm sido cada vez mais instrumentados com sensores ambientais (temperatura, umidade, qualidade do ar e ruído), câmeras distribuídas nas áreas internas e externas e sensores de ocupação e de fluxo instalados em catracas, arquibancadas e corredores. Um gêmeo digital voltado ao gerenciamento e à segurança de multidões, como o ilustrado na Figura 3.1, poderia manter uma representação dos principais espaços, equipamentos, sistemas e relações do ambiente físico, contribuindo para a segurança e a experiência dos espectadores. Um setor de arquibancada, por exemplo, poderia ser representado digitalmente com informações sobre capacidade, ocupação estimada, condições ambientais e equipamentos associados, enquanto câmeras forneceriam fluxos de vídeo para estimar o número de pessoas em cada área. A análise desses fluxos permitiria identificar, de forma preditiva, pontos de congestão antes que ocorressem, além de gerar comandos automatizados, por meio de sinalizações digitais dinâmicas, para orientar o deslocamento das pessoas, da chegada à saída do local. Exemplos reais de estádios inteligentes nos quais gêmeos digitais têm sido utili-

zados incluem o SoFi Stadium[1] e o Gillette Stadium[2], nos Estados Unidos, o Deutsche Bank Park[3], na Alemanha, e o Allianz Stadium[4], na Austrália.



Figura 3.1. Visão conceitual do gêmeo digital de um estádio inteligente

Esse cenário de um estádio inteligente demonstra que o desenvolvimento de gêmeos digitais envolve desafios que vão além da mera integração de sensores. Do ponto de vista dos sistemas multimídia e da Web, os gêmeos digitais constituem um campo de aplicação particularmente rico, pois sua construção e operação dependem da captura, do transporte, do armazenamento, da integração e da apresentação desses dados multimodais. Cabe ressaltar que grande parte desses dados é coletada, processada e disponibilizada por meio de arquiteturas e protocolos do ecossistema da Web, como APIs REST, *streaming* de eventos e WebSockets. Portanto, tecnologias de sistemas multimídia e Web não são meramente periféricas à construção de gêmeos digitais, mas sim frequentemente constituem a própria infraestrutura de integração, visualização e interação que os torna operacionalmente viáveis e acessíveis aos usuários finais e às aplicações.

Este capítulo tem por objetivo apresentar uma visão abrangente sobre gêmeos digitais, discutindo seus fundamentos conceituais, os principais paradigmas e as tecnologias que viabilizam sua construção, incluindo tecnologias multimídia e da Web, bem como algumas plataformas que possibilitam a conexão entre representações digitais e sistemas do mundo real. O capítulo também apresenta domínios reais de aplicação em que se observa a utilização consolidada de gêmeos digitais e discute os principais desafios científicos e tecnológicos relacionados ao seu desenvolvimento e sua implantação.

[1] *SoFi Stadium Builds Out Digital Twin for Operations and Maintenance*: https://www.enr.com/articles/50568-sofi-stadium-builds-out-digital-twin-for-operations-and-maintenance

[2] *Making Kraft Group architectural data more actionable*: https://www.sportsbusinessjournal.com/Articles/2024/06/26/Tech/gillette-stadium-digital-twin/

[3] *Arena of IoT: Eintracht's IoT center*: https://club.eintracht.de/eintrachttech/digital-center-arena-of-iot/

[4] *Optimizing Public Asset Lifecycle Using AWS IoT TwinMaker with John Holland*: https://aws.amazon.com/solutions/case-studies/john-holland-case-study/

O restante do capítulo está organizado da seguinte forma. A Seção 3.2 apresenta os conceitos fundamentais de gêmeos digitais e discute os principais paradigmas habilitadores, bem como o papel de tecnologias multimídia e da Web, linguagens, padrões e plataformas de *middleware* para sua construção e operação. A Seção 3.3 apresenta diferentes domínios em que gêmeos digitais têm sido empregados. A Seção 3.4 discute desafios de pesquisa relacionados à construção, operação e evolução de gêmeos digitais. Por fim, a Seção 3.5 sintetiza as principais discussões e contribuições do capítulo e aponta perspectivas futuras.

3.2. Conceitos, Paradigmas e Tecnologias para Gêmeos Digitais

3.2.1. Modelos, Sombras e Gêmeos Digitais

Um dos passos fundamentais para compreender os gêmeos digitais é distinguir três conceitos associados a diferentes níveis de integração entre os mundos físico e digital: **modelo digital**, **sombra digital** e **gêmeo digital**. Esses conceitos são frequentemente confundidos principalmente em decorrência da falta de uniformidade nas definições de gêmeos digitais na literatura [Kritzinger et al. 2018, Jones et al. 2020], levando diversos trabalhos a caracterizar como gêmeo digital algo que, na verdade, não se caracteriza como tal. A Figura 3.2 ilustra a distinção entre esses conceitos, explicados a seguir.

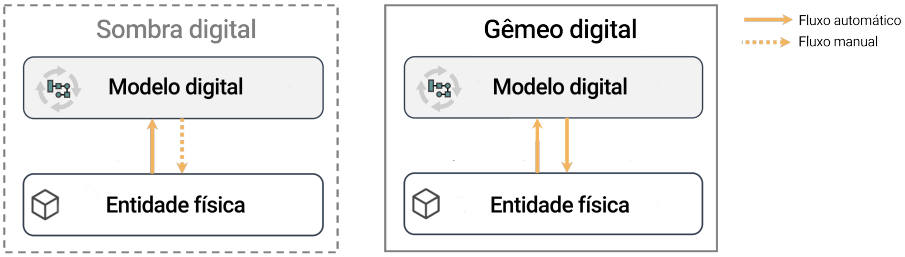


Figura 3.2. Diferença entre os conceitos de modelo, sombra e gêmeo digital

Um modelo digital é uma representação virtual de uma entidade, sistema ou processo físico, sem troca automática de dados entre as duas instâncias e, portanto, essencialmente estático. Um modelo tridimensional de um estádio, por exemplo, pode descrever sua geometria, seus setores e suas instalações sem depender de informações sobre o estado atual do edifício. Da mesma forma, um modelo matemático do comportamento térmico de uma arena pode simular diferentes condições sem estar conectado aos sensores nela instalados. Assim, alterações no estado da entidade física não se refletem automaticamente no modelo digital e vice-versa.

Uma sombra digital (*digital shadow*, em inglês) estabelece um primeiro nível de integração dinâmica entre a entidade física e sua representação digital: informações provenientes do sistema físico, tipicamente por meio de sensores, atualizam a representação digital, estabelecendo um fluxo automático no qual as alterações no estado físico são refletidas no domínio virtual. No exemplo do estádio inteligente, sensores de temperatura, câmeras, dispositivos de controle de acesso e medidores de consumo de energia poderiam alimentar uma representação que acompanha o estado atual do ambiente. Contudo, esse fluxo é unidirecional, pois a representação digital não tem capacidade de influenciar ou

alterar o estado da entidade correspondente.

Um gêmeo digital, em sua concepção mais completa, estabelece uma relação bidirecional entre os dois domínios. Dados do sistema físico atualizam a representação digital, enquanto análises, simulações, previsões ou decisões produzidas no domínio digital podem apoiar ou executar ações no sistema físico, por meio de atuadores, sistemas de controle ou recomendações operacionais. Portanto, um gêmeo digital é mais do que uma visualização ou ferramenta estática, constituindo-se em um sistema dinâmico que promove a integração efetiva entre os mundos real e virtual.

A distinção entre os conceitos de modelo, sombra e gêmeo digital também ajuda a evitar a interpretação de que qualquer modelo tridimensional conectado a sensores seria necessariamente um gêmeo digital. Uma representação 3D pode ser um componente importante do sistema, especialmente para fins de visualização espacial, mas o gêmeo digital depende de uma estrutura mais ampla, que inclui dados, modelos, relações, mecanismos de sincronização e, conforme o caso, análise e interação. No estádio inteligente, um modelo 3D com a geometria das arquibancadas e dos corredores é apenas uma representação virtual. Se esse modelo passar a receber continuamente informações sobre ocupação e temperatura, ele pode integrar uma sombra digital e, se o sistema utilizar essas informações para executar simulações de evacuação, prever condições ambientais e apoiar decisões operacionais, ele se aproxima das características de um gêmeo digital.

3.2.2. Paradigmas Habilitadores de Gêmeos Digitais

Gêmeos digitais reúnem ideias provenientes de paradigmas relacionados, embora com objetivos distintos. A Internet das Coisas (IoT, do inglês *Internet of Things*), a computação em nuvem e na borda, a modelagem e a simulação, a análise de dados e a IA contribuem para a construção de gêmeos digitais, mas nenhuma dessas tecnologias, isoladamente, os caracteriza. Na verdade, os gêmeos digitais resultam da integração complementar dessas capacidades em torno de uma representação digital associada a uma entidade ou a um processo do mundo real.

IoT. A IoT fornece uma das principais bases para conectar entidades do mundo físico ao domínio virtual. Sensores e dispositivos conectados coletam informações sobre temperatura, localização, vibração, consumo de energia, ocupação e outras propriedades, enquanto atuadores permitem materializar decisões em ações no mundo físico. No estádio inteligente, sensores podem monitorar as condições ambientais de cada setor, câmeras podem fornecer informações sobre a ocupação e a movimentação e sistemas de acesso podem registrar eventos de entrada e saída. Enquanto a IoT enfatiza tradicionalmente a conectividade dos dispositivos físicos, um gêmeo digital acrescenta uma camada de contexto, representação e relacionamento entre os elementos. Por exemplo, um sensor de temperatura isolado fornece uma leitura do ambiente, mas um gêmeo digital pode relacionar essa medição a um setor específico, ao histórico de medições, às condições de ocupação e aos modelos utilizados para prever o comportamento desse setor.

Computação em nuvem. A computação em nuvem (*cloud computing*, em inglês) oferece escalabilidade, poder computacional e capacidade de armazenamento para viabilizar o desenvolvimento e a operação de gêmeos digitais complexos e em larga escala. Essas

capacidades permitem concentrar informações históricas, executar análises mais avançadas e disponibilizar serviços a diferentes aplicações e usuários. No cenário do estádio inteligente, a nuvem pode armazenar históricos de ocupação, consumo de energia e condições ambientais, além de hospedar modelos analíticos e serviços Web utilizados por operadores e gestores.

Computação na borda. A computação na borda (*edge computing*, em inglês) desloca parte do processamento para locais próximos às fontes de dados, possibilitando sincronizar os mundos real e virtual com latência reduzida. Essa característica é especialmente relevante para dados multimídia, pois um conjunto de câmeras de alta resolução pode gerar um fluxo excessivo de dados para transmissão contínua a uma infraestrutura central na nuvem. Dessa forma, em vez de transmitir todo o vídeo, dispositivos de borda podem processá-lo localmente para extrair eventos ou características relevantes e enviar ao gêmeo digital apenas informações como ocupação estimada, objeto detectado ou evento anômalo. No estádio inteligente, esse processamento pode reduzir o consumo da largura de banda, a latência e os custos de armazenamento, além de permitir a tomada de decisões mesmo quando a conectividade com a nuvem estiver limitada. Assim, a infraestrutura de borda não deve ser entendida como substituta da infraestrutura de nuvem, mas como parte de uma arquitetura distribuída de processamento.

Modelagem e simulação. Modelos computacionais representam propriedades e comportamentos do sistema físico, enquanto a simulação os utiliza para explorar estados ou cenários que não necessariamente ocorrem no sistema real. Isso torna os gêmeos digitais particularmente úteis para a tomada de decisão, permitindo não apenas descrever o estado atual de uma entidade, mas também explorar o que aconteceria caso determinada condição fosse alterada. No estádio inteligente, seria possível estimar como a temperatura de um setor se comportaria diante de um aumento significativo na ocupação, simular estratégias de evacuação ou avaliar o impacto de alternativas de iluminação no consumo de energia. As simulações também permitem experimentar sem interferir diretamente no sistema físico, o que é particularmente relevante em sistemas críticos, nos quais testar certas condições diretamente poderia ser custoso ou causar prejuízos financeiros ou materiais. De todo modo, é importante destacar que gêmeos digitais podem incorporar capacidades de simulação, mas essa não é uma capacidade obrigatória de todo gêmeo digital [Wooley et al. 2023].

Análise de dados e IA. Técnicas de análise de dados e de IA extraem conhecimento a partir dos dados coletados, geram previsões e apoiam processos automatizados de tomada de decisão [Kreuzer et al. 2024]. Algoritmos de aprendizado de máquina (*machine learning*, em inglês) podem ser utilizados para classificação, detecção de anomalias, previsão de séries temporais e reconhecimento de padrões, enquanto técnicas de visão computacional extraem informações a partir de imagens e vídeos. No cenário de estádio inteligente, um sistema de visão computacional poderia estimar a ocupação de áreas, enquanto modelos de aprendizado de máquina poderiam prever a demanda energética ou detectar comportamentos anômalos. Esses resultados podem ser incorporados ao estado do gêmeo digital ou disponibilizados como serviços de análise.

3.2.3. O Papel de Tecnologias Multimídia e da Web em Gêmeos Digitais

A relação entre gêmeos digitais, tecnologias multimídia e a Web é particularmente relevante em três vertentes: **integração de dados multimodais**, **comunicação e interoperabilidade** e **visualização e interação**. Essas vertentes são discutidas a seguir.

Um gêmeo digital robusto geralmente não se limita à agregação de dados provenientes de sensores e à análise de séries temporais, mas sim frequentemente integra fluxos de vídeo e imagens de câmeras, sinais de áudio de microfones, trajetórias espaciais de dispositivos de localização e dados geoespaciais tridimensionais. Em um estádio inteligente, essas modalidades podem ser combinadas. Uma câmera detecta uma concentração anormal de pessoas em determinada área, sensores ambientais indicam um aumento de temperatura, o sistema de controle de acesso registra a entrada de muitas pessoas e um modelo 3D associa esses eventos à respectiva localização. Essa combinação permite ao gêmeo digital produzir uma representação contextualizada dificilmente obtida a partir de uma única modalidade. Por outro lado, surgem desafios adicionais, já que diferenças nas taxas de amostragem, latências, resoluções e características semânticas entre as modalidades podem causar dessincronização e inconsistências.

As arquiteturas e os protocolos da Web fornecem grande parte da infraestrutura de comunicação e de interoperabilidade entre os componentes de um gêmeo digital, frequentemente distribuídos. APIs baseadas no estilo arquitetural REST e protocolos de *streaming* de eventos, bem como mecanismos de publicação/assinatura, são amplamente empregados para transportar dados entre dispositivos de borda, plataformas de gêmeos digitais na nuvem e aplicações consumidoras. Além disso, padrões da Web Semântica, como ontologias e vocabulários RDF/JSON-LD, têm sido propostos para enriquecer semanticamente os modelos de informação de gêmeos digitais [Rico et al. 2023, Karabulut et al. 2024], favorecendo a interoperabilidade entre diferentes provedores de dados, serviços e sistemas.

Por fim, tecnologias de visualização e interação, em particular gráficos 3D acelerados por *hardware* acessíveis via navegador, padrões de realidade estendida para a Web e *frameworks* de visualização interativa de dados, constituem a principal interface por meio da qual os usuários podem explorar e interagir com um gêmeo digital. No estádio inteligente, um painel de controle acessível via navegador que exibe o modelo 3D do estádio e indicadores de ocupação atualizados em tempo real pode permitir que os operadores de segurança explorem interativamente os diferentes setores.

A Figura 3.3 ilustra uma arquitetura em camadas que reúne diferentes elementos relevantes à estruturação de gêmeos digitais. Nela, observa-se que as tecnologias de sistemas multimídia e Web se inserem mais diretamente em duas camadas: **comunicação**, responsável por transportar dados entre sensores, plataformas de gêmeos digitais e aplicações consumidoras; e **visualização e interação**, na qual *dashboards* Web e tecnologias de renderização 3D tornam o gêmeo digital acessível e explorável pelos usuários.

3.2.4. Linguagens e Padrões para Representação de Gêmeos Digitais

A construção de sistemas de gêmeos digitais requer mecanismos para descrever as entidades representadas, suas propriedades e as relações entre elas. Essa necessidade pode ser atendida por meio de linguagens de modelagem e padrões de representação, que esta-



Figura 3.3. Arquitetura em camadas com os diferentes elementos relevantes à estruturação de gêmeos digitais

belecem estruturas e vocabulários comuns para descrever os elementos de um ambiente. Dentre eles, destacam-se a *Digital Twins Definition Language* (DTDL) e o *Asset Administration Shell* (AAS), descritos a seguir.

DTDL. Proposta pela Microsoft, a DTDL[5] é uma linguagem aberta baseada em JSON-LD para descrever modelos e interfaces de gêmeos digitais em termos de propriedades, telemetria, componentes, comandos e relacionamentos. Na Azure Digital Twins, plataforma da Microsoft para gêmeos digitais que será brevemente introduzida na Seção 3.2.5, os modelos DTDL funcionam como esquemas para descrever os tipos de entidades de um ambiente e podem ser utilizados posteriormente para criar instâncias de gêmeos digitais, bem como relações entre elas.

O Código 3.1 ilustra um modelo simplificado, em DTDL, de um setor do estádio inteligente. As linhas 7 a 16 definem duas propriedades utilizadas para representar informações sobre o estado da entidade: *capacidade* representa a capacidade máxima da seção e *ocupacao* representa sua ocupação atual. As linhas 17 a 21 definem um relacionamento chamado *possuiSensor*, que indica que uma seção do estádio possui um sensor de temperatura instalado. Diferentemente de uma propriedade, que representa uma característica ou estado da entidade, um relacionamento representa uma conexão semântica com outra entidade. Neste exemplo, o atributo `target` (linha 20) restringe o relacionamento a entidades cujo modelo seja *SensorTemperatura*.

O Código 3.2 ilustra o modelo *SensorTemperatura* em DTDL, no qual a propriedade *temperatura*, definida nas linhas 7 a 11, representa a temperatura medida pelo sensor.

[5] https://github.com/Azure/opendigitaltwins-dtdl

```
{
  "@id": "dtmi:Estadio:Secao;1",
  "@type": "Interface",
  "@context": "dtmi:dtdl:context;3",
  "displayName": "Seção do Estádio",
  "contents": [
    {
      "@type": "Property",
      "name": "capacidade",
      "schema": "integer"
    },
    {
      "@type": "Property",
      "name": "ocupacao",
      "schema": "integer"
    },
    {
      "@type": "Relationship",
      "name": "possuiSensor",
      "target": "dtmi:stadium:SensorTemperatura;1"
    }
  ]
}
```

Código 3.1. Modelo DTDL de um setor do estádio inteligente

```
{
  "@id": "dtmi:Estadio:SensorTemperatura;1",
  "@type": "Interface",
  "@context": "dtmi:dtdl:context;3",
  "displayName": "Sensor de Temperatura",
  "contents": [
    {
      "@type": "Property",
      "name": "temperatura",
      "schema": "double"
    }
  ]
}
```

Código 3.2. Modelo DTDL de um sensor de temperatura existente no estádio inteligente

A descrição de uma entidade em DTDL não representa apenas suas propriedades isoladamente, mas também suas relações com outras entidades. Com isso, obtém-se um grafo em que as entidades aparecem como elementos conectados por relações semanticamente significativas. Essa abordagem é importante porque raramente sistemas complexos podem ser adequadamente representados como coleções de atributos isolados. Por exemplo, o significado de um sensor pode depender de sua relação com determinado equipamento; o equipamento, de sua localização; a localização, de sua associação a um setor, a um edifício ou a uma infraestrutura maior.

AAS. No contexto da manufatura, o AAS (*Asset Administration Shell*) fornece uma representação digital padronizada de um ativo físico ou digital e uma estrutura comum para acessar suas informações e serviços, enfatizando a interoperabilidade entre aplicações e sistemas industriais. A abordagem do AAS, definida pelo padrão internacional IEC 63278-1:2023 [IEC 2023], envolve dois elementos fundamentais: (i) o ativo (*asset*), que corresponde à entidade física ou digital a ser representada; e (ii) o AAS, que constitui a representação digital associada a esse ativo e fornece interfaces de acesso às informações

e aos serviços correspondentes.

O AAS pode ainda conter ou referenciar diferentes submodelos, cada um dedicado a um aspecto específico do ativo [IDTA 2026]. Em vez de concentrar todos os aspectos de um ativo em uma única estrutura, o AAS pode organizar diferentes perspectivas ou conjuntos de informações em submodelos independentes, de modo que cada submodelo represente um aspecto específico do ativo, como identificação, características técnicas, estado operacional, manutenção ou serviços [Zhang et al. 2025]. Os submodelos podem ainda utilizar modelos de submodelo padronizados e referências a dicionários de conceitos e ontologias, contribuindo para a interoperabilidade semântica. Essa organização modular permite ampliar progressivamente a representação à medida que novos requisitos são incorporados.

Considerando novamente o estádio inteligente como ilustração, uma representação baseada em AAS de um sistema de climatização poderia organizar diferentes aspectos do equipamento em submodelos independentes (ver Figura 3.4). Um submodelo de identificação poderia conter informações relativamente estáticas, como fabricante e modelo, enquanto um submodelo de dados operacionais poderia apresentar valores atualizados de temperatura, umidade e estado de funcionamento. Por sua vez, um submodelo de manutenção poderia registrar informações sobre intervenções realizadas e programadas e um submodelo de controle poderia disponibilizar operações para interagir com o equipamento, como o ajuste da temperatura desejada. Assim, por exemplo, uma aplicação de monitoramento poderia utilizar principalmente o submodelo de dados operacionais, enquanto uma aplicação de manutenção utilizaria as informações correspondentes ao ciclo de vida do equipamento e uma aplicação de automação poderia utilizar os serviços disponibilizados pelo submodelo de controle. Dessa forma, os submodelos permitem que diferentes aplicações acessem os aspectos do ativo relevantes aos seus objetivos, sem exigir que todas utilizem a representação completa. Além disso, submodelos padronizados favorecem a interoperabilidade ao estabelecer estruturas e conceitos comuns para a representação de informações e de serviços.

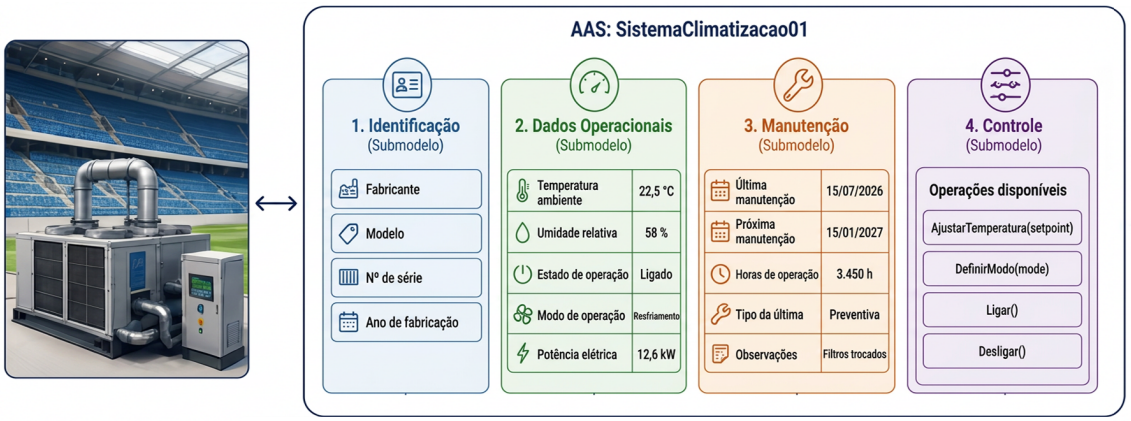


Figura 3.4. Representação conceitual de um AAS para um sistema de climatização de um estádio

O AAS define uma estrutura padronizada para a representação de ativos e seus aspectos, enquanto uma plataforma como o Eclipse BaSyx, que será descrita brevemente na

Seção 3.2.5, fornece mecanismos de infraestrutura para disponibilizar registros de AAS, hospedar e localizar submodelos, integrar dados e conectar as representações digitais aos sistemas físicos. Dessa forma, o AAS não deve ser confundido com uma plataforma de execução: ele define a estrutura e as interfaces da representação, enquanto componentes de infraestrutura, como os fornecidos pelo Eclipse BaSyx, podem implementá-la e conectá-la aos sistemas e aos dados do ambiente físico.

3.2.5. *Middleware* para a Construção e Operação de Gêmeos Digitais

Para concretizar o grande potencial dos gêmeos digitais, o seu desenvolvimento envolve diversos requisitos e desafios, incluindo a modelagem das entidades físicas, a integração de fontes de dados heterogêneas, a atualização contínua das representações digitais e a disponibilização de serviços de consulta, análise e interação.

Para atender a esses requisitos, plataformas de *middleware* têm sido propostas tanto pela academia quanto pela indústria para apoiar o desenvolvimento, a implantação e a operação de gêmeos digitais. Embora plataformas de *middleware* originalmente concebidas para domínios como a IoT [Razzaque et al. 2016, Toutsop et al. 2021] possam oferecer funcionalidades úteis nesse contexto, elas não necessariamente atendem a todos os requisitos de um sistema de gêmeo digital. Enquanto um *middleware* para IoT concentra-se, em geral, na conectividade de dispositivos, no transporte de dados e na abstração da heterogeneidade de *hardware* e de protocolos, um *middleware* para gêmeos digitais também precisa oferecer mecanismos para representar entidades e seus relacionamentos, gerenciar seu estado e integrar continuamente dados provenientes do ambiente físico [Almeida et al. 2026]. Além de apoiar a modelagem, integração e sincronização dos gêmeos digitais, essas plataformas também precisam fornecer mecanismos relacionados às vertentes discutidas na Seção 3.2.3, constituindo uma infraestrutura que articula os componentes do gêmeo digital e as tecnologias utilizadas para representar, comunicar e explorar suas informações.

Esta seção apresenta brevemente quatro plataformas utilizadas na construção de gêmeos digitais: *Azure Digital Twins* e *AWS IoT TwinMaker*, ambas soluções comerciais baseadas na nuvem, e *Eclipse BaSyx* e *Eclipse Ditto*, que são soluções de código aberto. Além dessas soluções, a literatura acadêmica apresenta algumas propostas de *middleware* para gêmeos digitais [Almeida et al. 2023, Almeida et al. 2026]. Todavia, de maneira geral, essas propostas tendem a concentrar-se em aspectos específicos, como comunicação, interoperabilidade industrial ou sincronização, enquanto as plataformas consideradas nesta seção oferecem conjuntos mais integrados de mecanismos para modelagem, integração e operação de gêmeos digitais.

Azure Digital Twins. O *Azure Digital Twins*[6] é uma plataforma em nuvem integrada ao ecossistema Azure, da Microsoft, que permite a criação de grafos de entidades de gêmeos digitais modeladas em DTDL, apresentada na Seção 3.2.4, e suas instâncias podem ser conectadas por relacionamentos que representam a estrutura do ambiente. A plataforma também oferece APIs para criação e gerenciamento de instâncias, consulta do grafo e integração com outros serviços.

[6] https://azure.microsoft.com/en-us/products/digital-twins

Uma vez materializadas na plataforma, as instâncias de entidades de gêmeos digitais podem ser atualizadas a partir de diferentes fontes de dados. Em arquiteturas baseadas no ecossistema Azure, o *Azure IoT Hub*[7] pode ser utilizado para conectar dispositivos e receber seus dados, que posteriormente são encaminhados ao Azure Digital Twins por mecanismos de integração apropriados. A plataforma também permite integrar dados provenientes de sistemas corporativos e de outras fontes por meio de APIs e de seu sistema de eventos. Dessa forma, o Azure Digital Twins funciona como uma camada de representação e contextualização dos dados, enquanto os serviços responsáveis pela aquisição e pelo processamento dos dados podem permanecer separados do grafo de gêmeos digitais.

A plataforma também oferece uma API de consulta que permite recuperar informações do ambiente de execução com base em propriedades, relacionamentos, propriedades de relacionamentos e modelos. Para visualização, o *Azure Digital Twins 3D Scenes* Studio[8] permite associar elementos de um modelo tridimensional às entidades do grafo e criar interfaces para monitorização e investigação de dados operacionais. As informações do grafo também podem ser integradas a outros serviços do Azure para armazenamento de histórico, análise e aplicações baseadas em IA.

A Figura 3.5 ilustra como o Azure Digital Twins poderia ser empregado no cenário do estádio inteligente. Cada setor poderia ser modelado como uma entidade (ver Código 3.1) enquanto sensores de temperatura, sistemas de controle de acesso, equipamentos de climatização e outros elementos seriam representados por tipos específicos definidos em DTDL. Os relacionamentos poderiam, por exemplo, indicar que um setor possui um determinado sensor ou que um equipamento de climatização atende a uma área específica, e os dados provenientes de sensores poderiam chegar por meio do Azure IoT Hub ou de outras fontes, atualizando as propriedades correspondentes das instâncias no grafo. Adicionalmente, o Azure Digital Twins 3D Scenes Studio poderia então associar o modelo tridimensional do estádio às entidades do grafo, permitindo selecionar visualmente um setor e consultar sua ocupação, temperatura ou outros indicadores operacionais.

AWS IoT TwinMaker. O *AWS IoT TwinMaker*[9] é um serviço da Amazon Web Services (AWS) voltado à construção de gêmeos digitais operacionais de sistemas físicos e digitais. A plataforma permite modelar dispositivos, equipamentos, espaços e processos como entidades, às quais são associados componentes que organizam propriedades e referências a fontes de dados. Os dados podem permanecer em diferentes sistemas e ser acessados por meio de conectores, o que permite integrar informações provenientes de sensores, câmeras e aplicações empresariais sem necessariamente transferi-las para um único repositório. As entidades e seus componentes são organizados em *workspaces*, que constituem o contexto de um sistema de gêmeo digital. Ademais, relacionamentos entre entidades podem ser representados por propriedades específicas e utilizados para construir um grafo de conhecimento no qual consultas podem ser realizadas. A plataforma oferece ainda uma API unificada para acesso aos dados conectados e mecanismos para integrar os resultados às aplicações que consomem o gêmeo digital.

A visualização constitui outro aspecto central da plataforma. O AWS IoT Twin-

[7] https://azure.microsoft.com/en-us/products/iot-hub

[8] https://learn.microsoft.com/en-us/azure/digital-twins/quickstart-3d-scenes-studio

[9] https://aws.amazon.com/iot-twinmaker/



Figura 3.5. Exemplo de utilização do Azure Digital Twins no cenário do estádio inteligente

Maker permite construir cenas tridimensionais em que modelos 3D são associados às respectivas entidades, possibilitando a apresentação de dados operacionais no contexto espacial do ambiente. Essas cenas podem ser utilizadas em aplicações Web, inclusive com componentes fornecidos pelo *AWS IoT AppKit*[10], e integradas a ferramentas de monitoramento e visualização, como o *Amazon Managed Grafana*[11].

A Figura 3.6 ilustra como o AWS IoT TwinMaker poderia ser empregado no cenário do estádio inteligente. Um *workspace* poderia representar todo o ambiente, com entidades correspondentes a setores, portões, sensores, câmeras e sistemas de climatização. Componentes poderiam associar, por exemplo, uma entidade referente a um setor do estádio aos dados de ocupação provenientes de um sistema de monitoramento e aos dados de temperatura armazenados em uma fonte de séries temporais, enquanto as relações entre setores, sensores e equipamentos comporiam o grafo de conhecimento do estádio. Por sua vez, uma cena 3D poderia representar espacialmente os setores e equipamentos, associando cada elemento do modelo tridimensional à entidade correspondente e permitindo que os operadores visualizem os dados operacionais diretamente sobre a representação do estádio.

Eclipse BaSyx. Entre as soluções de código aberto, o *Eclipse BaSyx*[12] constitui uma infraestrutura particularmente relevante para ambientes industriais [Kannoth et al. 2021] e para a implementação de gêmeos digitais baseados no AAS, apresentada na Seção 3.2.4. Originalmente desenvolvido no contexto de um projeto liderado pelo Fraunhofer IESE, na Alemanha, e atualmente mantido pela Eclipse Foundation, o Eclipse BaSyx oferece componentes para hospedar AAS e submodelos, realizar seu registro e sua descoberta,

[10] https://aws.amazon.com/blogs/iot/build-iot-applications-using-aws-iot-application-kit/
[11] https://aws.amazon.com/grafana/
[12] https://basyx.org/



Figura 3.6. Exemplo de utilização do AWS IoT TwinMaker no cenário do estádio inteligente

integrar dados provenientes de ativos físicos e disponibilizar informações e serviços para aplicações. A plataforma também oferece mecanismos de integração com tecnologias industriais e de IoT, incluindo OPC UA, MQTT e HTTP/REST.

A Figura 3.7 apresenta uma visão geral simplificada da arquitetura do Eclipse BaSyx, destacando os componentes responsáveis pela hospedagem de AAS, submodelos, registro, descoberta, integração com ativos físicos e disponibilização de dados às aplicações. Os elementos centrais da arquitetura, agregados no *AAS Environment*, incluem: o *AAS Repository*, que gerencia o ciclo de vida das entidades, armazenando a estrutura do AAS e permitindo operações de criação, leitura, atualização e remoção sobre o cabeçalho e os metadados do gêmeo digital; o *Submodel Repository*, responsável pelo armazenamento efetivo dos dados técnicos e operacionais, gerenciando os submodelos e seus elementos internos e atuando como o banco de dados onde as propriedades estáticas e dinâmicas do ativo residem; e o *ConceptDescription Repository*, que gerencia os dicionários semânticos locais, armazenando as definições que atribuem significado às propriedades e permitindo que sistemas clientes consultem o significado exato de cada dado armazenado. Além desses elementos, a plataforma permite conectar ativos reais ou simulados a submodelos dinâmicos e suporta diferentes protocolos de comunicação e oferece integrações para análise e visualização de dados, permitindo que aplicações externas consumam as informações disponibilizadas pelos AAS.

No cenário do estádio inteligente, cada equipamento ou sistema relevante poderia ser representado por um AAS. A título de exemplo, um sistema de climatização poderia possuir submodelos para identificação, dados operacionais, manutenção e controle, como ilustrado anteriormente na Figura 3.4. Por sua vez, sensores de temperatura e umidade seriam integrados aos submodelos dinâmicos, enquanto os sistemas de controle poderiam



Figura 3.7. Visão geral do Eclipse BaSyx

ser associados a operações disponibilizadas pelos submodelos correspondentes. Setores do estádio também poderiam ser representados por AAS e relacionados aos equipamentos que os atendem, permitindo construir uma estrutura hierárquica na qual as informações de diferentes equipamentos são agregadas para representar o estado de uma área do estádio. Assim, aplicaç de monitoramento poderiam consumir esses submodelos, enquanto aplicações de manutenção ou automação utilizariam, respectivamente, informações de ciclo de vida e serviços de controle.

Eclipse Ditto. O *Eclipse Ditto*[13] é uma plataforma de código aberto voltada à implementação de gêmeos digitais em ambientes IoT [Kherbache et al. 2022]. Seu elemento central é a *Thing*, que representa digitalmente um dispositivo ou um ativo físico e pode conter propriedades e características (*features*) que descrevem diferentes aspectos desse ativo. Como principais funcionalidades, a plataforma fornece mecanismos de persistência e de gerenciamento do estado das *Things*, controle de acesso, busca e integração com sistemas externos.

A arquitetura do Eclipse Ditto, apresentada na Figura 3.8, é composta por cinco microsserviços especializados: *Policies* gerencia as políticas de autorização; *Things* gerencia as representações digitais e suas respectivas características; *Things-Search* mantém índices para busca; *Gateway* disponibiliza interfaces de acesso, incluindo HTTP e WebSocket; e *Connectivity* integra a plataforma a sistemas e brokers externos. Esses serviços operam de forma distribuída e permitem que a plataforma seja implantada em diferentes ambientes de infraestrutura.

A comunicação com as *Things* utiliza o *Ditto Protocol*[14], uma especificação baseada em JSON que define a semântica de comandos, eventos e mensagens independentemente do protocolo de transporte. O Ditto Protocol adota dois canais principais de comu-

[13] https://github.com/eclipse-ditto/ditto
[14] https://eclipse.dev/ditto/protocol-overview.html



Figura 3.8. Serviços do Eclipse Ditto

nicação: o canal *Twin* permite consultar e atualizar o estado persistido da representação digital, enquanto o canal *Live* encaminha mensagens diretamente ao dispositivo ou ao gateway conectado, possibilitando a execução de comandos e a recepção de respostas em tempo real. Essa separação permite tratar, de maneira distinta, o estado persistido do gêmeo e a comunicação direta com o ativo físico, ao mesmo tempo em que o protocolo garante que o modelo digital preserve estados históricos e responda de forma imediata a interações, assegurando consistência, rastreabilidade e flexibilidade operacional.

No cenário do estádio inteligente, cada sensor, equipamento ou sistema poderia ser representado no Eclipse Ditto como uma *Thing*. Um sensor de temperatura poderia possuir uma propriedade *temperatura*, enquanto um sistema de climatização poderia apresentar características correspondentes ao estado operacional e às funções de controle. Já o serviço *Connectivity* poderia integrar as *Things* aos sensores e equipamentos por meio do protocolo MQTT ou de outros protocolos suportados, atualizando o estado persistido no canal *Twin*. Quando uma ação precisasse ser executada diretamente no equipamento, como alterar a temperatura de referência de um sistema de climatização, uma aplicação poderia utilizar o canal *Live* para encaminhar o comando ao dispositivo. O serviço *Things-Search* permitiria localizar equipamentos com base em propriedades ou metadados, enquanto o serviço *Gateway* disponibilizaria essas informações às aplicações de monitoramento e de operação.

Comparativo. A Tabela 3.1 apresenta um comparativo entre as quatro plataformas, considerando os principais aspectos relevantes para sua utilização na construção e operação de gêmeos digitais. Embora todas ofereçam mecanismos para representar entidades, integrar dados e disponibilizar o estado dos gêmeos digitais às aplicações, elas adotam abordagens distintas quanto à modelagem, conectividade, gerenciamento de estado, visualização e implantação. Em conjunto, essas capacidades mostram como um *middleware* pode articular os mecanismos de comunicação e interoperabilidade, a integração de dados e os recursos de visualização e interação discutidos na Seção 3.2.3, fornecendo a infraestrutura necessária para incorporar essas tecnologias aos gêmeos digitais.

Tabela 3.1. Comparativo de plataformas de *middleware* para construção e operação de gêmeos digitais

Aspecto	**Azure Digital Twins**	**AWS IoT TwinMaker**	**Eclipse BaSyx**	**Eclipse Ditto**
Modelo de representação	Modelos DTDL e grafo de gêmeos digitais	Entidades, componentes e grafo de conhecimento	AAS e submodelos	*Things*, propriedades e características
Principal contexto de uso	Ambientes e sistemas conectados	Gêmeos operacionais de ambientes físicos e digitais	Ambientes industriais e Indústria 4.0	IoT e gerenciamento de gêmeos de dispositivos/ativos
Representação de relações	Relacionamentos explícitos no grafo de gêmeos digitais	Relações entre entidades e componentes	Referências e estrutura de AAS e de submodelos	Relações podem ser modeladas por propriedades/metadados e integradas externamente
Integração de dados	APIs, sistema de eventos e integração com serviços Azure	Conectores e API unificada para fontes de dados	*Data Bridge*, APIs e protocolos industriais/IoT	Serviço *Connectivity* e *Ditto Protocol*
Protocolos e tecnologias de integração	APIs REST, eventos, *Azure IoT Hub* e serviços Azure	Conectores AWS e conectores personalizados	HTTP/REST, MQTT, OPC UA, entre outros	MQTT, AMQP, HTTP, Apache Kafka, entre outros
Gerenciamento de estado	Estado das instâncias no grafo de gêmeos digitais	Dados associados às entidades e componentes	Submodelos estáticos e dinâmicos	Estado persistido (canal *Twin*) e comunicação direta (canal *Live*)
Consultas e descoberta	API de consulta do grafo de gêmeos digitais	Grafo de conhecimento e consultas sobre entidades e relações	Registro e descoberta de AAS/submodelos	Serviço *Things-Search*
Visualização	Azure 3D Scenes Studio e aplicações personalizadas	Cenas 3D, *AWS IoT AppKit* e integrações de visualização	Integrações para dashboards e aplicações externas	Interfaces Web e aplicações externas
Controle/ações	Integração com serviços e sistemas externos	Integração com fontes e serviços AWS	Operações em submodelos e componentes de controle	Canal *Live* e comandos ao dispositivo
Implantação	Serviço gerenciado em nuvem (Azure)	Serviço gerenciado em nuvem (AWS)	Código aberto, implantável em infraestrutura própria	Código aberto, implantável em infraestrutura própria ou na nuvem
Padrão/tecnologia central	DTDL	Modelo proprietário de entidades e componentes	AAS	*Ditto Protocol*
Principal diferencial	Grafo semântico do ambiente integrado ao ecossistema Azure e visualização 3D	Integração de fontes heterogêneas e visualização operacional	Adoção do padrão AAS e arquitetura modular para ambientes industriais	Gerenciamento de estado, conectividade e interação com dispositivos

3.3. Aplicações de Gêmeos Digitais

Nos últimos anos, o conceito de gêmeos digitais tem sido explorado em uma ampla variedade de domínios de aplicação, o que se reflete no crescimento significativo da produção científica sobre o tema [Tao et al. 2019, Fuller et al. 2020, Sharma et al. 2022]. Embora os princípios fundamentais permaneçam semelhantes (representação digital, conexão com uma contraparte do mundo real, aquisição e integração de dados e utilização de modelos e serviços para análise ou tomada de decisão), os requisitos variam consideravelmente entre os domínios. Em alguns casos, o principal objetivo é monitorar o estado de um ativo; em outros, prever falhas, otimizar operações, testar cenários ou apoiar decisões humanas.

Essa diversidade também demonstra que não há um único tipo de gêmeo digital. Um gêmeo digital de uma máquina industrial pode operar com dados de sensores de alta frequência e modelos físicos específicos, enquanto um gêmeo digital em um ambiente urbano pode integrar informações geoespaciais, meteorológicas, ambientais, de mobilidade e de infraestrutura provenientes de milhares de fontes. Da mesma forma, um gêmeo digital aplicado à saúde pode combinar dados clínicos, sinais fisiológicos e imagens médicas, introduzindo requisitos de segurança, privacidade e governança significativamente diferentes dos observados em aplicações industriais.

Esta seção apresenta seis domínios em que os gêmeos digitais têm sido empregados (ver Figura 3.9), destacando como diferentes combinações de sensoriamento, dados, modelos, simulação, IA, tecnologias Web e recursos multimídia podem responder a problemas específicos. Embora as tecnologias habilitadoras variem conforme o contexto, observa-se uma estrutura conceitual recorrente, baseada na aquisição e integração de dados, na modelagem do sistema, na análise e simulação e na disponibilização de resultados para monitoramento, visualização, interação ou tomada de decisão. Todavia, a forma como esses elementos são combinados depende da finalidade e das características do sistema representado.



Figura 3.9. Alguns domínios de aplicação que utilizam gêmeos digitais

3.3.1. Manufatura

A manufatura é um dos domínios mais associados aos gêmeos digitais e permanece entre os principais ambientes de pesquisa e aplicação do paradigma [Kritzinger et al. 2018],

inclusive com esforços específicos de padronização, como a série ISO 23247 [ISO 2021]. Nesse contexto, o conceito de gêmeos digitais está estreitamente relacionado à transformação digital promovida pela Indústria 4.0 [Lu et al. 2020], na qual máquinas, produtos, processos e sistemas produtivos passam a ser conectados e representados digitalmente. Dessa forma, um gêmeo digital pode representar uma máquina individual, uma linha de produção, um produto ou uma fábrica inteira. Adicionalmente, sensores instalados nos equipamentos fornecem informações sobre temperatura, vibração, pressão, velocidade, consumo de energia e outras variáveis operacionais, que podem alimentar modelos capazes de monitorar o estado dos equipamentos e identificar alterações indicativas de degradação ou falha.

Um dos usos mais conhecidos de gêmeos digitais na manufatura é a manutenção preditiva. Em vez de realizar intervenções exclusivamente em intervalos predeterminados, o sistema pode estimar a condição do equipamento e apoiar a previsão de falhas. Para isso, o gêmeo digital pode combinar dados históricos e em tempo real com modelos de degradação e técnicas de análise, auxiliando na determinação do momento mais apropriado para a intervenção [Errandonea et al. 2020].

Outra aplicação envolve a otimização de processos produtivos, na qual um modelo digital da linha de produção pode ser utilizado para simular diferentes configurações antes de sua implementação física. É possível, por exemplo, investigar o efeito da alteração da velocidade de uma máquina, da reorganização de uma linha ou da introdução de um novo equipamento. A simulação permite experimentar virtualmente diferentes alternativas, o que é particularmente relevante quando as alterações físicas são caras, arriscadas ou provocam interrupções na produção [Lu et al. 2020].

O conceito também pode ser aplicado ao próprio produto. Um produto fabricado pode possuir uma representação digital associada ao seu ciclo de vida, contendo informações sobre projeto, fabricação, utilização e manutenção. Essa representação permite preservar a continuidade das informações entre diferentes etapas, aproximando engenharia, produção e operação. Nesse contexto, mecanismos de identificação, interoperabilidade e troca de informações tornam-se importantes para manter a correspondência entre o produto físico e sua representação digital ao longo de todo o ciclo de vida.

3.3.2. Saúde

A área de saúde apresenta possibilidades particularmente promissoras para a utilização de gêmeos digitais, mas também desafios significativamente maiores em razão da complexidade dos sistemas biológicos e da sensibilidade dos dados envolvidos. Recentes revisões da literatura identificam aplicações relacionadas tanto ao acompanhamento e à personalização do cuidado quanto à gestão de operações em instituições de saúde [Sun et al. 2023, Katsoulakis et al. 2024].

Um dos cenários discutidos na literatura é o desenvolvimento de representações digitais de pacientes. A ideia consiste em integrar informações dos indivíduos, como dados clínicos, exames, sinais fisiológicos, histórico de tratamentos e dados provenientes de dispositivos de monitoramento, a modelos capazes de representar ou prever determinados aspectos da condição do paciente. Por exemplo, dados coletados por dispositivos vestíveis poderiam ser combinados com informações clínicas e modelos preditivos para identificar

alterações relevantes e apoiar o acompanhamento do paciente.

Outro caso de aplicação de gêmeos digitais envolve o planejamento e a personalização de tratamentos. Modelos computacionais podem ser utilizados para simular diferentes alternativas terapêuticas e comparar possíveis respostas antes da aplicação de determinada intervenção. Ademais, dados provenientes de exames de imagem também podem contribuir para a construção de representações individualizadas de estruturas anatômicas e apoiar análises específicas. Nesse contexto, o gêmeo digital não tem necessariamente o propósito de reproduzir integralmente o paciente, mas de fornecer uma representação adequada à finalidade clínica ou operacional considerada, sendo, assim, uma questão diretamente relacionada ao nível de fidelidade necessário ao gêmeo digital. Em sistemas de saúde, reproduzir integralmente um organismo seria impraticável e, em muitos casos, desnecessário. Por essa razão, o nível de detalhe deve ser determinado pela finalidade do gêmeo digital e pelas características a serem observadas, modeladas ou previstas. Assim, diferentes aplicações podem demandar diferentes níveis de abstração, de acordo com as variáveis relevantes para diagnóstico, tratamento, monitoramento ou gestão.

Finalmente, privacidade, segurança, governança, viés e aspectos éticos tornam-se centrais nesse domínio. Dados de saúde podem ser altamente sensíveis, e erros de modelagem ou previsões incorretas podem ter consequências significativamente diferentes das observadas em aplicações como a manutenção de equipamentos. Esses requisitos condicionam não apenas os dados utilizados, mas também a forma de construção, validação e utilização dos gêmeos digitais na saúde.

3.3.3. Transporte

Sistemas de transporte constituem outro domínio no qual gêmeos digitais podem representar simultaneamente ativos físicos, infraestrutura, fluxos e processos. Um gêmeo digital pode representar um veículo individual, uma ferrovia, uma rodovia, um aeroporto, um porto ou uma rede de transporte. Sensores instalados em veículos e na infraestrutura podem fornecer informações sobre localização, velocidade, condições mecânicas, tráfego e condições ambientais, e dados históricos e em tempo real podem ser utilizados para identificar anomalias, prever necessidades de manutenção e avaliar diferentes estratégias de operação [Gao et al. 2021, Wu et al. 2025].

Em rodovias urbanas, por exemplo, um gêmeo digital pode representar não apenas a infraestrutura física, mas também o fluxo de veículos. Dados provenientes de câmeras, sensores, veículos conectados e sistemas de posicionamento podem ser combinados para estimar as condições de tráfego, enquanto modelos de simulação podem avaliar o impacto de alterações na sinalização ou nas condições de circulação [Kušić et al. 2023]. Nesse cenário, as tecnologias multimídia discutidas na Seção 3.2.3 assumem particular relevância: câmeras de trânsito fornecem informações visuais sobre congestionamentos, acidentes e comportamento dos veículos; sistemas de informação geográfica fornecem representações espaciais; e modelos tridimensionais podem representar pontes, túneis, estações e outros elementos da infraestrutura.

Uma característica importante do domínio de transporte é que o sistema representado não é necessariamente um objeto isolado, mas um sistema dinâmico no qual múltiplas entidades interagem. Dessa forma, o gêmeo digital pode integrar dados espaciais,

temporais e multimodais para apoiar o monitoramento e a operação do sistema. Essa característica aproxima os gêmeos digitais de sistemas de transporte inteligentes, nos quais comunicação, computação de borda, fusão de dados, IA e tecnologias de visualização podem atuar de forma conjunta.

3.3.4. Cidades Inteligentes

A aplicação de gêmeos digitais em cidades inteligentes amplia ainda mais a escala e a heterogeneidade dos desafios relacionados ao seu desenvolvimento, implantação e operação. Em vez de representar uma máquina, um edifício ou uma linha de produção, um gêmeo digital pode representar bairros, sistemas de transporte, redes de energia, edifícios, espaços públicos, infraestrutura e serviços, dando origem ao conceito de "gêmeos digitais urbanos" [Ferré-Bigorra et al. 2022, Weil et al. 2023]. Nesse contexto, diferentes fontes podem fornecer dados sobre mobilidade, consumo de energia, iluminação pública, gerenciamento de resíduos, redes de água, condições meteorológicas e qualidade ambiental.

Um exemplo de aplicação de gêmeos digitais urbanos refere-se ao planejamento urbano [Deng et al. 2021]. Modelos digitais podem representar edifícios e infraestrutura existentes e permitir a simulação de diferentes intervenções. Por exemplo, pode-se investigar o impacto da construção de um novo edifício na circulação, na iluminação natural, no consumo de energia ou em determinadas condições ambientais. Outra possibilidade envolve o monitoramento ambiental, no qual sensores distribuídos pela cidade fornecem dados sobre temperatura, umidade, qualidade do ar, ruído e outros indicadores. Esses dados podem ser associados espacialmente às representações digitais dos bairros e utilizados para identificar padrões ou áreas críticas.

No contexto de cidades inteligentes, tecnologias Web e multimídia podem desempenhar um papel relevante na integração e na exploração dessas informações. Dados geoespaciais, imagens, modelos tridimensionais, mapas interativos e diferentes fontes de sensoriamento podem ser combinados para produzir representações contextualizadas dos ambientes urbanos. Por sua vez, as interfaces Web e os recursos de visualização interativa podem permitir que diferentes usuários explorem o estado da cidade e os resultados das simulações. Essa combinação reforça a relação entre gêmeos digitais urbanos e as tecnologias de visualização e interação discutidas na Seção 3.2.3.

A escala e a complexidade do cenário urbano, contudo, introduzem um desafio fundamental para a construção de gêmeos digitais: determinar quais aspectos da cidade devem ser representados e com que nível de detalhe. Uma representação digital não precisa reproduzir todos os elementos e características do ambiente urbano, mas deve contemplar aqueles que são relevantes para a finalidade a que se destina. Por exemplo, um gêmeo digital voltado ao planejamento energético pode priorizar informações sobre edificações, consumo e geração de energia, enquanto um gêmeo destinado à análise de mobilidade pode enfatizar vias, veículos, transporte público e padrões de deslocamento. Assim, diferentes aplicações podem demandar diferentes níveis de abstração e conjuntos de informações, mesmo quando o objeto representado é o mesmo. Essa característica reforça a ideia de que um gêmeo digital é construído de acordo com uma finalidade específica, e que sua utilidade depende não da reprodução integral do sistema real, mas da representação dos aspectos relevantes para o contexto de uso, mesmo quando representa um sistema

de grande escala, como é o caso de uma cidade.

3.3.5. Energia

O setor de energia apresenta aplicações de gêmeos digitais na geração, transmissão, distribuição, armazenamento e consumo [Ghenai et al. 2022, do Amaral et al. 2023]. Usinas de geração de energia podem possuir representações digitais de turbinas, geradores e demais equipamentos, enquanto sensores fornecem informações sobre temperatura, pressão, vibração, velocidade e eficiência. Por sua vez, modelos físicos e estatísticos podem ser utilizados para monitorar a condição dos equipamentos, detectar anomalias e prever falhas. Em parques eólicos, por exemplo, um gêmeo digital pode integrar informações sobre turbinas, condições meteorológicas e produção de energia, permitindo estimar o desempenho esperado e identificar desvios em relação ao comportamento normal.

Na transmissão e na distribuição de energia, a escala do problema aumenta, uma vez que as redes elétricas são compostas por múltiplos equipamentos interdependentes. Como alterações em uma parte da rede podem afetar outras regiões, é importante utilizar modelos que representem não apenas ativos individuais, mas também suas interações. Um gêmeo digital pode, assim, integrar dados operacionais e modelos de rede para avaliar diferentes condições, identificar anomalias e apoiar decisões relacionadas à operação e ao planejamento [Heluany and Gkioulos 2023].

Gêmeos digitais também podem ser utilizados no gerenciamento de edificações e do consumo de energia [Cespedes-Cubides and Jradi 2024]. Um edifício pode possuir uma representação digital dos sistemas de iluminação, climatização, geração distribuída e armazenamento, enquanto dados de ocupação e condições ambientais são combinados com modelos energéticos para avaliar e otimizar o consumo de energia. Nesse contexto, recursos de visualização podem permitir o acompanhamento do desempenho energético e a exploração de diferentes cenários de operação, aproximando os gêmeos digitais das tecnologias de visualização interativa discutidas anteriormente.

3.3.6. Agricultura

Na agricultura, os gêmeos digitais podem representar propriedades agrícolas, áreas cultivadas, culturas, equipamentos e condições ambientais [Pylianidis et al. 2021, Wang 2024]. Sensores instalados no solo podem medir umidade, temperatura e características físico-químicas, como também estações meteorológicas fornecem informações sobre precipitação, radiação e temperatura. Por sua vez, drones e satélites produzem imagens que podem ser analisadas para identificar estresse hídrico, doenças ou diferenças no desenvolvimento das plantas. Dessa forma, a natureza multimodal dos dados é evidente: informações numéricas produzidas por sensores podem ser combinadas com imagens aéreas e dados geoespaciais, enquanto algoritmos de visão computacional e modelos agronômicos podem apoiar o monitoramento e a previsão de variáveis de interesse.

A título de exemplo, um gêmeo digital de uma área cultivada poderia manter uma representação espacial das diferentes regiões da propriedade e associar a cada uma delas informações sobre o solo, a umidade, a cultura e o histórico de produção. Com isso, modelos poderiam simular diferentes estratégias de irrigação ou de fertilização, permitindo avaliar alternativas antes de sua implementação física. Observam-se, nesse contexto, apli-

cações principalmente relacionadas ao monitoramento de culturas, manejo do solo, irrigação e máquinas agrícolas [Purcell and Neubauer 2023].

A utilização de tecnologias de borda também pode ser importante nesse domínio, especialmente em áreas rurais com conectividade limitada. Parte do processamento pode ocorrer próxima aos sensores ou equipamentos, reduzindo a necessidade de transmitir continuamente grandes volumes de dados para a nuvem, enquanto informações consolidadas podem ser sincronizadas posteriormente com serviços remotos. Essa combinação entre sensoriamento distribuído, processamento de dados, conectividade e modelos digitais demonstra como diferentes tecnologias habilitadoras podem ser integradas de acordo com as características do domínio e os requisitos da aplicação.

3.4. Desafios em Gêmeos Digitais

A diversidade de tecnologias, arquiteturas e aplicações discutida nas seções anteriores demonstra que os gêmeos digitais já ultrapassaram a condição de conceito experimental e constituem uma área de investigação e desenvolvimento com aplicações em diferentes domínios. Entretanto, a construção de um gêmeo digital útil, confiável e interoperável permanece um problema científico e tecnológico complexo [Sharma et al. 2022]. A dificuldade não decorre apenas da integração de sensores, redes e modelos, mas também da necessidade de manter, ao longo do tempo, uma relação consistente entre o sistema físico, sua representação digital, os dados, os modelos, as pessoas e os processos organizacionais.

Esses desafios tornam-se particularmente relevantes quando os gêmeos digitais incorporam tecnologias multimídia e da Web. Dados provenientes de vídeo, áudio, imagens, modelos tridimensionais, sensores e fontes geoespaciais apresentam características temporais, espaciais e semânticas distintas. Ao mesmo tempo, a Web amplia a necessidade de identificação, descoberta, interoperabilidade, exposição de serviços e composição de sistemas distribuídos. Assim, o desenvolvimento de gêmeos digitais exige não apenas tecnologias capazes de conectar o mundo físico ao digital, mas também mecanismos para integrar, interpretar, validar, proteger e utilizar essas informações de forma confiável.

Esta seção apresenta um conjunto não exaustivo de desafios atuais relacionados aos gêmeos digitais, resumidos na Figura 3.10. De modo geral, a principal dificuldade não está na indisponibilidade individual dos paradigmas e tecnologias habilitadoras, haja vista que estes já fornecem grande parte dos recursos necessários à construção e à operação desses sistemas. O problema central reside em integrar esses recursos de forma coerente, transformando representações digitais isoladas em gêmeos digitais confiáveis, interoperáveis, seguros e capazes de permanecer adequados à sua finalidade ao longo do tempo.

3.4.1. Padronização, Semântica e Interoperabilidade

Um dos desafios fundamentais relacionados aos gêmeos digitais está na coexistência de diferentes padrões, modelos, *frameworks*, linguagens e plataformas. Embora iniciativas como a ISO/IEC 30173:2023, a ISO 23247-1:2021, o AAS e a DTDL representem avanços importantes, elas possuem objetivos, escopos e comunidades de adoção distintos. A ISO/IEC 30173:2023 estabelece conceitos e terminologia para gêmeos digitais, enquanto



Figura 3.10. Alguns desafios atuais em gêmeos digitais

a ISO 23247-1:2021 define princípios gerais para um *framework* de gêmeos digitais aplicado à manufatura. O AAS e a DTDL, por sua vez, oferecem mecanismos para a representação estruturada de entidades e suas características em diferentes ecossistemas tecnológicos. Essa diversidade não constitui necessariamente um problema, pois diferentes modelos podem ser adequados a diferentes domínios e finalidades. O desafio está em permitir que essas representações coexistam e cooperem entre si.

Essa questão torna-se especialmente importante em sistemas compostos por múltiplos gêmeos digitais, nos quais equipamentos, processos, instalações e sistemas organizacionais podem possuir representações desenvolvidas segundo diferentes especificações. Um equipamento pode possuir uma representação específica do seu respectivo fabricante, enquanto uma fábrica pode utilizar outra representação, um sistema de manutenção pode utilizar um terceiro modelo e uma plataforma de gêmeos digitais pode adotar seu próprio modelo de entidades. Portanto, a integração entre esses elementos exige mais do que conectividade: é necessário estabelecer correspondências semânticas entre conceitos, propriedades, eventos e relações.

No contexto de sistemas multimídia e Web, esse problema também inclui a necessidade de associar diferentes recursos digitais às entidades representadas pelo gêmeo. Por exemplo, uma câmera de vídeo não deve ser compreendida apenas como uma fonte de vídeo, mas como um recurso contextualizado espacial e temporalmente, associado a uma entidade, a uma localização, a um campo de visão e a um conjunto de eventos observáveis. Abordagens baseadas em ontologias e tecnologias da Web Semântica podem contribuir para representar essas relações e favorecer a descoberta, a composição e a interoperabilidade entre gêmeos digitais heterogêneos [Rico et al. 2023, Karabulut et al. 2024].

Assim, o desafio não consiste em eliminar a diversidade de representações, mas em desenvolver mecanismos que permitam mapeá-las, relacioná-las e utilizá-las de forma interoperável [Acharya et al. 2024]. Entre as questões em aberto estão:

- Como estabelecer correspondências semânticas entre modelos de gêmeos digi-

tais definidos segundo diferentes padrões e ontologias?
- Como mecanismos baseados na Web podem favorecer a descoberta e a composição de gêmeos digitais heterogêneos?
- Em que medida ontologias, grafos de conhecimento e tecnologias da Web Semântica podem reduzir o custo de integração entre gêmeos desenvolvidos de forma independente?

3.4.2. Integração de Dados Multimodais e Sincronização

A interoperabilidade torna-se ainda mais complexa quando os dados utilizados pelos gêmeos digitais são heterogêneos e multimodais. Um sistema pode receber valores numéricos, eventos, textos, imagens, vídeos, áudio, modelos tridimensionais e dados geoespaciais, modalidades que apresentam diferentes estruturas, resoluções, frequências de atualização, latências e requisitos computacionais. Integrá-las exige estabelecer relações entre representações que descrevem o mesmo fenômeno sob diferentes perspectivas, preservando seu contexto, origem, qualidade e incerteza.

Em um estádio inteligente, uma estimativa de ocupação obtida por visão computacional precisa ser associada a uma região espacial e a um instante temporal, podendo ainda ser relacionada a informações provenientes de sensores ambientais, sistemas de acesso ou modelos de circulação. O problema, portanto, não consiste simplesmente em combinar diferentes fontes, mas em estabelecer um alinhamento espaço-temporal e semântico entre os dados e preservar as informações necessárias para sua interpretação e uso posterior. Essa questão é particularmente relevante quando dados derivados são utilizados por modelos analíticos ou mecanismos de decisão..

Outra questão nesse cenário consiste em determinar quais informações devem ser efetivamente incorporadas ao gêmeo digital. Em aplicações baseadas em vídeo, armazenar e transmitir continuamente todos os fluxos capturados pode ser desnecessário e computacionalmente dispendioso. Dependendo da finalidade, pode ser mais apropriado extrair eventos, características ou metadados próximos à fonte, mantendo referências aos dados originais. Assim, o processamento na borda pode transformar fluxos multimídia em informações estruturadas, enquanto serviços Web podem disponibilizar essas informações de forma seletiva para aplicações e outros gêmeos digitais.

A sincronização entre os mundos físico e virtual é fundamental para que a representação digital permaneça adequada ao estado, à estrutura e ao comportamento da entidade representada. Por outro lado, diferentes fontes de dados para alimentar gêmeos digitais apresentam frequências e latências distintas. Uma câmera pode produzir dezenas de quadros por segundo, sensores ambientais podem atualizar seus valores a cada segundo ou minuto, dados de consumo podem ser agregados periodicamente e informações de manutenção podem ser registradas somente quando ocorre uma intervenção. Consequentemente, a representação digital precisa lidar com estados que não são necessariamente observados simultaneamente.

De maneira adicional, sincronizações muito frequentes podem impor custos elevados de comunicação, processamento e armazenamento e, dependendo da aplicação, até mesmo ser desnecessárias, enquanto sincronizações pouco frequentes podem fazer com que a representação digital se desvie do sistema físico, comprometendo a precisão de aná-

lises, predições e decisões. Portanto, a frequência e a estratégia de sincronização devem ser compatíveis com a finalidade do gêmeo digital, sendo necessário também determinar quando e como realizar a sincronização, equilibrando a atualidade e a qualidade da representação com os recursos computacionais e de comunicação necessários para mantê-la [Tan and Matta 2023]

As principais questões em aberto aqui são:

- Como integrar dados multimodais em gêmeos digitais preservando, ao mesmo tempo, seu contexto, origem e eventuais incertezas?
- Quais mecanismos podem assegurar a sincronização de dados produzidos em diferentes escalas temporais e espaciais?
- Como determinar a frequência e as condições de sincronização que equilibrem a atualidade da representação digital e os requisitos de precisão com os custos de comunicação e processamento?
- Como determinar quais dados devem ser processados, transmitidos ou armazenados na borda, na infraestrutura local ou na nuvem, considerando simultaneamente requisitos de desempenho, qualidade e custo?

3.4.3. Fidelidade, Verificação, Validação e Incerteza

Outro desafio relevante está relacionado à adequação da representação digital ao propósito para o qual o gêmeo digital será utilizado. É tentador assumir que, quanto mais detalhado for um modelo, melhor será o gêmeo digital; todavia, essa relação não é necessariamente verdadeira. Um modelo excessivamente complexo pode demandar recursos computacionais elevados, ser difícil de validar ou representar detalhes que não contribuem para a finalidade da aplicação. Dessa forma, a questão mais apropriada é determinar se a representação possui fidelidade suficiente para o propósito em questão.

A avaliação da fidelidade torna-se particularmente importante quando o gêmeo digital é utilizado para previsão, experimentação ou tomada de decisão. Nesses casos, são necessárias atividades de verificação, validação e quantificação de incerteza (VVUQ, abreviação de *verification, validation, and uncertainty quantification*) para estabelecer a credibilidade dos resultados [Shao et al. 2023], o que foi, inclusive, destacado como um aspecto fundamental para gêmeos digitais em um recente relatório do National Institute of Standards and Technology (NIST), dos Estados Unidos [Frechette et al. 2026]. A verificação tem como objetivo determinar se o modelo foi implementado corretamente em relação à sua especificação, enquanto a validação busca avaliar se ele representa adequadamente o fenômeno ou sistema de interesse para determinado propósito. A quantificação de incerteza, por sua vez, procura caracterizar como as incertezas presentes nos dados, parâmetros, modelos e condições de execução afetam os resultados.

Em gêmeos digitais, as atividades de VVUQ estão fortemente relacionadas, pois quanto mais heterogêneo e distribuído o sistema, maior a necessidade de padrões e mecanismos de integração; quanto mais complexos os modelos, maior a necessidade de validação; e quanto mais o gêmeo digital participa de processos decisórios ou atua sobre o sistema físico, maiores são os requisitos de segurança, credibilidade e confiança. Além disso, elas não podem ser tratadas como procedimentos pontuais realizados apenas durante o desenvolvimento do modelo, uma vez que, como os dados são continuamente atu-

alizados e tanto o sistema físico quanto seus modelos podem evoluir ao longo do tempo, é necessário preservar a credibilidade do gêmeo digital durante sua operação. Isso também envolve identificar quando mudanças nos dados, nos parâmetros, nas condições de operação ou no próprio sistema físico podem comprometer a validade dos modelos e demandar novas atividades de VVUQ.

Finalmente, a correspondência entre o modelo digital e o sistema físico correspondente também pode se deteriorar ao longo do tempo. Por exemplo, uma câmera de vídeo pode ser reposicionada, uma máquina pode ser substituída ou algum tipo de infraestrutura pode ser modificado. Nesse caso, mesmo que o modelo digital permaneça internamente consistente, sua relação com o sistema físico pode deixar de ser válida. A manutenção da credibilidade exige, assim, mecanismos capazes de detectar mudanças, recalibrar modelos, atualizar representações e reavaliar a sua validade ao longo do ciclo de vida do gêmeo digital [Shao et al. 2023].

Entre as principais questões relacionadas a fidelidade e incerteza estão:

- Como determinar o nível de fidelidade necessário para que um gêmeo digital seja adequado a uma finalidade específica?
- Como estabelecer procedimentos sistemáticos de VVUQ ao longo do ciclo de vida de um gêmeo digital?
- Como identificar e tratar a perda de validade de modelos à medida que o sistema físico, seus dados e seu contexto de operação se modificam?

3.4.4. Experimentação e Tomada de Decisão

As aplicações apresentadas na Seção 3.3 mostram que um gêmeo digital pode desempenhar funções que vão além da mera representação e do monitoramento do estado atual de um sistema. Quando associado a modelos analíticos ou de simulação, um gêmeo digital também pode ser utilizado para explorar estados possíveis, avaliar alternativas e antecipar as consequências de diferentes condições ou ações antes de sua implementação no sistema físico. Entretanto, essa capacidade não é inerente a toda implementação de gêmeo digital, pois depende dos dados, modelos e mecanismos analíticos incorporados ao sistema, bem como da finalidade para a qual foi concebido [Wooley et al. 2023]. À medida que o gêmeo digital passa de uma função predominantemente descritiva para funções preditivas ou prescritivas, aumentam também as exigências de credibilidade dos resultados [Shao et al. 2023].

A diferença entre monitoramento, previsão e experimentação pode ser expressa pela natureza das perguntas às quais o sistema é capaz de responder. Um mecanismo convencional de monitoramento pode responder a uma pergunta do tipo: *o que está acontecendo agora?* Um gêmeo digital associado a modelos preditivos pode contribuir para responder: *o que provavelmente acontecerá?* Quando modelos de simulação e mecanismos de análise de cenários são incorporados, pode-se ainda investigar: *o que poderá acontecer se determinada ação for realizada?*

No contexto de um estádio inteligente, por exemplo, diferentes níveis de ocupação podem ser simulados para avaliar seus efeitos sobre o consumo de energia, a temperatura ambiente e a circulação de pessoas. Na manufatura, pode-se investigar o impacto da introdução de uma nova máquina; no transporte, os efeitos de alterações na infraestru-

tura ou nas condições de circulação; na agricultura, diferentes estratégias de irrigação. A experimentação virtual permite, assim, comparar alternativas e antecipar possíveis consequências antes da implementação efetiva [dos Santos et al. 2020, Wooley et al. 2023]. Contudo, uma previsão ou simulação somente será útil se o modelo for adequado ao propósito considerado e se as incertezas associadas aos dados e aos parâmetros forem suficientemente compreendidas [Shao et al. 2023]. Com isso, a experimentação virtual não elimina a necessidade de validação; pelo contrário, ela torna a credibilidade dos modelos ainda mais importante.

As tecnologias multimídia e da Web também participam desse processo. Representações tridimensionais, mapas, vídeos, *dashboards* e interfaces interativas podem permitir que os usuários explorem diferentes estados e cenários e compreendam suas possíveis consequências. A visualização, nesse contexto, não se limita a uma representação do estado atual do sistema, mas faz parte da cadeia pela qual os resultados computacionais são transformados em informações compreensíveis para apoiar a interpretação e a tomada de decisão. Uma visualização inadequada pode ocultar incertezas, dificultar a comparação entre alternativas ou transmitir uma falsa impressão de precisão.

Dessa forma, a ampliação do papel de um gêmeo digital não representa apenas uma expansão de suas capacidades tecnológicas, mas também modifica o tipo de evidência necessário para confiar em seus resultados. Entre as questões em aberto estão:

- Como avaliar a credibilidade de previsões e experimentos realizados sobre um gêmeo digital antes que seus resultados sejam utilizados para apoiar decisões?
- Quais evidências são necessárias para considerar um resultado suficientemente confiável para fundamentar uma decisão?
- Como interfaces multimídia e tecnologias da Web podem comunicar cenários, resultados e incertezas de forma compreensível a diferentes perfis de usuários?

3.4.5. Modelos de IA

A incorporação crescente de modelos de aprendizado de máquina em gêmeos digitais representa tanto uma oportunidade quanto um desafio [Kreuzer et al. 2024]. Esses modelos podem identificar padrões complexos, detectar anomalias, transformar dados multimídia em informações estruturadas e produzir previsões sobre estados futuros do sistema [Nele et al. 2024]. Em sistemas multimídia, técnicas de visão computacional podem transformar fluxos de vídeo em eventos associados a pessoas, objetos ou condições ambientais. Esses modelos também podem combinar dados visuais, temporais, espaciais e sensoriais para produzir estimativas complexas que seriam obtidas por meio de regras explícitas.

Entretanto, modelos de inteligência introduzem problemas adicionais de explicabilidade, generalização, atualização e validação. Um modelo treinado com dados históricos pode apresentar desempenho inadequado quando o sistema físico ou o contexto de operação se modifica. Da mesma forma, a qualidade das previsões depende das características dos dados utilizados no treinamento e na inferência, aspecto particularmente relevante em domínios críticos, como saúde e segurança. Uma possibilidade para lidar com essas limitações consiste no uso de modelos híbridos, que combinam conhecimento físico ou baseado em modelos com aprendizado orientado a dados. Em muitos sistemas,

dados históricos, isoladamente, não são suficientes para representar corretamente comportamentos em condições novas. Dessa forma, modelos físicos podem fornecer restrições e conhecimento estrutural que complementam os padrões aprendidos. No estádio inteligente, por exemplo, um modelo térmico baseado em princípios físicos pode ser combinado com aprendizado de máquina para estimar parâmetros ou corrigir erros sistemáticos, ampliando a capacidade preditiva sem eliminar o conhecimento sobre o comportamento físico do sistema.

Mais recentemente, modelos de IA generativa vêm sendo investigados em conjunto com gêmeos digitais, como para apoiar a geração de dados sintéticos, a exploração de cenários e novas formas de interação [Chiaro et al. 2025, Li et al. 2025]. Modelos de linguagem podem atuar como interfaces para consultar informações, explicar resultados ou auxiliar os usuários na exploração de cenários, enquanto sistemas baseados em agentes podem coordenar diferentes serviços analíticos e modelos especializados. Essas possibilidades, contudo, devem ser tratadas como uma direção emergente, não como uma evolução necessária dos gêmeos digitais. Quando sistemas baseados em IA recebem capacidade de executar ações ou acionar sistemas externos, surgem ainda requisitos adicionais de autorização, supervisão, verificação e segurança.

Uma direção recente que amplia essa discussão é a chamada IA física (*physical AI*, em inglês), conceito associado a sistemas de inteligência capazes de perceber, raciocinar e atuar diretamente no mundo físico [Li et al. 2023, Wu et al. 2026]. Diferentemente de aplicações convencionais de IA, nas quais as entradas e saídas permanecem predominantemente no domínio digital, os sistemas de IA física operam em ciclos fechados que envolvem sensores, modelos computacionais e atuadores e estão sujeitos às restrições temporais, físicas e operacionais do ambiente. Essa característica estabelece uma relação interessante com os gêmeos digitais, uma vez que estes podem fornecer ambientes virtuais para treinamento, simulação, avaliação e teste de sistemas inteligentes antes de sua execução no mundo físico.

Uma revisão recente da literatura sobre a integração entre gêmeos digitais e IA física destaca o papel dos gêmeos digitais na integração entre o mundo virtual e o real, no controle adaptativo e na avaliação de sistemas físicos inteligentes [Biswas et al. 2026]. Por outro lado, essa relação é, na verdade, de complementaridade: enquanto o gêmeo digital pode representar e simular o comportamento de um sistema físico, a IA física acrescenta mecanismos de percepção, decisão e atuação capazes de modificar esse sistema. A integração dos dois pode, assim, estabelecer um ciclo contínuo entre realidade e simulação, no qual os dados do sistema físico atualizam o gêmeo digital, o gêmeo digital fornece condições para treinamento e experimentação, e os modelos de IA resultantes retornam ao ambiente físico para executar ações. Essa integração também evidencia desafios já discutidos nesta seção, em particular quanto à necessidade de representar incertezas, preservar sincronização temporal e garantir verificação, validação e segurança quando decisões automatizadas passam a produzir efeitos físicos.

Diante do exposto, o desafio consiste em investigar como a IA pode ampliar as capacidades analíticas dos gêmeos digitais sem comprometer sua credibilidade, explicabilidade e controle. Entre as questões em aberto estão:

- Como combinar modelos físicos, modelos orientados por dados e modelos de

IA de modo a preservar a explicabilidade, a generalização e a confiabilidade?
- Como utilizar gêmeos digitais para treinar e avaliar sistemas baseados em IA física e, ao mesmo tempo, garantir que modelos treinados em ambientes virtuais mantenham comportamento seguro e válido quando transferidos para o mundo físico?
- Como realizar VVUQ em sistemas nos quais modelos de IA não apenas produzem previsões, mas também influenciam ou executam ações sobre o sistema físico?

3.4.6. Interação Humano-Gêmeo Digital, Segurança e Privacidade

Os gêmeos digitais são desenvolvidos não apenas para realizar processamento automatizado, mas também para auxiliar as pessoas em suas tarefas. Um operador pode precisar compreender rapidamente a situação atual, identificar anomalias, comparar alternativas e decidir sobre uma ação. Nesse contexto, uma representação excessivamente complexa pode dificultar a interpretação em vez de facilitá-la. Desse modo, a interação humano–gêmeo digital constitui uma dimensão importante para que as informações produzidas pelo sistema sejam efetivamente utilizadas [Barricelli and Fogli 2022]. Essa questão aproxima os gêmeos digitais de áreas como visualização de informação, interação humano-computador e sistemas multimídia. Interfaces tridimensionais podem facilitar a compreensão espacial, enquanto *dashboards* podem apoiar a comparação de indicadores, vídeos podem fornecer evidências associadas a eventos e tecnologias de realidade aumentada podem sobrepor informações digitais ao ambiente físico. A questão central, contudo, não é escolher a interface visualmente mais sofisticada, mas determinar qual representação fornece a informação necessária para cada tarefa ou decisão. Quando o gêmeo digital apresenta previsões ou recomendações, a interface também deve fornecer contexto e, quando apropriado, evidências e informações sobre as incertezas associadas aos resultados.

Segurança e privacidade constituem desafios igualmente importantes e devem ser consideradas desde a concepção do gêmeo digital, e não apenas ao final do desenvolvimento [Voas et al. 2025]. A natureza conectada desses sistemas, que podem manter fluxos contínuos de dados entre o mundo físico e sua representação virtual e, em alguns casos, entre a representação virtual e atuadores físicos, cria riscos de segurança que podem acarretar consequências no mundo físico. Por exemplo, no estádio inteligente, o comprometimento de um gêmeo digital responsável por apoiar ou controlar sistemas de acesso ou sinalização poderia transformar uma vulnerabilidade digital em um risco operacional.

Adicionalmente, muitas aplicações de gêmeos digitais envolvem dados potencialmente sensíveis, como localização, imagens e dados biométricos. Isso levanta questões relacionadas à finalidade da coleta, à resolução, ao período de retenção, ao controle de acesso e ao compartilhamento de dados. O desafio não consiste apenas em proteger os dados armazenados, mas em estabelecer mecanismos de governança que definam quais dados devem ser coletados, quem pode acessá-los, para quais finalidades e sob quais condições [Voas et al. 2025].

As principais questões em aberto aqui estão relacionadas a como projetar a relação entre usuários e gêmeos digitais sem transformar a interação em uma mera camada de

visualização:

- Como representar estados, previsões e incertezas de maneira adequada às necessidades de diferentes usuários?
- Como equilibrar riqueza multimídia, transparência e utilidade com requisitos de privacidade e segurança?
- Como estabelecer mecanismos de confiança e controle quando os resultados produzidos pelo gêmeo digital passam a influenciar ou automatizar decisões?

3.4.7. Escalabilidade e Composição de Gêmeos Digitais

Gêmeos digitais podem crescer significativamente em tamanho e complexidade. Um gêmeo de uma máquina pode envolver centenas de variáveis, enquanto um gêmeo de uma fábrica pode envolver dezenas ou centenas de equipamentos. Já um gêmeo digital urbano pode envolver milhares de entidades, eventos e relações, ou muito mais que isso. Quando dados multimídia são incorporados, a escala aumenta ainda mais, pois fluxos contínuos de vídeo, modelos tridimensionais, dados espaciais e eventos em tempo real podem exigir elevada capacidade de processamento, armazenamento e comunicação.

Esse cenário cria um problema de escalabilidade, pois não basta desenvolver um protótipo funcional para um número limitado de entidades. É necessário avaliar se a arquitetura continua adequada à medida que o número de dispositivos, usuários, modelos, eventos e fontes de dados aumenta. A questão também se manifesta na composição de múltiplos gêmeos digitais, uma vez que sistemas reais são compostos por ativos, processos e organizações interdependentes. Um gêmeo digital de uma máquina pode interagir com o gêmeo digital de uma linha de produção, que, por sua vez, pode fazer parte do gêmeo digital de uma fábrica. Em ambientes urbanos, gêmeos digitais de edifícios, transporte, energia e infraestrutura podem precisar cooperar entre si.

Nesse cenário, torna-se importante desenvolver arquiteturas capazes de descobrir, conectar, sincronizar e utilizar gêmeos digitais especializados sem exigir integrações específicas entre cada par de sistemas. A Web pode contribuir para identificação, descrição, descoberta e acesso a recursos distribuídos, enquanto plataformas de *middleware* podem abstrair aspectos de comunicação, modelagem e integração [Almeida et al. 2026]. Em vez de tratar cada gêmeo digital como um sistema isolado, arquiteturas futuras poderão oferecer serviços comuns para descoberta, sincronização, consulta, composição, segurança e gestão do ciclo de vida.

A ideia de operar múltiplos gêmeos digitais em uma perspectiva de cooperação desloca a pesquisa de instâncias individuais para a construção de ecossistemas e sistemas de sistemas de gêmeos digitais, nos quais diferentes representações podem cooperar entre si preservando, ao mesmo tempo, sua autonomia e especificidades [Rivera et al. 2022, Cavalcante et al. 2025]. A transição para esses ecossistemas levanta, assim, questões relacionadas à escalabilidade da composição:

- Como compor múltiplos gêmeos digitais sem introduzir dependências que comprometam sua autonomia?
- Como distribuir o processamento, o armazenamento e a comunicação entre a borda, a infraestrutura local e a nuvem em ecossistemas de grande escala?
- Como preservar o desempenho, a interoperabilidade e a consistência quando

múltiplos gêmeos precisam operar de forma coordenada?

3.5. Conclusão

Este capítulo teve como objetivo fornecer uma visão abrangente sobre gêmeos digitais, abordando seus conceitos fundamentais, os principais paradigmas habilitadores, padrões e plataformas para seu desenvolvimento e operação, bem como os diferentes domínios de aplicação em que têm sido empregados. A discussão também destacou o papel das tecnologias multimídia e da Web na construção de gêmeos digitais, em particular na integração de dados multimodais, na comunicação e na interoperabilidade entre sistemas distribuídos, e na visualização e na interação com suas representações digitais. Essa relação é relevante, pois gêmeos digitais tipicamente dependem de capacidades que incluem a integração de fluxos de dados heterogêneos, arquiteturas de comunicação distribuída, mecanismos de modelagem semântica e interfaces interativas capazes de representar informações complexas. Nesse sentido, a aproximação entre gêmeos digitais, sistemas multimídia e tecnologias da Web amplia as possibilidades de atuação de estudantes, pesquisadores e profissionais dessas áreas no desenvolvimento de novos sistemas e aplicações.

A análise das aplicações descritas neste capítulo mostrou, ainda, que os gêmeos digitais podem assumir diferentes formas e níveis de complexidade, desde a representação e o monitoramento de ativos individuais até a integração de processos, infraestruturas e sistemas interdependentes. Nesse contexto, encontram-se em aberto diversos desafios e questões relacionadas a padronização e interoperabilidade, integração e sincronização de dados multimodais, fidelidade e credibilidade dos modelos, apoio à experimentação e à tomada de decisão, incorporação de modelos de IA, interação humano–gêmeo digital, segurança, privacidade, escalabilidade e composição de gêmeos digitais.

A partir dessa perspectiva, uma questão abrangente para pesquisas futuras consiste em investigar como projetar ecossistemas de gêmeos digitais confiáveis, seguros e interoperáveis, capazes de integrar dados e modelos heterogêneos, incorporar recursos multimídia, operar sobre infraestruturas Web distribuídas e utilizar técnicas de inteligência computacional de maneira controlada e adequada às finalidades de cada aplicação. Essa questão não implica uma trajetória tecnológica única para os gêmeos digitais, mas sim salienta a necessidade de combinar diferentes paradigmas, modelos e tecnologias de acordo com os requisitos do sistema físico e do contexto em que o gêmeo será empregado.

Como principal contribuição, este capítulo buscou articular uma perspectiva integrada sobre gêmeos digitais que conecta fundamentos conceituais, tecnologias habilitadoras, padrões, plataformas, aplicações e desafios de pesquisa. Especialmente, procurou-se explicitar a contribuição de tecnologias multimídia e da Web para além de seu papel como mecanismos de apresentação ou de acesso, destacando sua participação na integração de dados multimodais, na comunicação, na interoperabilidade, na representação semântica e na interação com os gêmeos digitais. Ao reunir essas perspectivas e relacioná-las aos desafios que ainda limitam a adoção e evolução desses sistemas, este capítulo ofereceu uma base conceitual para orientar leituras especializadas e apoiar a identificação de oportunidades de pesquisa e desenvolvimento. Espera-se, assim, que esta discussão contribua para uma compreensão mais integrada dos gêmeos digitais e estimule novas investigações, aplicações e soluções que explorem o potencial das tecnologias multimídia e da

Web na construção de sistemas digitais capazes de representar, analisar, experimentar e apoiar decisões sobre sistemas físicos cada vez mais complexos.

Agradecimentos

Este trabalho possui apoio do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), processo nº 421339/2023-3, e do Instituto Nacional de Ciência e Tecnologia em Redes de Comunicação e Internet das Coisas Inteligentes (INCT ICoNIoT), financiado pelo CNPq, processo nº 405940/2022-0, e pela Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), processo nº 88887.954253/2024-00.

Declaração sobre o Uso de Inteligência Artificial

Ferramentas de Inteligência Artificial generativa, especificamente Google Gemini (Gemini 3.6 Flash e Nano Banana 2), Claude (Sonnet 5) e ChatGPT (GPT-5.6 Luna), foram utilizadas na elaboração deste capítulo como apoio à produção de algumas figuras e à revisão da organização e do fluxo textual. Todos os elementos produzidos ou modificados com auxílio dessas ferramentas foram cuidadosamente revisados e validados pelos autores, que permanecem integralmente responsáveis pelo conteúdo, pela precisão das informações e pela qualidade final do capítulo, em plena conformidade com o Código de Conduta para Autores em Publicações da Sociedade Brasileira de Computação (SBC).

Referências

Acharya, S., Khan, A. A., and Päivärinta, T. (2024). Interoperability levels and challenges of digital twins in cyber–physical systems. *Journal of Industrial Information Integration*, 42.

Adolphsen, M. (2025). *Digital twin technology: Revolutionizing stadium management for the ultimate fan experience*. https://www.simio.com/blog/digital-twin-technology-revolutionizing-stadium-management.

Almeida, A., Batista, T., Cavalcante, E., Delicato, F., Motta, R., and Vieira, M. (2023). Middleware for digital twins: A systematic mapping study. In *Proceedings of the 1st International Workshop on Middleware for Digital Twin*, pages 19–24, USA. ACM.

Almeida, A., Pereira, L., Batista, T., Cavalcante, E., Delicato, F. C., and Motta, R. (2026). Construção de sistemas de gêmeos digitais: Uma abordagem baseada em middleware. In Sampaio, L. N., Freitas, A. E. S., Munaretto, A., Moraes, I. M., and Mansilha, R. B., editors, *Minicursos do SBRC 2026*, pages 210–245. SBC, Brasil.

Barricelli, B. R., Casiraghi, E., and Fogli, D. (2019). A survey on digital twin: Definitions, characteristics, applications, and design implications. *IEEE Access*, 146.

Barricelli, B. R. and Fogli, D. (2022). Digital twins in human-computer interaction: A systematic review. *International Journal of Human–Computer Interaction*, 40(2):79–97.

Biswas, P., Kim, T. M., and Kim, W. S. (2026). Integration of digital twins and physical AI in cyber-physical systems. *Intelligent Systems with Applications*, 30.

Bokhtiar Al Zami, M., Shaon, S., Quy, V. K., and Nguyen, D. C. (2025). Digital twin in industries: A comprehensive survey. *IEEE Access*, 13:47291–47336.

Cavalcante, E., Batista, T., and Oquendo, F. (2025). Exploring synergies and challenges of system-of-systems digital twins. In *2025 IEEE 22nd International Conference on Software Architecture Companion*, pages 190–196, USA. IEEE.

Cespedes-Cubides, A. S. and Jradi, M. (2024). A review of building digital twins to improve energy efficiency in the building operational stage. *Energy Informatics*, 7(1).

Chiaro, D., Qi, P., Pescapè, A., and Piccialli, F. (2025). Generative AI-empowered digital twin: A comprehensive survey with taxonomy. *IEEE Transactions on Industrial Informatics*, 21(6):4287–4295.

Deng, T., Zhang, K., and Shen, Z.-J. M. (2021). A systematic review of a digital twin city: A new pattern of urban governance toward smart cities. *Journal of Management Science and Engineering*, 6(2):125–134.

Digital Twin Consortium (2022). *The DTC Glossary*. https://www.digitaltwinconsortium.org/glossary/glossary/.

do Amaral, J., dos Santos, C., Montevechi, J., and de Queiroz, A. (2023). Energy digital twin applications: A review. *Renewable and Sustainable Energy Reviews*, 188.

dos Santos, C. H., de Queiroz, J. A., Leal, F., and Montevechi, J. A. B. (2020). Use of simulation in the industry 4.0 context: Creation of a digital twin to optimise decision making on non-automated process. *Journal of Simulation*, 16(3):284–297.

Eramo, R., Bordeleau, F., Combemale, B., van den Brand, M., Wimmer, M., and Wortmann, A. (2022). Conceptualizing digital twins. *IEEE Software*, 39(2):39–46.

Errandonea, I., Beltrán, S., and Arrizabalaga, S. (2020). Digital twin for maintenance: A literature review. *Computers in Industry*, 123.

Ferré-Bigorra, J., Casals, M., and Gangolells, M. (2022). The adoption of urban digital twins. *Cities*, 131.

Fitzgerald, J., Gomes, C., and Larsen, P. G. (2024). *The engineering of digital twins*. Springer Cham, Switzerland.

Frechette, S., Shao, G., and Pease, M. (2026). Digital Twins Workshops Summary Report. Technical Report NISTIR 8620, National Institute of Standards and Technology, USA.

Fuller, A., Fan, Z., Day, C., and Barlow, C. (2020). Digital twin: Enabling technologies, challenges and open research. *IEEE Access*, 8:108952–108971.

Gao, Y., Qian, S., Li, Z., Wang, P., Wang, F., and He, Q. (2021). Digital twin and its application in transportation infrastructure. In *Proceedings of the IEEE 1st International Conference on Digital Twins and Parallel Intelligence*, pages 298–301, USA. IEEE.

Ghenai, C., Husein, L. A., Nahlawi, M. A., Hamid, A. K., and Bettayeb, M. (2022). Recent trends of digital twin technologies in the energy sector: A comprehensive review. *Sustainable Energy Technologies and Assessments*, 54.

Grieves, M. and Vickers, J. (2017). Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems. In Kahlen, F.-J., Flumerfelt, S., and Alves, A.,

editors, *Transdisciplinary perspectives on complex systems: New findings and approaches*, pages 85–113. Springer, Cham, Switzerland.

Heluany, J. B. and Gkioulos, V. (2023). A review on digital twins for power generation and distribution. *International Journal of Information Security*, 23(2):1171–1195.

IDTA (2026). *Specification of the Asset Administration Shell – Part 1: Metamodel, Version 3.2*. Industrial Digital Twin Association. https://industrialdigitaltwin.io/aas-specifications/IDTA-01001/v3.2/index.html.

IEC (2023). IEC 63278-1:2023: *Asset Administration Shell for industrial applications – Part 1: Asset Administration Shell structure*.

ISO (2021). ISO 23247-1:2021: *Automation systems and integration — Digital twin framework for manufacturing*. Part 1: Overview and general principles. https://www.iso.org/standard/75066.html.

ISO (2023). ISO/IEC 30173:2023: *Digital twin – Concepts and terminology*. https://www.iso.org/standard/81442.html.

Jones, D., Snider, C., Nassehi, A., Yon, J., and Hicks, B. (2020). Characterising the digital twin: A systematic literature review. *CIRP Journal of Manufacturing Science and Technology*, 29:36–52.

Kannoth, S., Hermann, J., Damm, M., Rübel, P., Rusin, D., Jacobi, M., Mittelsdorf, B., Kuhn, T., and Antonino, P. O. (2021). Enabling SMEs to Industry 4.0 using the BaSyx Middleware: A case study. In Biffl, S., Navarro, E., Löwe, W., Sirjani, M., Mirandola, R., and Weyns, D., editors, *Software Architecture*, volume 12857 of *Lecture Notes in Computer Science*, pages 277–294. Springer Cham, Switzerland.

Karabulut, E., Pileggi, S. F., Groth, P., and Degeler, V. (2024). Ontologies in digital twins: A systematic literature review. *Future Generation Computer Systems*, 153:442–456.

Katsoulakis, E., Wang, Q., Wu, H., Shahriyari, L., Fletcher, R., Liu, J., Achenie, L., Liu, H., Jackson, P., Xiao, Y., Syeda-Mahmood, T., Tuli, R., and Deng, J. (2024). Digital twins for health: a scoping review. *npj Digital Medicine*, 7(1).

Kherbache, M., Maimour, M., and Rondeau, E. (2022). Digital twin network for the IIoT using Eclipse Ditto and Hono. *IFAC-PapersOnLine*, 55(8):37–42.

Kreuzer, T., Papapetrou, P., and Zdravkovic, J. (2024). Artificial intelligence in digital twins – a systematic literature review. *Data & Knowledge Engineering*, 151.

Kritzinger, W., Karner, M., Traar, G., Henjes, J., and Sihn, W. (2018). Digital twin in manufacturing: A categorical literature review and classification. *IFAC-PapersOnLine*, 51(11):1016–1022.

Kušić, K., Schumann, R., and Ivanjko, E. (2023). A digital twin in transportation: Real-time synergy of traffic data streams and simulation for virtualizing motorway dynamics. *Advanced Engineering Informatics*, 55.

Li, T., Long, Q., Chai, H., Zhang, S., Jiang, F., Liu, H., Huang, W., Jin, D., and Li, Y. (2025). Generative AI empowered network digital twins: Architecture, technologies, and applications. *ACM Computing Surveys*, 57(6).

Li, Y., Li, Z., Duan, Y., and Spulber, A.-B. (2023). Physical Artificial Intelligence (PAI): The next-generation Artificial Intelligence. *Frontiers of Information Technology & Electronic Engineering*, 24(8):1231–1238.

Lu, Y., Liu, C., Wang, K. I.-K., Huang, H., and Xu, X. (2020). Digital twin-driven smart manufacturing: Connotation, reference model, applications and research issues. *Robotics and Computer-Integrated Manufacturing*, 61.

Nele, L., Mattera, G., Yap, E. W., Vozza, M., and Vespoli, S. (2024). Towards the application of machine learning in digital twin technology: a multi-scale review. *Discover Applied Sciences*, 6(10).

Purcell, W. and Neubauer, T. (2023). Digital twins in agriculture: A state-of-the-art review. *Smart Agricultural Technology*, 3.

Pylianidis, C., Osinga, S., and Athanasiadis, I. N. (2021). Introducing digital twins to agriculture. *Computers and Electronics in Agriculture*, 184.

Razzaque, M. A., Milojevic-Jevric, M., Palade, A., and Clarke, S. (2016). Middleware for Internet of Things: A survey. *IEEE Internet of Things Journal*, 3(1):70–95.

Rico, M., Taverna, M. L., Galli, M. R., and Caliusco, M. L. (2023). Context-aware representation of digital twins' data: The ontology network role. *Computers in Industry*, 7:167653–167671.

Rivera, L. F., Jimenez, M., Villegas, N. M., Tamura, G., and Muller, H. A. (2022). The forging of autonomic and cooperating digital twins. *IEEE Internet Computing*, 26(5):41–49.

Shao, G., Hightower, J., and Schindel, W. (2023). Credibility consideration for digital twins in manufacturing. *Manufacturing Letters*, 35:24–28.

Sharma, A., Kosasih, E., Zhang, J., Brintrup, A., and Calinescu, A. (2022). Digital twins: State of the art theory and practice, challenges, and open research questions. *Journal of Industrial Information Integration*, 30.

Sun, T., He, X., and Li, Z. (2023). Digital twin in healthcare: Recent updates and challenges. *Digital Health*, 9.

Tan, B. and Matta, A. (2023). The digital twin synchronization problem: Framework, formulations, and analysis. *IISE Transactions*, 56(6):652–665.

Tao, F., Zhang, H., Liu, A., and Nee, A. Y. C. (2019). Digital twin in industry: State-of-the-art. *IEEE Transactions on Industrial Informatics*, 15(4):2405–2415.

Toutsop, O., Kornegay, K., and Smith, E. (2021). A comparative analyses of current IoT middleware platforms. In *Proceedings of the 8th International Conference on Future Internet of Things and Cloud*, pages 413–420, USA. IEEE.

Voas, J., Mell, P., Laplante, P., and Piroumian, V. (2025). Security and ttust considerations for digital twin technology. Technical Report NIST IR 8356, National Institute of Standards and Technology, USA.

Wang, L. (2024). Digital twins in agriculture: A review of recent progress and open issues. *Electronics*, 13(11).

Weil, C., Bibri, S. E., Longchamp, R., Golay, F., and Alahi, A. (2023). Urban digital twin challenges: A systematic review and perspectives for sustainable smart cities. *Sustainable Cities and Society*, 99.

Wooley, A., Silva, D. F., and Bitencourt, J. (2023). When is a simulation a digital twin? a systematic literature review. *Manufacturing Letters*, 35:940–951.

Wu, D., Zheng, A., Yu, W., Cao, H., Ling, Q., Liu, J., and Zhou, D. (2025). Digital twin technology in transportation infrastructure: A comprehensive survey of current applications, challenges, and future directions. *Applied Sciences*, 15(4).

Wu, E.-H., Liu, Y.-Q., Xu, T.-C., Ren, L.-X., Qin, Y.-M., Wei, M.-Y., He, X.-W., Yuan, D.-Y., Hou, W.-C., Ma, Z.-W., and Sheng, B. (2026). Physical AI: Evolution, progress, challenges, and prospects. *Journal of Computer Science and Technology*, 41(1):271–288.

Zhang, J., Ellwein, C., Heithoff, M., Michael, J., and Wortmann, A. (2025). Digital twin and the Asset Administration Shell: An analysis of the three types of AASs and their feasibility for Digital Twin Engineering. *Software and Systems Modeling*, 24(3):771–793.

Capítulo 4

Personas e Acessibilidade Web: Desenvolvendo para Diferentes Usuários

Matheus A. S. Santos, Beatriz L. Silva, Marina F. Souza, Bruna C. R. Cunha

Abstract

Web accessibility is essential for digital inclusion, yet technical guidelines and tools do not, by themselves, guarantee accessible experiences. The chapter presents the fundamentals of Web accessibility: legal and regulatory aspects, WCAG, semantic HTML, ARIA, Assistive Technology, and evaluation tools. As its central focus, it discusses the creation and analysis of personas representing different disability profiles, used to align technical requirements with users' real needs and anticipate barriers in Web interfaces before implementation. A Barrier Walkthrough example illustrates this application, identifying problems and proposing solutions. Finally, it discusses the limitations of personas and of technical compliance itself, reinforcing that compliance is necessary but not sufficient, and that the participation of people with disabilities remains indispensable.

Resumo

A acessibilidade Web é essencial para a inclusão digital, mas diretrizes e ferramentas técnicas não garantem, por si só, experiências acessíveis. O capítulo apresenta os fundamentos da acessibilidade Web: aspectos legais e normativos, WCAG, HTML semântico, ARIA, Tecnologia Assistiva e ferramentas de avaliação. Como eixo central, discute a criação e análise de personas com diferentes perfis de deficiência, usadas para aproximar requisitos técnicos das necessidades reais dos usuários e antecipar barreiras em interfaces Web antes da implementação. Um exemplo de Percurso de Barreira ilustra essa aplicação, identificando problemas e propondo soluções. Por fim, discutem-se os limites das personas e da própria conformidade técnica, reforçando que ela é necessária, mas não suficiente, e que a participação de pessoas com deficiência continua indispensável

4.1. Introdução

Segundo a Pesquisa Nacional por Amostra de Domicílios (PNAD) Contínua 2022, cerca de 8,9% da população brasileira com dois anos ou mais possui algum tipo de deficiência,

correspondendo a aproximadamente 18,6 milhões de pessoas [14]. Esses dados evidenciam que as pessoas com deficiência representam uma parcela significativa da população brasileira e reforçam a necessidade de promover condições de acesso e participação em diferentes espaços da sociedade, incluindo os ambientes digitais. Por trás desse número, no entanto, existe uma diversidade de perfis (pessoas cegos, com baixa visão, surdos, com deficiência motora ou cognitiva, entre outros) cujas necessidades específicas frequentemente não são consideradas de forma concreta ao longo do processo de desenvolvimento, o que motiva a proposta deste capítulo de aproximar essa diversidade da prática de design e código por meio de personas.

Em um contexto no qual a tecnologia e a internet desempenham um papel fundamental no acesso à informação, à educação, aos serviços e à comunicação, a acessibilidade na Web torna-se um aspecto essencial para a promoção da inclusão digital. Nesse sentido, a acessibilidade Web busca reduzir ou eliminar barreiras que possam dificultar a interação de pessoas com diferentes tipos de deficiência com conteúdos e serviços disponibilizados na internet. Diretrizes como as *Web Content Accessibility Guidelines* (WCAG) estabelecem recomendações para tornar conteúdos Web mais acessíveis e constituem uma das principais referências para o desenvolvimento de interfaces acessíveis; entretanto, diretrizes técnicas por si só nem sempre traduzem, na prática do desenvolvedor, a experiência real de quem enfrenta essas barreiras, lacuna que a construção de personas busca preencher, ao dar rosto e contexto de uso a critérios que, de outra forma, permanecem abstratos.

Apesar da existência dessas diretrizes e da crescente importância dos ambientes digitais, a acessibilidade ainda apresenta limitações significativas nos sites brasileiros. De acordo com um relatório da BigDataCorp em parceria com o Movimento Web para Todos, que analisou a acessibilidade de sites ativos no Brasil por meio de testes automatizados, dos 26,3 milhões de sites analisados, apenas 2,9% foram considerados acessíveis, resultado que representa uma redução em relação aos 3,3% identificados em 2023 [15]. Além disso, o mesmo levantamento aponta que, mesmo com uma leve melhora frente ao ano anterior, 96% dos sites de notícias analisados ainda apresentam algum tipo de falha de acessibilidade [15].

Esses dados evidenciam que a acessibilidade ainda não é incorporada de maneira consistente ao desenvolvimento de projetos Web, e que o cumprimento formal de diretrizes não é, isoladamente, suficiente para garantir experiências verdadeiramente inclusivas. A presença de barreiras de acessibilidade pode comprometer a autonomia das pessoas com deficiência e dificultar seu acesso a informações e serviços disponíveis no ambiente digital. Dessa forma, torna-se necessário compreender como essas barreiras são produzidas na prática, e é justamente para exercitar esse olhar, situado no usuário real e não apenas na norma, que a criação de personas com diferentes perfis de deficiência ocupa um lugar central neste capítulo.

Este capítulo tem por objetivo apresentar, de forma integrada, (a) os fundamentos da acessibilidade Web e sua relação com a inclusão digital, incluindo aspectos legais e normativos brasileiros; (b) as principais diretrizes das *Web Content Accessibility Guidelines* (WCAG) e sua aplicação no desenvolvimento de interfaces acessíveis; (c) práticas de desenvolvimento inclusivo, abordando HTML semântico, formulários acessíveis, navega-

ção por teclado e uso adequado de atributos ARIA; (d) uma visão prática sobre recursos de TA e ferramentas automatizadas de avaliação de acessibilidade; e, como eixo central do capítulo, (e) atividades baseadas na criação e análise de personas com diferentes perfis de deficiência, utilizadas para exercitar a identificação de barreiras e a proposição de soluções de design e desenvolvimento a partir de necessidades concretas de usuários.

4.2. Acessibilidade Web, Direitos e Legislação

A trajetória da acessibilidade digital está intimamente ligada à evolução da própria tecnologia computacional. Nas décadas de 1970 e 1980, a popularização dos computadores pessoais (PCs) representou uma mudança significativa: dispositivos antes restritos a técnicos e a grandes estruturas compartilhadas passaram a ser operados diretamente por um único usuário. Essa transformação impulsionou as primeiras discussões mais amplas sobre acessibilidade digital.

Em 1990, surgiu a World Wide Web (WWW), que, em sua versão inicial, era limitada e restrita a poucos usuários [60]. A partir de 1994, no entanto, sua expansão se tornou expressiva, consolidando-a como um dos principais meios de acesso à informação na atualidade, à frente inclusive de mídias tradicionais como a televisão e o rádio. Nesse mesmo ano, foi criado o World Wide Web Consortium (W3C), organização responsável pelo desenvolvimento de padrões e diretrizes voltados à construção de uma Web baseada nos princípios de acessibilidade, internacionalização, privacidade e segurança.

À medida que a Web se expandia, tornou-se necessário discutir formas de garantir o acesso democrático a pessoas com diferentes necessidades. Nesse contexto, em 1997 o W3C lançou a Web Accessibility Initiative (WAI) [1], iniciativa internacional voltada à promoção da acessibilidade na Web. No ano seguinte, em 1998, os Estados Unidos aprovaram a Section 508 [2], que passou a exigir que órgãos federais oferecessem conteúdos digitais acessíveis. Esse processo teve continuidade em 1999, quando a WAI publicou a primeira versão das *Web Content Accessibility Guidelines* (WCAG) [3], documento que estabeleceu as bases técnicas ainda hoje utilizadas para orientar a construção de conteúdos acessíveis na Web.

4.2.1. Direitos e Legislação no Contexto Brasileiro

No Brasil, o primeiro marco legal totalmente dedicado ao tema foi a Lei nº 10.098, sancionada em 2000 [7], conhecida como Lei de Acessibilidade. Essa norma estabeleceu critérios gerais para a eliminação de barreiras nos meios de comunicação e informação, ainda que voltada, em grande parte, à acessibilidade física e urbana. Em 2004, o Decreto nº 5.296 [8] regulamentou essa lei e, em seu artigo 47, determinou explicitamente que os portais e sítios eletrônicos da administração pública deveriam ser tornados acessíveis, representando um dos primeiros dispositivos legais brasileiros a tratar diretamente da acessibilidade digital.

Em 2009, o Brasil promulgou a Convenção da ONU sobre os Direitos das Pessoas com Deficiência, por meio do Decreto nº 6.949 [9]. Por ter sido incorporada com status de emenda constitucional, a Convenção consolidou o acesso à informação digital como um direito fundamental no ordenamento jurídico brasileiro. Já em 2012, o Decreto nº 7.724 [10], que regulamenta a Lei de Acesso à Informação (LAI), reforçou essa diretriz

ao determinar que os sítios eletrônicos públicos garantissem o acesso a dados de forma clara e acessível a cidadãos com deficiência.

O maior marco da acessibilidade digital no país, no entanto, viria em 2015, com a promulgação da Lei nº 13.146 [11], conhecida como Lei Brasileira de Inclusão. Em seu artigo 63, a lei tornou obrigatória a acessibilidade em sítios e aplicativos mantidos por órgãos governamentais e por empresas privadas com sede no Brasil, consolidando um percurso de mais de uma década de avanços legislativos em direção à garantia do acesso digital como direito.

4.2.2. WCAG, eMAG e ABNT NBR 17225

Atualmente, a acessibilidade digital é orientada por um conjunto de padrões técnicos, internacionais e nacionais, que traduzem princípios legais em diretrizes práticas de implementação. No plano internacional, a principal referência é a WCAG (*Web Content Accessibility Guidelines*), publicada pelo W3C por meio da WAI. Sua primeira versão, a WCAG 1.0, foi lançada em 1999 [3], com 14 diretrizes voltadas a páginas estáticas. Em 2008, a WCAG 2.0 [4] ampliou substancialmente o escopo do documento, introduzindo os quatro princípios que sustentam a norma até hoje: conteúdo perceptível, operável, compreensível e robusto. Novas versões vieram em 2018 (WCAG 2.1 [5], com foco em dispositivos móveis e baixa visão) e em 2023 (WCAG 2.2 [6], a mais recente), sempre de forma aditiva e compatível com as versões anteriores, consolidando-se como o principal parâmetro técnico de acessibilidade adotado por governos e organizações em todo o mundo.

No Brasil, o primeiro padrão técnico nacional foi o eMAG (Modelo de Acessibilidade em Governo Eletrônico) [12], desenvolvido a partir de 2004 com base no estudo comparativo de catorze normas internacionais, entre elas a própria WCAG 1.0. Disponibilizado para consulta pública em 2005 e institucionalizado pela Portaria nº 3 de 2007, o eMAG tornou-se obrigatório nos sítios e portais do governo federal, passando por revisões que culminaram na versão 3.1, de 2014, já alinhada à WCAG 2.0.

Mais recentemente, em março de 2025, a ABNT publicou a NBR 17225 [13], primeira norma técnica brasileira dedicada especificamente à acessibilidade em conteúdos e aplicações Web. Alinhada à WCAG 2.2 e ao artigo 63 da Lei Brasileira de Inclusão, a norma reúne 146 diretrizes de alcance nacional, aplicáveis tanto a órgãos públicos quanto a empresas privadas, e representa o padrão técnico mais abrangente já produzido no país sobre o tema.

Juntos, esses três documentos, um internacional e dois nacionais, formam a base técnica que orienta, hoje, a construção de sites e aplicações digitalmente acessíveis.

4.3. Diretrizes WCAG

Conforme apresentado no capítulo anterior, a WCAG constitui uma das principais referências para a acessibilidade Web [6]. Sua estrutura é organizada em quatro níveis hierárquicos: os princípios, que representam os pilares fundamentais da acessibilidade; as diretrizes, que detalham objetivos gerais dentro de cada princípio; os critérios de sucesso, que traduzem essas diretrizes em requisitos testáveis e objetivos; e os níveis de confor-

midade (A, AA e AAA), atribuídos a cada critério conforme seu grau de exigência. Essa estrutura hierárquica permite que a WCAG seja utilizada tanto como referência conceitual quanto como checklist técnico ao longo do processo de desenvolvimento.

4.3.1. Princípios POUR

Os princípios POUR (*Perceivable, Operable, Understandable, Robust*) constituem os quatro pilares fundamentais da WCAG e orientam, de forma geral, o design de conteúdos acessíveis. Mais do que garantir a conformidade técnica com a norma, esses princípios buscam assegurar que o conteúdo seja genuinamente utilizável por pessoas com diferentes tipos de deficiência.

Tabela 4.1. Princípios de acessibilidade da WCAG.

Princípio	O que significa	Como aplicar
Perceptível	As informações e os componentes da interface devem ser apresentados de forma que possam ser percebidos pelos usuários.	Fornecer alternativas em texto para imagens, legendas para conteúdos em vídeo e garantir contraste adequado entre texto e fundo.
Operável	Os componentes de navegação e da interface devem ser operáveis por diferentes meios de interação.	Permitir a navegação completa por teclado, evitar conteúdos que possam causar convulsões e fornecer tempo suficiente para leitura e interação.
Compreensível	As informações e a operação da interface devem ser apresentadas de maneira clara e de fácil entendimento.	Utilizar linguagem clara, manter o comportamento das páginas previsível e fornecer instruções e mensagens para prevenção e correção de erros.
Robusto	O conteúdo deve ser compatível com diferentes navegadores, dispositivos e recursos de TA, atuais e futuras.	Utilizar HTML semântico e código válido, garantindo que o conteúdo possa ser corretamente interpretado por leitores de tela e outros recursos de TA.

4.3.2. Critérios de Sucesso

Dentro de cada princípio, a WCAG estabelece diretrizes e, dentro de cada diretriz, critérios de sucesso: regras práticas e testáveis, identificadas por numeração própria (como 1.1.1, 1.4.3 ou 2.1.1), que especificam exatamente o que deve ser implementado no código ou no design para que o conteúdo funcione corretamente com leitores de tela, navegação por teclado e demais recursos de TA. A Tabela 4.2 apresenta alguns exemplos de critérios de sucesso associados a cada princípio.

Tabela 4.2. Exemplos de critérios de sucesso da WCAG.

Princípio	Critério	Exemplo
Perceptível	1.1.1	Conteúdo Não Textual: toda imagem informativa deve possuir um texto alternativo (atributo `alt`) que descreva seu propósito.
Perceptível	1.4.3	Contraste (Mínimo): o contraste entre texto e plano de fundo deve ser de, no mínimo, 4,5:1 para texto padrão.
Operável	2.1.1	Teclado: toda funcionalidade da página deve poder ser operada exclusivamente por teclado, sem depender do *mouse*.
Operável	2.4.7	Foco Visível: quando um elemento recebe foco via teclado, deve haver uma indicação visual clara desse foco.
Compreensível	3.3.1	Identificação de Erro: quando um erro é detectado em um formulário, o item com problema deve ser identificado e descrito ao usuário em texto.
Robusto	4.1.2	Nome, Função, Valor: componentes de interface (como botões e campos) devem ter seu nome e função corretamente expostos a recursos de TA, geralmente por meio de HTML semântico ou atributos ARIA.

4.3.3. Níveis de Conformidade

Cada critério de sucesso está associado a um nível de conformidade, A, AA ou AAA, que indica seu grau de exigência e o impacto de seu descumprimento sobre a experiência do usuário.

Tabela 4.3. Níveis de conformidade da WCAG.

Nível	Descrição
A (Básico)	Remove as barreiras mais críticas de acessibilidade. Sem o atendimento a esse nível, parte dos usuários pode não conseguir utilizar o site ou acessar determinados conteúdos.
AA (Recomendado)	Resolve as principais barreiras de acessibilidade encontradas em situações comuns de uso. É o nível geralmente adotado como referência em leis, normas e diretrizes de acessibilidade, incluindo a legislação brasileira.
AAA (Avançado)	Representa o nível mais elevado de conformidade com as diretrizes de acessibilidade. Pode ser indicado para públicos ou conteúdos específicos, ainda que sua aplicação integral seja, muitas vezes, difícil em sites complexos.

4.4. HTML Semântico e Recursos de Acessibilidade

HTML semântico refere-se ao uso de elementos que descrevem o *significado* e o papel do conteúdo dentro da página, e não apenas sua aparência visual [30]. Em vez de estruturar um documento inteiro com elementos genéricos como `<div>` e `<span>` – prática conhecida informalmente como *"div soup"* –, o HTML5 disponibiliza elementos como `<header>`, `<nav>`, `<main>`, `<article>`, `<section>`, `<aside>`, `<footer>` e `<figure>`, cada um comunicando explicitamente sua função dentro do documento [31]. Essa separação entre estrutura e apresentação é um dos princípios fundadores da própria Web, reforçado desde as primeiras recomendações do W3C sobre acessibilidade [1] e formalizado nas primeiras versões das diretrizes de conteúdo Web acessível [3].

O uso de marcação semântica não é apenas uma boa prática estética: navegadores, motores de busca e, principalmente, tecnologias assistivas (leitores de tela, linhas Braille,

softwares de reconhecimento de voz) dependem dessa semântica para construir uma *árvore de acessibilidade* (*accessibility tree*) que representa corretamente a página para o usuário [32].

4.4.1. Importância da Estrutura Correta da Página

A estrutura correta de uma página HTML contribui para que seu conteúdo seja organizado de maneira lógica e possa ser interpretado adequadamente por diferentes agentes de usuário e mecanismos de busca [30, 32]. O uso de elementos semânticos permite representar a função e o significado de cada região do documento em vez de sua mera aparência visual, definindo claramente áreas como cabeçalho (`<header>`), navegação (`<nav>`), conteúdo principal (`<main>`) e rodapé (`<footer>`) [30, 33]. Essa separação entre estrutura e apresentação torna a organização do código mais transparente e previsível, tanto para navegadores quanto para recursos de TA [30, 32].

Essa estruturação também é determinante para a navegabilidade do documento. Leitores de tela, por exemplo, utilizam elementos semânticos e marcas estruturais (*landmarks*) como pontos de referência para sinalizar as regiões da página, permitindo que o usuário salte diretamente entre elas [30, 33]. Essa forma de navegação é particularmente relevante na prática: levantamentos periódicos sobre o uso de leitores de tela mostram que navegar por landmarks e por cabeçalhos está entre os métodos mais utilizados pelos usuários para explorar rapidamente uma página, ao lado da leitura sequencial e da navegação por listas de links [19]. Do mesmo modo, uma organização hierárquica adequada dos títulos possibilita compreender as relações de subordinação entre as diferentes seções, funcionando como um mapa mental do conteúdo [30, 33].

Do ponto de vista da engenharia de software, a adesão ao HTML semântico também impacta a manutenção do código ao longo do ciclo de vida do projeto. Quando os elementos são empregados conforme sua finalidade nativa, elimina-se a necessidade de encadeamentos excessivos de contêineres genéricos (a prática de *"div soup"*) ou do uso de classes, scripts e estilos adicionais apenas para simular um significado que o próprio elemento nativo já expressa por padrão [30, 33]. Isso reduz a quantidade de código a ser mantido, facilita a leitura por outros desenvolvedores e diminui o risco de que mudanças futuras quebrem o comportamento esperado por recursos de TA, já que a semântica passa a fazer parte da própria estrutura do documento, e não de convenções implícitas de estilo ou de comportamento adicionadas via JavaScript.

Portanto, a utilização adequada dos elementos estruturais do HTML ultrapassa a mera organização de código: ela estabelece a base para que o conteúdo seja interpretado de forma consistente por uma grande diversidade de navegadores, dispositivos e tecnologias de interação [30, 32]. A Tabela 4.4 resume os principais elementos estruturais do HTML5 e o papel que cada um assume perante recursos de TA.

Tabela 4.4. Principais elementos semânticos do HTML5 e seu papel de acessibilidade.

Elemento	Função	Landmark/papel implícito
`<header>`	Cabeçalho da página ou de uma seção.	`banner` (quando associado ao documento inteiro).
`<nav>`	Bloco de links de navegação principal ou secundária.	`navigation`.
`<main>`	Conteúdo principal e único da página.	`main`.
`<article>`	Conteúdo autocontido e reutilizável (post, notícia, comentário).	`article`.
`<section>`	Agrupamento temático de conteúdo, geralmente com título próprio.	`region` (quando rotulado).
`<aside>`	Conteúdo relacionado, mas tangencial ao fluxo principal.	`complementary`.
`<footer>`	Rodapé da página ou de uma seção.	`contentinfo` (quando associado ao documento inteiro).
`<figure>`	Conteúdo ilustrativo (imagem, gráfico, código) com legenda associada.	`figure`.

4.4.2. Estrutura e Elementos Acessíveis

Nesse contexto, destacam-se alguns elementos e práticas que contribuem para a construção de páginas HTML acessíveis. A seguir, são apresentados exemplos de implementações não acessíveis e acessíveis, acompanhados de uma breve explicação sobre sua importância.

Títulos e Hierarquia.

Os elementos `<h1>` a `<h6>` devem refletir a hierarquia lógica do conteúdo, sem saltos de nível (por exemplo, de `<h2>` direto para `<h4>`). Leitores de tela permitem que o usuário navegue de título em título, de modo que uma hierarquia mal formada compromete a compreensão geral da página [5, 12].

Links e Botões.

Elementos de *link* (`<a>`) devem ser usados para navegação, enquanto `<button>` deve ser usado para ações dentro da própria página. O texto do link deve ser descritivo por si só (evitando construções como “clique aqui”), pois muitos usuários de leitor de tela navegam ouvindo apenas a lista de links da página [6, 2]. Elementos interativos criados a partir de `<div>` ou `<span>` com eventos de clique, sem função ou papel semântico associado, não recebem foco de teclado nem são anunciados corretamente por recursos de TA.

Como exemplo, um botão implementado com um elemento genérico `<div>` não possui, por padrão, o comportamento e a semântica de um botão:

```
<div onclick="salvar()">Salvar</div>
```

Código 4.1. Não acessível: div como botão

A utilização do elemento nativo `<button>` fornece semântica e suporte à interação por teclado de forma nativa:

```
<button type="button" onclick="salvar()">Salvar</button>
```

Código 4.2. Acessível: elemento nativo

Formulários.

Cada campo de formulário deve possuir um `<label>` explicitamente associado (via atributo `for`/`id` ou envolvendo o campo), e grupos de campos relacionados devem usar `<fieldset>` e `<legend>`. Mensagens de erro e campos obrigatórios devem ser comunicados de forma programática (por exemplo, com `aria-describedby` e `aria-required`), e não apenas por cor [5, 12].

Um campo sem um rótulo associado dificulta a identificação de sua finalidade por recursos de TA:

```
<p>Nome:</p>
<input type="text" name="nome">
```

Código 4.3. Não acessível: campo sem rótulo associado

A associação explícita entre `<label>` e `<input>` permite que recursos de TA identifiquem corretamente o campo:

```
<label for="nome">Nome:</label>
<input type="text" id="nome" name="nome" autocomplete="name">
```

Código 4.4. Acessível: label associado ao campo

Imagens e Texto Alternativo.

Toda imagem com valor informativo precisa de um atributo `alt` descritivo; imagens puramente decorativas devem usar `alt=` para serem ignoradas por leitores de tela. Imagens complexas, como gráficos, infográficos e diagramas, podem exigir uma descrição estendida para transmitir informações que não podem ser adequadamente apresentadas apenas no atributo `alt`. A descrição estendida fornece detalhes adicionais sobre o conteúdo da imagem, como valores, relações, tendências ou outras informações relevantes para sua compreensão. Ela pode ser apresentada como texto adjacente à imagem ou associada por meio de elementos como `<figure>` e `<figcaption>` [3, 2].

Uma imagem sem texto alternativo não fornece aos usuários de leitores de tela informações sobre seu conteúdo:

```
<img src="grafico-vendas.png">
```

Código 4.5. Nao acessivel: sem texto alternativo

Quando a imagem possui valor informativo, o atributo `alt` deve apresentar uma descrição adequada ao seu conteúdo. Para imagens simples, essa descrição pode ser suficiente:

```
<img src="grafico-vendas.png"
     alt="Grafico de vendas mensais, com crescimento de 12% em marco">
```

Código 4.6. Acessivel: com texto alternativo descritivo

Em imagens complexas, entretanto, o atributo `alt` pode ser insuficiente para transmitir todas as informações apresentadas. Nesses casos, recomenda-se fornecer uma descrição estendida. Por exemplo, um gráfico pode ter um texto alternativo resumido e uma descrição adicional com os valores e as principais tendências:

```
<figure>
  <img src="grafico-vendas.png"
       alt="Grafico de vendas mensais de janeiro a maio">
  <figcaption>
    As vendas foram de 120 unidades em janeiro,
    150 em fevereiro, 180 em marco, 210 em abril
    e 250 em maio, indicando crescimento continuo.
  </figcaption>
</figure>
```

Código 4.7. Acessivel: imagem complexa com descricao estendida

Assim, o atributo `alt` fornece uma identificação resumida da imagem, enquanto a descrição estendida disponibiliza informações adicionais necessárias para compreender seu conteúdo.

Tabelas de Dados.

Tabelas usadas para apresentar dados (e não para fins de *layout*) devem expor sua estrutura de forma programática, e não apenas visualmente. O elemento `<caption>` fornece um título acessível para a tabela, enquanto o atributo `scope` em células de cabeçalho (`<th scope="col»` ou `<th scope="row»`) associa cada célula de dado ao seu respectivo cabeçalho de coluna ou linha, permitindo que leitores de tela anunciem esse contexto ao navegar célula a célula [31, 6].

Uma tabela sem cabeçalhos adequadamente definidos dificulta a compreensão da relação entre os dados:

```
<table>
  <tr><td>Ms</td><td>Vendas</td></tr>
  <tr><td>Maro</td><td>R$ 12.000</td></tr>
</table>
```

Código 4.8. Não acessível: tabela sem caption nem scope

A utilização de `<caption>` e de `scope` permite estruturar semanticamente a tabela e estabelecer a relação entre cabeçalhos e dados:

```
<table>
  <caption>Vendas mensais por perodo</caption>
  <tr><th scope="col">Ms</th><th scope="col">Vendas</th></tr>
  <tr><th scope="row">Maro</th><td>R$ 12.000</td></tr>
</table>
```

Código 4.9. Acessível: caption e scope associando cabeçalhos

Navegação por Teclado.

Todo elemento interativo deve ser alcançável e operável apenas com teclado (`Tab`, `Shift+Tab`, `Enter`, `Espaço`), sem "armadilhas de foco" que impeçam o usuário de

sair de um componente. A ordem de tabulação deve seguir a ordem visual/lógica do conteúdo (evitando valores positivos de `tabindex`), o indicador de foco deve permanecer visível, e recomenda-se a inclusão de um link de "pular para o conteúdo principal" no início da página [5, 12].

4.4.3. Recursos de ARIA

Quando os elementos nativos do HTML não são suficientes para descrever um componente de interface – como abas, *modais*, *comboboxes* ou barras de progresso –, a especificação WAI-ARIA (*Accessible Rich Internet Applications*) oferece atributos que comunicam papel, estado e propriedades a recursos de TA [28]. A diferença entre esses dois casos é a de papel *implícito* e papel *explícito*: elementos nativos como `<button>` ou `<nav>` já carregam, por definição da especificação HTML, um papel, um estado e um comportamento de teclado reconhecidos automaticamente por leitores de tela; já um `<div>` não possui papel algum até que este seja atribuído explicitamente via `role`, e mesmo assim o ARIA só descreve *o que* o componente é, sem implementar *como* ele deve se comportar diante do teclado ou do foco [28, 29]. Entre os atributos mais utilizados estão:

Tabela 4.5. Principais atributos ARIA e sua finalidade.

Atributo	Finalidade
`role`	Define o papel semântico de um elemento (ex.: `role="dialog"`, `role="alert"`).
`aria-label` / `aria-labelledby`	Fornecem um nome acessível quando não há texto visível suficiente.
`aria-describedby`	Associa uma descrição adicional a um elemento.
`aria-live`	Anuncia atualizações dinâmicas de conteúdo (ex.: mensagens de erro carregadas via JavaScript).
`aria-expanded`	Indica se um componente expansível (menu, acordeão) está aberto ou fechado.
`aria-hidden`	Oculta um elemento da árvore de acessibilidade, sem afetar sua exibição visual.
`aria-current`	Indica o item atual dentro de um conjunto (ex.: página ativa em um menu de navegação).

É importante observar a chamada "primeira regra do ARIA": sempre que existir um elemento HTML nativo com a semântica e o comportamento necessários, ele deve ser preferido a recriar esse comportamento com `<div>`/`<span>` e atributos ARIA, já que ARIA descreve papéis e estados, mas não implementa comportamento (como resposta a teclado) por si só [28, 29, 6].

4.5. Tecnologia Assistiva e Interação com a Web

O Estatuto da Pessoa com Deficiência (Lei N° 13.146, de 2015) define Tecnologia Assistiva (TA) como:

> "(...) produtos, equipamentos, dispositivos, recursos, metodologias, estratégias, práticas e serviços que objetivem promover a funcionalidade, relacionada à atividade e à participação da pessoa com deficiência ou com mobilidade reduzida, visando à sua autonomia, independência, qualidade de vida e inclusão social." [11]

Alternativamente, a TA pode ser entendida como o conjunto de serviços e recursos que contribuem para oferecer ou ampliar habilidades funcionais de pessoas com deficiência e, consequentemente, facilitar a vida dessas pessoas [23].

No contexto digital e na Web, os recursos de TA funcionam como pontes de mediação entre o usuário e a interface, tornando o formato do conteúdo adequado para o usuário e permitindo diferentes alternativas para a interação com ele [1].

4.5.1. Conceitos e Ferramentas de Tecnologia Assistiva

Na Web, os recursos de TA manifestam-se em diferentes formas de hardware e software para atender a diversas necessidades funcionais. Algumas delas estão listadas a seguir:

- **Leitores de Tela (*Screen Readers*):** Softwares que convertem texto e elementos estruturais de uma interface gráfica em fala sintetizada ou em braille tátil. Exemplos populares incluem o NVDA (*NonVisual Desktop Access*), JAWS e o VoiceOver (nativo em ecossistemas Apple) [26, 35, 34, 1].

- **Softwares de Ampliação de Tela:** Recursos voltados para pessoas com baixa visão que aumentam porções da tela e ajustam o contraste de cores, como o ZoomText [36, 1].

- **Dispositivos de Entrada Alternativos:** Teclados adaptados, acionadores de pressão, sistemas de rastreamento ocular (*eye-tracking*) e navegação por comando de voz, que atendem pessoas com restrições motoras severas [1].

- **Ferramentas de Tradução Automática para Libras:** Tradutores virtuais como o VLibras, que convertem textos em língua portuguesa para a Língua Brasileira de Sinais por meio de avatares tridimensionais [27].

4.5.2. Relação entre Tecnologia Assistiva e Acessibilidade Web

Existe uma relação de interdependência direta entre a Tecnologia Assistiva e a Acessibilidade Web. Enquanto a TA provê o meio técnico para que o usuário interaja com o conteúdo, a Acessibilidade Web determina como o conteúdo Web deve ser desenvolvido para que ele seja utilizável por pessoas com deficiência [1].

Sem a aplicação das diretrizes de acessibilidade recomendadas pelo W3C através das *Web Content Accessibility Guidelines* (WCAG), o desempenho dos recursos de TA é comprometido [5]. Por exemplo, um leitor de tela não consegue descrever o propósito de um botão interativo se o desenvolvedor utilizar apenas uma imagem sem texto alternativo ou sem rótulos estruturais adequados.

4.5.3. Como Recursos de TA Interpretam uma Página Web

A interpretação de uma página por um recurso de TA não ocorre por meio do processamento visual da interface gráfica renderizada na tela, mas sim pela navegação através da estrutura do documento subjacente. O processo ocorre através das seguintes etapas e mecânicas:

1. **Construção do DOM (*Document Object Model*):** Quando o navegador carrega o documento HTML, XML ou SVG, ele monta uma árvore estrutural de elementos. O DOM possui métodos que permitem acesso à árvore, e, consequentemente, que possibilitam a sua modificação [24].

2. **Árvore de Acessibilidade (*Accessibility Tree*):** A partir do DOM, a API do sistema operacional cria uma representação paralela chamada *Árvore de Acessibilidade*. Essa árvore simplifica o código e expõe apenas informações relevantes. Seus objetos apresentam quatro propriedades principais [25]:

   - **Nome (*Name*):** O rótulo legível do elemento (ex.: "Enviar"e "Ler mais").
   - **Descrição (*Description*):** A descrição do elemento, caso se pretenda informar uma descrição para além do nome do objeto (ex.: a descrição de uma tabela pode explicar que tipo de informação está presente no objeto).
   - **Papel (*Role*):** A função do elemento na interface (ex.: *button*, *heading*, *checkbox*).
   - **Estado (*State*):** A condição atual do elemento (ex.: *expanded*, *checked*, *disabled*).

   Além dessas propriedades, o objeto também pode informar "o que ele faz". Por exemplo, uma entrada de texto pode receber informações do teclado [25].

3. **Uso de Especificações ARIA:** Quando os elementos HTML nativos não são suficientes para expressar a semântica de componentes dinâmicos (como menus retráteis ou *modais*), utilizam-se os atributos WAI-ARIA (*Accessible Rich Internet Applications*) para enriquecer a árvore de acessibilidade [28].

4.6. Ferramentas de Avaliação de Acessibilidade

A avaliação de acessibilidade consiste no processo de verificar se um site ou aplicação Web atende aos critérios estabelecidos por normas como a WCAG [6], identificando barreiras que possam impedir ou dificultar o uso por pessoas com deficiência. Esse processo pode ser conduzido de diferentes formas, sendo a avaliação automática uma das mais utilizadas por sua rapidez e baixo custo de aplicação. Atualmente, existem diversas ferramentas que auxiliam desenvolvedores a testar a acessibilidade de páginas Web com base nos critérios de sucesso da WCAG, entre as quais se destacam o WAVE, o axe DevTools e o Lighthouse.

Essas ferramentas realizam uma varredura automatizada do código da página e conseguem identificar problemas objetivos e programaticamente verificáveis, como a ausência de texto alternativo em imagens, contrastes de cor insuficientes entre texto e plano de fundo, ausência de rótulos em campos de formulário, uso incorreto de hierarquia de cabeçalhos e elementos interativos que não podem ser acessados via teclado. Um levantamento realizado com o WebAIM Million, que analisa anualmente a acessibilidade de um milhão dos principais sites do mundo, aponta que a grande maioria das páginas analisadas apresenta falhas detectáveis por ferramentas automáticas, evidenciando a utilidade dessas ferramentas para identificar rapidamente problemas técnicos recorrentes [18].

Entretanto, a avaliação automática não substitui a avaliação humana, uma vez que apresenta limitações significativas. Um estudo conduzido pelo Government Digital Service do Reino Unido testou treze ferramentas automáticas de avaliação contra uma página com 142 barreiras de acessibilidade previamente documentadas: a ferramenta com melhor desempenho identificou 40% dos problemas, enquanto as demais apresentaram taxas de detecção ainda menores [17]. De forma semelhante, o relatório da Deque Systems, baseado em mais de 2.000 auditorias, abrangendo mais de 13.000 páginas ou estados de páginas e aproximadamente 300.000 problemas de acessibilidade, identificou que, em média, 57,38% dos problemas registrados foram detectados por testes automatizados. Entretanto, a análise também identificou problemas automatizáveis em apenas 16 dos 50 critérios de sucesso da WCAG 2.1 de níveis A e AA, evidenciando que determinados aspectos da acessibilidade ainda requerem avaliação manual. Entre os critérios analisados, problemas relacionados à ordem de foco, navegação por teclado e visibilidade do foco apresentaram predominância de detecção manual [16].

Por esse motivo, a literatura da área recomenda uma combinação entre avaliação automática e avaliação manual, sendo a primeira utilizada para identificar rapidamente problemas técnicos objetivos, e a segunda, conduzida por especialistas ou pelas próprias pessoas com deficiência, responsável por avaliar aspectos qualitativos e de experiência que nenhuma ferramenta é capaz de mensurar sozinha.

4.6.1. WAVE Web Accessibility Evaluation Tools

O WAVE consiste em um conjunto de ferramentas de avaliação que objetivam aumentar a acessibilidade de conteúdo Web. Entre as suas funções, destacam-se a identificação de erros de acessibilidade e de conformidade com as Diretrizes de Acessibilidade para Conteúdo Web (WCAG), bem como a facilitação de avaliação humana do conteúdo Web. O WAVE está disponível através de extensões de navegador e de Interfaces de Programação de Aplicações (APIs). As extensões de navegador estão disponíveis para Chrome, Firefox e Edge, e permitem testar acessibilidade diretamente no navegador. Já as APIs possibilitam a coleta de dados de teste de acessibilidade em múltiplas páginas Web. Adicionalmente, elas possuem mecanismos de teste que podem ser integrados à infraestrutura do sistema. Ao final do processo, o WAVE retorna um relatório de avaliação de Impacto de Acessibilidade (AIM), que fornece dados dos testes, pontuação AIM e resultados de testes manuais especializados. A pontuação AIM visa medir o impacto no usuário comparativamente, em relação páginas Web em geral e considerando regras e diretrizes definidas por testadores humanos. [22].

4.6.2. axe DevTools

O axe DevTools é um verificador de acessibilidade no Google Chrome direcionado para designers, desenvolvedores e testadores. A ferramenta permite a avaliação de acessibilidade em soluções Web e *mobile*. O axe DevTools foi a primeira ferramenta de acessibilidade a usar automação e Inteligência Artificial (IA). Há três ferramentas principais: (i) axe MCP Server (ou Servidor de Modelo de Contexto e Protocolo axe): conecta o motor de testes de acessibilidade Web a Ambientes de Desenvolvimento (IDEs), auxiliando no desenvolvimento do código; (ii) Assistente axe: um chatbot de IA, treinado com uma base robusta de conhecimento em acessibilidade; e (iii) Extensão Axe DevTools: utiliza

Testes Guiados Inteligentes (IGTs) baseados em IA dentro da extensão do navegador para identificar erros de acessibilidade que a automação não foi capaz de identificar. Além disso, a ferramenta também oferece auditorias realizadas por especialistas que avaliam a acessibilidade e padrões de conformidade do sistema. [21]

4.6.3. Lighthouse

O Lighthouse é uma ferramenta de código aberto que objetiva a melhoria da qualidade de páginas Web automaticamente. Para utilizá-lo, é possível fornecer uma URL à ferramenta - através das ferramentas do desenvolvedor do Chrome, da linha de comando, de um módulo Node ou de uma interface da Web - para a realização da auditoria. O Lighthouse, em seguida, executa uma série de auditorias na página Web e, a partir dos resultados obtidos, gera um relatório sobre o desempenho dessa página. Para cada auditoria, há uma explicação de sua importância e, se necessário, do que se precisa fazer para corrigi-la. O Lighthouse apresenta duas ferramentas que o personalizam de acordo com as necessidades do desenvolvedor: (i) pacotes de pilhas: o Lighthouse consegue detectar em qual plataforma o site foi criado e apresenta recomendações específicas, estabelecidas e definidas por especialistas, com base na pilha utilizada; e (ii) plug-ins: permitem a adição de novas categorias no relatório através da implementação de módulos de nó. [20]

4.7. Prática de Criação com Personas

A criação de personas é uma técnica consolidada do design centrado no usuário, introduzida por Cooper [43] e posteriormente sistematizada como método de pesquisa e comunicação de requisitos por Pruitt e Grudin [44]. Uma persona é um indivíduo fictício, porém realista, que representa as características, necessidades, comportamentos, objetivos e expectativas de um grupo de usuários-alvo. Sua principal contribuição não é descritiva, mas comunicativa: personas traduzem dados de pesquisa em uma linguagem narrativa que as equipes de desenvolvimento conseguem incorporar às decisões cotidianas de projeto, funcionando como substituto viável quando o acesso direto aos usuários reais é inviável ou o recrutamento é particularmente difícil [55].

No contexto da acessibilidade, essa técnica assume um papel específico. Personas que representam pessoas com deficiência são utilizadas sobretudo nas fases de pesquisa e design, com o objetivo de antecipar barreiras que, de outro modo, só seriam identificadas após a implementação. A ausência desse tipo de representação faz com que produtos e serviços digitais sejam projetados sem considerar barreiras que afetam pessoas com deficiência, ao passo que sua inclusão nas etapas de planejamento e design permite que a acessibilidade seja tratada de forma proativa, e não reativa [55]. Personas também podem ser empregadas como instrumento de conscientização sobre desenho universal e sobre as necessidades de acessibilidade desse público, incluindo o uso de recursos de TA [45].

Um aspecto frequentemente negligenciado na construção dessas personas é a caracterização do recurso de TA utilizado e do nível de proficiência do usuário com ela. Conforme discutido na Seção 4.5, a experiência de navegação de um usuário iniciante em leitores de tela difere substancialmente da de um usuário experiente na mesma página. Uma persona que registre apenas o diagnóstico, sem descrever o contexto de uso, o dispositivo, a tecnologia assistiva e o grau de familiaridade com ela, tende a produzir

análises de barreiras genéricas e pouco acionáveis.

4.7.1. Limitações e riscos da técnica

A literatura de Interação Humano-Computador documenta que personas são uma ferramenta de construção de empatia controversa, que pode produzir efeitos prejudiciais ao design de produtos acessíveis quando mal construída ou empregada de forma inadequada [53]. Quatro riscos merecem atenção explícita.

O primeiro diz respeito à **natureza estática do artefato**. Edwards, Sum e Branham [47], a partir de um estudo qualitativo com nove participantes que possuíam múltiplas deficiências ou uma deficiência associada a outras identidades marginalizadas, apontam que personas são tipicamente documentos fixos, enquanto as identidades de pessoas com deficiência são dinâmicas e em constante transformação. Como consequência, personas não se adaptam nem respondem como uma pessoa real durante o processo de design, o que pode reforçar estereótipos em vez de dissolvê-los.

O segundo é o **problema da cobertura**. O número de personas empregado em um projeto é inerentemente limitado, o que torna impossível representar adequadamente a diversidade e a complexidade das experiências vividas [47]. Uma persona "usuário cego" não representa a população de pessoas cegas; representa, no máximo, um recorte específico dela.

O terceiro risco refere-se a atividades de **simulação de deficiência**, frequentemente associadas a dinâmicas de sensibilização e por vezes utilizadas como complemento à construção de personas. Nario-Redmond, Gospodinov e Cobb [49] avaliaram experimentalmente esse tipo de atividade e verificaram que as simulações deixaram os participantes mais confusos, constrangidos, desamparados e mais vulneráveis à percepção de que poderiam se tornar deficientes. Embora a preocupação empática tenha aumentado, as atitudes relativas à interação com pessoas com deficiência não melhoraram; em um dos experimentos, os participantes julgaram-se menos competentes, expressaram mais pena e maior desconforto de interação após a simulação. Os autores concluem que simular deficiências promove sofrimento e falha em melhorar atitudes, minando esforços de inclusão mesmo quando os próprios participantes relatam ter compreendido melhor a experiência da deficiência. Por essa razão, este minicurso não adota dinâmicas de simulação como recurso pedagógico.

O quarto risco é de ordem conceitual. Bennett e Rosner [48] analisam criticamente a fase de "empatia" do processo de design centrado no ser humano e argumentam que atividades de construção de empatia podem, paradoxalmente, afastar pessoas com deficiência dos processos nos quais se pretendia incluí-las, ao substituir a participação efetiva pela representação feita por terceiros. A implicação prática é direta: personas informam decisões de projeto, mas não substituem a participação de usuários reais.

4.7.2. Estratégias de mitigação

Três estratégias respondem a essas limitações.

A primeira é a **co-criação**. Neate et al. [46] investigaram o uso de personas cocriadas com pessoas com doença de Parkinson, demência e afasia, e observaram que a

técnica incentivou usuários com necessidades diversas a se engajarem no co-design, além de produzir benefícios adicionais de empoderamento dentro de um processo de design mais acessível. No estudo, as personas não eram fixas: podiam ser modificadas à medida que o escopo do projeto se tornava mais definido. Os autores registram ainda que, ao trabalhar apenas com os co-designers presentes, a equipe fica limitada à experiência existente na sala, e as personas co-criadas servem justamente para ampliar esse alcance.

A segunda é a **fundamentação em dados empíricos**. Recomenda-se que personas de acessibilidade partam de pesquisa real com usuários, seja por meio de entrevistas com pessoas com deficiência, seja pelo aproveitamento de evidências provenientes de estudos de acessibilidade, testes de usabilidade e organizações de defesa de direitos [55]. Um exemplo direto de aplicação dessa estratégia em contexto próximo ao deste minicurso é o de Heitmeier et al. [53], que descrevem uma abordagem de co-design aninhado, incluindo pessoas com deficiência em múltiplas etapas e funções do processo, aplicada especificamente à melhoria de materiais didáticos de acessibilidade digital em instituições de ensino superior, um contexto educacional diretamente análogo ao deste minicurso.

A terceira, aplicável quando não há acesso direto a usuários, é o **apoio em guias de recomendações derivados de revisão sistemática da literatura**. No contexto brasileiro, o GAIA constitui exemplo relevante para o perfil cognitivo: Britto e Pizzolato [50] derivaram suas recomendações de uma revisão de 17 trabalhos publicados entre 2005 e 2015, identificando inicialmente 107 recomendações agrupadas em 10 categorias por meio de diagrama de afinidades, posteriormente normalizadas em um conjunto final de 28 recomendações únicas voltadas ao Transtorno do Espectro Autista.

4.7.3. Exemplo de Aplicação com o Percurso de Barreira

Apresenta-se um cenário de avaliação com um uso prático de Percurso de Barreira (*Barrier Walkthrough*). Este exemplo demonstra como o uso de personas apoia o desenvolvimento de interfaces de usuário, configurando-se como uma ferramenta de inspeção para o desenvolvedor.

Considere a persona apresentada na Figura 4.1, baseada no modelo de [62]. Lucas, persona desenvolvida para este exemplo, é uma pessoa cega desde a infância. Ele é desenvolvedor de *software*, apresentando facilidade com tecnologia. Lucas utiliza os seguintes recursos de TA: leitor de tela, teclado e linha Braille, sendo proficiente em Braille. As principais formas de interação de Lucas com soluções *web* são: auditiva, teclado e toque. Ele apresenta um perfil independente, paciente, adaptável, objetivo e um pouco cauteloso. As suas necessidades específicas são compatibilidade total com leitor de tela e navegação fluida por teclado.

Tarefa Avaliada: O objetivo é avaliar o fluxo em um **aplicativo de *e-commerce* de eletrônicos**, em que a persona, Lucas, precisa encontrar e realizar a compra de um **combo de *mouse* e teclado** para uso residencial. A avaliação realiza o percurso do usuário e confronta as exigências do sistema com as capacidades do usuário.

Durante a execução do percurso, a interface é analisada considerando as características, necessidades e formas de interação descritas na persona de Lucas. Sempre que o sistema exigir uma habilidade incompatível com essas características, uma barreira de

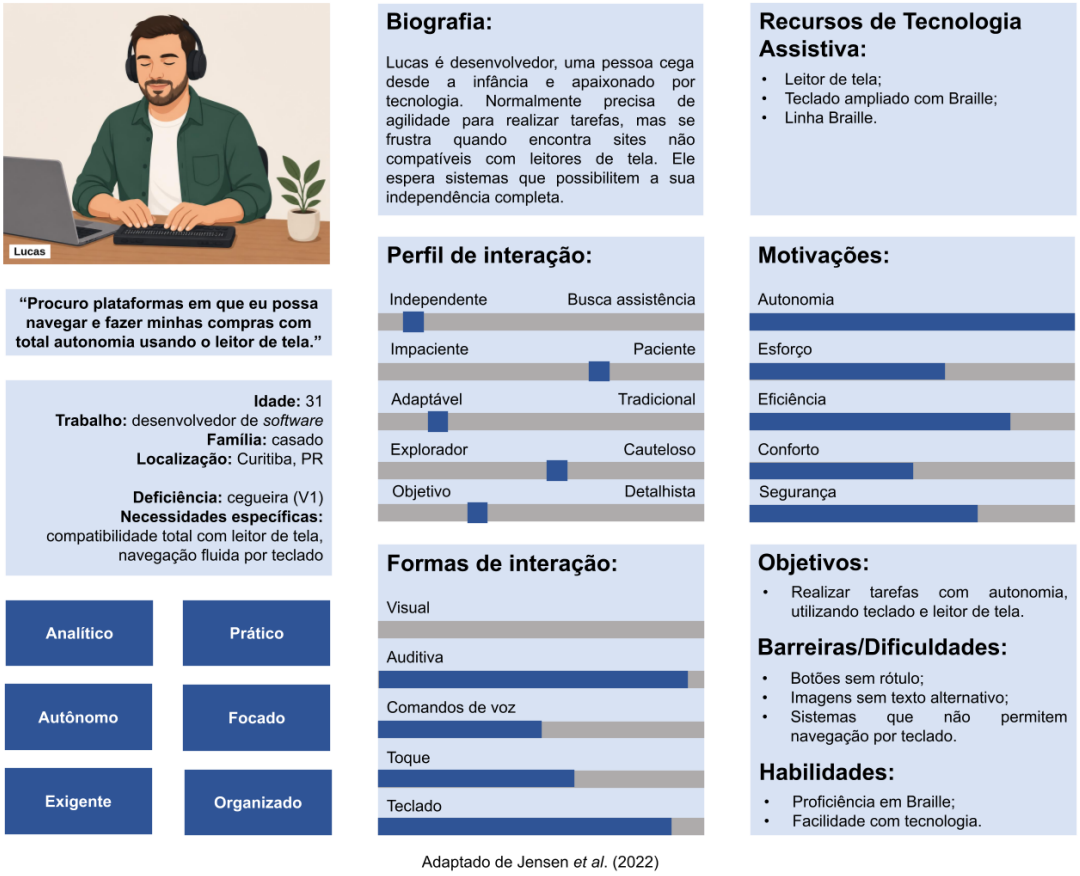


Figura 4.1. Exemplo de persona PCD.

acesso é identificada, registrada e catalogada.

Definiu-se um percurso com quatro passos sequenciais da jornada de compra no *e-commerce*. A análise considera as ações esperadas de um usuário que navega pela interface utilizando um leitor de tela e teclado, conforme as características definidas na persona.

1. **Passo 1 — Seleção e Validação do Produto:** Lucas navega até a página do combo de *mouse* e teclado. Ele tenta verificar se o item está disponível para o seu endereço. A disponibilidade do produto, porém, é indicada exclusivamente por meio de uma informação visual baseada em cores (a bolha colorida), sem uma alternativa textual equivalente acessível ao leitor de tela. Como essa informação não é comunicada pelo leitor de tela, Lucas assume que há um erro na página ou que o produto pode estar indisponível.

2. **Passo 2 — Leitura dos Detalhes Técnicos:** Ao tentar entender as especificações do teclado (*layout* das teclas, conexões), Lucas busca a descrição nas imagens do produto. Como as imagens não possuem texto alternativo adequado, o leitor de

tela informa apenas que se trata de uma imagem, sem fornecer detalhes sobre seu conteúdo. Ele precisa confiar apenas no título curto do produto, sem conseguir validar os detalhes técnicos necessários para o seu uso como desenvolvedor.

3. **Passo 3 — Revisão no Carrinho de Compras:** Após adicionar o item, Lucas acessa o carrinho para conferir o valor. Ele decide ajustar o pedido, mas o leitor de telas anuncia apenas a palavra genérica "botão" repetidas vezes. Trata-se do botão de exclusão, representado apenas pelo ícone de uma lixeira, sem um nome acessível associado. Sem saber qual botão remove o item ou avança para o pagamento, o fluxo é interrompido por falta de semântica.

4. **Passo 4 — Tentativa de *Checkout* e Cadastro:** Ao tentar digitar o endereço, Lucas espera que o aplicativo auxilie sua digitação de forma ágil. No entanto, a ausência do uso adequado do atributo `autocomplete` nos campos do formulário o obriga a interagir campo por campo de forma lenta e manual, resultando em uma quebra na eficiência do sistema.

A Tabela 4.6 ilustra esse caminho prático, conectando a persona às falhas de usabilidade e acessibilidade encontradas na interface do *e-commerce*. Para simplificar o preenchimento do modelo de persona (Figura 4.1), as barreiras associadas à navegação por teclado foram agrupadas sob a condição de "sistemas que não permitem navegação por teclado".

Tabela 4.6. Matriz de Tradução de Barreiras na Compra de Eletrônicos.

Atributo da Persona	Barreira e Impacto Gerado (Interface)
Ausência de Interação Visual	**Nível de estoque definido por cor:** Na tela de informações do produto (combo de *mouse* e teclado), a disponibilidade do item é indicada apenas por um círculo colorido, sem texto alternativo ou legenda. Como a informação é apresentada exclusivamente de forma visual, ela não é comunicada pelo leitor de telas, impossibilitando que o usuário saiba qual é a disponibilidade do produto.
Perfil Tecnológico / Ágil	**Formulário sem marcação semântica e sem preenchimento automático:** Ao preencher os dados de entrega, os campos não possuem identificação semântica adequada e não utilizam corretamente o atributo `autocomplete` (por exemplo, no campo de CEP). Isso impede que o navegador ofereça preenchimento automático, além de forçar o usuário a digitar manualmente dados repetitivos de forma sequencial e lenta. Como consequência, a eficiência do sistema é reduzida consideravelmente.
Imagens do produto	**Imagens sem texto alternativo:** As imagens dos produtos (*mouse* e teclado) não possuem texto alternativo. Isso faz com que essas imagens se tornem "invisíveis" para o leitor de telas e prejudica a experiência do usuário.
Acessibilidade através do teclado	**Navegação através do teclado não é possível:** Lucas utiliza leitor de tela associado a teclado. Isso faz com que, para que a sua experiência como usuário seja positiva, o sistema precise ser inteiramente navegável por teclado. Além disso, é necessário que exista ordem lógica na página, foco visível e a garantia de que o usuário consegue entrar e sair de qualquer componente apenas com o teclado.
Uso de Leitor de Tela	**Botão de remover item sem rótulo:** No carrinho de compras, o botão para excluir um item é representado apenas pelo ícone de uma lixeira, sem um nome acessível associado (por exemplo, via atributo `aria-label` ou texto equivalente). O leitor de telas reproduz apenas a palavra genérica "botão", o que impede a continuidade do fluxo de pagamento com segurança.

Ao final do levantamento de barreiras, as dificuldades mapeadas evoluem de falhas técnicas abstratas para a compreensão de seu impacto real na experiência de compra de um usuário. Isso permite que a equipe de desenvolvimento priorize correções de *UI/UX* com o objetivo de eliminar os bloqueios que impedem a inclusão de diferentes tipos de usuário.

4.8. Limitações da WCAG e Requisitos Além da Conformidade

A Seção 4.6 estabeleceu que ferramentas automatizadas identificam apenas parte dos problemas de acessibilidade e que a avaliação manual é necessária para complementá-las. Esta seção avança um argumento distinto e mais amplo: mesmo a conformidade plena e corretamente verificada com as diretrizes não garante, por si só, uma experiência acessível. A distância entre conformidade técnica e experiência real de uso constitui um objeto de investigação próprio, com evidência empírica acumulada há quase duas décadas.

4.8.1. Duas definições de acessibilidade

Essa distância tem origem em uma ambiguidade conceitual explicitada por Rømen e Svanæs [39]. Os autores observam que a própria WCAG define acessibilidade de forma minimalista: "content is accessible when it may be used by someone with a disability", ou

seja, basta que seja *possível* a um usuário com deficiência utilizar o conteúdo. Essa definição, que os autores chamam de *WCAG-accessibility*, contrasta com a definição adotada pela norma ISO 9241-171:2008, que trata acessibilidade como usabilidade para pessoas com deficiência, isto é, a capacidade de "achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use". Essa segunda definição, chamada pelos autores de *ISO-accessibility*, é substancialmente mais exigente: um site pode "poder ser usado" por uma pessoa com deficiência (satisfazendo a WCAG) e, ainda assim, apresentar problemas de usabilidade severos que impeçam esse mesmo usuário de atingir seus objetivos concretos [39]. É precisamente essa diferença, entre poder ser usado e poder ser usado com eficácia, eficiência e satisfação, que a evidência empírica a seguir documenta.

4.8.2. Evidências empíricas de cobertura parcial

Três estudos são particularmente relevantes para dimensionar essa distância.

Petrie e Kheir [37] investigaram a relação entre acessibilidade e usabilidade por meio de testes com doze participantes, seis pessoas cegas e seis pessoas sem deficiência visual, como grupo de controle, em dois sites comerciais de telefonia móvel (Orange e T-Mobile, Reino Unido). Os problemas encontrados pelos dois grupos formaram conjuntos que se intersectam em aproximadamente 14% quando os dados dos dois sites são combinados (17% no site Orange e 10,6% no T-Mobile) [37]. Mais significativo do que o percentual de sobreposição é o segundo achado do estudo: a correlação entre a severidade atribuída pelos participantes cegos aos problemas e a classificação de importância fornecida pelas diretrizes de usabilidade do *U.S. Department of Health and Human Services* (HHS) foi *negativa* e estatisticamente significativa ($r = -0{,}244$ a $-0{,}273$, $p \leq 0{,}05$), ou seja, problemas que as diretrizes classificam como mais importantes tenderam a ser avaliados pelos próprios usuários como *menos* severos. Não houve, tampouco, concordância significativa entre as avaliações de participantes ou pesquisadores e os níveis de prioridade da WCAG [37].

Rømen e Svanæs [39] conduziram testes de usabilidade controlados com sete pessoas com deficiência (três com deficiência visual, dois com deficiência motora e dois com dislexia) e um grupo de controle de seis pessoas, em dois sites de administrações municipais norueguesas. Do total de 80 problemas distintos identificados, 47 (58,75%) foram experimentados apenas por pessoas com deficiência. Desses 47 problemas de acessibilidade, apenas 13 (27%) poderiam ter sido identificados pela aplicação da WCAG 1.0, e apenas 15 (32%) pela WCAG 2.0, uma melhoria marginal de apenas 5 pontos percentuais entre versões. Combinando as duas versões da norma, a cobertura sobe para 38%. Compensando o número reduzido de participantes por meio de testes de confiança, os valores permaneceram baixos: há 95% de confiança de que a conformidade com a WCAG 1.0 fique abaixo de 42%, com a WCAG 2.0 abaixo de 49%, e com as duas versões combinadas abaixo de 50% [39]. O dado mais contundente do estudo, porém, refere-se à severidade: dos seis problemas classificados como críticos, aqueles que impedem completamente a realização de uma tarefa, apenas um seria identificado pela WCAG 1.0 e apenas dois pela WCAG 2.0; mesmo combinando as duas versões, três dos seis problemas críticos permaneceriam sem cobertura normativa [39]. Os autores concluem que a aplicação das diretrizes da WAI não é suficiente para garantir a acessibilidade de um site e recomendam

que versões futuras sejam fundamentadas e validadas empiricamente, além de defenderem a ampliação da definição de acessibilidade adotada pela WAI de modo a incorporar a noção de ISO-accessibility descrita acima.

O estudo de maior escala é o de Power, Freire, Petrie e Swallow [38], que analisou os problemas encontrados por 32 pessoas cegas em avaliações baseadas em tarefas realizadas em 16 sites, gerando 1.383 instâncias de problemas. Apenas 50,4% dessas instâncias eram cobertas por critérios de sucesso (*Success Criteria*, SC) da WCAG 2.0. Mais relevante para o argumento desta seção é o segundo resultado: dos problemas cobertos pela norma, 16,7% — equivalentes a 8,4% de todos os problemas relatados — corresponderam a casos em que a técnica recomendada pela WCAG 2.0 *havia sido implementada corretamente* e, ainda assim, o problema persistiu para o usuário [38]. Um exemplo concreto ilustra o mecanismo: na categoria "destino do link não é claro", que respondeu por 117 problemas (8,5% do total), 35% dos casos ocorreram em páginas que haviam implementado corretamente o critério de sucesso 2.4.4 (*Link Purpose*), e os usuários ainda assim não conseguiam determinar para onde o link levava. Os próprios autores concluem que "as técnicas suficientes para este critério [...] não são, de fato, suficientes" [38]. Os autores identificam ainda seis categorias de problemas com cobertura *nula* (0%) pela WCAG 2.0, que juntas correspondem a 18,5% de todos os problemas relatados: conteúdo encontrado onde não era esperado pelo usuário, conteúdo não encontrado onde era esperado, páginas lentas para carregar, ausência de alternativa a formatos de documento (como PDF), arquitetura de informação excessivamente complexa, e links quebrados [38]. Com base nesses achados, os autores defendem a transição de uma abordagem baseada em problemas para uma abordagem baseada em princípios de design para a acessibilidade Web.

4.8.3. Escopo das diretrizes: um contraponto necessário

Esses resultados não devem ser interpretados como evidência de falha das diretrizes, e sim como delimitação de seu escopo. Os próprios Rømen e Svanæs reconhecem essa defesa possível da WCAG: "in defence of WCAG one could argue that WCAG in its very definition of accessibility never promised that conformance to its guidelines would result in usability for disabled users [...], it only promised that the resulting sites 'may be used by someone with a disability' " [39]. De modo semelhante, o próprio grupo de trabalho da WCAG registrou uma resposta ao estudo de Power et al., argumentando que parte dos problemas relatados constitui questões de usabilidade apresentadas como questões de acessibilidade, e que há concordância entre revisores de que os problemas apontados como não cobertos pela WCAG 2.0 não seriam algo que diretrizes de acessibilidade pudessem ou devessem abordar especificamente [56].

A questão pertinente, portanto, não é se a WCAG falha, mas onde termina o escopo legítimo de uma norma verificável e onde começa a responsabilidade de projeto. Essa reformulação, entre WCAG-accessibility e ISO-accessibility, é o que sustenta o restante desta seção.

4.8.4. A testabilidade como origem estrutural das lacunas

A principal restrição de escopo da WCAG é metodológica e está documentada pelo próprio W3C. Todo critério de sucesso da WCAG precisa ser testável, de modo que autores possam determinar objetivamente se estão ou não em conformidade, e essa exigência é apontada pela própria força-tarefa de acessibilidade cognitiva como uma das razões pelas quais determinadas necessidades de usuários não foram incorporadas às diretrizes [42].

O efeito dessa restrição é assimétrico entre perfis de deficiência. Necessidades de pessoas com deficiência visual, auditiva ou motora tendem a se traduzir com relativa facilidade em condições binárias verificáveis no código; necessidades relacionadas a compreensão, memória, atenção e carga cognitiva, não. Como resultado, apenas três dos novos critérios de sucesso introduzidos na WCAG 2.1 podem ser diretamente atribuídos ao trabalho da *Cognitive and Learning Disabilities Accessibility Task Force* (COGA): os critérios 1.3.5 (Identify Input Purpose), 1.3.6 (Identify Purpose) e 2.2.6 (Timeouts) [57, 58, 59].

O reconhecimento mais explícito dessa lacuna está no documento *Making Content Usable for People with Cognitive and Learning Disabilities* [40], produzido pelo próprio W3C. O documento declara que suas orientações e padrões constituem material suplementar aos requisitos da WCAG, que segui-los não é exigido para conformidade, e que eles endereçam barreiras de acessibilidade não contempladas na WCAG 2.x normativa. Existe, portanto, um corpo consolidado de orientação produzido pelo próprio organismo normatizador que permanece deliberadamente fora do escopo da conformidade. Guias específicos por perfil, como o GAIA [50] no contexto brasileiro, cumprem função análoga.

4.8.5. O modelo de conformidade e sua revisão em curso

Uma segunda limitação decorre do modelo de conformidade adotado pela WCAG 2.x, que é binário e organizado por página: cada critério de sucesso é aprovado ou reprovado, sem gradação intermediária que reflita a severidade do impacto sobre o usuário ou o progresso alcançado.

Essa limitação é reconhecida pelo grupo de trabalho responsável pela próxima geração das diretrizes. O documento explicativo da WCAG 3.0 registra que o grupo tem explorado classificações de severidade que possam contribuir para pontuação e priorização, bem como classificações adjetivais que permitam que resultados de teste vão além de “aprovado” ou “reprovado”, indicando progresso em direção a uma meta ou sua superação, em uma escala exemplificada como *Fail*, *Progress*, *Pass*, *Better* e *Exceptional* [41]. Cabe ressaltar que a WCAG 3.0 encontra-se em estágio de rascunho de trabalho (*Working Draft*) e não constitui norma vigente nem referência para conformidade legal; a WCAG 2.2 [6] permanece a recomendação em vigor.

4.8.6. Conformidade performática: o caso dos *overlays*

A distância entre conformidade declarada e acessibilidade efetiva manifesta-se de forma particularmente evidente no caso dos *overlays* de acessibilidade, soluções de terceiros, tipicamente em JavaScript, que prometem corrigir automaticamente barreiras de acessibilidade a partir da inserção de um único trecho de código na página, frequentemente comercializadas também como proteção contra litígio.

A evidência disponível é consistentemente desfavorável. Na pesquisa da WebAIM com 758 profissionais de acessibilidade, conduzida em janeiro de 2021, 67% dos respondentes classificaram essas ferramentas como pouco ou nada eficazes; entre respondentes com deficiência, essa proporção sobe para 72%, e apenas 2,4% as consideraram muito eficazes [51]. Em estudo revisado por pares que combinou questionário e entrevistas semiestruturadas com pessoas cegas e com baixa visão, Makati, Tigwell e Shinohara [52] relatam que a maioria dos participantes considerou os overlays inúteis, e que eles frequentemente agravavam os problemas de acessibilidade em vez de resolvê-los; os participantes relataram com frequência que os overlays entravam em conflito com suas próprias recursos de TA, reduzindo a funcionalidade e aumentando a frustração. Os autores defendem o abandono de correções superficiais em favor de práticas de design centrado no usuário que incorporem a acessibilidade desde a concepção do produto [52].

O caso é ilustrativo porque explicita o risco de tratar a acessibilidade como problema de conformidade a ser resolvido no final do ciclo de desenvolvimento, e não como requisito de qualidade incorporado desde a concepção.

4.8.7. Necessidades de acesso conflitantes

Uma limitação final não é atribuível à WCAG nem a qualquer conjunto de diretrizes, mas à natureza do problema. Lidar com necessidades de acesso variadas e, por vezes, conflitantes tornou-se uma consideração central no design para acessibilidade: quando um recurso beneficia algumas pessoas ao mesmo tempo em que restringe outras, não existe critério normativo capaz de resolver o impasse. Tang e Piper [54], a partir de um estudo de campo de seis meses com uma organização sem fins lucrativos de composição mista de usuários, argumentam pela reformulação dessa questão: em vez de tratar necessidades conflitantes como problemas técnicos a serem eliminados pelo design, os autores defendem repensar esses "conflitos" como locais-chave para revelar estruturas de poder e criar oportunidades de responsabilização e reparação [54].

Esse é precisamente o tipo de decisão que a atividade com personas descrita na Seção 4.7 procura tornar visível: há escolhas de projeto que exigem deliberação e priorização explícitas, e que nenhuma lista de verificação pode substituir.

4.8.8. Síntese

A discussão apresentada não invalida as diretrizes existentes, tampouco autoriza sua relativização. Considerando o quadro normativo brasileiro discutido na Seção 4.2, em particular o artigo 63 da Lei Brasileira de Inclusão [11] e a ABNT NBR 17225 [13], a conformidade constitui obrigação legal e piso mínimo. O que a evidência empírica indica é que ela é condição necessária, mas não suficiente, a distinção entre WCAG-accessibility e ISO-accessibility de Rømen e Svanæs [39] sintetiza precisamente esse limite.

Quatro implicações práticas decorrem dessa discussão: (i) a conformidade deve ser tratada como ponto de partida, e não como objetivo final; (ii) a avaliação deve ser conduzida em camadas, combinando verificação automatizada, avaliação manual especializada e testes com usuários reais; (iii) orientações suplementares específicas por perfil, como as produzidas pela COGA [40] e pelo GAIA [50], devem complementar a norma naquilo que ela deliberadamente não cobre; e (iv) pessoas com deficiência devem partici-

par do processo de desenvolvimento, e não apenas de sua etapa final de validação.

4.9. Conclusão

Este capítulo percorreu, do início ao fim, o caminho que a acessibilidade Web trilhou até se tornar o que é hoje: partiu das primeiras discussões dos anos 1970 e 1980 sobre computação pessoal, passou pela criação do W3C e da WAI nos anos 1990, atravessou décadas de avanços legislativos no Brasil até chegar ao artigo 63 da Lei Brasileira de Inclusão e à recém publicada ABNT NBR 17225, e desembocou nas diretrizes técnicas da WCAG, no HTML semântico, nos atributos ARIA, nas recursos de TA e nas ferramentas automatizadas de avaliação apresentadas ao longo do texto. Esse percurso mostra que a acessibilidade digital não é um tema recente nem um modismo passageiro, mas o resultado de décadas de construção coletiva, técnica e legal, em direção a uma Web mais justa.

Ainda assim, todo esse arcabouço de normas e recursos só ganha sentido pleno quando encontra o usuário real, e é exatamente esse encontro que a criação de personas proporciona. Ao dar rosto, contexto e tecnologia assistiva a critérios que, isolados, permanecem abstratos, as personas transformam diretrizes técnicas em decisões concretas de projeto e permitem enxergar, antes mesmo da implementação, barreiras que de outra forma só apareceriam depois que o produto já estivesse no ar, quando corrigi-las custa mais caro e afeta mais pessoas. Por isso, entre tudo o que foi discutido neste capítulo, a construção e a análise de personas com diferentes perfis de deficiência funcionam como o verdadeiro eixo central: é o ponto em que a norma deixa de ser checklist e passa a ser empatia orientada por evidência.

Ao mesmo tempo, é preciso reforçar que conformidade técnica não é sinônimo de acessibilidade real. Passar em uma ferramenta automatizada, atingir o nível AA da WCAG ou seguir à risca uma norma como a NBR 17225 garante um piso mínimo, necessário e legalmente exigido, mas não garante, por si só, que a pessoa com deficiência consiga usar o site com autonomia, eficiência e satisfação. A distância entre o que a norma exige e o que o usuário de fato vive é o espaço que caberá sempre às equipes de desenvolvimento e design preencherem, e é justamente aí que a escuta de pessoas reais, a construção cuidadosa de personas e a avaliação manual complementar às ferramentas automáticas se tornam indispensáveis.

Por fim, cabe uma reflexão dirigida às próprias empresas e organizações que constroem produtos digitais. A acessibilidade ainda costuma ser tratada como item secundário, adicionado às pressas no fim do desenvolvimento ou terceirizado a soluções superficiais que prometem resolver tudo com uma linha de código, quando na verdade deveria ser incorporada desde a concepção do produto, como requisito de qualidade e não como obrigação a ser cumprida no último momento. Investir em acessibilidade não é apenas questão de conformidade legal ou de evitar litígios: é reconhecer que milhões de pessoas dependem de experiências digitais bem construídas para estudar, trabalhar, se comunicar e exercer sua autonomia. Quanto mais cedo as empresas entenderem que acessibilidade é parte do produto e não um adicional, mais perto a Web estará de cumprir sua promessa original: ser, de fato, para todos.

Agradecimentos

Este trabalho contou com o apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), por meio de bolsa de mestrado, processo nº 88887.371070/2026-00.

Referências

[1] W3C (1997) "Web Accessibility Initiative (WAI)", World Wide Web Consortium, https://www.w3.org/WAI/, acesso em: 4 set. 2026.

[2] UNITED STATES CONGRESS (1998) "Section 508 of the Rehabilitation Act", 29 U.S.C. § 794d, Washington, D.C.

[3] CHISHOLM, W.; VANDERHEIDEN, G.; JACOBS, I. (1999) "Web Content Accessibility Guidelines (WCAG) 1.0", W3C Recommendation, 5 May, https://www.w3.org/TR/WAI-WEBCONTENT/, acesso em: 3 set. 2026.

[4] CALDWELL, B.; COOPER, M.; GUARINO REID, L.; VANDERHEIDEN, G. (2008) "Web Content Accessibility Guidelines (WCAG) 2.0", W3C Recommendation, 11 December, https://www.w3.org/TR/WCAG20/, acesso em: 3 set. 2026.

[5] KIRKPATRICK, A.; O CONNOR, J.; CAMPBELL, A.; COOPER, M. (2018) "Web Content Accessibility Guidelines (WCAG) 2.1", W3C Recommendation, 5 June, https://www.w3.org/TR/WCAG21/, acesso em: 3 set. 2026.

[6] KIRKPATRICK, A.; O CONNOR, J.; CAMPBELL, A.; COOPER, M. (2023) "Web Content Accessibility Guidelines (WCAG) 2.2", W3C Recommendation, 5 October, https://www.w3.org/TR/WCAG22/, acesso em: 3 set. 2026.

[7] BRASIL (2000) "Lei nº 10.098, de 19 de dezembro de 2000", Diário Oficial da União, Brasília.

[8] BRASIL (2004) "Decreto nº 5.296, de 2 de dezembro de 2004", Diário Oficial da União, Brasília.

[9] BRASIL (2009) "Decreto nº 6.949, de 25 de agosto de 2009", Diário Oficial da União, Brasília.

[10] BRASIL (2012) "Decreto nº 7.724, de 16 de maio de 2012", Diário Oficial da União, Brasília.

[11] BRASIL (2015) "Lei nº 13.146, de 6 de julho de 2015 (Lei Brasileira de Inclusão)", Diário Oficial da União, Brasília.

[12] BRASIL. Ministério do Planejamento, Desenvolvimento e Gestão (2014) "Modelo de Acessibilidade em Governo Eletrônico (eMAG), versão 3.1", Departamento de Governo Eletrônico, Brasília.

[13] ABNT (2025) "NBR 17225: Acessibilidade em conteúdos e aplicações web", Associação Brasileira de Normas Técnicas, Rio de Janeiro.

[14] INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA (IBGE) (2023) “Pessoas com deficiência têm menor acesso à educação, ao trabalho e à renda”, Agência de Notícias, 7 jul., https://agenciadenoticias.ibge.gov.br/agencia-noticias/2012-agencia-de-noticias/noticias/37317-pessoas-com-deficiencia-tem-menor-acesso-a-educacao-ao-trabalho-e-a-renda, acesso em: 5 set. 2026.

[15] MORO, P.; HENRIQUES, M. (2025) “Falta de acessibilidade digital ainda é desafio para pessoas com deficiência: pesquisa da UFSM investiga consequências da falta de acessibilidade em veículos de comunicação”, Agência de Notícias UFSM, 7 ago., https://www.ufsm.br/2025/08/07/falta-de-acessibilidade-digital-ainda-e-desafio-para-pessoas-com-deficiencia, acesso em: 5 set. 2026.

[16] DEQUE SYSTEMS (2021) “The Automated Accessibility Coverage Report”, https://www.deque.com/automated-accessibility-coverage-report/, acesso em: 2 set. 2026.

[17] GOVERNMENT DIGITAL SERVICE (GDS) (2018) “Accessibility tools audit results”, GDS Accessibility Team, United Kingdom, https://alphagov.github.io/accessibility-tool-audit/, acesso em: 3 set. 2026.

[18] WEBAIM (2026) “The WebAIM Million: An Annual Accessibility Analysis of the Top 1,000,000 Home Pages”, https://webaim.org/projects/million/, acesso em: 4 set. 2026.

[19] WEBAIM (2024) “Screen Reader User Survey”, https://webaim.org/projects/screenreadersurvey/, acesso em: 5 set. 2026.

[20] Lighthouse (2026) “Introduction to Lighthouse”, https://developer.chrome.com/docs/lighthouse/overview, acesso em: 4 set. 2026.

[21] Axe DevTools (2026) “Axe DevTools - The ultimate digital accessibility testing solution”, https://www.deque.com/axe/devtools/, acesso em: 4 set. 2026.

[22] WAVE (2026) “WAVE web accessibility evaluation tool”, https://wave.webaim.org/, acesso em: 4 set. 2026.

[23] Bersch, R. (2017) “Introdução à Tecnologia Assistiva”, CVI-Compreender Centro de Acessibilidade Assistiva, Porto Alegre, RS.

[24] MDN Web Docs (2023) “Document Object Model (DOM)”, Mozilla Developer Network, https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model, acesso em: 4 set. 2026.

[25] MDN Glossary (2026) “Accessibility tree”, Mozilla Developer Network, https://developer.mozilla.org/en-US/docs/Glossary/Accessibility_tree, acesso em: 5 set. 2026.

[26] NV Access (2023) “About NVDA”, NV Access, https://www.nvaccess.org/about-nvda/, acesso em: 4 set. 2026.

[27] VLibras (2023) “Suíte VLibras”, Governo do Brasil, https://www.gov.br/governodigital/pt-br/vlibras, acesso em: 4 set. 2026.

[28] W3C (2023) “Accessible Rich Internet Applications (WAI-ARIA) 1.2”, World Wide Web Consortium, https://www.w3.org/TR/wai-aria-1.2/, acesso em: 4 set. 2026.

[29] W3C (2024) “ARIA Authoring Practices Guide (APG)”, World Wide Web Consortium, https://www.w3.org/WAI/ARIA/apg/, acesso em: 5 set. 2026.

[30] MDN Web Docs (2024) “Semântica – Glossário do MDN Web Docs”, Mozilla, https://developer.mozilla.org/pt-BR/docs/Glossary/Semantics, acesso em: 5 set. 2026.

[31] MDN Web Docs (2024) “Referência de elementos HTML”, Mozilla, https://developer.mozilla.org/pt-BR/docs/Web/HTML/Element, acesso em: 5 set. 2026.

[32] W3C (2024) “HTML Specification: Semantics, structure, and APIs of HTML documents”, World Wide Web Consortium, https://www.w3.org/TR/html52/semantics.html, acesso em: 5 set. 2026.

[33] Google Developers (2024) “Learn HTML: Semantic HTML”, Google, https://web.dev/learn/html/semantic-html, acesso em: 5 set. 2026.

[34] Apple (2026) “Manuel de Uso do VoiceOver”, Apple, https://support.apple.com/pt-br/guide/voiceover/welcome/mac, acesso em: 5 set. 2026.

[35] Tecassistiva (2026) “Jaws Screen Reader”, Tecassistiva, https://tecassistiva.com.br/produtos/jaws/, acesso em: 5 set. 2026.

[36] Vispero (2026) “ZoomText Screen Magnification”, Vispero, https://vispero.com/zoomtext-screen-magnifier-software/, acesso em: 5 set. 2026.

[37] PETRIE, H.; KHEIR, O. (2007) “The relationship between accessibility and usability of websites”, In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI ’07)*, ACM, New York, p. 397–406. https://doi.org/10.1145/1240624.1240688

[38] POWER, C.; FREIRE, A. P.; PETRIE, H.; SWALLOW, D. (2012) “Guidelines are only half of the story: accessibility problems encountered by blind users on the web”, In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI ’12)*, ACM, New York, p. 433–442. https://doi.org/10.1145/2207676.2207736

[39] RØMEN, D.; SVANÆS, D. (2012) “Validating WCAG versions 1.0 and 2.0 through usability testing with disabled users”, *Universal Access in the Information Society*, v. 11, n. 4, p. 375–385. https://doi.org/10.1007/s10209-011-0259-3

[40] W3C (2021) “Making Content Usable for People with Cognitive and Learning Disabilities”, W3C Group Note, https://www.w3.org/TR/coga-usable/, acesso em: 4 set. 2026.

[41] W3C (2025) “Explainer for W3C Accessibility Guidelines (WCAG) 3.0”, https://w3c.github.io/wcag3/explainer/, acesso em: 4 set. 2026.

[42] W3C COGA TASK FORCE “COGA FAQ”, Cognitive and Learning Disabilities Accessibility Task Force, https://www.w3.org/WAI/GL/task-forces/coga/wiki/COGA-FAQ, acesso em: 4 set. 2026.

[43] COOPER, A. (1999) *The Inmates Are Running the Asylum*, Macmillan Publishing, Indianapolis.

[44] PRUITT, J.; GRUDIN, J. (2003) “Personas: practice and theory”, In: *Proceedings of the 2003 Conference on Designing for User Experiences (DUX ’03)*, ACM, New York, p. 1–15.

[45] SCHULZ, T.; FUGLERUD, K. S. (2012) “Creating Personas with Disabilities”, In: *Computers Helping People with Special Needs (ICCHP 2012)*, LNCS 7383, Springer, Berlin, p. 145–152. https://doi.org/10.1007/978-3-642-31534-3_22

[46] NEATE, T.; BOURAZERI, A.; ROPER, A.; STUMPF, S.; WILSON, S. (2019) “Co-Created Personas: Engaging and Empowering Users with Diverse Needs Within the Design Process”, In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI ’19)*, ACM, New York, p. 1–12. https://doi.org/10.1145/3290605.3300880

[47] EDWARDS, E. J.; SUM, C. M.; BRANHAM, S. M. (2020) “Three Tensions Between Personas and Complex Disability Identities”, In: *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems (CHI EA ’20)*, ACM, New York, p. 1–9. https://doi.org/10.1145/3334480.3382931

[48] BENNETT, C. L.; ROSNER, D. K. (2019) “The Promise of Empathy: Design, Disability, and Knowing the ‘Other”’, In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI ’19)*, ACM, New York, Paper 298. https://doi.org/10.1145/3290605.3300528

[49] NARIO-REDMOND, M. R.; GOSPODINOV, D.; COBB, A. (2017) “Crip for a Day: The Unintended Negative Consequences of Disability Simulations”, *Rehabilitation Psychology*, v. 62, n. 3, p. 324–333. https://doi.org/10.1037/rep0000127

[50] BRITTO, T. C. P.; PIZZOLATO, E. B. (2016) “GAIA: uma proposta de um guia de recomendações de acessibilidade de interfaces Web com foco em aspectos do Autismo”, *Revista Brasileira de Informática na Educação*, v. 26, n. 2, p. 102–123.

[51] WEBAIM (2021) “Survey of Web Accessibility Practitioners #3”, https://webaim.org/projects/practitionersurvey3/, acesso em: 2 set. 2026.

[52] MAKATI, T.; TIGWELL, G. W.; SHINOHARA, K. (2024) "The Promise and Pitfalls of Web Accessibility Overlays for Blind and Low Vision Users", In: *Proceedings of the 26th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '24)*, ACM, New York, Article 38, 12 p. https://doi.org/10.1145/3663548.3675650

[53] HEITMEIER, K.-A.; KERSKEN, V.; PISKOREK, P.; BÖHM, A.-K.; EGGER, N.; LANG, M.; ZIMMERMANN, G. (2023) "Persona Co-Design for Improving Digital Accessibility", In: *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems (CHI EA '23)*, ACM, New York, Article 227, 7 p. https://doi.org/10.1145/3544549.3585857

[54] TANG, X.; PIPER, A. M. (2026) "Designing for Collective Access: In Search of a Solution to Accessible Communication in a Mixed-Ability Non-Profit", In: *Proceedings of the 2026 Designing Interactive Systems Conference (DIS '26)*, ACM, New York, p. 4400–4415. https://doi.org/10.1145/3800645.3813075

[55] SECTION508.GOV "Sample Personas for Users With Disabilities", U.S. General Services Administration, https://www.section508.gov/develop/sample-personas/, acesso em: 4 set. 2026.

[56] CARLSON, L. "Review: Guidelines are only half of the story", WCAG Working Group Wiki, W3C, https://www.w3.org/WAI/GL/wiki/Review:_Guidelines_are_only_half_of_the_story, acesso em: 2 set. 2026.

[57] W3C (2023) "Understanding Success Criterion 1.3.5: Identify Input Purpose", Web Accessibility Initiative (WAI), https://www.w3.org/WAI/WCAG22/Understanding/identify-input-purpose.html, acesso em: 4 set. 2026.

[58] W3C (2018) "Understanding Success Criterion 1.3.6: Identify Purpose", Web Accessibility Initiative (WAI), https://w3c.github.io/wcag21/understanding/identify-purpose.html, acesso em: 4 set. 2026.

[59] W3C (2018) "Understanding Success Criterion 2.2.6: Timeouts", Web Accessibility Initiative (WAI), https://www.w3.org/WAI/WCAG22/Understanding/timeouts.html, acesso em: 4 set. 2026.

[60] BERNERS-LEE, T. (1999) "Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web by Its Inventor", *HarperSanFrancisco*.

[61] JANSEN, B. J.; JUNG, S.-G; NIELSEN, L.; GUAN, K. W.; SALMINEN, J. (2022) "How to Create Personas: Three Persona Creation Methodologies with Implications for Practical Employment", *Pacific Asia Journal of the Association for Information Systems*, v. 14, n. 3, p. 1–28

[62] JANSEN, B. J.; JUNG, S.-G; NIELSEN, L.; GUAN, K. W.; SALMINEN, J. (2022) "How to Create Personas: Three Persona Creation Methodologies with Implications

for Practical Employment", *Pacific Asia Journal of the Association for Information Systems*, v. 14, n. 3, p. 1–28

[63] BILLESTRUP, J.; STAGE, J.; BRUUN, A.; NIELSEN, L.; NIELSEN, K. S. (2014) "Creating and Using Personas in Software Development: Experiences from Practice", In: *Sauer, S., Bogdan, C., Forbrig, P., Bernhaupt, R., Winckler, M. (eds) Human-Centered Software Engineering. HCSE 2014. Lecture Notes in Computer Science, vol 8742)*, Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-662-44811-3$_1$6

Chapter 5

Wikidata in the Era of Generative AI: Interfaces, Techniques, and Applications

Marcelo Machado, João Pedro Porto Campos, Alexya Silva, Gabriel Banaggia, and Guilherme Lima

Abstract

We present an introduction to the Wikidata knowledge graph with emphasis on the interfaces and techniques used to build generative AI applications. Wikidata is one of the largest knowledge graphs available on the Web and plays a crucial role today as a provider of authenticated information and equitable representation of world knowledge to AI applications. We start by presenting Wikidata itself, its data model, RDF encoding, and access interfaces, and then shift the focus to techniques and applications that combine LLMs and Wikidata. Two perspectives are explored. In the Wikidata for AI perspective, we discuss how Wikidata can be used to ground and contextualize AI applications, while in the AI for Wikidata perspective, we discuss the ways in which AI models can assist with the construction, enrichment, verification, and curation of Wikidata itself.

Resumo

Apresentamos uma introdução ao grafo de conhecimento Wikidata com foco nas interfaces e técnicas utilizadas para construção de aplicações de IA generativa. O Wikidata é um dos maiores grafos de conhecimento disponíveis na Web e desempenha hoje um papel crucial como fornecedor de informação autenticada e representação equitativa de conhecimento de mundo para aplicações de IA. Começamos apresentando o Wikidata em si, seu modelo de dados, codificação RDF e interfaces de acesso, e em seguida mudamos o foco para técnicas e aplicações que utilizam LLMs juntamente com o Wikidata. São exploradas duas perspectivas. Na perspectiva Wikidata para IA, discutimos como o Wikidata pode ser usado para ancorar factualmente e contextualizar aplicações de IA. Já na perspectiva IA para Wikidata, discutimos as formas pelas quais modelos de IA podem ajudar na construção, enriquecimento, verificação, e curadoria do próprio Wikidata.

5.1. Introduction

Wikidata[1] [Vrandečić and Krötzsch 2014] is a sister project of Wikipedia[2] and it is also one of largest bases of structured knowledge publicly-available on the Web. Strictly speaking, Wikidata is a knowledge graph or KG, that is, a database structured as a (directed) graph in which nodes represent entities and edges represent relationships between pairs of entities. But there is more to Wikidata than just the knowledge graph. As Wikipedia, Wikidata is a Wikimedia Foundation[3] project maintained by a large community of volunteers (at the time of writing 43,357 active users and 326 bots[4]). Since its launch in 2012, Wikidata has grown into one of the largest general-purpose knowledge graphs, providing multilingual, machine-readable descriptions of entities and their relationships which are reused across other Wikimedia Foundation projects. For example, some of the data of the "infoboxes" of Wikipedia come directly from Wikidata [Sáez and Hogan 2018]. Many other projects also reuse Wikidata content including digital libraries, cultural heritage institutions, and countless third-party applications.

More recently, the role of Wikidata as a source of authoritative and equitable knowledge has taken on a new significance with the rise of generative artificial intelligence (AI) and large language models (LLMs). LLMs have exhibited remarkable text manipulation and reasoning abilities, often far surpassing those of traditional natural language processing methods. At the same time, LLMs are inherently stochastic and are known to suffer from hallucinations [Ji et al. 2023], biases [Navigli et al. 2023], and overreliance on outdated information [Mousavi et al. 2026]. So, while LLMs have enabled applications to navigate and process large quantities of text, they still lack grounding: their outputs while seemingly plausible might be factually incorrect, biased, or simply outdated. This is where Wikidata comes in. It provides precisely the kind of structured, authoritative, equitable, and up-to-date knowledge needed to ground the output of LLMs.

But using Wikidata as a source of trusted information for AI systems is just one side of the story. Another interesting possibility is leveraging the natural language understanding and generation capabilities of LLMs to support the development and growth of Wikidata itself. For instance, LLMs can be used to extract information from unstructured sources in order to predict missing entities or relationships in the graph, enrich statements with references and qualifiers, and help in identifying incorrect or misleading statements.

In this chapter, we present Wikidata in detail and discuss how emerging AI techniques can be used to construct richer AI applications using or targeting Wikidata. The chapter is divided into two parts. The first part (Section 5.2) presents the Wikidata data model, RDF encoding, and main access interfaces, including official ones (SPARQL, REST APIs, and data dumps) but also alternatives interfaces (TPF and KIF). The second part (Section 5.3), discusses the intersection between Wikidata and generative AI from the complementary perspectives mentioned to above. In *Wikidata for AI* (Section 5.3.1), we discuss how the knowledge graph can be used to ground and contextualize AI applications. In *AI for Wikidata* (Section 5.3.2), we discuss the ways in which AI models can assist with

[1] https://www.wikidata.org/

[2] https://www.wikipedia.org/

[3] https://wikimediafoundation.org/

[4] https://www.wikidata.org/wiki/Special:Statistics

the construction, enrichment, verification, and curation of the knowledge graph. At the end of the chapter, we discuss some projects maintained by the Wikidata community which aim to ease the integration of Wikidata with generative AI systems. These include Wikidata Embedding and Vector Database projects, for semantic retrieval, and the Wikidata Model Context Protocol (MCP), for interfacing with AI agents.

5.2. The Wikidata Knowledge Graph

The Wikidata knowledge graph is the central repository of structured data for Wikimedia Foundation projects, including Wikipedia, and beyond. Although the user interface (UI) of Wikidata is organized in pages describing particular subjects, under the hood Wikidata is structured as a graph. The building blocks of this graph are entities and statements and the pages we see in Wikidata's UI are essentially human-friendly way to visualize and edit parts of this graph.

In this section, we take a deep dive into the structure of the Wikidata knowledge graph. We start in by presenting in detail Wikidata's conceptual data model. We then shift to the technologies on top of which this data model is implemented. One of these technologies is RDF (Resource Description Framework [Kellogg et al. 2026]) which is at the core of the Semantic Web standard[5]. RDF is a graph-based representation language that provides a vocabulary and constructors for describing sets of triples denoting directed graphs. The Wikidata infrastructure leverages RDF and other Semantic Web technologies (such as SPARQL) to encode and provide standard access interfaces to the Wikidata knowledge graph. As we will see, the Wikidata knowledge graph can also be accessed through other non-standard means, such as via data dumps or REST APIs, which might be more convenient depending on the application.

5.2.1. Wikidata Data Model

The data model of Wikidata consists of entities and statements about entities.[6] In the Wikidata user interface, statements are grouped into "entity pages". Figure 5.1 depicts a part of the page of entity `Q1638425`, which stands for the Brazilian city of Lavras[7], located in the state of Minas Gerais. (At the time of writing, there are more than 123 million entity pages like this one in Wikidata.) Every Wikidata entity has a label, a description, and one or more aliases in a given language. In Figure 5.1, these are shown in the header at the top of the page. So, the English label of entity `Q1638425` is "Lavras", its English description is "town and municipality in the state of Minas Gerais, Brazil", and it as a single English alias "Arraial de Santana das Lavras do Funil".

After the header comes the "Statements" section which groups statements about the entity being described. A *statement* consists of two parts: subject and snak. The *subject* is the entity about which the statement is made. The *snak* is the statement's claim. It associates a property with either a specific value, some unspecified value, or no value.

In Figure 5.1, two statements about the population of Lavras are depicted. These can be read as follows:

[5]https://en.wikipedia.org/wiki/Semantic_Web_Stack

[6]https://www.mediawiki.org/wiki/Wikibase/DataModel

[7]https://www.wikidata.org/wiki/Q1638425

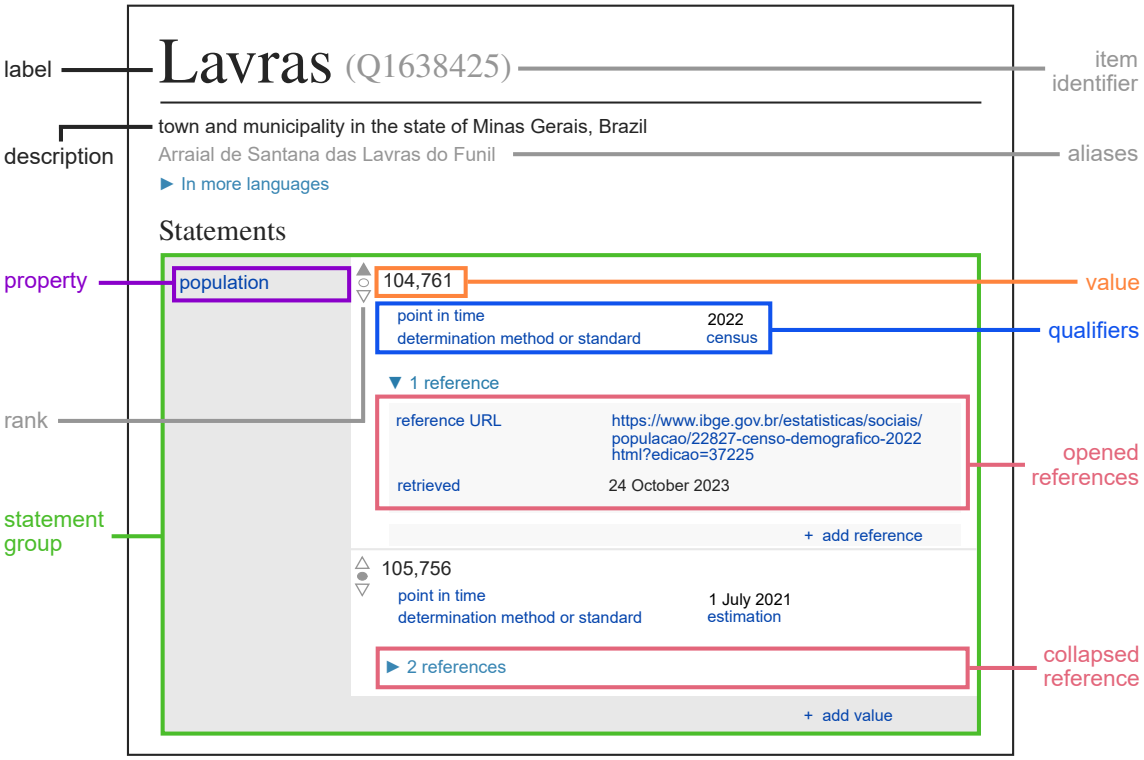


Figure 5.1: Part of Wikidata's entity page of Lavras (Q1638425). (Adapted from [Odell et al. 2022].)

1. "Lavras has a population of 104,761"; and
2. "Lavras has a population of 105,756".

The subject of statements (1) and (2) is the same, the entity "Lavras" (Q1638425). Their snak is of the form property-value. The property of both is "population" (P1082). The value of statement (1) is "104,761" and the value of statement (2) is "105,756". Note that the data model distinguishes between two types of entities: *items* (identified with "Q") and *properties* (identified with "P"). Lavras (Q1638425) is an example of an item, while population (P1082) is an example of a property. Only properties can occur as the first component of snaks.

Figure 5.1 also shows the qualifiers and references associated with each statement. These appear the statement value in the boxes "qualifiers" and "references". *Qualifiers* are extra snaks which qualify what is being said by the statement's main snak. *references* (or *reference records*) are sets of snaks which keep provenance information. In the case of Figure 5.1, the qualifiers convey information which is crucial to interpret the statement, namely, the point in time at which the population counts were performed and method used (census in the case of statement (1) and estimation in the case of statement (2)).

The last piece of metadata associated with statements is the *rank* which can be either "preferred", "normal", or "deprecated". In Figure 5.1, the rank is represented symbolically by the two triangles and circle which occur on the left of the statement's value. A filled upper triangle means preferred rank; a filled circle means normal rank; and a filled lower

triangle means deprecated rank. As can be seen in Figure 5.1, statement (1) has a preferred rank and (2) has normal rank.

As we said previously, the page depicted in Figure 5.1 is just a human-friendly way to visualize statements about a given subject. Under the hood, statements are actually stored as the edges of a graph, whose nodes are Wikidata entities (items or properties) To concretely describe this graph, Wikidata adopts RDF. It encodes the concepts of the data model (item, properties, statements, snaks, etc.) as an RDF triples, which can be manipulated and queried using other standard Semantic Web technologies. We discuss RDF and the encoding of the Wikidata knowledge graph in RDF in detail next.

5.2.2. Resource Description Framework (RDF)

One of the main standards of the Semantic Web is the *Resource Description Framework* (RDF) [Wood et al. 2014]. RDF is a graph-based representation language which provides a vocabulary and constructors for describing sets of triples denoting directed graphs. These triples are called *subject-predicate-object (spo) triples* and have the form

$$\langle \textit{subject}, \textit{predicate}, \textit{object} \rangle .$$

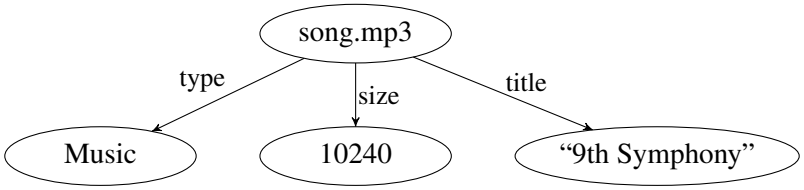
An *spo* triple denotes a labeled edge of a directed graph. For instance, the three *spo* triples below describe the type, size in bytes, and title associated with an audio file:

$\langle$song.mp3, type, Music$\rangle$,

$\langle$song.mp3, size, 10240$\rangle$,

$\langle$song.mp3, title, 9th Symphony$\rangle$.

Here is a diagram depicting the graph corresponding to these three *spo* triples:



Each triple becomes a directed edge in the graph. The subject of the triple is the source of the edge, the object is its target, and the predicate is the edge label. So, the resulting graph is a directed edge-labeled graph.

Of course, this is just an abstract example. In an actual RDF graph, the components of the triples, that is, the labels of nodes and edges in the graph, must obey a specific format (or *syntax*). A popular plain text syntax for RDF is Turtle (Terse RDF Triple Language) [Carothers and Prud'hommeaux 2014]. Turtle is an alternative to the widely supported but more cumbersome RDF/XML syntax. (All of our RDF examples will be written in Turtle.)

The nodes in an RDF graph can be of three kinds: IRI nodes, literal nodes, or blank nodes. An *IRI node* is a node labeled by an IRI (Internationalized Resource Identifier); an IRI is an URI extended with support for Unicode characters. Here is an example IRI:

`https://pt.wiktionary.org/wiki/maçã`

Note the non-ASCII characters "ç" and "ã" above. In an RDF graph, IRIs are used to denote arbitrary *resources*, such as physical objects, documents, images, abstract concepts, numbers, strings, etc.

The second kind of node which can appear in an RDF graph is the literal node. A *literal node* is any node labeled by a literal, that is, a quoted string denoting a value in-place. In the Turtle syntax, literals are written enclosed in double quotes optionally accompanied with language tags and type tags. Here are three RDF literals (written in Turtle):

```
"Rock",   "maçã"@pt,   "10240"^^xsd:integer.
```

The first is a plain literal, the second is a literal with a language tag (Portuguese in this case), and the third is a literal with a type tag, also called typed literal. The prefix `xsd:`, which occurs in the third literal above, stands for the namespace where the type tag `integer` is defined. We will have more to say about namespaces in a moment.

The third kind of node that can occur in an RDF graph is the blank node. A *blank node* has no specific denotation; it is an unlabeled node which is neither an IRI nor literal node. Blank nodes are often used as dummy nodes whose only purpose is to ease the encoding of complex structures in graphs.

In an RDF triple $\langle s, p, o \rangle$, the subject s must be an IRI or blank node, the predicate p must be an IRI, and the object o must be an IRI, literal, or blank node. When we assert an RDF triple we are saying informally that "the relationship indicated by the predicate holds between the resource denoted by the subject and the resource or value denoted by the object".

Figure 5.2 presents a possible Turtle encoding for the graph describing the audio file "song.mp3" depicted a few paragraphs ago.[8] Each edge in the graph (that is, triple in the set) becomes a corresponding triple in the Turtle document (lines 5–7). The subject, predicate, and object parts are separated by spaces and each triple is terminated with a dot (`.`). Note that IRIs are written between angle brackets `<...>`.

```
1 @base <http://example.org/data/> .
2 @prefix ex: <http://example.org/#> .
3 @prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
4 <song.mp3> ex:type ex:Music .
5 <song.mp3> ex:size "10240"^^xsd:integer .
6 <song.mp3> ex:title "9th Symphony" .
```

Figure 5.2: Turtle encoding of the RDF triples describing "song.mp3".

The directives in lines 1–3 of Figure 5.2 define the namespaces of the document. That is, they specify how relative and prefixed IRIs are to be interpreted. The directive `@base` defines the base IRI of the whole Turtle document. Any relative IRI occurring in the document is to be interpreted as relative to this base IRI. So, for instance, the occurrences of `<song.mp3>` in lines 4–6 of Figure 5.2 are to be interpreted as the absolute IRI:

```
http://example.org/data/song.mp3
```

[8] Sometimes literals are represented as rectangles when depicting an RDF graph. Incidentally, we did not follow this convention when depicting the graph describing "song.mp3".

The directive `@prefix` introduces prefix labels which can be used to abbreviate IRIs. So, in Figure 5.2, the labels `ex:` and `xsd:` stand for the following IRI prefixes:

```
http://example.org/#
http://www.w3.org/2001/XMLSchema#
```

Note that prefixed IRIs are *not* written between angle brackets.

Turtle has a number of other features. Some of these are useful for avoiding tedious repetitions within the document. For instance, Figure 5.3 uses some of these features to define in fewer lines the same graph as Figure 5.2. Note the semicolons ";" in lines 3 and 4 of Figure 5.3. These are used to separate predicate-object pairs referring to the same subject; in this case, the IRI `http://example.org/data/song.mp3`. This type of construction is called a *predicate list*. Note also the typed literal 10240 in line 4. Some common types of literals, such as integers and floating-point numbers, can be written in Turtle in their usual syntaxes.

```
1 @prefix ex: <http://example.org/#> .
2 <http://example.org/data/song.mp3>
3   ex:type ex:Music ;
4   ex:size 10240 ;
5   ex:title "9th Symphony" .
```

Figure 5.3: Example of Figure 5.2 rewritten using predicate lists.

It is important to realize that an RDF graph is just a data structure, and that Turtle, RDF/XML, and the other syntaxes for RDF are just ways to encode this data structure in a string. More importantly, the meanings of the labels which occur in an RDF graph are not defined in the RDF standard. These labels might indicate something to the person reading them, but to a computer that understands only RDF these labels are meaningless strings. They serve only to identify and distinguish specific components of the graph.

There is another Semantic Web standard, called RDF Schema (RDFS) [Hayes and Patel-Schneider 2014], which is defined on top of RDF and which specifies a basic vocabulary (set predefined terms) and associated meanings. This vocabulary can be found in the namespace:

`http://www.w3.org/2000/01/rdf-schema#`

RDFS provides a set of basic terms which can be used to define more specialized terms whose meaning is given by describing their interrelations with other terms. The Wikidata RDF encoding, which we discuss next, reuses very little of standard vocabularies, such as RDFS. Instead it adopts its own vocabulary for most representation needs.

5.2.3. Wikidata RDF Encoding

The RDF encoding of Wikidata's data model is an extension of the encoding used by Wikibase[9]. The format varies slightly depending on whether it is used in a data dump or observed from Wikidata's query service. The version we describe here is that of the query service.

[9]https://www.mediawiki.org/wiki/Wikibase/Indexing/RDF_Dump_Format

Throughout this section we will use two statements about the toxicity of the chemical compound benzene (Q2270) as running examples. These two statements are shown in Figure 5.4 as part of benzene's entity page. Both concern property median lethal dose, or LD50, (P2240) which is standard toxicity measure used in chemistry.[10] The first statement (1) says that the median lethal dose of benzene is approximately 4,700 milligrams per kilogram when the route of administration is oral and the target taxon is laboratory mouse. Note that route of administration and target taxon are given in the statement's qualifier. Also, the statement has two references, which are detailed in the figure and which give its provenance.

The second statement (2) shown in Figure 5.4 is similar. It says that the median lethal dose of benzine is approximately 88 milligrams per kilogram when the route of administration is intravenous infusion and defusion and the target taxon is rabbit. Again, the route of administration and target taxon are given in the statement's qualifiers. The second statement has a single reference which is not detailed in the figure.

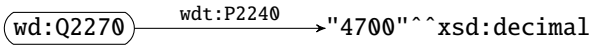
The rationale for using these two statements as examples is that they illustrate some representation features of Wikidata which are not present in standard vocabularies, such as RDF, RDFS, and OWL [W3C OWL WG 2012], nor popular knowledge graphs, such as DBpedia [Lehmann et al. 2015]. One of these features is the native support for numerical data values with associated units (mg/kg) and precision ranges (±1).



Figure 5.4: Part of Wikidata's entity page of benzene (Q2270).

[10]More precisely, LD50 is a toxicity unit that measures the dose of a substance that is required to kill half the members of a given population.

Back to the RDF encoding, Wikidata adopts a two-level representation of statements. The first level, called *truthy*, keeps a shallow representation of the statement as a single RDF triple. For example, the truthy encoding of statement (1) of Figure 5.4, namely, "benzene (Q2270) has an LD50 (P2240) of 4,700±1 mg/kg", consists of the single triple:



(†)

The namespace `wd:` indicates an entity and `wdt:` indicates a truthy application of a property. Some statements are fully characterized at the truthy level. But, as illustrated by (†), this is not always the case. Note that the unit, lower-, and upper-bounds associated with the value 4700 are not represented in (†). In general, when the statement's value is a structured data-value, like a quantity or time value, a single literal is used to represent it at the truthy level. This is the so called *simple value* of the statement. For quantity values, the simple value is is just the numerical amount.

The second level of the encoding keeps the full representation of the statement. It uses reification to capture the *deep value* of the statement plus its qualifiers, references, and rank. Figure 5.5 depicts the full representation of statement (1) of Figure 5.4 considering only one qualifier and one reference record.

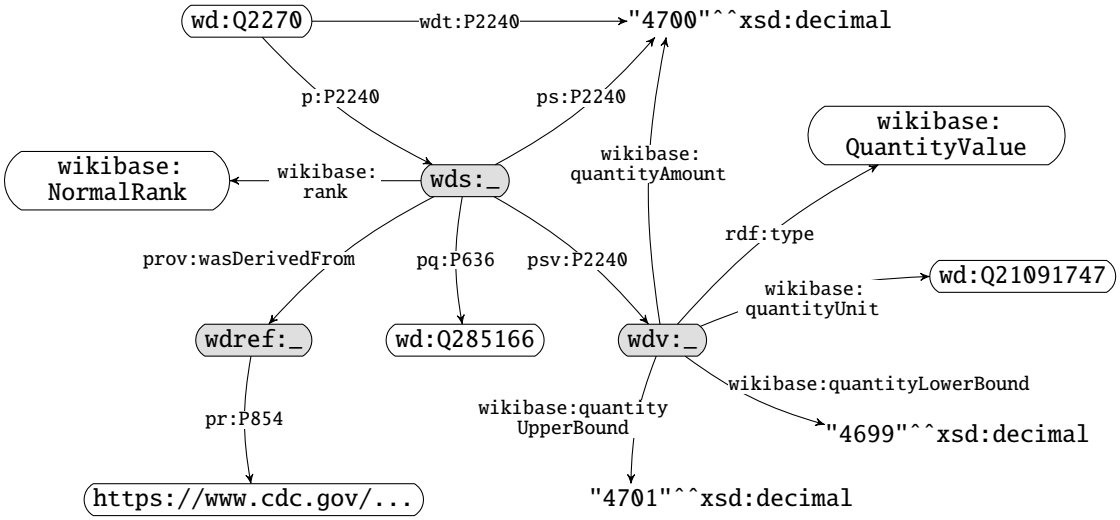


Figure 5.5: RDF representation of the statement "Benzene (Q2270) has an LD50 (P2240) of 4,700 ±1 mg/kg (Q21091747)" considering only the qualifier "route of administration (P636) is oral administration (Q285166)" and the reference record "reference URL (P854) is https://www.cdc.gov/niosh-rtecs/CY155CC0.html".

In Figure 5.5, the shaded nodes are the reified ones. The single underscore (_) indicates that their ids are opaque (hence not shown in the figure). Node `wds:_` represents the statement itself. Predicates `p:P2240` and `ps:P2240` are used to connect the subject "benzene" (`wd:Q2270`) to the statement and the statement to its simple value, i.e., the number 4700 in decimal notation.

The deep value of the statement is represented by node `wdv:_`. It has type `wikibase:QuantityValue` and is connected to the unit mg/kg (`wd:Q21091747`), the lower-

bound 4699, and the upper-bound 4701. The rank of the statement is connected via predicate `wikibase:rank`.

In Figure 5.5, predicate `pq:P636` connects to the statement the qualifier "route of administration" (`P636`) with value "oral administration" (`wd:Q285166`). Similarly, predicate `prov:wasDerivedFrom` connects to the statement the reference record represented by node `wdref:_` whose content (the snak "reference URL" (`P854`) with value "`https://www.cdc.gov/...`") is encoded using predicate `pr:P854` and a (simple) IRI value.

5.2.4. Wikidata Access Interfaces

We now move to the interfaces for accessing the Wikidata knowledge graph. We cover the traditional interfaces first, including the official SPARQL query service, REST APIs, and data dumps. Later, we cover more exotic interfaces, such as the triple pattern fragments (TPF) interface and the one provided by KIF [Lima et al. 2024].

5.2.4.1. SPARQL

So far we have been working with fragments of RDF graphs consisting of some small number of nodes and edges. When working with such small graphs one hardly needs to worry about efficiency. In practice, however, RDF graphs are rarely small. It is not uncommon to find applications that deal with graphs containing millions or even billions of nodes and edges. At the time of writing, for example, the Wikidata knowledge graph contains close to 9 billion triples. At this scale simple tools will not do. One needs specialized RDF databases, called *triplestores*, designed to efficiently store and process large amounts of RDF data. Currently, the Wikidata relies on Blazegraph[11], a high-performance RDF triplestore. But there are plans to migrate the infrastructure to other RDF solutions, as the open-source version of Blazegraph is no longer maintained.

SPARQL (SPARQL Protocol and RDF Query Language) [Harris and Seaborne 2013] is to triplestores what SQL is to relational databases. It is a declarative interface for querying and manipulating the contents of the graph stored in the database. There are two basic types of SPARQL queries: graph pattern matching and graph navigation. We discuss each of these two types of queries next, using the RDF graph of Figure 5.6 as an example target graph. This graph describes the film *Taxi Driver*.

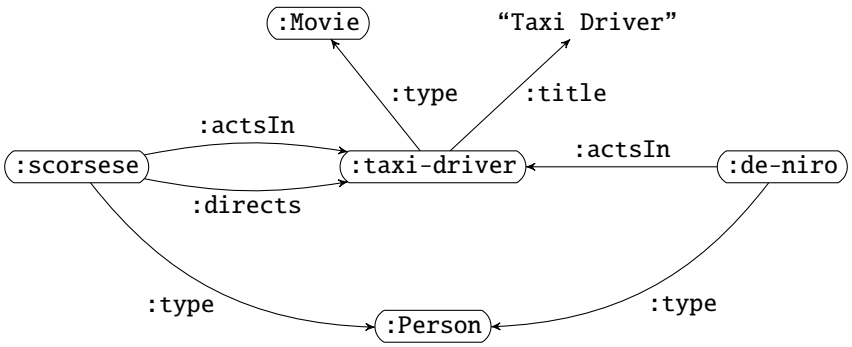


Figure 5.6: Example RDF graph describing the movie *Taxi Driver*.

[11] https://blazegraph.com/

Graph pattern matching A graph pattern matching query is a query that searches for a given (bounded) structural pattern in the graph. In SPARQL, these structural patterns are expressed as *triple patterns*, which are RDF triples in which the subject, predicate, or object may be a variable (name starting with "?"). A *basic graph pattern* (BGP) is the combination of one or more triple patterns.

Consider the RDF graph depicted in Figure 5.6, which describes the film *Taxi Driver*. Suppose we want to list all pairs of distinct persons who acted in this film. We can specify this query in SPARQL as follows:

```
1 PREFIX: <http://example.org/#>
2 SELECT ?x1 ?x2
3 WHERE {
4   ?x1 :actsIn ?x3 . ?x1 :type :Person .
5   ?x2 :actsIn ?x3 . ?x2 :type :Person .
6   ?x3 :title "Taxi Driver" . ?x3 :type :Movie .
7   FILTER (?x1 != ?x2)
8 }
```

This query expresses a BGP consisting of six triple patterns. It select all pairs of labels x_1 and x_2 (line 2) with $x_1 \neq x_2$ (line 7) such that, for some label x_3, each of the following triples occur in the graph:

$\langle x_1, \texttt{:actsIn}, x_3 \rangle$	and	$\langle x_1, \texttt{:type}, \texttt{:Person} \rangle$	(line 4)
$\langle x_2, \texttt{:actsIn}, x_3 \rangle$	and	$\langle x_2, \texttt{:type}, \texttt{:Person} \rangle$	(line 5)
$\langle x_3, \texttt{:title}, \texttt{"Taxi Driver"} \rangle$	and	$\langle x_3, \texttt{:type}, \texttt{:Movie} \rangle$	(line 6)

If we apply this query to the graph of Figure 5.6 we get the following result:

```
1 ?x1         ?x2
2 ---------------------
3 :scorsese   :de-niro
4 :de-niro    :scorsese
```

That is, our *answer set* consists exactly of two pairs of labels: one where x_1 is `:scorsese` and x_2 is `:de-niro` (line 3 in the result), and another where x_1 is `:de-niro` and x_2 is `:scorsese` (line 4 in the result).

Figure 5.7 shows the results of running a similar query over Wikidata. The figure depicts an screenshot of the Wikidata SPARQL Query Service[12]. The query in this case references the items Taxi Driver (`Q47221`), person (`Q5`), and the properties cast member (`PP161`) and human (`Q5`). The truthy namespace `wdt:` is used for matching over the truthy graph, and the `SERVICE` clause is used to obtain the labels of the retrieved entities via Wikidata's custom label service.

Graph navigation The other common kind of query supported by SPARQL are graph navigation queries. These are queries that navigate the topology of the graph through paths

[12] https://query.wikidata.org/

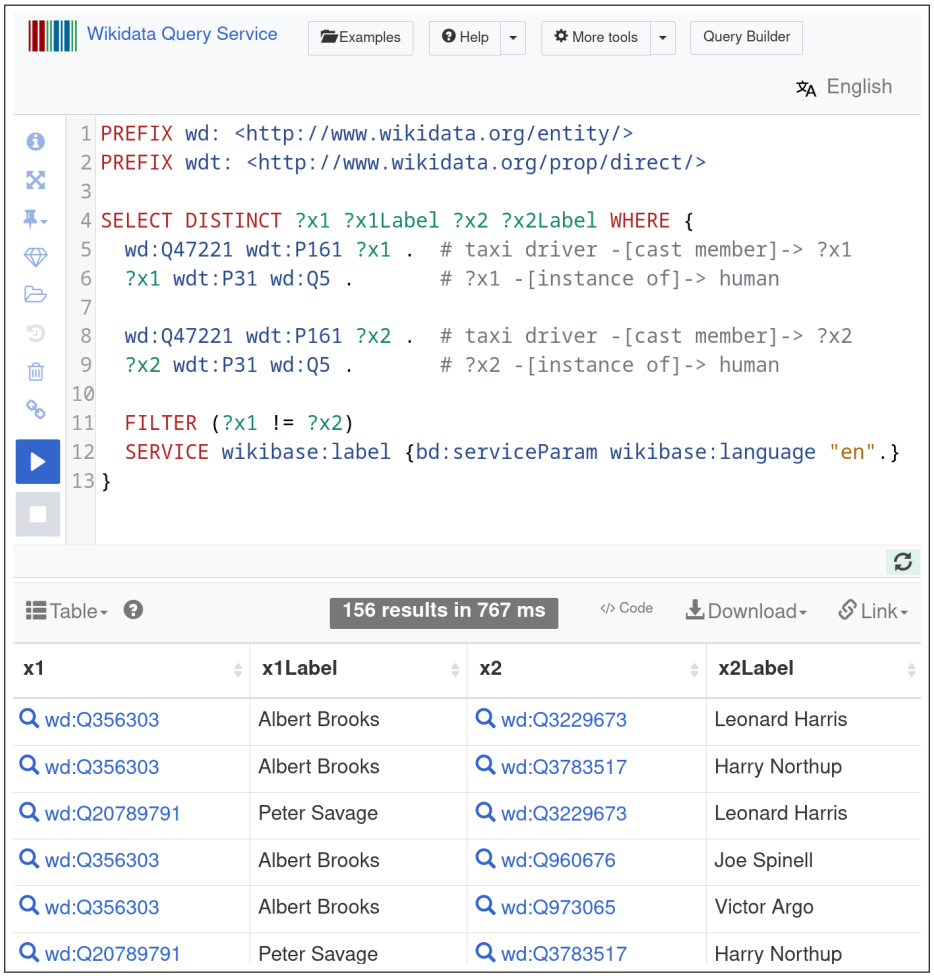


Figure 5.7: Screenshot of the Wikidata SPARQL Query Service showing the results for "All pairs of distinct persons who acted in *Taxi Driver*".

of potentially arbitrary lengths. A *path query* is a basic navigation query of the form $x \xrightarrow{\alpha} y$ that retrieves all pairs $\langle x, y \rangle$ such that there is a path in the graph from x to y which satisfies a given condition α.

For instance, the path queries

$$x \xrightarrow{\texttt{:directs}} y \quad \text{and} \quad x \xrightarrow{\texttt{:actsIn} \circ \texttt{:type}} y$$

search, respectively, for all nodes x and y such that there is an edge labeled `:directs` from x to y, and for all nodes x and y such that there is path of size two from x to y where the first edge in the path is labeled `:actsIn` and the second edge is labeled `:type`. (Recall that a *path* in a graph is a sequence of edges joining a sequence of nodes.).

We can specify the latter query in SPARQL as follows:

```
SELECT ?x ?y
WHERE {
  ?x (:actsIn/:type) ?y .
}
```

Here the symbol "/" stands for chaining and corresponds to the "∘" we used in the abstract notation. If we apply this SPARQL query to the graph of Figure 5.6 we get answer:

```
?x          ?y
------------------
:scorsese   :Movie
:de-niro    :Movie
```

For a more interesting example, consider the path query

$$x \xrightarrow{(:\texttt{actsIn} \circ :\texttt{actsIn}^{-})^{+}} y.$$

This query searches for all actors who have a finite collaboration distance[13], that is, actors x and y such that x and y are co-stars in the same movie (distance 0), or x and y are co-stars of the same z in some movie (distance 1), or x is co-star of some z_1 in some movie and y is co-star of some z_2 in some movie and z_1 and z_2 are co-stars in the same movie (distance 2), and so on. The following SPARQL code implements this query:

```
SELECT ?x ?y
WHERE {
  ?x (:actsIn/^:actsIn)+ ?y .
}
```

Here the "^" denotes the inverse of an edge label (it corresponds to operator "−" used in the abstract notation) and "+" stands for one or more repetitions of the pattern immediately preceding it.

Suppose we add the following triples to the graph of Figure 5.6:

$$\langle :\texttt{de-niro}, :\texttt{actsIn}, :\texttt{cassino}\rangle \quad \text{and} \quad \langle :\texttt{sheron-stone}, :\texttt{actsIn}, :\texttt{cassino}\rangle.$$

If we run the above SPARQL query over this updated graph we get the answer:

```
?x              ?y
-----------------------------
:scorsese       :de-niro
:de-niro        :scorsese
:de-niro        :sharon-stone
:sharon-stone   :de-niro
:scorsese       :sharon-stone
:sharon-stone   :scorsese
```

Figure 5.8 shows the results of running a query similar to the last one over Wikidata. This time we use the QLever SPARQL query interface[14] which, although not being an official Wikidata endpoint, is significantly faster for this type of query.

SPARQL has many more features, such as groups, options, alternatives, filters, etc., which can be combined with basic graph patterns and navigation operators in order

[13]When x is the actor Kevin Bacon this distance is known as the *Bacon distance* or *Bacon degree*. See: https://en.wikipedia.org/wiki/Six_Degrees_of_Kevin_Bacon

[14]https://qlever.dev/wikidata/

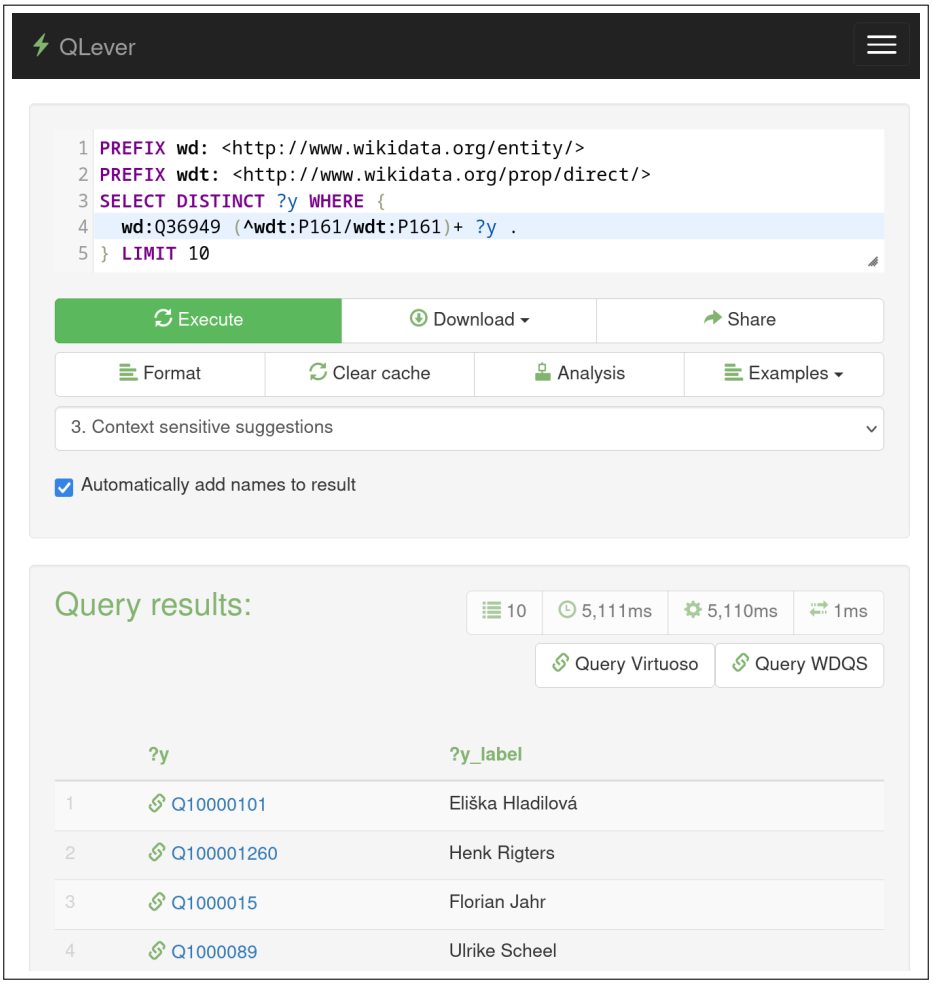


Figure 5.8: Screenshot of QLever SPARQL query interface showing the results for "All actors who have a finite collaboration distance with Robert De Niro".

to build complex queries. Our goal here was to provide a glimpse of SPARQL queries. For a detailed introduction to SPARQL, see [Hitzler et al. 2010]. Besides SPARQL, there are many other languages for querying graphs. See [Angles 2012] for a general discussion of these. For a practical introduction to graph databases and triple stores, we recommend [Robinson et al. 2013].

5.2.4.2. REST APIs

As an alternative to SPARQL, Wikidata offers REST APIs for searching and fetching information about entities and properties. Three different REST APIs are available [Wikidata 2026a]. The first one is MediaWiki Action API[15], which is intended for obtaining general information about Wikidata entities in JSON format. Currently, requests to the MediaWiki Action API can carry up to 50 entities per request.

Here is an example call to the MediaWiki Action API:

[15]https://wikidata.org/w/api.php

https://www.wikidata.org/w/api.php?action=wbgetentities&ids=Q47221&format=json

This request fetches all data necessary to build the page of entity Taxi Driver (`Q47221`), namely, its labels, descriptions, and aliases together with all statements (claims) whose subject is the entity `Q47221`. Here is part of the response to this request:


```
{
    "entities": {
        "Q47221": {
            "modified": "2026-09-12T19:52:54Z",
            "type": "item",
            "id": "Q47221",
            "labels": {
                "en": {
                    "language": "en",
                    "value": "Taxi Driver"
                },
                ...
            },
            "descriptions": {
                "en": {
                    "language": "en",
                    "value": "1976 film directed by Martin Scorsese"
                },
                ...
            },
            "aliases": {
                ...
            },
            "claims": {
                "P57": [
                    {
                        "mainsnak": {
                            "snaktype": "value",
                            "property": "P57",
                            "datavalue": {
                                "value": {
                                    "entity-type": "item",
                                    "id": "Q41148"
                                },
                            },
                            "datatype": "wikibase-item"
                        },
                        "type": "statement",
                        "id": "q47221$EB940392-6851-48AB-A348-4C16A35471E8",
                        "rank": "normal",
                        "references": [
                            {
                                "snaks": {
                                    "P248": [
                                        {
                                            "snaktype": "value",
                                            "property": "P248",
                                            "datavalue": {
```

```
                                        "value": {
                                            "entity-type": "item",
                                            "id": "Q2638147"
                                        },
                                    },
                                    "datatype": "wikibase-item"
                                }
                            ],
                            ...
                        },
                    },
                    ...
                ]
            }
        ],
        ...
    },
    "sitelinks": {
        ...
    }
}
},
"success": 1
}
```


The second REST API provided by Wikidata is the Wikibase REST API[16]. This is an OpenAPI interface for interacting with items and statements in Wikibase instances, including Wikidata. This Wikibase REST API is still under development and it is intended to eventually replace the MediaWiki Action API.

The third REST API provided by Wikidata is not exactly a REST API but a GraphQL service[17] which can be reached through a MediaWiki Action API call (`action=wbgraphql`). The GraphQL interface aims to reduce the load on the SPARQL query service by serving some common use cases, such as fetching the labels of linked entities, reading items in batch, finding items by property or value, etc.

5.2.4.3. Data Dumps

When live querying Wikidata is not needed, for instance, when one is only interested in processing large quantities of data offline, the best approach is to use a data dump [Wikidata 2026a]. Wikidata publishes complete copies of its content in different formats, with the JSON and RDF dumps being regarded the stable interfaces [Wikidata 2026b].

JSON is currently the recommended format for Wikidata dumps. The JSON dumps are created weekly and contain all Wikidata entities unsorted in a single JSON array. One detail of the JSON dump format that can help speed up processing is that each entity occurs on a separate line in the JSON file. This way files can be read line by line with each line decoded separately as an individual JSON object.

Another popular format is RDF. The canonical Wikidata RDF dumps use the

[16]https://www.wikidata.org/w/rest.php/wikibase/v1
[17]https://www.wikidata.org/wiki/Wikidata:Wikibase_GraphQL

Turtle and N-Triples formats. There are two versions of the RDF dumps. The "all" version corresponds to the full graph, which besides information about entities and their relationships carries the qualifiers, references, ranks and other metadata. In contrast, the "truthy" version corresponds to the truthy graph (see Section 5.2.3) which contains only information about the entities and their relationships. The choice between "all" and "truthy" depends on the application, or on how much detail one is willing to lose in exchange for a significant reduction in the size of the graph.

There are also incremental dumps, or dumps of additions and changes, which contain what was added to the graph in the last 24 hours. Applications that need to be kept in sync with the latest version of the graph should use these.

5.2.4.4. Triple Pattern Fragments

We now move to more experimental access interfaces, starting with TPF. TPF (Triple Pattern Fragments) [Verborgh et al. 2016] is a lightweight interface to RDF graphs. They are part of the Linked Data Fragments (LDF) initiative [Verborgh et al. 2014] and were proposed as an intermediate alternative to RDF data dumps and SPARQL endpoints. LDF itself is a framework for the conceptual analysis of Linked Data interfaces. According to LDF, any such interface publishes only parts, or *(linked data) fragments*, of a given knowledge graph. These fragments are considered the "units of response" of the interface and are assumed to consist of three things:

1. Data: a subset of the triples of the graph;
2. Metadata: triples describing the data; and
3. Controls: links and forms that can be used to retrieve other fragments of the same or other knowledge graphs.

For example, a data dump of a knowledge graph can be described as single linked data fragment where the data is the whole content of the dump, the metadata consists of things like version, author, etc., and the controls are empty. Similarly, the response to a SPARQL CONSTRUCT query can be seen as a fragment where the data are the resulting triples, the metadata is empty, and the controls are any parameters used to paginate the result, such as limits and offsets.

A TPF interface provides access to an RDF graph based on a single triple-pattern. A *triple-pattern* is a pattern of the form *(s, p, o)* where *s*, *p*, and *o* are either fixed values (URIs or literals) or anonymous variables. When given a triple-pattern, the TPF interface responds with a fragment in which (i) the data are triples in the graph that match the pattern; (ii) the metadata are an estimate of the total number such triples; and (iii) the controls are a hypermedia form that allows clients to retrieve other fragments matching the same pattern.

Wikidata and some other popular knowledge graphs, such as DBpedia [Lehmann et al. 2015], provide TPF interfaces. Other public TPF interfaces can be found here[18].

We can query the Wikidata TPF interface by submitting a GET HTTP request to an URL such as this one:

[18]https://linkeddatafragments.org/data/

```
https://query.wikidata.org/bigdata/ldf?predicate=wdt:P31\&object=wd:Q5
```

This request asks for every triple in Wikidata whose predicate component is `wdt:P31` (instance of) and object component is `wd:Q5` (human). That is, it selects the triples matching the pattern

$$(s, \texttt{wdt:P31}, \texttt{wd:Q5})$$

for any value of s. In other words, the triples such that subject s is an instance of human. The response to this specific request is a single linked data fragment containing usually 100 triples matching the pattern. For instance, if you open the above URI in a Web browser, an HTML page with the matched triples will be shown. At the bottom of the page, you will see links (controls) to other pages (fragments) which contain the rest of the result. The actual format of the response depends on the value of the "Accept" header provided to the server in the HTTP request. This can be set to any of the popular RDF serialization formats, such as Turtle and JSON-LD.

Some TPF interfaces also support quad-patterns of the form (s, p, o, g) which take an additional parameter g specifying a named graph of an RDF dataset. If not set, g is assumed to be the default graph. Note that the Wikidata RDF encoding does not use named graphs.

The advantage of TPF over other Linked Data interfaces is that it offers some query capability while being extremely lightweight. For some applications, it is more convenient than a data dump and provides a more reasonable Web API than SPARQL in the sense that it exposes only a limited interface to the underlying database. A further advantage has to do with caching. Because of their restricted syntax TPF requests are more cache-friendly than SPARQL requests. For instance, just by looking at the requested URI, it is straightforward for an HTTP proxy to determine whether two TPF requests are identical. The same cannot be said about SPARQL where there are different ways to write essentially the same query.

5.2.4.5. Knowledge Integration Framework

KIF (Knowledge Integration Framework) [Lima et al. 2024] is an open-source framework for knowledge integration based on Wikidata. It adopts the data model and vocabulary of Wikidata as a lingua franca for integrating heterogeneous knowledge sources. The integration done by KIF is virtual in the sense that the data model and vocabulary translations happen at query time, guided by user-provided mappings.

KIF is in essence a Python library (kif-lib)[19] for evaluating queries over abstract engines and obtaining objects of the Wikidata data model as a result. KIF engines are implemented following a plugin architecture. There are various predefined (built-in) engine plugins which can be extended by applications and loaded dynamically into the library. The two basic types of KIF engines are stores and searchers.

Stores A KIF *store* is an interface to a knowledge source, typically but not necessarily a knowledge graph. The most common type of KIF store is the SPARQL store, which reads Wikidata-like statements from a given SPARQL endpoint. To create a SPARQL store, one needs specify the address of the target SPARQL endpoint and the SPARQL

[19]https://pypi.org/project/kif-lib/

mappings to be used to generate a concrete SPARQL query (i.e., one adhering to the RDF encoding of the target graph). Besides mappings for Wikidata itself, KIF comes with SPARQL mappings for DBpedia [Lehmann et al. 2015], PubChem (RDF) [Kim et al. 2023], UniProt [Consortium 2025], among others. New mappings can be added programmatically.

We can use the KIF command-line interface (CLI), which is distributed together with the library, to run a simple query over a KIF store. For instance:

```
1 $ kif filter --subject='wd.Q(7251)' --property=wd.doctoral_advisor
2 (Statement
3   (Item Alan Turing)
4   (ValueSnak (Property doctoral advisor)
5              (Item Alonzo Church)))
```

The command on line 1 runs a KIF *filter* (a statement match query) using the default store plugin, `wikidata`, which targets the public Wikidata SPARQL Query Service (WDQS) and uses KIF's Wikidata SPARQL mappings. This particular filter instructs KIF to search for statements whose subject is `wd:Q7251` (Alan Turing) and property is `wd:P184` (doctoral advisor). To evaluate a filter, the SPARQL store uses the provided mappings to compile it into a SPARQL query and then runs the query over the target endpoint. The resulting bindings are used to construct Wikidata-like statements. The returned statement in this case is shown on lines 2–5 and stands for the assertion "Alan Turing's doctoral advisor is Alonzo Church".

As we have seen, in the Wikidata data model, which is adopted by KIF, every statement consists of a subject (which can be an item or property) and a *snak*. The subject of the above statement is the item Alan Turing and its snak is a *value snak* (i.e., property-value pair) with property equal to doctoral advisor and value equal to the item Alonzo Church.

KIF distinguishes between unannotated statements, such as the one above, and annotated statements. The latter, besides a subject and a snak, also carry a set of qualifiers, a set of reference records, and a rank. For instance, the filter below matches *annotated statements* (due to the `--annotated` flag) with property `wd:P2054` (density) and value equal to 0.88 grams per cubic centimeter:

```
1  $ kif filter --property=wd.density \
2               --value='Quantity("0.88", wd.Q(13147228))' --annotated
3  (AnnotatedStatement
4    (Item benzene)
5    (ValueSnak (Property density) 0.88 ±0.01 g/cm³)
6    (QualifierRecord
7      (ValueSnak (Property temperature) 20 ±1 °C)
8      (ValueSnak (Property phase of matter) (Item liquid)))
9    (ReferenceRecordSet
10     (ReferenceRecord
11       (ValueSnak (Property HSDB ID) "35#section=TSCA-Test-Submissions")
12       (ValueSnak (Property stated in) (Item Hazardous Substances Data Bank))))
13   NormalRank)
```

The annotated statement shown on lines 2–13 asserts that "benzene's density is 0.88±0.01 g/cm³". Its qualifier record (lines 6–8) qualifies this assertion, i.e., says in addition that

this is the case when “the temperature is 20±1 °C” and “the phase of matter is liquid”. Its reference record set (lines 9–12) contains a single reference record (lines 7–9) which indicates the provenance of the statement, namely, the entry with the given HSDB ID in the Hazardous Substances Data Bank. Finally, the value NormalRank for its rank (line 13) is the default one and means “neutral” (neither preferred nor deprecated).

More complex filters So far, we have used simple values to match the components of statements in filters. In KIF, we can also specify the desired subject, property, or value indirectly, by giving a constraint that it must satisfy. For instance:

```
$ kif filter \
    --subject='(wd.shares_border_with(wd.Argentina)|
                wd.shares_border_with(wd.Paraguay))&
                wd.located_in_or_next_to_body_of_water(wd.Atlantic_Ocean)' \
    --property='wd.official_language'
(Statement
  (Item Brazil)
  (ValueSnak (Property official language)
             (Item Portuguese)))
(Statement
  (Item Uruguay)
  (ValueSnak (Property official language)
             (Item Spanish)))
```

This filter matches statements such that (i) the subject “shares border with Argentina *or* Paraguay” *and* “is located in or next to the Atlantic Ocean”; and (ii) the property is `wd:P37` (official language). The only matching subjects in this case are `wd:Q155` (Brazil) and `wd:Q77` (Uruguay) and their official languages, as indicated by the statements above, are `wd:Q5146` (Portuguese) and `wd:Q1321` (Spanish).

Besides single-hop checks, KIF supports constraints involving n-hop property paths. These can be combined via conjunction (&) and disjunction (|) operators, as shown above. Simple datatype checks and projection on the statement components (subject, property, value) are also supported.

Targeting other graphs To query a graph other than Wikidata, all we need to do is change the store plugin. For instance:

```
$ kif filter --store=dbpedia --subject='db.r("Alan_Turing")'
             --value='(db.almaMater/db.city)(db.r("Princeton,_New_Jersey"))'
(Statement
    (Item dbr:Alan_Turing)
    (ValueSnak (Property dbo:doctoralAdvisor)
               (Item dbr:Alonzo_Church)))
(Statement
    (Item dbr:Alan_Turing)
    (ValueSnak (Property doctoral advisor)
               (Item dbr:Alonzo_Church)))
```

The `--store=dbpedia` argument instructs KIF to run the filter using the `dbpedia` store plugin, which targets the public DBpedia SPARQL endpoint and uses KIF’s DBpedia SPARQL mappings. The filter in this case matches statements whose subject is

`dbr:Alan_Turing` and value is "someone whose `dbo:almaMater` is located in the `dbo:city` of `dbr:Princeton,_New_Jersey`".

Despite querying DBpedia, as expected, we get as a result two Wikidata-like statements (lines 3–10) asserting that "Alan Turing's doctoral advisor is Alonzo Church". We say these statements are "Wikidata-like" because although following Wikidata syntax, their components mostly have DBpedia URLs. This happens because KIF's DBpedia SPARQL mappings do not attempt to convert entities in the DBpedia namespace into that of Wikidata. The exception are properties. If there exists a Wikidata property which is said to be the same as (`owl:sameAs`) a given DBpedia property in the DBpedia graph, then the generated SPARQL query will match it. This is why on line 9 we get a statement whose property is `wd:P184` (doctoral advisor) even though we are querying DBpedia.

Searchers Besides stores, the other kind of engine in KIF are searchers. A KIF *searcher* is an interface to a similarity search method within a namespace. For example, given a string, the `wikidata-wapi` search plugin looks in the Wikidata namespace for entities with a similar label or alias:

```
1 $ kif search --search=wikidata-wapi 'pizza' --item
2 (Item pizza)
3 (Item Mariagrazia Pizza)
4 (Item Pizza Hut)
```

The `--search=wikidata-wapi` argument specifies the search plugin to be used,"pizza" is the search string, and `--item` indicates that we want to search for items. The top three items found by the search are shown on lines 2–4.

In addition to Wikidata, KIF comes with built-in plugins to search for entities in the namespaces of DBpedia and PubChem.

5.3. Wikidata in the Era of Generative AI Models

The rapid development of LLMs has changed the way users interact with information systems. Instead of formulating queries in a formal language or navigating predefined interfaces, users can express their information needs in natural language and expect AI systems to interpret their requests, retrieve relevant information, and generate natural language answers. This shift in interaction mode has renewed interest in KGs. Their structured and explicitly represented knowledge can complement the information encoded in LLMs, providing external sources of factual knowledge that can be retrieved and inspected. At the same time, the ability of LLMs to interpret and generate natural language creates new opportunities for enriching, curating, verifying, and maintaining KGs.

Wikidata is particularly well positioned in this ecosystem because it combines broad domain coverage with an openly accessible, collaboratively maintained, and continuously evolving representation of structured knowledge. Moreover, Wikidata can be accessed through multiple interfaces and representations, supporting different ways of querying and consuming its knowledge. This flexibility is particularly relevant for AI applications, where the requirements for accessing a KG may vary across systems and use cases.

Two complementary perspectives are useful for understanding the relationship

between Wikidata and AI. In *Wikidata for AI*, the Wikidata KG is used to contextualize and ground AI systems. In *AI for Wikidata*, AI models assist with knowledge completion, verification, and curation. Together, these perspectives can form a feedback loop in which Wikidata supports AI systems which in turn help improve Wikidata.

5.3.1. Wikidata for AI: Why KGs Matter for Generative AI?

LLMs are trained on vast quantities of unlabeled data, rich in world knowledge [Youssef et al. 2023]. However, this knowledge is embedded in the model's parameters, making it difficult to access it through deterministic queries [Li et al. 2025b]. More importantly, a model may generate statements that are linguistically convincing but factually incorrect. This phenomenon, commonly referred to as hallucination, illustrates a fundamental limitation of relying solely on the model's internal knowledge.

RAG Retrieval-Augmented Generation (RAG) addresses part of the aforementioned limitations by complementing the model's parametric knowledge with information obtained from an external source [Lewis et al. 2020]. Rather than requiring the model to answer entirely from what was learned during training, a retrieval component first identifies potentially relevant evidence and provides it as context for generation. The model then use this retrieved information as context when formulating its response.

KGs serve as an interesting source of information for this purpose. While a conventional document-oriented RAG system retrieves passages of text, a KG exposes knowledge as entities connected by explicit relationships. This difference becomes important when answering questions that require reasoning. For instance, consider the following question:

> *Which researchers work on the Semantic Web and are affiliated with universities?*

A language model may be able to produce a plausible list of researchers from its internal knowledge, but there is no guarantee that the listed people are correctly identified, the relationships hold, and the resulting set is complete. However, answering this reliably is not simply a matter of retrieving relevant passages from documents. It requires identifying people who are researchers working on the Semantic Web, determining their affiliations, and connecting these facts. A KG makes these connections explicit and allows them to be traversed and evaluated through a formal query (see Figure 5.9).

Wikidata is particularly useful in this setting not because it is complete or error-free, but because it makes knowledge available in a structured and inspectable form. As discussed in Section 5.2, its data model represents knowledge through items, properties, and statements, which can be further enriched with qualifiers and references. Qualifiers provide additional context about a statement, while references indicate the sources supporting it. For example, when answering the question above, a statement about a researcher's affiliation can include qualifiers indicating the period during which the affiliation holds. This enables an AI system not only to retrieve the relevant information, but also to inspect the context and provenance associated with it. Moreover, because Wikidata is continuously maintained by a large community of contributors, changes to the underlying knowledge graph are reflected in subsequent queries, allowing AI systems to generate answers that

evolve as the knowledge itself is updated.

KGQA What we described above is the goal of a classic task in natural language processing, called Knowledge Graph Question Answering (KGQA). A KGQA system takes natural language questions as input and attempts to answer them by retrieving and reasoning over a knowledge graph, such as Wikidata. KGQA pipelines are usually employed in the implementation of chatbots, either as standalone applications or as components of domain-specific question-answering systems. One advantage of KGQA over other forms of QA is that the existence of an external representation of knowledge makes it possible to separate the process of interpreting natural language from the processes of obtaining relevant information. This separation is useful in cases where the output of the system is not necessarily a natural language answer.

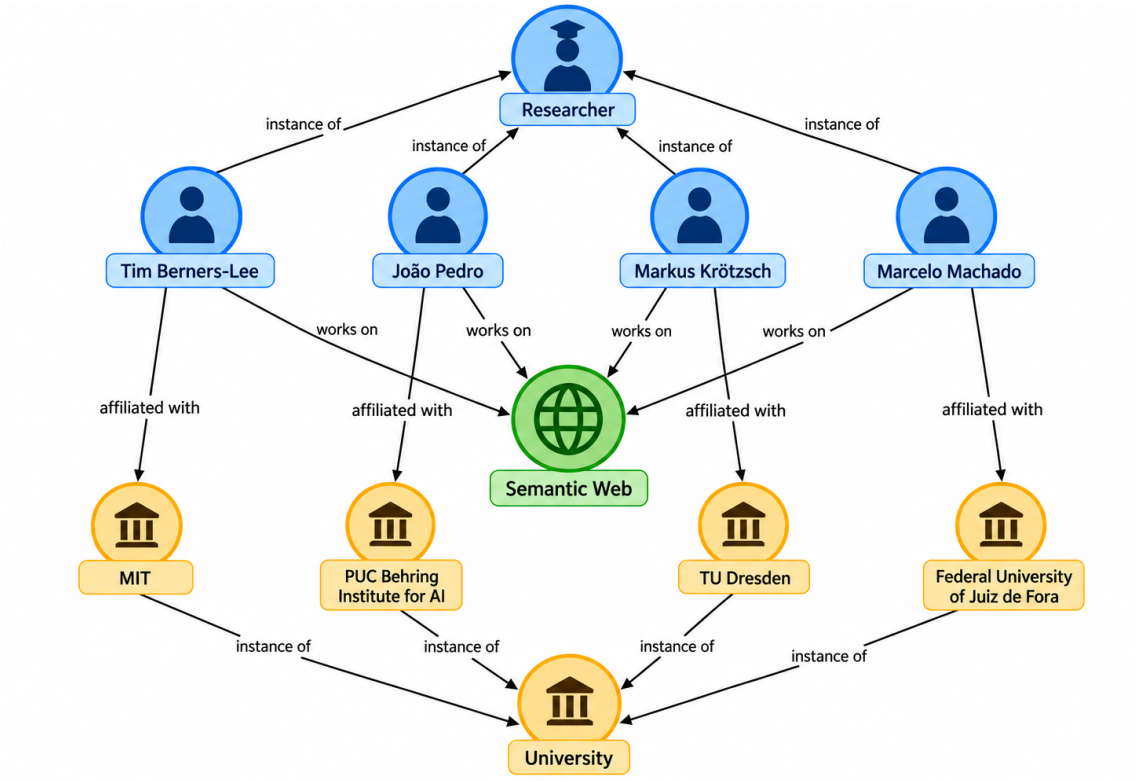


Figure 5.9: Knowledge graph representing researchers, their affiliations, and their relationship with the Semantic Web.

KIF-QA [Machado et al. 2025, Campos et al. 2025], for example, is a system that answers simple one-hop questions by retrieving statements from Wikidata (and other KGs). Rather than relying on the language model to provide the factual answer itself, the system uses it to interpret the natural language question and construct a structured query, while the KG provides the factual knowledge required to answer it. This illustrates a broader class of applications in which LLMs serve as an interface to structured knowledge, with the KG remaining responsible for providing the underlying facts. The pipeline of KIF-QA consists of five main steps (see Figure 5.10).

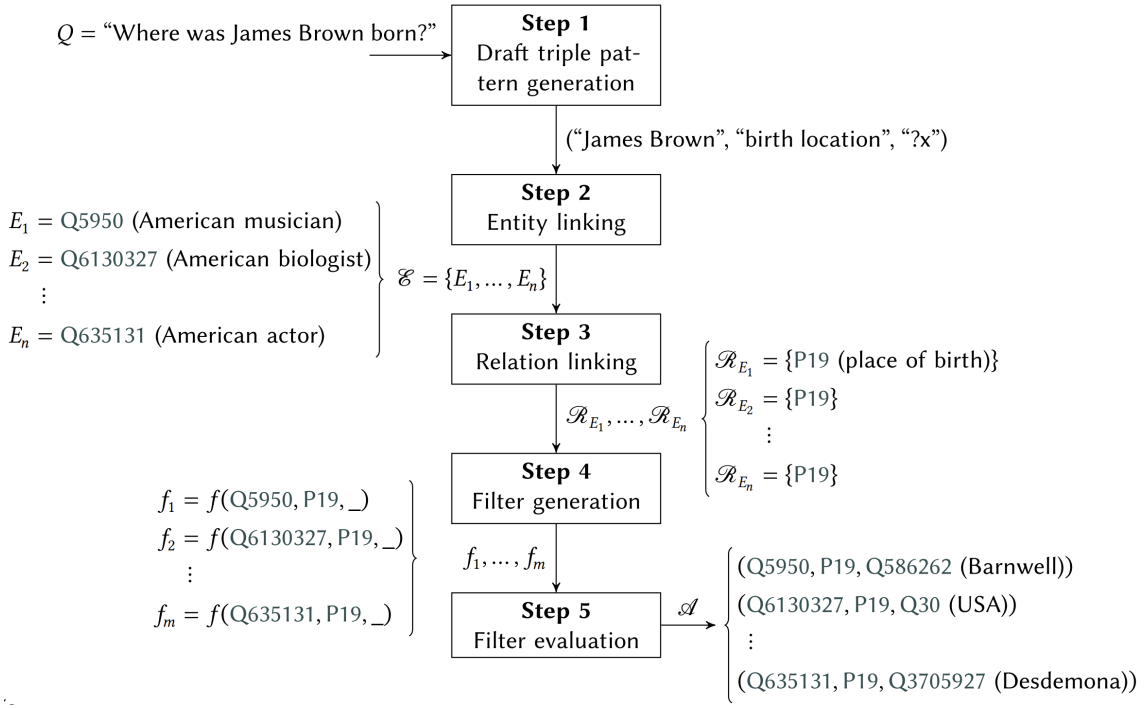


Figure 5.10: A typical KQGA pipeline for transforming natural language questions into structured queries over a knowledge graph [Machado et al. 2025].

We can summarize each of the steps of the pipeline of Figure 5.10 as follows:

- *Step 1: Draft triple pattern generation*. Given a simple question Q in natural language, KIF-QA prompts the LLM to generate a draft triple pattern (s, p, o) using text labels for the known components (s-p or p-o) and the variable "?x" for the unknown component (either s or o). When available, examples of "(question, draft triple pattern)" pairs are also provided in the prompt.

- *Step 2: Entity linking*. Given a draft triple pattern (s, p, o), KIF-QA performs entity linking on its *topic entity label*, i.e., the known s or o. The system searches for entities with a similar label or alias in the namespace of the target graph. The result is a set of candidate entities accompanied by their descriptions. The LLM then *disambiguates* the topic entity label against the candidate entities, i.e., it selects among the candidates those entities whose label and description best match the topic entity label and the text of the question. The result of this step is a refined set $\mathcal{E}$ of *candidate topic entities*.

- *Step 3: Relation linking*. To resolve the predicate component p of the draft triple pattern (s, p, o), KIF-QA proceeds as follows. For each candidate topic entity E in $\mathcal{E}$, the system obtains candidate relations R_i and R_j such that, for some x, $R_i(E, x)$ or $R_j(x, E)$ holds in the graph. The LLM then *disambiguates* the predicate label p against the label and description of each candidate relation R_k, the label and description of E, and the text of question Q. The result of this step is, for each topic entity E, a refined set $\mathcal{R}_E$ of *candidate relations for* E. This set is constructed such

that, for each R_k in $\mathcal{R}_E$, it is possible to determine whether E occurs as the source or target of R_k in the graph.

- *Steps 4 and 5: Filter generation and evaluation.* For each candidate topic entity E in $\mathcal{E}$ and candidate relation R_k in $\mathcal{R}_E$, KIF-QA generates a SPARQL query with a single triple pattern (filter) $f(E, R_k, \_)$ if E occurs as the source of R_k, or a filter $f(\_, R_k, E)$ if E occurs as the target. The system then executes all filters generated for the candidate entities in parallel. The resulting statement set $\mathcal{A}$ constitutes the *answer set* associated with the original natural language question Q.

Fact checking A similar pipeline can be employed in fact checking applications. For example, Wikidata has been used to verify facts generated by LLMs in a geospatial question-answering pipeline [Norouzi and Hitzler 2025], distinguishing generated facts for which supporting information can be retrieved from Wikidata from those for which no corresponding information is found. More generally, rather than determining whether a claim is true or false in general, such a system can check whether the KG contains a statement supporting the claim. The LLM interprets the natural language claim and, together with the entity linking process described above, translates it into a formal SPARQL query. In particular, an ASK query can be used to determine whether a matching statement exists in the KG, returning a Boolean value indicating whether the queried fact is represented. For example, given the claim "Tim Berners-Lee is an employee of the World Wide Web Consortium", the fact-checking pipeline could generate the following SPARQL query targeting Wikidata:

```
ASK {
  # ("Tim Berners-Lee", "employer", "World Wide Web Consortium")
  wd:Q80 wdt:P108 wd:Q37033 .
}
```

This particular query evaluates to true, indicating that Wikidata contains a statement supporting the fact.

In addition to query-level checks, verification can also be performed against Wikidata's ontology constraints. Chepurova et al. [Chepurova et al. 2024] propose an LLM-based knowledge extraction pipeline that uses Wikidata's property constraints to verify extracted triples. For example, consider the triple:

$$\langle \text{Tahiti Honey, screenwriter, Frederick Kohner} \rangle.$$

After linking its entities to Wikidata, the approach checks whether the types of the subject and object are compatible with the constraints defined for the screenwriter (P58) property. Such constraints specify which types of entities are expected in each position of a triple. For example, a property such as screenwriter (P58) could require its subject to be an entity representing a film or audiovisual work and its object to be an entity representing a person. The types of the linked items are determined from their instance of (P31) and subclass of (P279) hierarchies, allowing the system to assess whether the types of Tahiti Honey (Q20949975) and Frederick Kohner (Q2656718) satisfy the corresponding subject and object constraints. In this case, Wikidata is used not only as a source of factual knowledge, but also as a semantic schema for validating the knowledge produced by LLMs.

Summary Taken together, the previous use cases illustrate a broader pattern in the integration of Wikidata and language models. Wikidata can serve not only as a source for retrieving factual information, but also as a structured layer for grounding, verifying, constraining, and supporting the generation of knowledge-based responses. At the same time, the different approaches demonstrate that LLMs can interact with such structured knowledge at several layers, from interpreting natural language questions to generating queries and validating extracted or generated information.

5.3.2. AI for Wikidata: How Generative AI can support KGs?

In Section 5.3.1, we discussed how Wikidata can be used as an external source of structured knowledge to support AI systems. In this section, we present a complementary perspective. Here LLMs are no longer used just to interpret a query and retrieve existing knowledge. Instead, they become a collaborative component in the process of producing, enriching, and validating structured knowledge that is fed back to the KG itself.

5.3.2.1. Knowledge Graph Completion

KGs usually operate under a open-world assumption. This means that any missing relationships or facts are treated as unknown rather than false—a semantic choice that makes KGs are inherently incomplete [Ji et al. 2021]. Wikidata is no different: by its nature it is incomplete, even more so because it attempts to model the real world, which is dynamic, vast, and continuously evolving. As knowledge evolves, so does the need for automated approaches to identify and complete missing knowledge in KGs.

Knowledge graph completion generally refers to the task of predicting missing entities or relationships in a KG [Chen et al. 2020]. Early KG completion approaches relied primarily on the structural information encoded in the graph, including embedding- and path-based methods [Li et al. 2025a]. Subsequent approaches incorporated additional sources of information, such as textual descriptions and neural representations. More recently, LLM-based approaches have reformulated KG completion as a language model task, leveraging the semantic and linguistic capabilities of LLMs [Ibrahim et al. 2024].

An example system implementing KG completion pipeline is LLM Store [Machado et al. 2024b, Machado et al. 2024a]. Although not primarily designed for this purpose, LLM Store illustrates how the completion task can be preformed over Wikidata using an LLM as a knowledge source. LLM Store was developed as a KIF plugin (Section 5.2.4.5). It takes a structured query in a SPARQL-like syntax, translates it into a natural-language prompt and sends it to an LLM. The LLM answer is then translated back into structured Wikidata-like statements. Interestingly, LLM Store may be thought as a counterpart to KIF-QA [Machado et al. 2025, Campos et al. 2025]: while KIF-QA starts from a natural-language question and uses an LLM to construct a structured query whose answer is retrieved from the KG, LLM Store starts from a structured query pattern and uses an LLM to generate candidate facts that are translated into Wikidata statements.

For example, consider the query about Ada Lovelace's (`Q7259`) occupation (`P106`):

```
SELECT ?o WHERE {
  wd:Q7259 wdt:P106 ?o .
}
```

LLM Store translates it into a prompt which is then fed into an LLM. In this case, the LLM is asked to predict the missing value (?o) for the occupation of Ada Lovelace. Here is the actual prompt sent to the LLM by LLM Store:

```
# SYSTEM
You are a helpful and honest assistant that resolves a TASK. Please, respond
↪  concisely, with no further explanation, and truthfully.

# USER
TASK: Fill in the gap to complete the relation:
Ada Lovelace occupation ___

The output should be only a list containing the answers, such as `["answer_1",
↪  "answer_2",..., "answer_n"]`.  Do not provide any further explanation and
↪  avoid false answers. Return an empty list, such as `[]`, if no information is
↪  available.
```

The LLM may predict "mathematician" as one of the missing values. However, for this prediction to become structured knowledge, the generated string must be linked to the corresponding Wikidata item. This is precisely the entity linking step used in KIF-QA (see Figure 5.10). In this case, the missing value is resolved to `Q170790` (mathematician) and the resulting statement corresponds to the triple:

$$\langle \texttt{wd:Q7259}, \texttt{wdt:P106}, \texttt{wd:Q170790} \rangle.$$

LLM Store is capable of handling more complex queries involving constraints on the subject or the object. For example, consider the query below which asks for the entity representing the "official sign language of Brazil":

```
SELECT ?o WHERE {
  # ("Brazil", "official language", ?o)
  wd:Q155 wdt:P37 ?o .
  # (?o "instance of",  sign language)
  ?o wdt:P31 wd:Q34228
}
```

This query gives rise to the following user prompt:

```
...
# USER
TASK: Fill in the gap to complete the relation:
Brazil official language ___ where ___ is an instance of sign language
...
```

For Wikidata-like KGs, LLM Store leverages the property constraints to improve the process of linking items and constructing the resulting statements. For instance, the property date of birth (`P569`) is used to link an item (usually a person) to a date value (its date of birth). So, if during completion for the value of this property, an LLM generates something other than a date value, LLM Store can detect the inconsistency and trigger a new generation round or report the error. Also, because the expected value is a literal

(date value), when an actual date value is generated, LLM Store knows it can skip the item-linking process entirely.

The KG completion approach we just discussed can be naturally complemented with RAG (see Section 5.3.1) to increase the likelihood of the LLMs generating sound answers. In fact, LLM Store offers a native solution for RAG which uses a plugin mechanism to connect to external sources at query time. Custom plugins can be developed by users, but LLM Store comes with a set of predefined plugins targeting popular sources, such as Wikipedia.

We conclude this section on knowledge graph completion by remarking that one of the main limitations of existing approaches, including LLM Store's, is that they largely focus on completing only the basic components of a statement, namely, its subject, property, or object. More comprehensive approaches could be developed to generate and validate other components, such as qualifiers and references.

5.3.2.2. Knowledge Graph Construction

"Completion" is only one way in which LLMs can contribute to KGs. LLMs can also be used to extract information from unstructured sources in bulk and represent it as a graph, according to the schema (data model and vocabulary) of a target KG. In this setting, the task is not to infer a missing component of an existing statement in a KG, but rather to construct a graph from scratch, by extracting a set of statements from unstructured sources.

The implementation of KG construction pipeline typically involves three steps: (i) extracting the information from the sources, (ii) grounding the extracted entities and relationships in the target KG's schema, and (iii) validating the resulting statements before possibly incorporating them back into the target KG [Zhong et al. 2023]. We detail these steps and how LLMs can assist with each of them next (see Figure 5.11).

Extraction The extraction step is usually performed through open information extraction (oIE) in which the LLM is instructed to return a list of *spo* triples from the input text. Usually no predefined schema used at this point, that is, the admissible relations and entity types are unconstrained. The instruction to the LLM is typically accompanied by examples, so that the model returns triples in a consistent format rather than in free-form prose. For example, consider the instruction below, we will use this as the running example throughout this section.

```
# SYSTEM
Your task is to extract triples of the form `[Subject, Relation, Object]` from a
↪  given piece of text.

# EXAMPLE
[examples go here]

# USER
Extract triples from the following text:

Albert Einstein was born in Ulm on March 14, 1879.
```

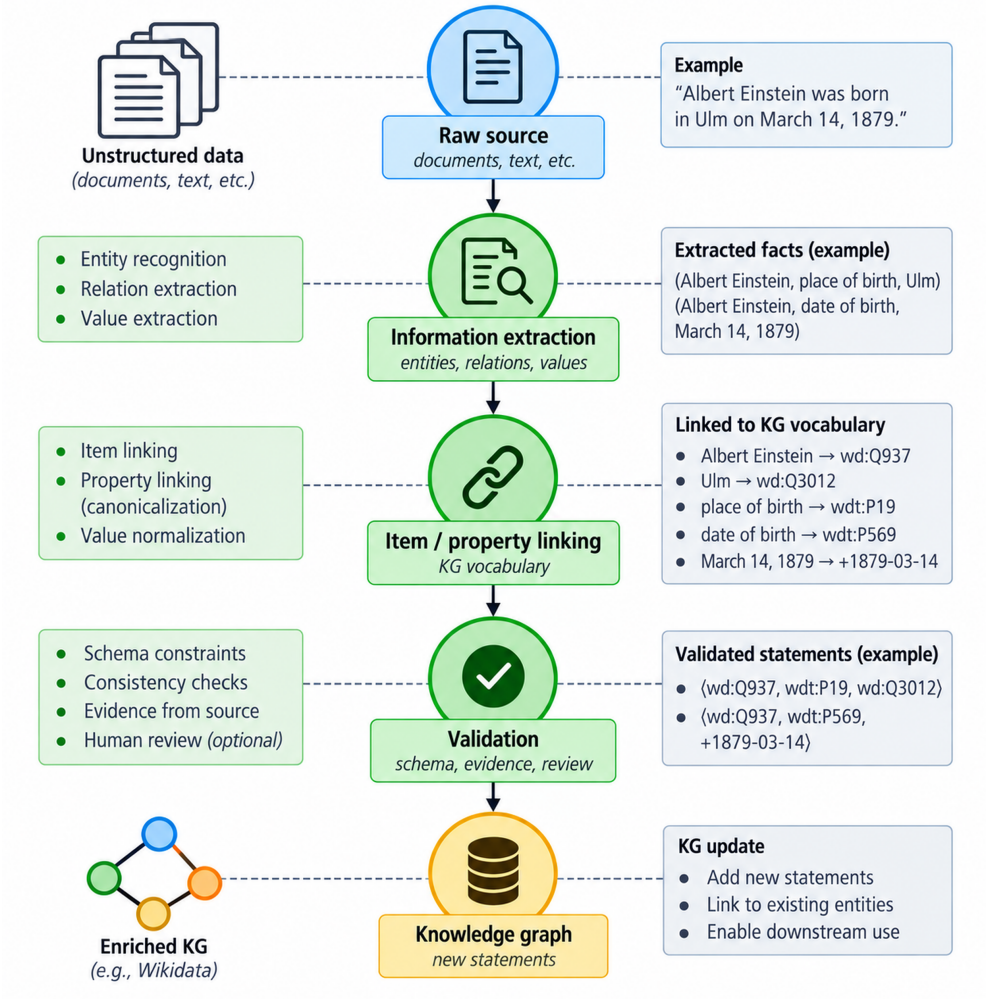


Figure 5.11: LLM-assisted pipeline for constructing KGs from unstructured data.

In this case, the model might respond with the triples:

$$\langle \text{Albert Einstein}, \text{born in}, \text{Ulm} \rangle,$$
$$\langle \text{Albert Einstein}, \text{born on}, \text{March 14, 1879} \rangle.$$

The EDC (Extract-Define-Canonicalize) [Zhang and Soh 2024] is a proposal that instantiates the design we just described. In its first phase, EDC prompts an LLM to extract draft triples freely, independently of any target schema. The advantage of schema-free extraction is that it allows for both matching the results against a preexisting target schema or creating a new schema on-the-fly. The disadvantage is that the resulting triple is still a triple of surface forms, that is, "born in" rather than `P19`, "Albert Einstein" rather than `Q937`, etc. This means that different fragments of the input text describing the a fact may yield different surface forms for the same entities. Also, a prompt that asks only for a core triple components might discard important contextual information (the kind of thing that goes into qualifiers and references).

Grounding The grounding step is responsible for transforming surface-form triples into KG statements. This transformation usually involves two related tasks: entity linking and relation linking. Entity linking attempts to map the surface forms in subject and object positions to existing KG entities, while relation linking (or canonicalization) attempts to map surface forms in predicate position to properties in the target KG.

So, for example, in a KB construction pipeline targeting Wikidata, the surface form "Albert Einstein" can be linked to `Q937`, "Ulm" to `Q3012`, "born in" to `P19` (place of birth), and "born on" to `P569` (date of birth). The resulting statements of our running example can thus be represented as:

$$\langle \texttt{wd:Q937}, \texttt{wdt:P19}, \texttt{wd:Q3012} \rangle,$$
$$\langle \texttt{wd:Q937}, \texttt{wdt:P569}, \texttt{"+1879-03-14T00:00:00Z"} \rangle.$$

The approach adopted in the extraction stage of a KB construction pipeline often determines how difficult the grounding stage will be. At one extreme, closed information extraction (cIE) assumes a fixed KG schema and generates triples that conform to this schema. At the other extreme, oIE (the setting we have been describing) imposes no constraints and attempts build the graph from scratch. Each of these extremes fail in a characteristic way. Closed extraction is precise but brittle: it is hard to adapt it to a new domain and it tends to silently discard whatever the schema cannot express. Open extraction is flexible but produces graphs whose nodes and edges are surface forms: as the graph grows, synonymy, coreference, and predicate variation accumulate and erode the properties that motivated building a KG in the first place (precision, interpretability, and logical consistency) [Chepurova et al. 2026].

A compromise between the two extremes, cIE and oIE, is what is sometimes called ontology-guided extraction. In ontology-guided extraction, one extracts freely, as in oIE, and only then align the surface forms against a given schema. This separation of concerns motivates the treatment of grounding as an independent, first-class stage, rather simply post-processing. And it is also why a vocabulary such as Wikidata's remains useful even for systems that never intend to contribute statements to Wikidata itself. Table 5.1 summarizes the three approaches we just described.

Setting	**Schema**	**Typical failure mode**
Closed IE (cIE)	Fixed, given a priori	Low recall; costly domain adaptation
Open IE (oIE)	None; induced from text	Redundancy; inconsistent surface forms
Ontology-guided	External, applied after extraction	Mismatch between text and ontology

Table 5.1: Positions along the information extraction spectrum for KG construction.

The previously mentioned EDC (Extract-Define-Canonicalize) [Zhang and Soh 2024] proposal supports a mixture of three approaches, cIE, oIE, and ontology-guided. If one provides a target schema, EDC becomes ontology-guided, as it aligns the open triples against the schema after extraction. But if no target schema is provided, EDC builds and canonicalizes its own schema on-the-fly. In either case, the pipeline is the same. EDC first uses an LLM to generate a natural-language definition for each relation induced by

the extraction phase. These definitions are then used to (i) identify semantically related elements in the schema, and (ii) validate whether an extracted relation can be replaced by one of the candidate relations.

For example, given the sentence "Alan Shepard was a member of the Apollo 14 crew", EDC extracts the surface-form triple

$$\langle \text{Alan Shepard},\ \text{participated in},\ \text{Apollo 14} \rangle.$$

The relation "participated in" in this case is assumed to have its usual meaning (definition), namely, that the subject took part in the event specified by the object entity. During canonicalization, EDC compares this definition is to those of the relations in the target schema and identifies, say, "mission", as a suitable candidate. EDC then uses the LLM to verify whether the replacement of "participated in" by "mission" is appropriate in the given context, and if so, and produces the canonical triple

$$\langle \text{Alan Shepard},\ \text{mission},\ \text{Apollo 14} \rangle.$$

This example illustrates that, rather than requiring the full target schema during extraction, EDC performs the alignment after the initial extraction, allowing the same pipeline to be used with both a target scheme or an initially empty schema which is generated on the fly.

Grounding natively against Wikidata Wikontic [Chepurova et al. 2026] is an interesting proposal for grounding surface-forms against Wikidata. Instead of treating Wikidata as one of many possible target schemas, it uses Wikidata as the backbone of the grounding method. More specifically, Wikontic uses a custom ontology derived from Wikidata which contains 2,464 factual properties such that, for each of these, Wikontic records the subject and object type constraints with entity types expanded recursively via instance of (P31) and subclass of (P279) relationships, so that a constraint declared for a superclass, such as "audiovisual work", still applies to its subclasses, such as "film".

The custom Wikidata-derived ontology is used by Wikontic to guide both extraction and alignment, which are carried out more thoroughly than in the generic proposals discussed previously, such as EDC. In particular, Wikontic's extraction stage does not stop at the bare triple: it uses the LLM to generate statement qualifiers (time, location, or other conditions under which the statement holds). So, given the sentence "In 2010, Christopher Nolan directed the science fiction movie Inception", Wikontic might generate as a candidate the surface-form triple & qualifier:

$$\langle \text{Nolan},\ \text{directed},\ \text{Inception} \rangle$$
$$\textit{qualifier:}\ \langle \text{point in time},\ 2010 \rangle$$

with the additional constraints that "Nolan" is an instance of "human" and "Inception" is an instance of "film".

In Wikontic, entity and relation linking are driven by type constraints rather than by semantic similarity definitions alone. Candidate types for both subject and object are retrieved from a dense index and selected by the LLM, with supertypes added so that constraints declared at a higher level of abstraction still apply. Wikidata's own constraints

are then used to enumerate every relation that can possibly connect the chosen types, with inverses (such as "directed" vs "director") also considered. Only the admissible set together with the source text are submitted to the LLM for a final decision. So, in the previous example, Wikontic's approach repairs both the surface form and the direction of the candidate, yielding

$$\langle \text{Inception},\ \text{director},\ \text{Christopher Nolan} \rangle$$
$$\textit{qualifier:}\ \langle \text{point in time},\ 2010 \rangle$$

Differently from EDC, Wikontic handles entity deduplication. Subject and object mentions are linked to existing entries construction that share a compatible type in the current graph under. This is done using embeddings over previously seen aliases. If a match is found, the mention is replaced by the canonical label and its surface form is stored as a new alias; otherwise a new entity is created. This is what makes the merging of "Nolan" and "Christopher Nolan" possible in the example above.

Note that although we said that Wikontic is an example of an ontology-guided system, the "entities" it produces are not Wikidata items: instead they are canonical names together with aliases, and the final entity-linking step is delegated to users. Also, although Wikontic generates qualifiers it does not attempt to generate references or ranks.

Validation The grounding stage produces well-formed statements, but not necessarily correct statements: the LLM may link a surface-form to the wrong entity, select a property that only approximates the relation in the source, or generate a value the source does not support. Candidate statements should therefore be validated before being added to the target graph [Zhong et al. 2023, Chen et al. 2024].

Validation draws on at least four complementary sources of evidence. The first one are *schema constraints*, such as domain and ranges of properties. For instance, `P19` (place of birth) expects a human as subject and a geographic place as object. These constraints have been formalized in SHACL and SPARQL [Ferranti et al. 2024] and used to verify LLM-extracted triples [Chepurova et al. 2024]. The second source of evidence for statement validation is the *existing knowledge* in the graph. For instance, a generated statement might contradict a statement that already exists in the graph, which serves as evidence to deem the generated statement invalid. The third source of evidence is the *source text* which was used to produce the statement as, in principle, the former should entail the latter. Finally, the fourth source of validation is *human review*, which is costly but might be necessary in mission-critical applications.

Note that statements that fail validation need not be discarded. They can be flagged as such and recorded together with the constraint violated and the input text that was used to generate them.

5.3.2.3. Verifying and Curating Wikidata with LLMs

So far, we discussed knowledge graph completion and construction. A complementary line of work uses LLMs not to add new knowledge, but to assess and improve knowledge that is already present in Wikidata. Quality assurance at Wikidata's scale is not a new problem. For over a decade, the graph has relied on machine-learning classifiers trained

on edit metadata and content features to catch vandalism and low-quality edits in near real time [Heindorf et al. 2016]. What LLMs add to this picture is not merely a faster classifier, but a qualitatively different kind of check. They can be fed a statement alongside a piece of text and asked to decide on whether the text supports the statement. Alternatively, they can be asked to evaluate a statement against the ontology in order to determine if the combination is sound. These checks depend on language and semantic understanding rather than edit-history features alone.

Verifying statements against external evidence One way to check a statement is to ask whether the world, as described in some independent source, actually supports it. Adam and Kliegr [Adam and Kliegr 2025] propose a method in which an LLM does not rely on its own internal knowledge to verify a statement. Instead, the statement is compared against externally retrieved grounding text, allowing the system to produce a traceable justification for its assessment rather than an unexplained verdict. This distinction matters. An LLM asked in isolation whether "Tim Berners-Lee is an employee of the World Wide Web Consortium" is true answers from the same parametric knowledge that makes LLMs unreliable narrators of fact in the first place (Section 5.3.1). Conversely, an LLM asked to compare the statement against a retrieved passage performs a task closer to reading comprehension, where a wrong verdict can at least be traced back to a specific piece of evidence. The approach is demonstrated on Wikidata by automatically identifying statements that require verification, prioritizing the pool of candidates a human curator would otherwise have to review from scratch.

Verifying statements against the ontology A second, complementary check does not need an external document at all. It asks whether a statement is internally consistent with the ontology it is supposed to belong to. Wikidata's property constraints specify expectations about how a property may be used, such as the subject and object type constraints introduced in Section 5.3.1. A violation of these constraints is a structural signal that a statement may be wrong, independently of whether it is factually true. Formalizing these constraints does not by itself require an LLM. Ferranti et al. [Ferranti et al. 2024] encode Wikidata's property constraints in SHACL and SPARQL, providing a systematic, fully symbolic way to identify violations across the graph. Nevertheless, this formalization is what makes LLM-based curation checkable rather than merely plausible-sounding. It supplies explicit, machine-checkable criteria against which a statement can be evaluated, whether it is authored by a human editor or proposed by an LLM-based construction pipeline (Section 5.3.2.2).

The two checks are not alternatives so much as complementary signals that recent systems combine directly. MAKGED [Li et al. 2025c], for instance, detects erroneous triples by having several LLM-backed agents examine a candidate from different angles. Each agent is conditioned on a different bidirectional view of the triple's local subgraph structure, settling disagreements through a multi-round discussion rather than relying on a single opaque score. This is a step beyond earlier, purely structural error detectors. Those models score a triple's plausibility from graph embeddings alone and struggle precisely on cases that look structurally normal but are semantically wrong [Liu et al. 2024]. Replacing a director with another plausible-looking director is exactly the kind of error that structure alone misses but a language-aware check catches. Combining a structural signal with an

LLM's reading of an entity's textual description, as MAKGED does, is one way to close that gap. Comparing a statement against retrieved external text, as Adam and Kliegr do, is another. Both methods replace a single confidence score with something a curator can inspect.

Curating structure, not just statements LLMs can also be used to address problems in the semantic structure of the KG itself, rather than in any one statement. WiKC [Peng et al. 2024] combines LLMs with graph-mining techniques to refine the Wikidata taxonomy. The approach identifies problematic taxonomic relations and performs operations such as cutting spurious links or merging classes that were only superficially distinct, producing a cleaned version of the taxonomy. This extends the role of LLMs beyond the statement-level completion and verification discussed in the previous sections. It supports the maintenance of the class hierarchy that those very statements are typed against. This hierarchy forms the *instance of* and *subclass of* backbone that both KG completion (Section 5.3.2.1) and KG construction (Section 5.3.2.2) rely on to generalize constraints across levels of abstraction. An error in the taxonomy propagates silently into every statement typed under it, making this structural layer a target worth curating in its own right.

From proposal to insertion: what automation cannot yet close These approaches also highlight an important distinction between AI-assisted curation and fully automated editing. Because LLMs can generate plausible but incorrect statements, their output cannot be assumed to be suitable for direct insertion into Wikidata. The very hallucination risk that motivates using Wikidata to ground LLM outputs elsewhere in this chapter reappears on the writing side, once an LLM is itself proposing what the graph should say [Lavrinovics et al. 2025]. [Regino and dos Reis 2025] offer a direct test of whether an LLM can be trusted to validate a triple before it is inserted, rather than just completing or extracting one. They evaluate LLMs on four concrete curation tasks: class and property alignment, URI standardization, semantic consistency, and syntactic correctness. Their results are informative less for a particular accuracy number than for where the models struggle. They highlight challenges with domain generalization across ontologies, semantic drift over longer chains of reasoning, and cases that still require a human in the loop even when the model performs well on individual tasks. Consequently, the more defensible role for LLMs in this setting is as assistants that propose candidate statements, identify likely errors, retrieve supporting evidence, or prioritize items for human review. This aligns exactly with the role played by the statement-level and structural checks described above. Wikidata's constraints, references, and collaborative editing mechanisms provide complementary safeguards for evaluating any AI-generated contribution before it becomes part of the graph.

Taken together, these approaches illustrate the complementary nature of the two perspectives introduced above. In Wikidata for AI, the LLM acts primarily as an interface through which structured knowledge can be interpreted, queried, and incorporated into an AI application. In AI for Wikidata, the LLM becomes a component in the construction and maintenance of the knowledge itself, generating, extracting, enriching, or validating statements. The interaction is therefore not a one-way flow of information from the KG to the model, but a feedback loop in which structured knowledge can support AI systems while AI can assist in expanding and improving the KG.

5.3.3. Wikidata Infrastructure for AI Applications

The interaction between Wikidata and AI applications depends not only on techniques, but also on infrastructure that makes the knowledge graph accessible to modern AI systems. While conventional interfaces provide precise mechanisms for querying and retrieving Wikidata, applications based on LLMs and AI agents often benefit from semantic retrieval and more flexible forms of interaction.

In this section, we introduce two community-driven initiatives that address these requirements: the Wikidata Vector Database and the Wikidata Model Context Protocol. They provide complementary capabilities for semantic retrieval and agentic interaction, extending the traditional access mechanisms discussed in Section 5.2.4. Together, they illustrate how the Wikidata ecosystem is evolving to accommodate the needs of generative AI applications.

5.3.3.1. The Wikidata Vector Database

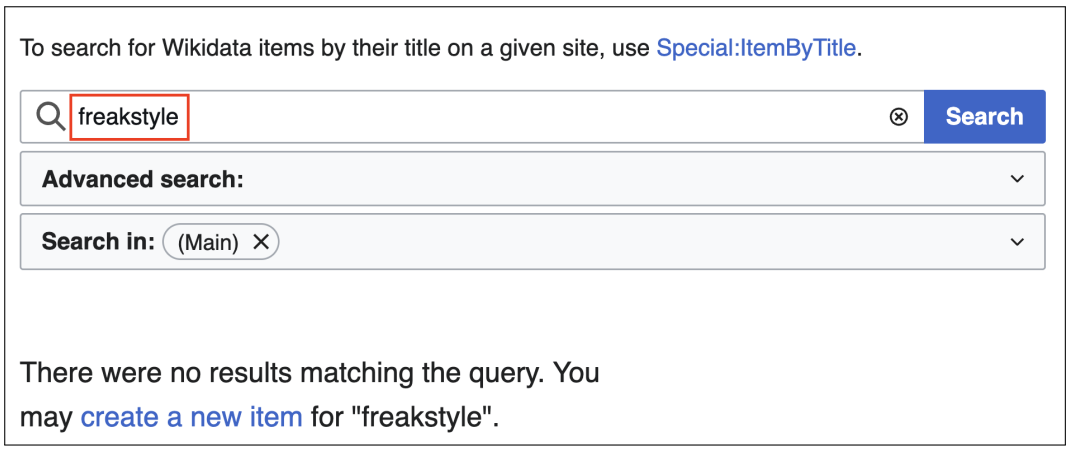
In both directions of collaboration between Wikidata and AI, the applications mentioned in previous sections have demonstrated the need to query Wikidata to capture entities present in the graph. While traditional mechanisms generally serve this purpose, they are often limited to structured searches or exact keyword matching; this hinders interaction with applications that primarily handle natural language, where terms may differ grammatically yet share the same meaning. To bridge this gap between traditional structured search and continuous semantic spaces, the *Wikidata Vector Database* (Wikidata VDB)[20] operationalizes semantic access to Wikidata by storing and serving dense vector representations of items and properties [Wikimedia Deutschland 2026].

Wikidata VDB is developed as the core operational layer of the *Wikidata Embedding Project*, which is an initiative led by Wikimedia Deutschland in collaboration with Jina AI[21] and DataStax[22] intended to support AI and machine-learning applications [Wikimedia Deutschland 2025]. While the traditional Wikidata keyword search, via CirrusSearch[23], is strictly bound to surface text and SPARQL is precise and accurate, but its steep learning curve and complexity make it challenging for many users to leverage, vector representations allow semantic search even when queries differ syntactically or orthographically from canonical labels.

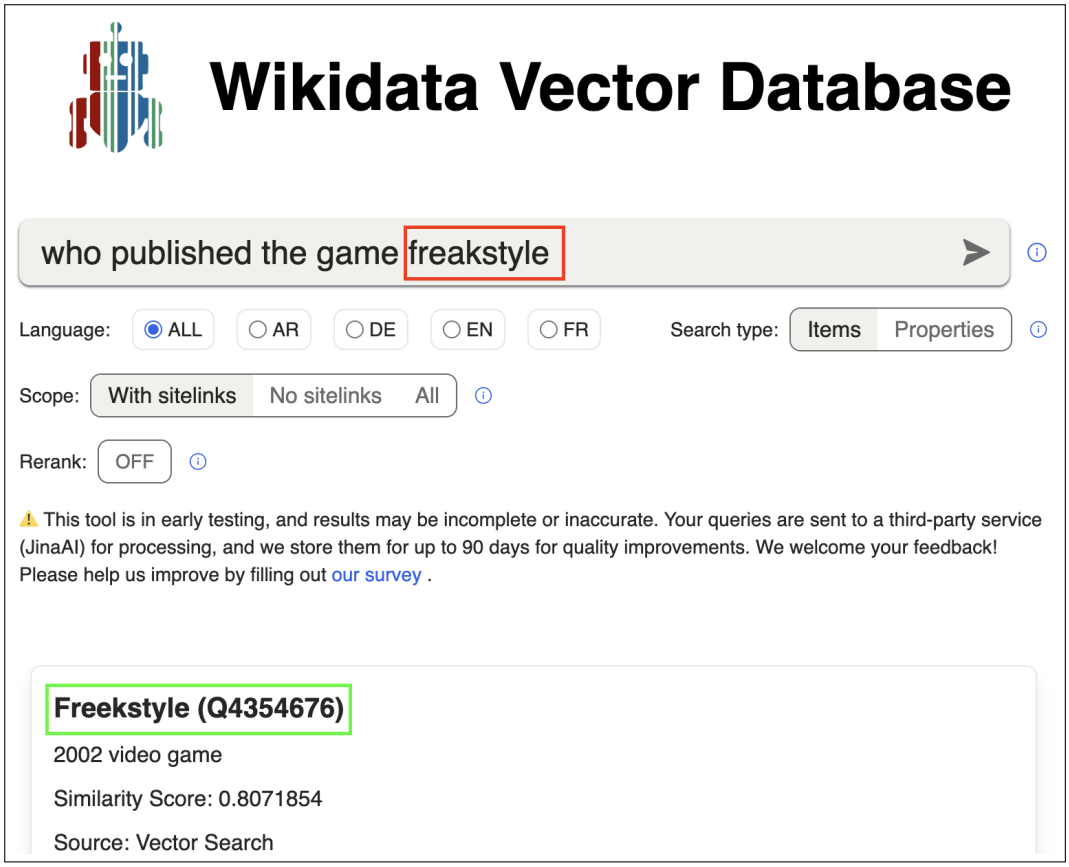
The basic idea is simple: information associated with Wikidata entities is transformed into textual representations and then encoded as dense vectors. Queries can be embedded in the same space, allowing the system to retrieve semantically related entities even when the wording of the query does not exactly match the labels stored in Wikidata. For example, a user might search for "scientists who investigated the relationship between heredity and evolution" without knowing the name of the relevant person or the precise vocabulary used in Wikidata. Semantic retrieval can identify entities whose textual representations are conceptually close to the query.

[20] https://wd-vectordb.wmcloud.org
[21] https://jina.ai/
[22] https://www.ibm.com/products/datastax
[23] https://www.mediawiki.org/wiki/Help:CirrusSearch

KIF-QA, for example, uses the traditional Wikidata API to retrieve candidate entities during the entity and relation linking processes. As the authors noted, KIF-QA may fail to answer correctly if the LLM generates a surface form that, does not yield any can–



(a) Searching the Wikidata traditional lookup interface.



(b) Searchin the Wikidata Vector Database interface.

Figure 5.12: Comparison of Wikidata interfaces for the term “freakstyle” (with typo). The traditional lookup interface returns no results (a), whereas the Vector Database interface retrieves relevant results (b).

didate entities in Wikidata. For instance, the paper reports that Llama 3.3 misrecognized "fre**e**kstyle" as "fre**a**kstyle" (with 'a') in the question "who published the game freekstyle" (from SimpleQuestionsQA dataset [Bordes et al. 2015]).Consequently, a keyword-based search cannot retrieve the correct entity from Wikidata (see Figure 5.12a). However, as shown in Figure 5.12b, semantic search through the Wikidata VDB can retrieve the correct entity despite the typo as dense embeddings capture conceptual context beyond brittle surface orthography.

5.3.4. Data Serialization and Vector Indexing. Turning a symbolic entity into something an embedding model can consume requires a text representation of it, and the choice of representation matters as much as the choice of model. Each item's label, description, aliases, and claims are concatenated into a short, human-readable paragraph, so that the text given to the embedding model reads like a compact entity summary rather than a bare list of triples. For complex entities with extensive claim networks, a statement-level chunking strategy splits claims into smaller segments while maintaining the core metadata (label, description, aliases, and *instance of* [`P31`]) across all chunks to fit model context windows. Furthermore, Wikidata properties (PIDs) are encoded using a dual representation: one treats the property itself as an entity, combining its own label, description, and aliases; the other pairs the property's label and description with a handful of example statements that use it, so that a property such as `P166` (`award received`) is represented alongside concrete instances of its use, giving the embedding model usage context that the property's bare definition would not convey on its own. All text passages are encoded using *Jina Embedding V3* (`retrieval.passage` mode), a multilingual embedding model supporting more than 100 languages and input sequences of up to 8,192 tokens. The resulting embeddings are indexed in *DataStax Astra DB*, partitioned separately by entity type (items vs. properties) and language. For unsupported languages, input queries are automatically translated into English via the MinT machine translation engine (based on NLLB-200) before searching across language shards.

5.3.5. Hybrid Retrieval Pipeline. Rather than replacing lexical lookup, the database combines vector-based retrieval with traditional keyword search through a hybrid execution pipeline (Figure 5.13). When a query is received, vector candidate retrieval (via Jina Embedding V3 `retrieval.query`) and keyword retrieval (via CirrusSearch with stop-word removal and logical `OR` expansion) run in parallel. The system can filter results dynamically using parameters such as `instanceof` or `exclude_external_ids`. Candidates from both channels are merged using Reciprocal Rank Fusion (RRF) [Cormack et al. 2009], computed as:

$$\mathrm{RRF}(e) = \sum_{l \in L} \frac{1}{k + r_l(e)} \tag{1}$$

where L is the set of ranked result lists, $r_l(e)$ is the rank position of entity e in list l, and $k = 60$ is a smoothing constant. Optionally, candidates undergo secondary reranking using *Jina Reranker v2 Base Multilingual*, a cross-encoder model that scores query-entity alignment at the cost of slightly higher latency.

5.3.6. Accessing Wikidata VDB. The database is exposed through a small REST API. Three endpoints cover the two ways one might want to use it: searching for new candidates,

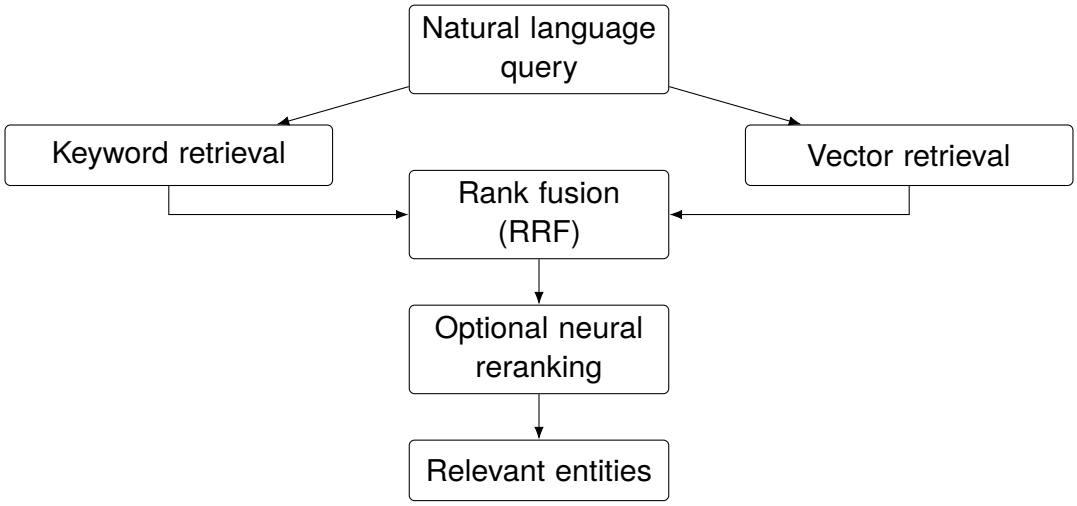


Figure 5.13: Hybrid pipeline to access Wikidata through lexical and semantic search.

and scoring a fixed list of candidates one already has.

- `GET /item/query/` — hybrid vector-and-keyword search over item shards, returning QIDs ranked by their fused score.
- `GET /property/query/` — the same hybrid search restricted to property shards, returning PIDs.
- `GET /similarity-score/` — given a text query and an explicit list of QIDs and/or PIDs, computes and returns similarity scores only for those entities, without retrieving any new candidates.

A request for items related to the earlier example query can be issued as:

https://wd-vectordb.wmcloud.org/item/query/?query=scientists+who+investigated+heredity+and+evolution&lang=en

which returns a ranked list of QIDs, such as `wd:Q1035` (Charles Darwin) together with their vector-similarity (0.72215676). An `instanceof` parameter can restrict item search to instances of a given class, and property search accepts an `exclude_external_ids` flag to drop properties whose datatype is `external-id`, which are rarely useful for semantic matching.

5.3.7. Limitations. The project describes itself as an alpha release, and two limitations follow directly from that status. Coverage is bounded to roughly 30 million items, those connected to at least one Wikipedia article, which excludes the long tail of Wikidata items with no corresponding Wikipedia page. And unlike the weekly and incremental dumps discussed in Section 5.2.4.3, the embeddings are not refreshed on a regular cadence: at the time of writing they are built from a May 2026 Wikidata snapshot, so an application relying on the vector database alone may miss statements added or edited since then.[24]

[24] In a LinkedIn post Wikidata profile posted that coverage for more Wikidata items is being prioritized, targeting September 2026: https://lnkd.in/dzG4NbHt

The project's own documentation is explicit about the intended use that follows from this:

> The vector search is primarily designed for exploration and discovery. It helps users and researchers identify related entities that might not appear through traditional keyword search. The results are not intended to serve as direct answers but as starting points for further investigation and analysis. [Wikimedia Deutschland 2026]

5.3.7.1. Wikidata MCP: Wikidata as an Agentic Tool Environment

The transition from static Retrieval-Augmented Generation (RAG) to agentic AI paradigms introduces a critical requirement: Large Language Models (LLMs) need standardized, dynamic interfaces to query and explore external knowledge environments. While LLMs are generally capable of generating syntactically valid SPARQL queries, they frequently fail in zero-shot execution over Wikidata due to a lack of domain-specific schema knowledge, unfamiliarity with entity identifiers (QIDs) and property predicates (PIDs), and complex taxonomic hierarchies. To bridge this gap, the Wikimedia community developed the *Wikidata MCP* server [Wikidata Community 2026], an implementation of Anthropic's open-standard Model Context Protocol (MCP) [Anthropic 2024]—frequently described as the "USB-C for AI" tools. By standardizing communication, MCP transforms Wikidata from a passive database accessed via hardcoded programmatic endpoints into an interactive *tool environment* that an LLM agent can actively explore.

Operationalizing agentic access via a standardized HTTP streaming interface (`streamable_http`) hosted at `https://wd-mcp.wmcloud.org/mcp/`, the service allows client environments to register the endpoint directly within their system configuration:

```
{
  "mcpServers": {
    "wikidata": {
      "type": "streamable_http",
      "url": "https://wd-mcp.wmcloud.org/mcp/"
    }
  }
}
```

Crucially, the interface operates in a read-only state, establishing a formal governance boundary that permits multi-turn graph exploration without exposing the underlying Knowledge Graph to unauthorized edits or operational risks. Within this secure execution context, the server exposes a specialized suite of tools designed to mirror a human researcher's exploratory workflow rather than forcing immediate query synthesis:

- `search_items(query, lang)`: Executes hybrid (vector and lexical) retrieval over Wikidata items to discover candidate entity identifiers (QIDs) from natural language queries.
- `search_properties(query, lang)`: Performs hybrid retrieval over properties to identify relevant relation predicates (PIDs) prior to constructing structured queries.

- `get_statements(entity_id, include_external_ids, lang)`: Retrieves direct graph assertions (triplets) for an entity, deliberately excluding qualifiers and references to provide a fast, low-latency structural overview.

- `get_statement_values(entity_id, property_id, lang)`: Fetches full claim details for a specific entity-property pair, including statement qualifiers, provenance references, and deprecation statuses when fine-grained metadata is required.

- `get_instance_and_subclass_hierarchy(entity_id, max_depth, lang)`: Traverses class relationships via *instance of* (`P31`) and *subclass of* (`P279`) up to a configured depth (defaulting to 5), validating ontological boundaries before query generation.

- `execute_sparql(sparql, K)`: Executes the compiled SPARQL query against the Wikidata Query Service (WDQS), returning up to K result rows or diagnostic syntax error messages.

The true architectural value of this modular suite lies in facilitating an autonomous reasoning loop, replacing rigid single-step prompt-to-SPARQL generation with iterative context gathering. As illustrated in Figure 5.14, an agent processing a complex prompt such as *"Who are the presidents of France?"* does not guess identifiers; it first invokes `search_items` and `search_properties` to resolve France (`Q142`) and president (`Q191954`), verifies class boundaries using `get_instance_and_subclass_hierarchy`, and only then constructs and executes the SPARQL query via `execute_sparql`. This multi-stage interaction ensures ground-truth verification and eliminates hallucinations, as facts are anchored strictly to deterministic graph execution. If execution yields syntax errors or empty result sets, the feedback loop routes the error message back to the LLM for self-correction.

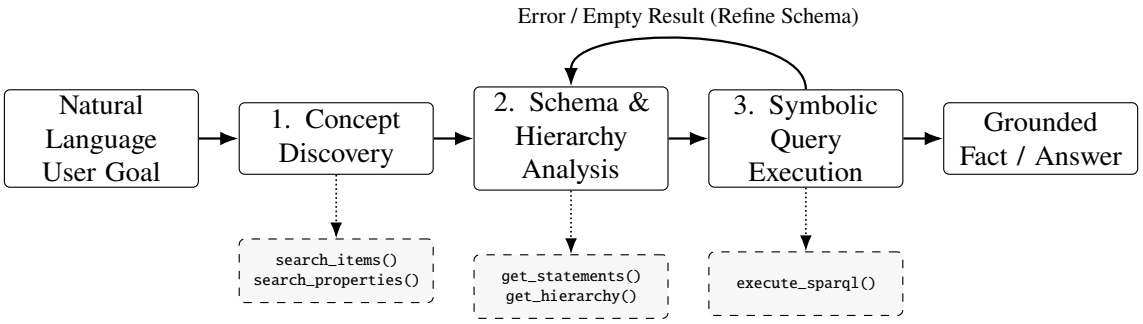


Figure 5.14: Agentic reasoning workflow enabled by Wikidata MCP: moving iteratively from fuzzy entity discovery to schema verification and deterministic SPARQL execution.

Beyond standalone research, the deployment of Wikidata MCP establishes a foundational bridge between structured open data and the broader developer ecosystem. The protocol has enabled integrations within agentic execution frameworks such as LangChain, as well as specialized workflow plugins like the OpenCode Wikidata SPARQL workflow (bringing direct Wikidata access to over 7.5 million developers). Furthermore, this paradigm

extends beyond Wikidata itself: through *Wikibase MCP* (`wb-mcp.wmcloud.org`), third-party Wikibase instances can register dedicated endpoints, extending standardized, agentic graph exploration to private enterprise knowledge bases.

5.4. Conclusion

This chapter has presented Wikidata as a KG and examined its role in the emerging ecosystem of generative AI. We first introduced the main concepts of the Wikidata data model and its RDF representation, showing how entities, statements, qualifiers, references, and ranks are represented as structured knowledge. We then discussed the different mechanisms through which this knowledge can be accessed, ranging from SPARQL and REST APIs to data dumps, Triple Pattern Fragments, and KIF. These interfaces expose the same underlying knowledge at different levels of abstraction and support different requirements in applications that consume or integrate Wikidata.

We then turned to the intersection between Wikidata and generative AI from two complementary perspectives. In *Wikidata for AI*, Wikidata can provide an external source of structured and inspectable knowledge that complements the parametric knowledge of LLMs. We discussed how this role appears in RAG, KGQA, and systems such as KIF-QA, where the language model is used to interpret a natural-language question while the KG provides the factual information. In these settings, Wikidata is not merely an additional text source: its explicit entities and relationships make it possible to retrieve, inspect, and query knowledge in ways that are difficult to guarantee from the model alone.

In the complementary *AI for Wikidata* perspective, the language model is instead used as a component in the construction and maintenance of the KG. We discussed approaches for KG completion and construction, where LLMs can predict missing information or extract candidate statements from unstructured sources, as well as approaches for verification and curation. Examples such as LLM Store illustrate how an LLM can be used to generate candidate knowledge, while other approaches use external evidence, ontology constraints, or graph structure to identify potentially incorrect statements. These examples also highlight an important limitation: the ability of LLMs to generate plausible information does not make their output reliable enough for unrestricted insertion into a collaboratively maintained KG. Human review, provenance, structural constraints, and other forms of validation therefore remain important parts of the process.

Finally, we examined recent infrastructure developed around Wikidata to support AI-oriented use cases. The Wikidata Vector Database extend traditional lexical retrieval with semantic retrieval, while the Wikidata MCP provides an interface through which AI agents can dynamically search, inspect, explore, and query the graph. These mechanisms do not replace Wikidata's traditional interfaces. Rather, they provide additional layers through which AI systems can interact with the same underlying structured knowledge, combining semantic retrieval with precise graph-based querying when necessary (Table 5.2 summarizes the Wikidata access interfaces).

Taken together, these perspectives show that the relationship between Wikidata and generative AI is bidirectional. Wikidata can provide AI systems with structured, multilingual, inspectable, and provenance-aware knowledge, while AI models can assist with discovering, constructing, verifying, and maintaining knowledge in the graph. The

Access mechanism	Paradigm	Typical role in AI systems
Keyword search	Lexical retrieval	Exact entity names, labels, and known terminology
SPARQL	Symbolic querying	Precise filtering, joins, aggregation, and graph reasoning
Vector Database	Semantic retrieval	Semantic search, candidate generation, RAG, and graph exploration
Wikidata MCP	Agentic tool use	Dynamic search, graph exploration, query execution, and orchestration

Table 5.2: Complementary mechanisms for accessing Wikidata in AI applications.

goal, therefore, is not to replace one technology with the other, but to combine their complementary capabilities. At the same time, this integration must preserve the properties that make Wikidata valuable as a collaborative KG, particularly the explicit representation of knowledge, provenance, openness, and community-based curation.

Declaration on Generative AI

During the preparation of this work, the authors used generative AI tools for machine translation from Portuguese to English, grammar and spelling checking, and to assist in paraphrasing and improving the writing style. After using these tools, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[Adam and Kliegr 2025] Adam, D. and Kliegr, T. (2025). Traceable llm-based validation of statements in knowledge graphs. *Information Processing & Management*, 62(4):104128.

[Angles 2012] Angles, R. (2012). A comparison of current graph database models. In *2012 IEEE 28th International Conference on Data Engineering Workshops*, pages 171–177. IEEE.

[Anthropic 2024] Anthropic (2024). Introducing the model context protocol. https://www.anthropic.com/news/model-context-protocol. Accessed: 2026-08-27.

[Bordes et al. 2015] Bordes, A., Usunier, N., Chopra, S., and Weston, J. (2015). Large-scale simple question answering with memory networks. *arXiv preprint arXiv:1506.02075*.

[Campos et al. 2025] Campos, J. P. P., Machado, M., Lima, G., and da Silva, V. T. (2025). Uma ferramenta web baseada em llms para respostas a perguntas simples sobre bases de conhecimento heterogêneas. In *Simpósio Brasileiro de Sistemas Multimídia e Web (WebMedia)*, pages 159–162. SBC.

[Carothers and Prud'hommeaux 2014] Carothers, G. and Prud'hommeaux, E. (2014). RDF 1.1 Turtle. W3c recommendation, W3C. http://www.w3.org/TR/2014/REC-turtle-20140225/.

[Chen et al. 2024] Chen, H., Shen, X., Lv, Q., Wang, J., Ni, X., and Ye, J. (2024). SAC-KG: Exploiting Large Language Models as Skilled Automatic Constructors for Domain Knowledge Graph. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4345–4360.

[Chen et al. 2020] Chen, Z., Wang, Y., Zhao, B., Cheng, J., Zhao, X., and Duan, Z. (2020). Knowledge graph completion: A review. *IEEE Access*, 8:192435–192456.

[Chepurova et al. 2026] Chepurova, A., Bulatov, A., Burtsev, M., and Kuratov, Y. (2026). Wikontic: Constructing wikidata-aligned, ontology-aware knowledge graphs with large language models. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8304–8319.

[Chepurova et al. 2024] Chepurova, A., Kuratov, Y., Bulatov, A., and Burtsev, M. (2024). Prompt me one more time: A two-step knowledge extraction pipeline with ontology-based verification. In *Proceedings of TextGraphs-17: Graph-based Methods for Natural Language Processing*, pages 61–77.

[Consortium 2025] Consortium, T. U. (2025). Uniprot: the universal protein knowledgebase in 2025. *Nucleic Acids Research*, 53(D1):D609–D617.

[Cormack et al. 2009] Cormack, G. V., Clarke, C. L. A., and Büttcher, S. (2009). Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 758–759. ACM.

[Ferranti et al. 2024] Ferranti, N., De Souza, J. F., Ahmetaj, S., and Polleres, A. (2024). Formalizing and validating Wikidata's property constraints using SHACL and SPARQL. *Semantic Web*, 15(6):2333–2380.

[Harris and Seaborne 2013] Harris, S. and Seaborne, A. (2013). SPARQL 1.1 query language. W3c recommendation, W3C. http://www.w3.org/TR/2013/REC-sparql11-query-20130321/.

[Hayes and Patel-Schneider 2014] Hayes, P. and Patel-Schneider, P. (2014). RDF 1.1 semantics. Recommendation, W3C. http://www.w3.org/TR/2014/REC-rdf11-mt-20140225/.

[Heindorf et al. 2016] Heindorf, S., Potthast, M., Stein, B., and Engels, G. (2016). Vandalism detection in Wikidata. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management (CIKM)*, pages 327–336. ACM.

[Hitzler et al. 2010] Hitzler, P., Krötzsch, M., and Rudolph, S. (2010). *Foundations of Semantic Web Technologies*. Chapman & Hall.

[Ibrahim et al. 2024] Ibrahim, N., Aboulela, S., Ibrahim, A., and Kashef, R. (2024). A survey on augmenting knowledge graphs (KGs) with large language models (LLMs): models, evaluation metrics, benchmarks, and challenges. *Discover Artificial Intelligence*, 4(1):76.

[Ji et al. 2021] Ji, S., Pan, S., Cambria, E., Marttinen, P., and Yu, P. S. (2021). A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE transactions on neural networks and learning systems*, 33(2):494–514.

[Ji et al. 2023] Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y. J., Madotto, A., and Fung, P. (2023). Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

[Kellogg et al. 2026] Kellogg, G., Hartig, O., Champin, P.-A., and Seaborne, A. (2026). RDF 1.2 concepts and abstract data model. https://www.w3.org/TR/rdf12-concepts/. Publicado em 7 abr. 2026. Versão desta data: https://www.w3.org/TR/2026/CR-rdf12-concepts-20260407/. A Recomendação vigente continua sendo RDF 1.1. Acesso em: 25 ago. 2026.

[Kim et al. 2023] Kim, S., Chen, J., Cheng, T., Gindulyte, A., He, J., He, S., Li, Q., Shoemaker, B. A., Thiessen, P. A., Yu, B., Zaslavsky, L., Zhang, J., and Bolton, E. E. (2023). Pubchem 2023 update. *Nucleic Acids Research*, 51(D1):D1373–D1380.

[Lavrinovics et al. 2025] Lavrinovics, E., Biswas, R., Bjerva, J., and Hose, K. (2025). Knowledge graphs, large language models, and hallucinations: An NLP perspective. *Journal of Web Semantics*, 85:100844.

[Lehmann et al. 2015] Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P. N., Hellmann, S., Morsey, M., van Kleef, P., Auer, S., and Bizer, C. (2015). Dbpedia – a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 6(2):167–195.

[Lewis et al. 2020] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., Riedel, S., and Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474.

[Li et al. 2025a] Li, D., Liang, Y., Qian, S., Wu, H., Jia, W., Fu, Y., and Sun, Y. (2025a). A review of background, methods, limitations and opportunities of knowledge graph completion. *Computer Science Review*, 58:100809.

[Li et al. 2025b] Li, M., Zhao, Y., Zhang, W., Li, S., Xie, W., Ng, S. K., Chua, T.-S., and Deng, Y. (2025b). Knowledge boundary of large language models: A survey. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5131–5157.

[Li et al. 2025c] Li, Y., Huang, Y., Qi, G., Feng, J., Hu, N., Zhai, S., Xue, H., Chen, Y., Shen, R., and Wu, T. (2025c). Harnessing diverse perspectives: A multi-agent framework for enhanced error detection in knowledge graphs. In *Database Systems for Advanced Applications (DASFAA 2025), Part VI*, volume 15991 of *Lecture Notes in Computer Science*, pages 446–458. Springer.

[Lima et al. 2024] Lima, G., Rodrigues, J., Machado, M., Soares, E., Fiorini, S. R., Thiago, R., Azevedo, L. G., da Silva, V. T., and Cerqueira, R. (2024). KIF: A Wikidata-Based Framework for Integrating Heterogeneous Knowledge Sources. *arXiv preprint arXiv:2403.10304*.

[Liu et al. 2024] Liu, X., Liu, Y., and Hu, W. (2024). Knowledge graph error detection with contrastive confidence adaption. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 8824–8831.

[Machado et al. 2025] Machado, M. d. O. C., Porto Campos, J. P., Lima, G. A. F., and Silva, V. T. d. (2025). KIF-QA: Using Off-the-Shelf LLMs to Answer Simple Questions over Heterogeneous Knowledge Bases. In *Proceedings of the 5th Wikidata Workshop*, volume 4108 of *CEUR Workshop Proceedings*. CEUR-WS.org.

[Machado et al. 2024a] Machado, M. d. O. C., Rodrigues, J. M. B., Lima, G., and da Silva, V. T. (2024a). LLM Store: A KIF Plugin for Wikidata-Based Knowledge Base Completion via LLMs. In *KBC-LM/LM-KBC@ ISWC*.

[Machado et al. 2024b] Machado, M. d. O. C., Rodrigues, J. M. B., Lima, G., and Silva, V. T. d. (2024b). LLM Store: A KIF Plugin for Wikidata-Based Knowledge Base Completion via LLMs. In *Proceedings of the 4th Workshop on Knowledge Base Construction from Texts: The LM-KBC Challenge @ ISWC 2024*, volume 3853 of *CEUR Workshop Proceedings*. CEUR-WS.org.

[Mousavi et al. 2026] Mousavi, S. M., Alghisi, S., and Riccardi, G. (2026). LLMs as Repositories of Factual Knowledge: Limitations and Solutions. *IEEE Transactions on Audio, Speech and Language Processing*, 34:2213–2226.

[Navigli et al. 2023] Navigli, R., Conia, S., and Ross, B. (2023). Biases in Large Language Models: Origins, Inventory, and Discussion. *J. Data and Information Quality*, 15(2).

[Norouzi and Hitzler 2025] Norouzi, S. S. and Hitzler, P. (2025). Knowledge-Enhanced Geospatial QA: Integrating Wikidata Fact Verification with LLMs. In *Proceedings of the AAAI Symposium Series*, volume 5, pages 334–339.

[Odell et al. 2022] Odell, J., Lemus-Rojas, M., and Brys, L. (2022). *Wikidata for Scholarly Communication Librarianship*. IUPUI University Library. DOI: 10.7912/9Z4E-9M13.

[Peng et al. 2024] Peng, Y., Bonald, T., and Alam, M. (2024). Refining Wikidata taxonomy using large language models. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management (CIKM)*. ACM.

[Regino and dos Reis 2025] Regino, A. G. and dos Reis, J. C. (2025). Can LLMs be knowledge graph curators for validating triple insertions? In *Proceedings of the Workshop on Generative AI and Knowledge Graphs (GenAIK)*, pages 87–99, Abu Dhabi, UAE. International Committee on Computational Linguistics.

[Robinson et al. 2013] Robinson, I., Webber, J., and Eifrem, E. (2013). *Graph Databases*. O'Reilly.

[Sáez and Hogan 2018] Sáez, T. and Hogan, A. (2018). Automatically generating wikipedia info-boxes from wikidata. In *Companion Proceedings of the The Web Conference 2018*, WWW '18, page 1823–1830, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

[Verborgh et al. 2014] Verborgh, R., Vander Sande, M., Colpaert, P., Coppens, S., Mannens, E., and Van de Walle, R. (2014). Web-scale querying through Linked Data Fragments. In Bizer, C., Heath, T., Auer, S., and Berners-Lee, T., editors, *Proceedings of the 7th Workshop on Linked Data on the Web*, volume 1184 of *CEUR Workshop Proceedings*. CEUR.

[Verborgh et al. 2016] Verborgh, R., Vander Sande, M., Hartig, O., Van Herwegen, J., De Vocht, L., De Meester, B., Haesendonck, G., and Colpaert, P. (2016). Triple pattern fragments: A low-cost knowledge graph interface for the web. *Journal of Web Semantics*, 37-38:184–206.

[Vrandečić and Krötzsch 2014] Vrandečić, D. and Krötzsch, M. (2014). Wikidata: a free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85.

[W3C OWL WG 2012] W3C OWL WG (2012). OWL 2 Web Ontology Language document overview (second edition). Recommendation, W3C. http://www.w3.org/TR/2012/REC-owl2-overview-20121211/.

[Wikidata 2026a] Wikidata (2026a). Wikidata:Data access. https://www.wikidata.org/w/index.php?title=Wikidata:Data_access. Revisão 2497420605. Acesso em: 25 ago. 2026.

[Wikidata 2026b] Wikidata (2026b). Wikidata:Database download. https://www.wikidata.org/w/index.php?title=Wikidata:Database_download. Revisão 2479770906. Acesso em: 25 ago. 2026.

[Wikidata Community 2026] Wikidata Community (2026). Wikidata: Mcp. https://www.wikidata.org/wiki/Wikidata:MCP. Accessed: 2026-08-27.

[Wikimedia Deutschland 2025] Wikimedia Deutschland (2025). Wikidata embedding project. https://www.wikidata.org/wiki/Wikidata:Embedding_Project. Accessed: 2026-08-27.

[Wikimedia Deutschland 2026] Wikimedia Deutschland (2026). Wikidata vector database. https://www.wikidata.org/wiki/Wikidata:Vector_Database. Accessed: 2026-08-27.

[Wood et al. 2014] Wood, D., Lanthaler, M., and Cyganiak, R. (2014). RDF 1.1 concepts and abstract syntax. W3c recommendation, W3C. http://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/.

[Youssef et al. 2023] Youssef, P., Koraş, O., Li, M., Schlötterer, J., and Seifert, C. (2023). Give me the facts! a survey on factual knowledge probing in pre-trained language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 15588–15605.

[Zhang and Soh 2024] Zhang, B. and Soh, H. (2024). Extract, define, canonicalize: An llm-based framework for knowledge graph construction. In *Proceedings of the 2024 conference on empirical methods in natural language processing*, pages 9820–9836. Association for Computational Linguistics.

[Zhong et al. 2023] Zhong, L., Wu, J., Li, Q., Peng, H., and Wu, X. (2023). A comprehensive survey on automatic knowledge graph construction. *ACM Computing Surveys*, 56(4):1–62.