

# **Minicursos da ERCEMAPI e EAComp 2019**

Organização

**Alex Oliveira Barradas Filho**

**Pedro Porfirio Muniz Farias**

**Ricardo de Andrade Lira Rabêlo**





Prof. Dr. Alex Oliveira Barradas Filho - UFMA

Prof. Dr. Pedro Porfirio Muniz Farias - UNIFOR

Prof. Ricardo de Andrade Lira Rabêlo - UFPI

## **Minicursos da ERCEMAPI e EAComp 2019**

ISBN: 978-85-7669-489-2

Sociedade Brasileira da Computação

Porto Alegre

2019

Barradas Filho, Alex Oliveira.

Minicursos da ERCEMAPI e EAComp 2019 / Alex Oliveira Barradas Filho, Pedro Porfírio Muniz Farias, Ricardo de Andrade Lira Rabêlo . — Porto Alegre: SBC, 2019.

112 p.: il.

ISBN 978-85-7669-489-2

1. Computação - Minicurso. 2. Robótica. 3. Modelo CEK. 4. Tecnologias. I. Farias, Pedro Porfírio Muniz. II. Rabêlo, Ricardo de Andrade Lira. III. Título.

CDD 004

CDU 004

Elaborada pela bibliotecária Marcia Cristina da Cruz Pereira CRB 13 / 418



## Prefácio

A VII Escola Regional de Computação Ceará, Maranhão, Piauí (ERCEMAPI 2019) tem como objetivo disseminar o conhecimento técnico e científico sobre temas e assuntos de vanguarda na área de Computação. A escola busca incentivar a produção, publicação e disseminação de material técnico, didático e científico de alta qualidade. É um fórum de integração entre coordenadores de curso, pesquisadores, alunos de graduação e pós-graduação, e representantes dos setores governamental, industrial e de serviços do Maranhão, Ceará, Piauí e região. Nesta sua sétima edição, sediada mais uma vez em São Luís, MA, a ERCEMAPI resgatará uma jornada de sucesso iniciada em 2007. Esse resgate visa consolidar a ERCEMAPI como um evento de referência nos Estados do Maranhão, Ceará e Piauí e região. Além disso, o evento colabora e contribui com a consolidação dos Programas de Pós-Graduação da Região num momento em que todos os três estados possuem programas de Pós-Graduação em Computação nos níveis de Mestrado e Doutorado.

Em conjunto com a ERCEMAPI 2019, será realizado o VII Encontro Acadêmico de Computação (EAComp 2019). Este evento de caráter acadêmico e técnico-científico com o objetivo de propiciar um espaço para os alunos dos cursos de Computação da UFMA, apresentando as oportunidades dentro da graduação/universidade, intercâmbio entre os diversos atores dos setores de Informática da região e oportunidades de desenvolvimento tecnológico e de inovação em Computação. O EAComp é organizado pelo Programa de Educação Tutorial do curso de Ciência da Computação (PETComp) do Departamento de Informática (DEINF) da UFMA.

Os minicursos têm a duração de 4 horas e permitem que participantes recebam informações sobre novas tecnologias e tópicos atuais de pesquisa em áreas da computação. Assim, minicursos aparecem como uma oportunidade para familiarização dos participantes da ERCEMAPI com novos temas de pesquisa que podem vir a ser úteis em suas vidas profissionais. O processo de seleção de minicursos foi feito a partir de várias chamadas públicas divulgadas na lista eletrônica de emails da SBC, bem como ampla divulgação no site oficial do evento e em redes sociais. Foram recebidas 7 (sete) propostas de minicursos, avaliadas por comitê composto de professores com conhecimento nos temas abordados. Cada minicurso foi avaliado por três avaliadores, que pontuaram notas para quesitos como relevância para o evento, expectativa do público, atualidade e conteúdo de cada minicurso. Ao final, 5 (cinco) propostas foram selecionadas, cujos conteúdos abordados constituem os capítulos deste livro.

O primeiro capítulo, intitulado “Modelo CEK (Canvas + EAP + Kanban): Integração de Ferramentas de Gestão para transformar ideias em projetos colaborativos, ágeis, controláveis e efetivos” aborda o modelo CEK e sua dinâmica do funcionamento do Modelo CEK, com detalhes de sua dinâmica, sendo apresentado um exemplo de sua aplicação prática, como forma de sedimentar nos alunos o conhecimento e com a pretensão de ser uma chamada à ação, incentivando sua aplicação.



O Capítulo dois, “SCRATCH FOR ARDUINO: Desenvolvendo o pensamento computacional com auxílio da robótica educacional”, tem como objetivo a confecção de um modelo didático com Arduino e afins para o desenvolvimento do pensamento computacional e aprendizado introdutório a programação de computadores.

O terceiro capítulo é intitulado “Introdução ao V-REP: Uma Plataforma Virtual para Simulação de Robôs”. Em especial, o minicurso tem como objetivo principal uma visão introdutória sobre a plataforma V-REP (Virtual Robot Experimentation Platform) apresentando conceitos básicos da ferramenta, assim como alguns exemplos de simulações.

O capítulo quatro é intitulado “Planejamento de Caminho para Sistemas Robóticos Autônomos”. Este minicurso aborda aspectos introdutórios sobre planejamento de caminho para sistemas robóticos, apresentando as principais técnicas, bibliotecas, e aplicações, com destaque na robótica autônoma.

Por fim, o capítulo 5, apresenta o minicurso intitulado “Os cinco passos para transformar qualquer site de dados públicos em uma API de dados abertos”. Este minicurso aborda de maneira prática, cinco passos para transformar sites com dados públicos em portais de dados abertos. Esses passos incluem a (1) modelagem e (2) extração dos dados de um site web; (3) O desenvolvimento e (4) documentação de uma API REST; E a (5) implantação em um serviço de nuvem.

Esperamos que este livro seja útil para todos aqueles interessados nos tópicos abordados da Escola Regional.

São Luís, setembro de 2019.

Prof. Dr. Alex Oliveira Barradas Filho (UFMA)

Prof. Dr. Pedro Porfirio Muniz Farias (UNIFOR)

Prof. Ricardo de Andrade Lira Rabêlo (UFPI)

Coordenadores dos Minicursos da ERCEMAPI 2019

## Capítulo

# 1

## **Modelo CEK (Canvas + EAP + Kanban): Integração de Ferramentas de Gestão para transformar ideias em projetos colaborativos, ágeis, controláveis e efetivos**

Fernando Escobar, Washington Henrique Carvalho Almeida, Luciano Aguiar Monteiro, Pedro César Vieira Barbosa e Aislan Rafael Rodrigues Souza.

### *Abstract*

*This chapter presents problems that historically impacted software development. The CEK Model, based on the approach of project management applied to software engineering through the integration of management tools like Canvas, Work Breakdown Structure, and Kanban, is proposed as one of the possible answers to software development problems. About the CEK Model, the management tools that compose it are described through a dense theoretical foundation. Following, we feature the dynamics of CEK operation, with the details of its application and present an example of its use, as a form of remain the knowledge and with the pretense of being a call to action, encouraging its application.*

### *Resumo*

*Neste capítulo são apresentados problemas que historicamente impactaram o desenvolvimento de software. O Modelo CEK, fundamentado na abordagem por projetos aplicada à engenharia de software, por meio da integração das ferramentas de gestão Canvas, Estrutura Analítica do Projeto e Kanban, é proposto como uma das possíveis respostas aos problemas do desenvolvimento de software. Acerca do Modelo CEK, as ferramentas de gestão as quais o compõe são descritas por meio de uma densa fundamentação teórica. Na sequência, é descrita a dinâmica do funcionamento do Modelo CEK, com detalhes de sua dinâmica, sendo apresentado um exemplo de sua aplicação prática, como forma de sedimentar nos alunos o conhecimento e com a pretensão de ser uma chamada à ação, incentivando sua aplicação.*

## 1.1. Introdução

Desde muito cedo, têm-se buscado a aplicação de técnicas para aprimorar o processo de desenvolvimento de software. Na década de 1930, um especialista da Bell Labs propôs a adoção de uma versão rudimentar do ciclo PDCA (*Plan - Do - Check - Act*) voltada à melhoria da qualidade do processo de software, sendo mais tarde explorada e promovida por William Edwards Deming. Já na década de 1960, no âmbito do projeto Mercury na NASA, a equipe adotou iterações curtas, com revisões técnicas, com testes descritos antes da implementação, em uma prática aderente ao que hoje classificamos como métodos ágeis e TDD (Test Driven Development) [Larman and Basili 2003].

Na década de 1970 e 1980, em busca de respostas aos problemas de estouro de orçamento, baixa qualidade do código e não satisfação aos requisitos dos clientes, as empresas começaram a adotar métodos sistematizados de desenvolvimento, entretanto sem resultados concretos [Robiolo 2014]. Nesta mesma época, o modelo em cascata (waterfall) começou a ser adotado [Royce 1987].

Na década de 1990, novas metodologias começaram a surgir, com destaque para os métodos ágeis, que se baseiam na premissa de um software de alta qualidade, adaptando-se aos requisitos do cliente, baseado em rápido *feedback*, com melhorias precoces, em contraponto à abordagem tradicional, na qual os requisitos precisam estar totalmente especificados, antes da implementação, conforme o modelo em cascata [Sharma and Kotwal 2016].

Entretanto, segundo [Pressman and Maxim 2016], algumas questões ainda persistem no campo do desenvolvimento de software:

- Por que a conclusão de um software leva tanto tempo?
- Por que os custos de desenvolvimento são tão altos?
- Por que não conseguimos encontrar todos os erros antes de entregarmos o software aos clientes?
- Por que gastamos tanto tempo e esforço realizando a manutenção de programas existentes?
- Por que ainda temos dificuldades de medir o progresso de desenvolvimento e a manutenção de um software?

[Sommerville 2019] apresenta que existem, ainda, muitos relatos de projetos equivocados e de “falhas de software”. O autor registra que a engenharia de software é criticada como inadequada para o desenvolvimento de software moderno. Entretanto, ainda segundo este mesmo autor, muitas dessas alegadas falhas são consequência de dois fatores: complexidade crescente dos sistemas e a não utilização de métodos de engenharia de software.

Corroborando o diagnóstico, [Shneiderman and Plaisant 2006] afirmam que 60% dos projetos de software falham na definição dos objetivos, em decorrência de falha de comunicação entre integrantes da equipe e os usuários finais.

[Pressman and Maxim 2016] conceituam o que pode ser uma resposta aos problemas diagnosticados, a partir da abordagem do projeto de software, que engloba de forma sistematizada princípios, conceitos e práticas, que podem levar ao desenvolvimento de sistema ou produto com alta qualidade, a partir da prevalência dos princípios de projeto sobre as atitudes e ações do desenvolvimento, orientando as atividades para realizar o projeto.

Corroborando a relevância da abordagem por projetos, [Vargas 2005] pontua que a adoção da abordagem do gerenciamento de projetos torna a resposta das organizações mais ágil e competente, aumentando a probabilidade de atingir seus resultados.

Em linha com esse raciocínio, [Archibald and Archibald 2016] defendem que a adoção da abordagem por projetos é um ponto de inflexão nas organizações, indutor da capacidade de geração de valor e da inovação. Os autores, ainda, evidenciando a relevância da temática, reconhecem que todas as inovações significativas são alcançadas por meio de projetos.

Como resposta a este cenário, fundamentado na abordagem por projetos aplicada à engenharia de software, em um método híbrido que combina elementos das abordagens tradicional e ágil, apresentamos neste capítulo o Modelo CEK (Canvas + EAP + Kanban). Trata-se da proposição da integração de diferentes ferramentas de gestão, de forma a mitigar os problemas relacionados ao desenvolvimento de software, em especial às falhas no levantamento de requisitos e definição de objetivos, à comunicação ineficiente entre equipe e clientes, com foco na ideação, planejamento e monitoramento da execução, reduzindo desperdícios, transformando ideias em projetos colaborativos, ágeis, controláveis e efetivos.

Esse capítulo está organizado da seguinte forma: 1.1-Introdução; 1.2-Ferramentas de Gestão; 1.3 Canvas; 1.4 EAP; 1.5 Kanban; 1.6 Modelo CEK; e 1.7 Conclusões.

## **1.2. Ferramentas de Gestão**

[Fernandes Vendruscolo 2011] consolida a visão de diversos autores ao conceituar as ferramentas de gestão como aquelas utilizadas pelos gestores, tanto na elaboração e na definição da estratégia da organização, quanto no desdobramento desta em projetos e ações que alterem a realidade e contribuam para a operação e a agregação de valor na organização.

[Rigby and Others 2009] afirmam que, ao longo das últimas décadas, as ferramentas de gestão se tornaram comuns na vida das organizações, auxiliando os executivos a atingirem seus objetivos, sejam eles aumentar receitas, reduzir custos, inovar, melhorar a qualidade ou se planejar para o futuro.

Corroborando o raciocínio, [Daychoum 2018] lista ferramentas e técnicas de apoio à gestão, pontuando que elas se aplicam às mais diversas áreas de conhecimento, reconhecendo a possibilidade de adaptações de seus usos, tendo como objetivo a busca da melhoria da gestão.

Dentre as muitas ferramentas de gestão existentes, este estudo centra-se no Canvas, em especial Project Design Canvas, no Gerenciamento de Projetos, em especial na Estrutura Analítica do Projeto (EAP) e no Quadro Kanban.

## **1.3. Canvas**

Canvas é uma palavra de origem inglesa que significa uma tela de pintura. A abordagem do Canvas tem sido utilizada, como ferramenta de gestão, no mundo dos negócios e dos projetos. O Canvas funciona como uma representação visual que apoia o design (projetização) e a comunicação [Lüftenegger et al. 2012]]. Essa abordagem emergiu do domínio dos Sistemas de Informação, em princípio no mundo acadêmico, e

atualmente tem sido amplamente aceita nos mais diferentes segmentos da indústria, negócios, governo e terceiro setor [Osterwalder and Pigneur 2010].

### 1.3.1. Canvas de Modelo de Negócio

O Canvas de Modelo de Negócio, ou Business Model Canvas – BMC é uma ferramenta de gestão que ajuda a descrever como uma organização cria, gera e entrega valor, por meio de um modelo de negócio viável que gere riqueza à organização [Osterwalder et al. 2019].

O BMC foi desenvolvido por Alexander Osterwalder e Yves Pigneur, como tese de doutorado, com o objetivo de propor algo que viesse a descomplicar a concepção de um modelo de negócio, com uma abordagem colaborativa, incentivando empreendedores a inovar e prototipar novos negócios [Leandro and Vieira 2018]. Ainda segundo os autores, o BMC inovou ao aplicar uma abordagem visual – com ilustrações e poucos textos –, simplificando em uma única página a forma de conceber negócios, tornando o processo divertido e intuitivo ao propor que os participantes da dinâmica registrassem suas ideias em *post-its* e os colassem nos blocos do Canvas.

O Canvas de Modelo de Negócio está ilustrado, com suas dimensões (blocos), na Figura 1.



Figura 1. Canvas de Modelo de Negócio. Fonte: [Osterwalder et al. 2019]

A forma inovadora, visual e com dinâmica colaborativa de conceber negócios foi a inspiração inicial, a partir da qual os diversos outros modelos propostos de Canvas, de negócios e de projetos foram derivados [Leandro and Vieira 2018].

### 1.3.2. Project Model Canvas – PMCanvas

A ferramenta de gestão visual de projetos, nomeada Project Model Canvas – PMCanvas foi desenvolvida pelo professor José Finocchio Júnior, baseado no Business Model Canvas, utilizando uma orientação específica para projetos, constituindo-se em uma metodologia [De Sousa Neto 2014; Leandro and Vieira 2018].



- **Comunicar/compartilhar:** o Canvas servirá como base para gerar outros documentos importantes para o projeto.

[Abdollahyan 2013] apud [Silva et al. 2015] afirma que o PMCanvas se relaciona fortemente com os documentos Termo de Abertura do Projeto e Plano de Gerenciamento do Projeto, ambos previstos no Guia do Conhecimento em Gerenciamento de Projetos – PMBOK, abordando principalmente os processos de iniciação e planejamento.

Corroborando esse conceito, [ProjectBuilder 2017] afirma que o PMCanvas é uma metodologia que permite conceber projetos em uma sistemática de página única e transformar os integrantes do time em agentes de inovação, envolvendo e engajando toda a equipe, aumentando o comprometimento e produtividade.

### 1.3.3. Canvas Project Design

O Canvas de Projeto – Canvas Project Design foi proposto pelos professores Wankes Leandro Ribeiro e Helber Ricardo Vieira, a partir da síntese entre modelos teóricos, boas práticas e diversas sessões de tentativa e erro.

De acordo com [Leandro and Vieira 2018], a síntese que culminou no Canvas Project Design incorporou conceitos, abordagens e características do Design Thinking, da Programação Neurolinguística, dos Guias PMBOK e PRINCE2, das Abordagens Ágeis e do Princípio MECE<sup>1</sup> (Mutuamente Exclusivo, Coletivamente Exaustivo).

A ferramenta de gestão do Canvas Project Design foi desenvolvida para compor uma técnica idealizada de conceber projetos de um modo simples e prático, visual, participativo e colaborativo [Leandro and Vieira 2018].

O Canvas Project Design é assim estruturado, conforme a Figura 3:

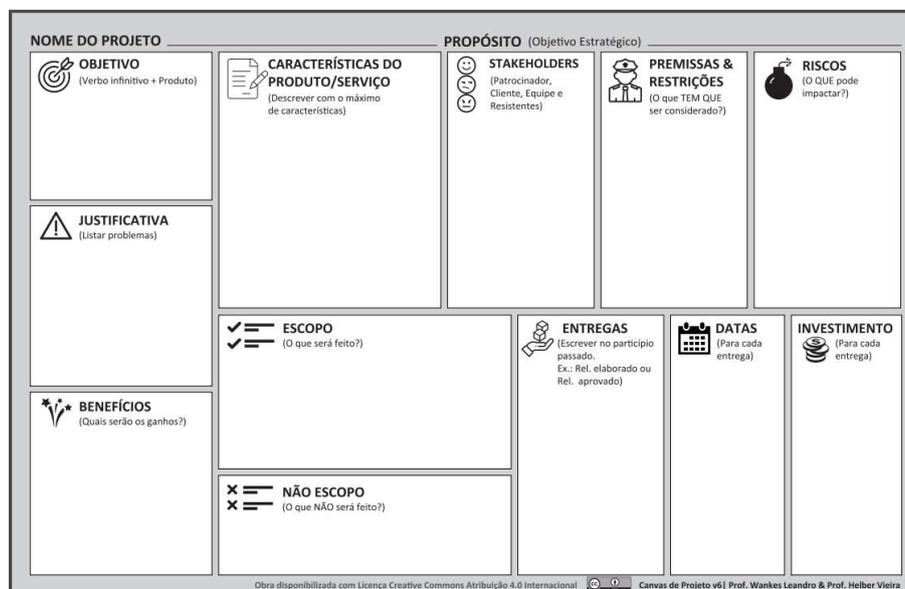


Figura 3. Canvas de Projeto – Canvas Project Design. Fonte: [Leandro and Vieira 2018]

A Figura 3 ilustra os blocos do Canvas Project Design, assim detalhados:

<sup>1</sup> Para maiores detalhes, consulte RASIEL, E. The McKinsey way. McGraw-Hill Companies: 1999.

- **Nome do projeto** – o nome expressa a identidade de um grupo para com o projeto.
- **Propósito** – propósitos são inspiradores e podem listar o objetivo estratégico a que um projeto está vinculado.
- **Objetivo** – o objetivo do projeto deve ser simples, curto, direto e escrito com verbo no infinitivo, seguido pelo produto, serviço ou resultado que se pretende entregar ao final do projeto.
- **Justificativa** – projetos nascem tanto de problemas como de oportunidades, ou de uma combinação de ambos.
- **Benefícios** – os benefícios devem descrever os ganhos que o projeto trará para as partes interessadas quando de sua entrega.
- **Características do Produto / Serviço** – descrever o máximo de características do produto ou serviço é fundamental para alinhar os esforços no escopo do projeto.
- **Escopo** – todo o trabalho que o time do projeto terá que fazer em um projeto para que ele seja entregue.
- **Não Escopo** – declaração do trabalho que não será realizado no âmbito do projeto.
- **Stakeholders** – pessoa (física ou jurídica) que tem algum interesse positivo ou negativo no projeto.
- **Premissas e Restrições** – sem diferenciar conceitualmente premissas de restrições, deve-se explicitar tudo o que tem que ser considerado para que o projeto seja realizado, do ponto de vista do gerente do projeto, dos patrocinadores e demais partes interessadas.
- **Riscos** – identificar o que pode impactar o projeto, com foco principalmente no que pode dar errado e impactar negativamente o projeto.
- **Entregas** – resultados intermediários, que são produzidos ao longo do projeto, devendo ser mensuráveis, como marcos.
- **Datas** – datas nas quais as entregas devem acontecer.
- **Investimento** – estimar o montante que será investido para cada entrega.

A dinâmica de concepção do Canvas Project Design deve ser conduzida em sessão composta das fases de divergência (nas quais se criam opções) e convergência (a partir de escolhas), assim detalhadas [Leandro and Vieira 2018]:

- **Fase 1 – Divergência – Individual** – cada pessoa do Time cola seus próprios *post-its*, individualmente e sem intermediários, no bloco do Canvas que quiser, com o conteúdo que achar pertinente.
- **Fase 2 – Convergência – Coletiva** – quando as contribuições da Fase 1 deixarem de acontecer, o Time começa a interagir, sob a facilitação do Gerente do Projeto, revisando cada *post-it*, refletindo em grupo se a contribuição precisa ser ajustada, consolidada, desdobrada, alterada de bloco (realocada), excluída ou até mesmo gerada nova contribuição.

Ao analisar as diferenças entre a proposta do Canvas Project Design e o PMCanvas, [Leandro and Vieira 2017] assim pontuaram:

**Tabela 1. Diferenças entre abordagens de Canvas**

<b>Bloco do Canvas</b>	<b>Canvas Project Design</b>	<b>PMCanvas</b>	<b>Detalhe da diferenciação</b>
<i>Objetivo</i>	Objetivo FAST <sup>2</sup>	Objetivo SMART <sup>3</sup>	Os objetivos FAST são especialmente importantes para o Canvas Project Design nos momentos de proposição dos objetivos e definição das entregas, com o verbo sendo proposto no [infinitivo] + [produto, serviço ou resultado].
<i>Características do produto/serviço</i>	Características	Produto	O Produto no PMCanvas tem a orientação de indicar o que será entregue, enquanto que no Canvas Project Design este bloco tem a orientação de descrever as características do produto/serviço que será entregue.
<i>Entregas</i>	Entregas	Grupo de entregas	No PMCanvas tem orientação de ser o produto final dividido em partes menores. No Canvas Project Design tem orientação de ser o escopo dividido em partes menores e ser escrito no participio passado.
<i>Partes Interessadas</i>	Stakeholders	Stakeholders externos	O PMCanvas tem a orientação de listar somente as partes interessadas que não sejam a equipe do projeto. O Canvas Project Design tem a orientação de listar todas as partes interessadas, incluindo a equipe do projeto, além listar as partes interessadas negativas/resistentes.
<i>Premissas e Restrições</i>	Premissas e Restrições formam bloco único	Premissas e Restrições são blocos distintos	O PMCanvas aborda as Premissas e as Restrições, as conceituam e as distinguem, devendo ser colocadas em blocos específicos. O Canvas Project Design aborda as Premissas e as Restrições, mas não as conceituam ou distinguem, devendo ser listado o que tem que ser considerado para a existência do projeto.
<i>Atividades</i>	Não lista	Linha do tempo	No PMCanvas, as atividades são dispostas na Linha do tempo. O Canvas Project Design não aborda as atividades.
<i>Datas</i>	Para cada entrega	Não lista	O campo Datas é exclusivo do Canvas Project Design.

Conforme reconhecido pelos criadores, o Canvas Project Design é ideal para conceber projetos, mas peca na ausência de maior detalhamento, sendo aplicável em projetos simples, mas que demandaria o uso de outra ferramenta de gestão, para detalhar o planejamento, como suporte para a etapa de execução [Leandro and Vieira 2018].

Em decorrência dos argumentos dos professores Wankes Leandro Ribeiro e Helber Ricardo Vieira, destacando a simplicidade do modelo do Canvas Project Design, com sua abordagem visual, participativa e colaborativa, além de suas características únicas, combinadas com a experiência didática dos autores, que aborda a aplicação em diversas turmas, tanto do PMCanvas quanto do Canvas Project Design, o Modelo CEK adota o Canvas Project Design para sua aplicação prática. A “lacuna” reconhecida do

<sup>2</sup> quatro atributos: F (Frequentemente discutida), A (Ambiciosa), S (eSpecífica), T (Transparente).

<sup>3</sup> cinco atributos: S (Específico), M (Mensurável), A (Atingível), R (Relevante) e T (Temporal).

Canvas Project Design, referente ao detalhamento do planejamento, é endereçada pela adoção da Estrutura Analítica do Projeto, conforme detalhado na sequência.

#### 1.4. Estrutura Analítica do Projeto (EAP)

O gerenciamento de projetos, como ferramenta de gestão, é uma disciplina gerencial na qual são aplicados conhecimentos, habilidades, ferramentas e técnicas às atividades do projeto, com o objetivo de facilitar o cumprimento de seus requisitos.

O gerenciamento de projetos é realizado por meio da aplicação e da integração apropriadas dos processos de gerenciamento de projetos identificados para o projeto. Os processos de gerenciamento de projetos estão listados no Guia do Conhecimento em Gerenciamento de Projetos – PMBOK 6. O PMBOK 6 baseia-se no Padrão de Gerenciamento de Projetos, do American National Standards Institute (ANSI), sendo uma referência fundamental para os programas de desenvolvimento profissional e a prática do gerenciamento de projetos [PMI 2017].

No PMBOK 6, os processos de gerenciamento de projetos estão organizados em áreas de conhecimento, a seguir listadas e brevemente explicadas, conforme [PMI 2017]:

- **GERENCIAMENTO DA INTEGRAÇÃO DO PROJETO** - contempla os processos e as atividades indispensáveis para identificar, definir, combinar, unificar e coordenar os diversos processos e atividades de gerenciamento de projetos.
- **GERENCIAMENTO DO ESCOPO DO PROJETO** - aborda os processos imprescindíveis para assegurar que o projeto contemple todo o trabalho necessário, e apenas o necessário, para que o mesmo termine com sucesso.
- **GERENCIAMENTO DO CRONOGRAMA DO PROJETO** - abrange os processos indispensáveis para gerenciar o término pontual do projeto.
- **GERENCIAMENTO DOS CUSTOS DO PROJETO** - contempla os processos envolvidos no planejamento, estimativas, orçamentos, financiamentos, gerenciamento e controle dos custos, de modo que o projeto possa ser terminado dentro do orçamento aprovado.
- **GERENCIAMENTO DA QUALIDADE DO PROJETO** - aborda os processos para aplicação da política de qualidade da organização com relação ao planejamento, gerenciamento e controle dos requisitos de qualidade do projeto e do produto para atender às expectativas das partes interessadas.
- **GERENCIAMENTO DOS RECURSOS DO PROJETO** - abrange os processos para identificar, adquirir e gerenciar os recursos necessários para a finalização bem-sucedida do projeto.
- **GERENCIAMENTO DAS COMUNICAÇÕES DO PROJETO** - contempla os processos indispensáveis para asseverar que as informações do projeto sejam planejadas, coletadas, criadas, distribuídas, armazenadas, recuperadas,

gerenciadas, controladas, monitoradas e finalmente organizadas de maneira oportuna e apropriada.

- **GERENCIAMENTO DOS RISCOS DO PROJETO** - aborda os processos de direção do planejamento, identificação e análise do gerenciamento de risco, planejamento de resposta, implementação de resposta e monitoramento de riscos em um projeto.
- **GERENCIAMENTO DAS AQUISIÇÕES DO PROJETO** - abrange os processos imprescindíveis para comprar ou adquirir produtos, serviços ou resultados externos à equipe do projeto.
- **GERENCIAMENTO DAS PARTES INTERESSADAS DO PROJETO** - inclui os processos exigidos para identificar as pessoas, grupos ou organizações que podem impactar ou serem impactados pelo projeto, analisar suas expectativas e seu impacto no projeto, e propor estratégias de gerenciamento adequadas para o seu engajamento eficaz nas decisões e na execução do projeto.

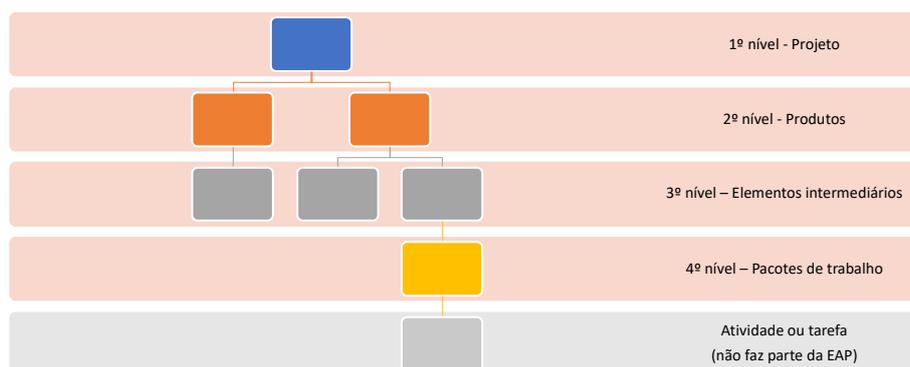
De acordo com [Vargas 2018], o gerenciamento do escopo tem como principal objetivo definir e controlar os trabalhos que serão realizados no âmbito do projeto, de modo a assegurar que o produto, serviço ou resultado que é o objetivo do projeto seja realizado por meio da menor quantidade de trabalho possível, atendendo a todos os requisitos. Nesse sentido, ainda segundo o autor, se não se conhece o escopo, torna-se impossível derivar a estimativa de custo e prazo.

No contexto do escopo, é fundamental estabelecer a diferença entre escopo do produto e escopo do projeto. O PMBOK 6 define escopo do produto como as características e funções que descrevem o produto, serviço ou resultado; já escopo do projeto é o trabalho que precisa ser realizado para a entrega de um produto, serviço ou resultado, atendendo aos seus requisitos [PMI 2017].

No PMBOK 6, a área de conhecimento do gerenciamento do escopo abrange os processos: (1) Planejar o gerenciamento do escopo; (2) Coletar os requisitos; (3) Definir o escopo; (4) Criar a EAP; (5) Validar o escopo; (6) Controlar o escopo.

De acordo com PMBOK 6 (PMI, 2017, p. 156), Criar a Estrutura Analítica do Projeto – EAP, um dos processos de gerenciamento do escopo do projeto, é o processo de “decompor as entregas e o trabalho do projeto em componentes menores e mais facilmente gerenciáveis”. A execução desse processo tem como principal benefício fornecer uma visão estruturada do que deve ser entregue.

A EAP é uma decomposição hierárquica de todo o escopo do projeto (trabalho total) que precisa ser executado pelo time do projeto a fim de produzir as entregas necessárias e satisfazer às necessidades das partes interessadas – cada nível descendente da EAP representa uma definição mais detalhada do nível mais alto correspondente e, ao final, de todo o trabalho do projeto. A Figura 4 apresenta uma estrutura genérica de uma EAP, com seus níveis hierárquicos.



**Figura 4. Estrutura da EAP**

Conforme pode ser observado na Figura 4, o trabalho planejado deve estar contido no pacote de trabalho (4º nível) – o nível mais baixo de componentes da EAP, que deve refletir os produtos de trabalho ou entregas do projeto. O pacote de trabalho pode ser utilizado para agrupar as diversas atividades ou tarefas a partir das quais o trabalho pode ser agendado, estimado, monitorado e controlado. As atividades ou tarefas não fazem parte da EAP.

Ainda com base no ilustrado pela Figura 4, os pacotes de trabalho podem ser agrupados em elementos intermediários (3º nível), também conhecidos como contas de controle; o agrupamento de diversos elementos intermediários irá compor os produtos ou entregas (2º nível) do projeto, que juntos atenderão ao escopo total do projeto (1º nível).

#### 1.4.1. Decomposição

Decomposição é uma das ferramentas e técnicas do processo de Criar a EAP, aplicada para dividir e subdividir o escopo do projeto e suas entregas em partes menores e mais facilmente gerenciáveis, até o nível do pacote de trabalho.

De acordo com o PMBOK [PMI 2017], a decomposição dos componentes do nível mais alto da EAP demanda a subdivisão do trabalho para cada uma das entregas ou subcomponentes em seus componentes básicos, em que os componentes da EAP representam produtos, serviços ou resultados verificáveis.

O processo de decomposição continua de forma recorrente e recursiva até que haja consenso de que os pacotes de trabalho sejam os necessários e os suficientes para a completude das entregas do nível mais alto correspondente. Conforme o [PMI 2017], à medida que o trabalho é decomposto em mais níveis, a capacidade de planejar, gerenciar e controlar aumenta; entretanto, uma decomposição excessiva pode resultar em improdutividade, ineficiência e futura dificuldade de agregação de dados nos diferentes níveis da EAP.

Corroborando com essa busca do equilíbrio na decomposição, [Vargas 2018] afirma ser fundamental trabalhar com um escopo que garanta o produto, serviço ou resultado do projeto, sem ser demasiadamente detalhado, para que seu gerenciamento não se torne complexo.

O nível de decomposição é pautado pelo grau de controle que o time do projeto julga apropriado no caso concreto, para gerenciar o projeto de forma eficaz, podendo variar de acordo com a complexidade e o “tamanho” do projeto. A “árvore” resultante da técnica de composição não precisa ser simétrica – isso significa que os produtos constantes da EAP podem ter diferentes níveis de decomposição.

### 1.4.2. Estimativa Bottom-up

Estimativa *bottom-up* é um método para estimar um componente de trabalho. A duração ou o custo de cada tarefa ou atividade é estimado, com o maior nível de detalhes especificados. As estimativas da duração ou custo do projeto são obtidas pela agregação das estimativas de cada tarefa ou atividade, agrupadas, em uma abordagem de baixo para cima, em seus respectivos pacotes de trabalho; a partir daí, as estimativas dos pacotes de trabalho são agregadas aos componentes de nível mais acima na EAP, até o projeto como um todo.

Quando a duração ou custo de uma tarefa ou atividade não puder ser estimado com um grau razoável de confiança, o trabalho dentro da tarefa ou atividade deve ser decomposto em mais detalhes – até que sejam suficientes. Com isso, as durações ou custos de cada tarefa ou atividade podem ser estimados.

De acordo com o PMBOK 6 [PMI 2017], o método de estimativa *bottom-up* é aplicável aos processos de Estimar as Durações das Atividades, Estimar os custos e Estimar os Recursos das Atividades. Estimar as Durações das Atividades é o processo de estimativa do número de períodos de trabalho (horas, dias, semanas, meses, etc.) que serão necessários para finalizar atividades individuais, com o uso do quantitativo de recursos estimados. Estimar os custos é o processo de desenvolver uma aproximação dos recursos financeiros necessários para finalizar o trabalho do projeto. Estimar os Recursos das Atividades é o processo de estimar recursos da equipe, o tipo e as quantidades de materiais, equipamentos e suprimentos mandatórios para realizar o trabalho do projeto.

### 1.4.3. Agregação

Com as estimativas de todas as atividades ou tarefas (custos e/ou duração), essas estimativas podem ser agregadas em uma quantidade total para cada um dos pacotes de trabalho da EAP e, ainda com o uso da abordagem *bottom-up*, agregadas nos níveis de componentes mais altos da EAP, até contemplar todo o projeto. A Figura 5 representa essa dinâmica.

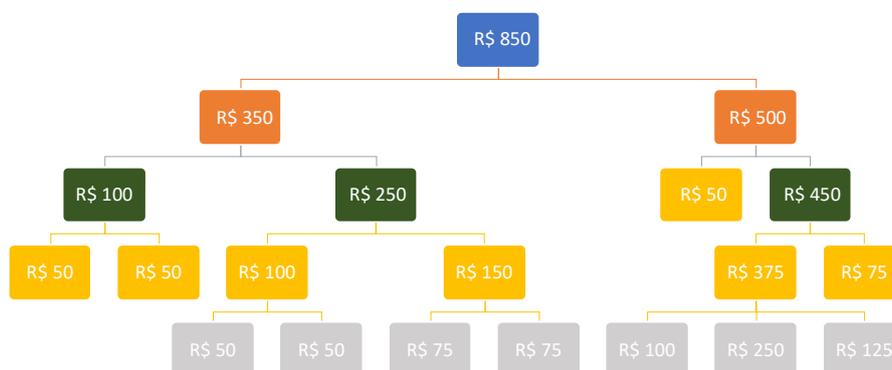


Figura 5 - Exemplo de abordagem bottom-up com agregação de custos

A Figura 5 ilustra a dinâmica da abordagem *bottom-up* sendo definidas as estimativas de custos das atividades (destacadas em cinza) – e que não fazem parte da EAP, sendo demonstradas aqui apenas para fins de entendimento da dinâmica. A partir

das estimativas de custos das atividades, elas são agregadas para os pacotes de trabalho (destacados em laranja). Observe que alguns pacotes de trabalho podem ter seus custos estimados, sem necessidade de decomposição das atividades – quem define quais se encaixam nesse quesito é o próprio time do projeto. Os custos estimados dos pacotes de trabalho são então agregados nos elementos intermediários (destacados em verde), que são agregados nos Produtos (destacados em vermelho) e, por fim, no projeto como um todo (em azul).

De acordo com o PMBOK 6 [PMI 2017], a ferramenta de agregação tem aplicação no processo Determinar o Orçamento, o qual agrega os custos estimados de atividades individuais ou pacotes de trabalho para estabelecer uma linha de base dos custos autorizada.

Em decorrência das aplicações descritas, e de sua abordagem visual e colaborativa, o Modelo CEK adota o processo de criação da Estrutura Analítica do Projeto (EAP), em especial com aplicação de sua ferramenta de Decomposição, como forma de decomposição das Entregas (resultados intermediários) concebidas no âmbito do Canvas Project Design, até o nível do pacote de trabalho, habilitando as estimativas *bottom-up* de duração e de custos. Para conclusão de sua aplicação, com aplicação da Agregação, estas estimativas são sumarizadas nos níveis superiores da EAP, até derivar nos respectivos blocos de Datas e Investimentos do Canvas Project Design.

## **1.5. Kanban**

A essência conceitual dos métodos ágeis é a adaptação, buscando eliminar desperdícios e custos desnecessários. [Agarwal 2018] explica o Kanban como um método visual de gerenciamento de fluxo de trabalho para gerenciamento de maneira efetiva, aplicado com os objetivos de eliminação de desperdício e de organização do trabalho.

A história do Kanban é datada de 1940; [Campbell 2018] descreve que Taiichi Ohno, um empresário japonês e engenheiro que trabalhou para a Toyota, é reconhecido como o “pai” do conceito. Entretanto, para entender o Kanban é necessário retornar às suas origens e falar do sistema Lean.

### **1.5.1. Lean**

Segundo [Lopes and Frota 2015], o Lean tem como objetivo a busca pela eliminação de perdas no processo produtivo. O conceito evoluiu e se transformou numa metodologia com foco em eficiência e eficácia, buscando a redução de custos, utilizando conceitos do ciclo PDCA, para o aumento da produtividade.

O Lean, também conhecido por Sistema Toyota de Produção, surgiu no Japão e alguns de seus princípios nasceram para atender anseios de melhoria de qualidade na indústria para recuperação econômica no pós-guerra.

Na época, os EUA enviaram ao Japão o estatístico William Edwards Deming, que trabalhava na AT&T. Deming ensinou aos líderes de empresas e autoridades japoneses a importância do controle estatístico de processos e do ciclo PDCA. Esses conceitos se tornaram base para a empresa Toyota.

### **1.5.2. Just-In-Time (JIT)**

Nesse contexto surge o Just In Time – JIT, com o objetivo de eliminar qualquer função desnecessária no sistema de manufatura que traga custos indiretos, que não acrescentem

valor para a empresa e que impeça melhor produtividade ou agregue despesas desnecessária no sistema operacional do cliente [Cient et al. 2011].

Na engenharia de software, o desperdício estará presente na construção de funcionalidades de baixo valor agregado ao negócio, perda de tempo com foco no processo de desenvolvimento, ao invés da entrega de software funcional e retrabalho – em resumo, perda de recursos com atividades desnecessárias.

### **1.5.3. Pull System and Push System**

Uma das principais características do modelo Toyota de Produção é a mudança implementada com a ideia de sistema puxado “Pull System”, em substituição ao antes dominante sistema empurrado “Push System”.

No sistema “Push System”, a linha produção funciona basicamente com a sequência de tarefas, com novos itens sendo adicionados assim que um novo produto é concluído; esse tipo de produção gera um grande estoque. Dentro da filosofia Lean, isso é desperdício e deve ser evitado.

Segundo [Heller and Johnson 2010], o Kanban é um sistema puxado, que implementa conceitos do Just In Time (JIT) e do Lean, e funciona, para melhor entendimento, de forma similar a uma impressora, onde o papel é puxado somente quando tem trabalho, ou seja, uma nova impressão é demandada.

Assim surgiu o conceito do Kanban e do “Pull System”, nos quais os itens vão sendo produzidos assim que um novo é entregue (e não apenas concluído). Dessa forma, o desperdício é eliminado ou radicalmente diminuído, pois novos itens só serão produzidos quando tiver uma demanda.

O Kanban foi projetado para visualizar o fluxo de trabalho de uma linha de produção, todavia, esta sistemática pode ser adaptada para qualquer projeto. Quando transposto para a produção de software, seu funcionamento é similar, só que os itens de trabalho são puxados pelos desenvolvedores ou pela equipe de TI, diminuindo os desperdícios.

[Agarwal 2018] discorre que a representação visual de itens de trabalho em um Quadro Kanban permite que os membros da equipe conheçam o estado atual de cada item de trabalho a qualquer momento; ele ainda cita como benefícios da adoção do Kanban os seguintes aspectos:

- aumento da eficiência;
- melhoria no foco;
- menos desperdícios;
- comunicação eficiente;
- maior transparência;
- maior flexibilidade de planejamento;
- menos defeitos;
- entrega contínua;
- ciclos de tempo encurtado.

[Khaled Yacoub et al. 2016] apresentaram em seu estudo o resultado de uma Pesquisa do National Institute of Standards and Technology – NIST, que corrobora com os benefícios elencados anteriormente, em que a implementação efetiva de técnicas Lean como Kanban pode gerar benefícios significativos, como melhoria da qualidade

(ganhos entre 25-75%), redução do trabalho em andamento (até 90%), melhoria da produtividade (entre 10-40%), redução do tempo de espera (até 95%), além de vários outros benefícios relacionados à melhoria do fluxo do produto e aprimoramento do trabalho em equipe e da comunicação.

#### 1.5.4. Implementação do Kanban

Ainda segundo [Khaled Yacoub et al. 2016], o Kanban tem cinco princípios fundamentais: (1) visualizar o fluxo de trabalho; (2) limitar o trabalho em andamento; (3) medir e gerenciar o fluxo; (4) tornar explícitas as políticas do processo; e (5) usar modelos para reconhecer melhorias e oportunidades.

Os princípios Kanban são aplicados usando um Quadro Kanban, que visualiza o fluxo de atividades do processo em várias colunas. Os cartões são usados para cada item de trabalho no Quadro Kanban, para mostrar seu estado atual. A Figura 6 a seguir ilustra um Quadro Kanban.

A FAZER	FAZENDO	FEITO
2.2	3.1	5.2
1.1		
1.3	5.3	5.1
2.1		
2.3		
1.4	1.2	4.2
4.1		

Figura 6. Quadro Kanban

A Figura 6 mostra um exemplo de cartões dispostos em um Quadro Kanban. Os itens ainda não iniciados, que compõem o backlog do projeto, são dispostos na coluna A FAZER; conforme priorizados e com seu trabalho iniciado são movidos para a coluna FAZENDO; uma vez concluídos, são movidos para a coluna FEITO. A sistemática, quando aplicada, garante um sistema puxado e de fluxo contínuo.

Um aspecto importante na metodologia do Kanban é evitar a execução de multitarefas ao longo do processo, conceitualmente dentro da metodologia. Esta característica é chamada de Limite de Trabalho em Andamento (Work in Progress – WIP).

[Agarwal 2018] discorre que o WIP força o Time a priorizar algumas tarefas e as completarem em vez de iniciar novas. Quanto menor o limite de WIP, mais rápido o trabalho fluirá por meio do sistema. Altos limites de WIP podem impedir que a equipe descubra impedimentos no sistema. O autor ainda lista outros benefícios do WIP:

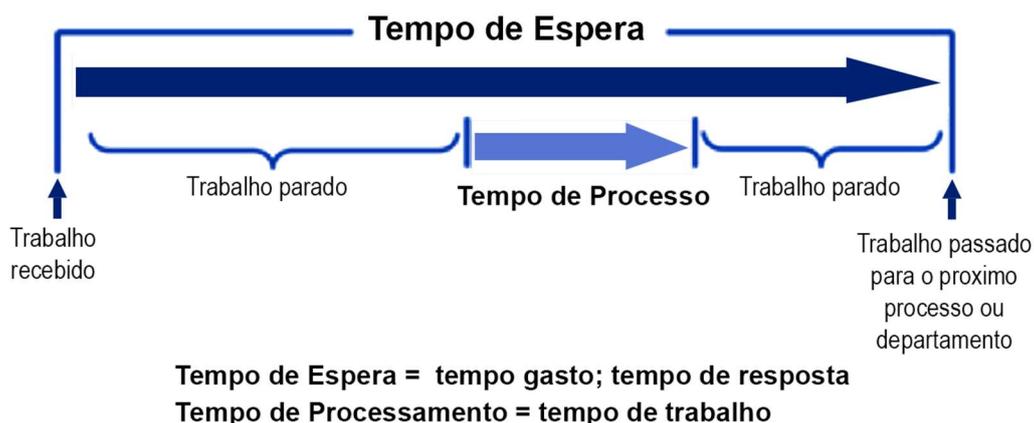
- Melhor qualidade no código;
- Reduz a troca de tarefas;
- Promove a colaboração;
- Aumenta a previsibilidade.

Para concluir a caracterização da implementação do Kanban, [Klipp 2015] destaca e detalha três regras básicas para sua implementação:

1. **Visualize o Fluxo de Trabalho** - Uma representação visual do processo permite que você veja exatamente como as tarefas mudam de “não concluídas” para “feitas corretamente”. Quanto mais complexo é um processo, mais útil e importante é criar um fluxo de trabalho visual. O Kanban pode ser usado se houver apenas algumas etapas (a fazer, fazendo, feito) ou várias etapas (planejar, projetar, rascunhar, aprovar, programar, implementar, testar, integrar, implantar).
2. **Limite de Trabalho em Andamento (WIP)** - Faça mais fazendo menos. Pode parecer contra intuitivo, mas é uma ideia poderosa que foi comprovada verdadeira. Há um limite para o número de coisas que você pode estar trabalhando e ainda fazê-las bem, e esse limite é geralmente menor do que você pensa.
3. **Medir e Melhorar o Fluxo** - A melhoria deve sempre ser baseada em medições objetivas, e o Kanban não é diferente. Encontrar (e aplicar) boas métricas geralmente é uma etapa difícil, mas elas podem fornecer as informações necessárias para otimizar o fluxo e a eficiência.

O tempo total entre o início e fim para desenvolvimento de um item do *backlog* do projeto é chamado de Tempo de Espera (*Lead Time*) – conforme ilustrado na Figura 7. Tempo de Espera é a métrica principal de eficiência do sistema Kanban e após medido, deve ser monitorado com foco na sua redução contínua.

## Tempo de Espera vs. Tempo de Processamento



**Figura 7. Tempo de Espera**

O Modelo CEK adota a sistemática do Quadro Kanban, com os Pacotes de Trabalho da EAP, ainda não priorizados, sendo dispostos na coluna A FAZER (compondo o *backlog* do projeto); os Pacotes de Trabalho já priorizados e que tiveram seu trabalho iniciado (limitados a 1 Pacote de Trabalho por equipe ou responsável – em atenção ao limite do Work In Progress) são movidos para a coluna FAZENDO, até que sejam concluídos e pontuados como FEITO – à medida que os cartões são movidos para

a coluna FEITO, novos espaços são liberados para a coluna FAZENDO, em um sistema puxado e de fluxo contínuo.

O fundamento da adoção do Kanban, que complementa a abordagem proposta pelo Modelo CEK – de transformar ideias em projetos colaborativos, ágeis, controláveis e efetivos – é justamente controlar a execução do projeto por meio do acompanhamento da execução dos pacotes de trabalho, definidos de forma colaborativa na EAP, derivados da concepção idealizada, também de forma colaborativa no Canvas, buscando a redução de desperdícios, a melhoria na comunicação, a agilidade do fluxo contínuo e puxado, e a efetividades dos projetos de software.

## 1.6. Modelo CEK

A gênese da proposição do Modelo CEK foi a experiência didática dos autores, combinada com a atuação profissional e com a vivência de sessões de facilitação de Canvas (tanto o PMCanvas quanto o Canvas Project Design), da gerência de projetos e da utilização de práticas ágeis, em especial Kanban.

O Modelo CEK adota o Canvas Project Design<sup>4</sup>, pela sua abordagem de página única, visual e colaborativa, como ferramenta de gestão para conceber projetos de Engenharia de Software.

Complementarmente, o Modelo CEK adota a ferramenta de gestão, também visual e de dinâmica colaborativa, da Estrutura Analítica do Projeto (EAP), como forma de mitigar a reconhecida deficiência do Canvas Project Design, no tocante ao detalhamento do planejamento de projetos mais complexos. A EAP é aplicada por meio da técnica de Decomposição, como forma de decompor as Entregas concebidas no âmbito do Canvas Project Design, em partes menores, até o nível do pacote de trabalho.

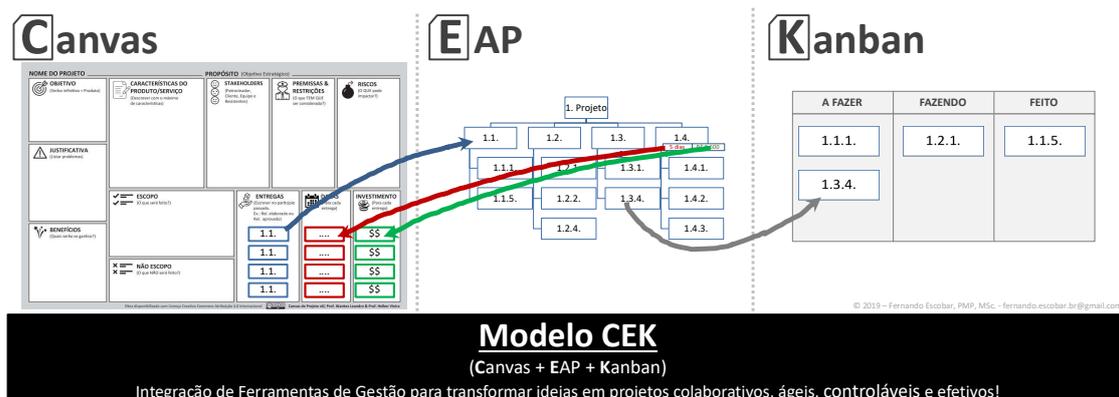
A decomposição nessa granularidade permite ao time estimar duração e custo de cada pacote de trabalho – seja diretamente, seja pela decomposição do pacote de trabalho em atividades. A partir dessas estimativas, com aplicação das técnicas de Agregação e *Bottom-up* será possível agregar duração e custo aos níveis intermediários da EAP e ao projeto como um todo. O resultado desta agregação, em exercício iterativo e incremental, permitirá refinamentos nos valores dos blocos de Datas e Investimentos do Canvas Project Design, relacionados às respectivas Entregas.

Por fim, o Modelo CEK agrega a ferramenta de gestão do Kanban, como complemento visual e ágil, habilitando o monitoramento e acompanhamento, permitindo que os projetos sejam controláveis e efetivos.

O Modelo CEK está esquematizado na Figura 8 a seguir, com destaque para as interações entre as ferramentas, representadas pelas setas:

---

<sup>4</sup> O Canvas Project Design pode ser baixado a partir do site de seus idealizadores: <https://www.canvasdeprojeto.com.br/>



**Figura 8. Modelo CEK**

### 1.6.1. Preenchendo o Canvas

A sessão de preenchimento do Canvas deve ser iniciada com uma exposição prévia, pelo Gerente do Projeto ao Time, acerca de seu conhecimento sobre o projeto – com duração máxima de 5 minutos. Além da exposição prévia, o Gerente do Projeto atua como facilitador da dinâmica do Canvas.

De acordo com “dica” dos criadores do Canvas Project Design, não é necessário (nem recomendável) haver um alinhamento sobre conceitos de gerenciamento de projetos, nem entendimento completo sobre o projeto nesta etapa [Leandro and Vieira 2018].

#### Fase 1 – Divergência

Esta etapa é individual, com cada integrante do Time colocando sua ideia no Canvas. Nesta Fase, o facilitador deve evitar esforços de modo a evitar que opiniões sejam expressas em voz alta, que conversas ocorram entre os integrantes ou que críticas sejam verbalizadas, de modo que o processo criativo e divergente não fique cerceado ou enviesado.

A dinâmica prevê que cada integrante cole, livremente, individualmente e sem intermediários, cada *post-it* com sua contribuição (cada *post-it* deve ter apenas uma contribuição), em qualquer bloco do Canvas – a colaboração com diversos *post-its* (quanto mais melhor) deve ser incentivada.

De acordo com “dica” dos criadores do Canvas Project Design, se eventualmente algum bloco do Canvas ficar em branco (sem contribuições) ao final desta Fase, não tem problema – pois na Fase 2, naturalmente surge conteúdo para preenchê-lo [Leandro and Vieira 2018].

O que se busca nesta Fase é divergir, contemplando uma diversidade de opiniões e pontos de vista, em um processo colaborativo, rico e criativo.

Quando as contribuições começarem a deixar de acontecer, deve-se avançar para a Fase 2.

#### Fase 2 – Convergência

A partir deste momento, o Time começa a interagir, sob a facilitação do Gerente do Projeto, que deve indicar por qual bloco a revisão irá começar.

Durante a interação, a partir de cada bloco do Canvas, cada *post-it* deve ser revisado e debatido com o Time, refletindo em grupo se a contribuição precisa ser ajustada, consolidada, desdobrada, alterada de bloco (realocada), excluída ou até mesmo ser gerada nova contribuição.

A partir da abordagem integrada proposta pelo Modelo CEK, os blocos Datas e Investimentos podem ser abordados na dinâmica do Canvas, mas, necessariamente, precisam ser revistos ao final da decomposição e da agregação *bottom-up* do processo da Estrutura Analítica do Projeto – EAP.

De acordo com “dica” dos criadores do Canvas Project Design, as sessões de Canvas não devem exceder 4 horas – se eventualmente este prazo expirar, sugere-se interromper a sessão e retomar no dia seguinte, evitando a exaustão que pode vir a prejudicar o processo criativo e de colaboração [Leandro and Vieira 2018].

Ao final desta Fase – quando todos os blocos e todos os *post-its* tiverem sido revisados – o projeto estará concebido, pronto para ser decomposto na Estrutura Analítica do Projeto.

A Figura 9 apresenta um Canvas Project Design preenchido para o projeto Solução para Serviço de Moto Táxi, o qual irá exemplificar a aplicação do Modelo CEK neste capítulo.

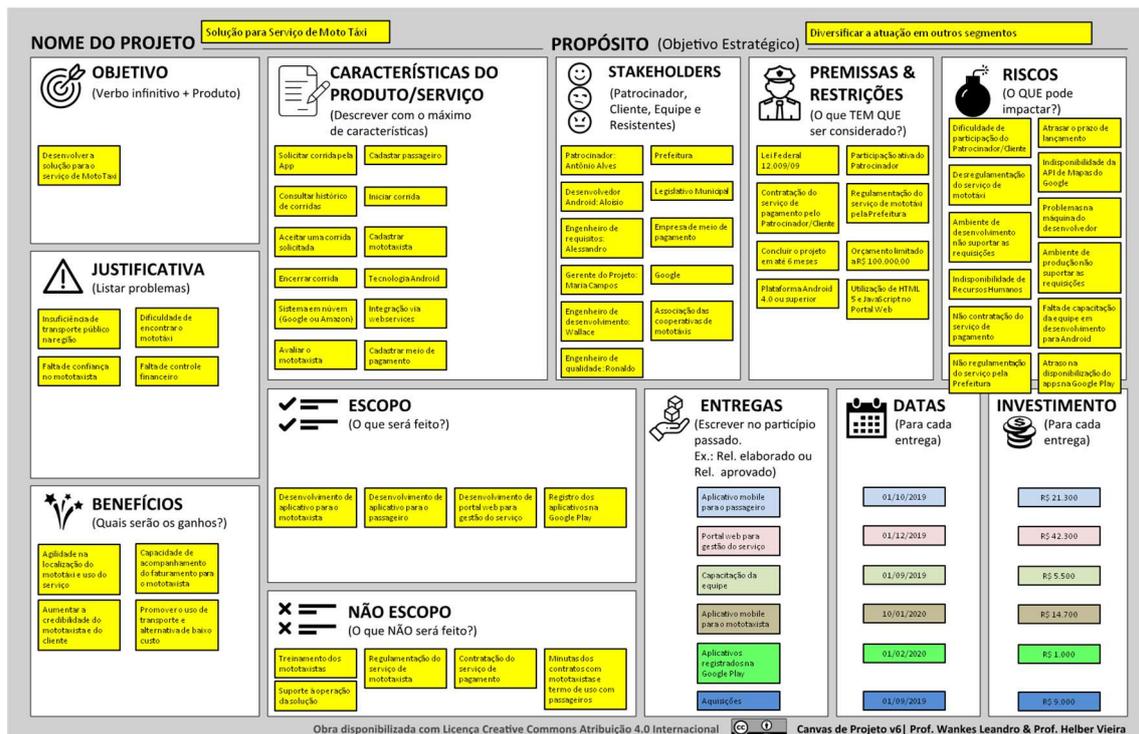


Figura 9. Exemplo de Canvas preenchido para o projeto Moto Táxi

A partir do exemplo do Canvas preenchido na Figura 9, observa-se um esquema de cores para realçar o relacionamento “um para um”, entre as Entregas, suas respectivas Datas e seus respectivos Investimentos. Lembre-se que, após a decomposição da EAP, o Time deve retornar ao Canvas, revisando e atualizando os *post-its* dos blocos Datas e Investimentos.

### 1.6.2. Decompondo a EAP

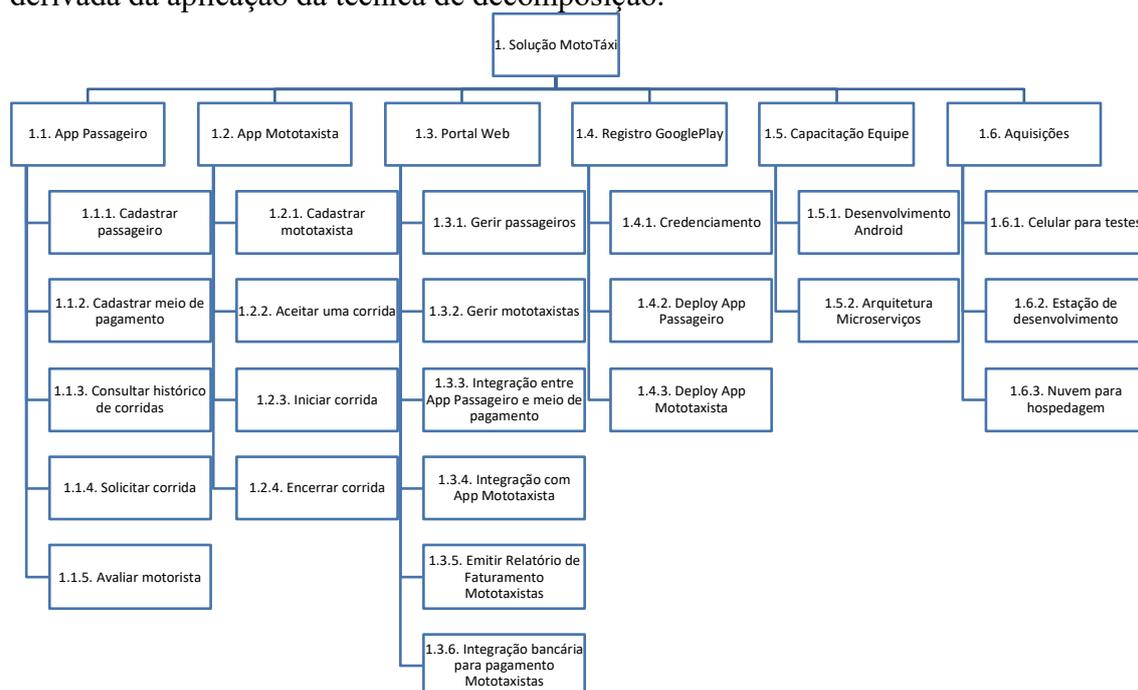
Por ser, também, um processo colaborativo e visual, a recomendação da limitação de janela de tempo (máximo 4 horas), tratada em dinâmica apartada da do Canvas, também se estende ao processo de decomposição da EAP, de modo a não exaurir o Time e prejudicar a qualidade do resultado.

Uma característica específica do processo de decomposição da EAP é a possibilidade de (ou incentivo a) convidar outros integrantes a compor o Time e fornecer a opinião especializada – aquele conhecimento e experiência, de pessoas ou grupos, em projetos similares anteriores ou na execução dos trabalhos previstos no projeto –, necessária para o detalhamento mais realista das estimativas de duração e custo.

Tendo o nome do projeto como 1º nível da EAP e as Entregas conforme concebidas na dinâmica do Canvas como 2º nível, o Time, com a incorporação de novos integrantes especialistas, aplica a técnica da decomposição em uma abordagem descendente (*top-down*), dividindo e subdividindo as Entregas em partes menores e mais facilmente gerenciáveis, até no nível do Pacote de Trabalho.

Uma numeração única que permita identificar cada elemento da EAP, evidenciando a hierarquia, deve ser adotada.

A Figura 10 exemplifica a EAP do projeto Solução para Serviço de Moto Táxi, derivada da aplicação da técnica de decomposição.



**Figura 10. Exemplo de EAP para o projeto Moto Táxi**

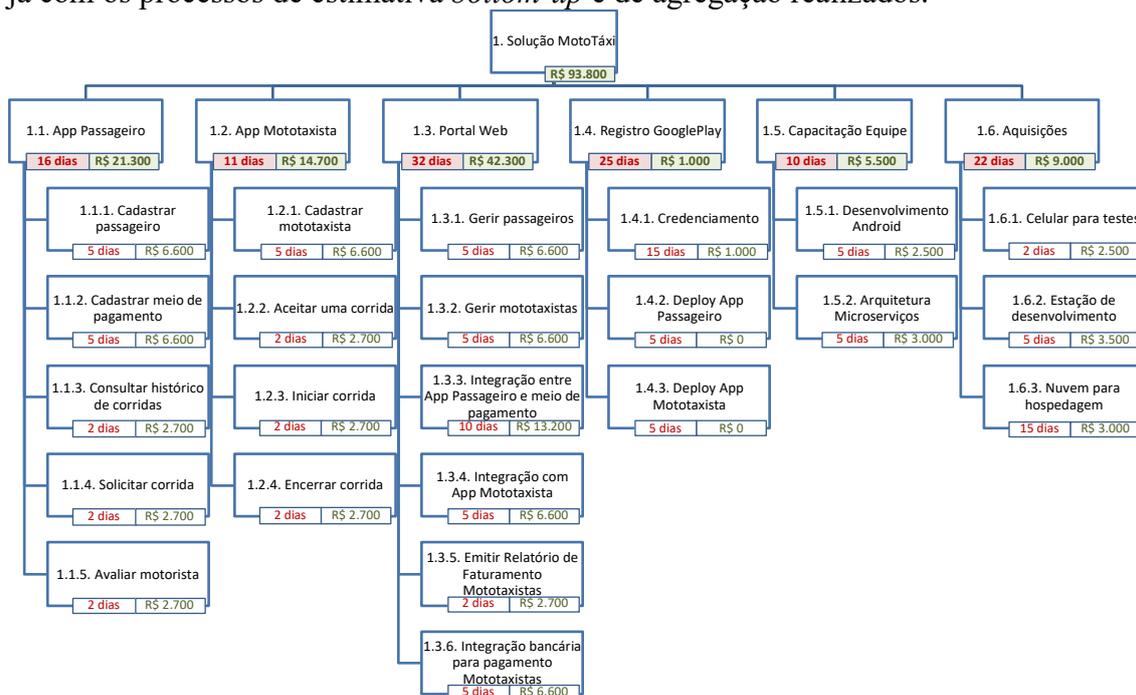
### 1.6.3. Estimativa Bottom-up e Agregação

Concluída a decomposição, o Time parte para estimar o número de períodos de trabalho (duração) que serão necessários para concluir cada Pacote de Trabalho e os custos dos recursos necessários para executar o respectivo trabalho associado ao Pacote de Trabalho. Quando a duração ou o custo não puder ser estimado com um grau razoável de confiança, o trabalho dentro do Pacote de Trabalho deve ser decomposto em mais detalhes.

Uma melhor prática disseminada para o processo de Estimativa é a participação efetiva das pessoas que irão realizar as atividades e produzir os Pacotes de Trabalho. O envolvimento delas nas estimativas aumenta a precisão e favorece o engajamento.

Ao finalizar o processo de estimativa de duração e de custos de cada Pacote de Trabalho, as estimativas devem ser agregadas nos níveis de componentes mais altos da EAP, até contemplar todo o projeto.

A Figura 11 exemplifica a EAP do projeto Solução para Serviço de Moto Táxi, já com os processos de estimativa *bottom-up* e de agregação realizados.



**Figura 11. Exemplo de EAP para o projeto Moto Táxi, com estimativas**

Conforme pode ser observado na Figura 11, a agregação de custos é realizada para todo o projeto, pois fornece uma medida interessante e relevante para seu planejamento e aprovação da execução. Entretanto, a agregação de duração não é necessária para todo o projeto, pois pode haver paralelismo entre alguns trabalhos, não evidenciado na EAP.

O término da construção da EAP com as estimativas deve ser sucedido pela revisão dos blocos Datas e Investimentos do Canvas, harmonizando as estimativas do Canvas com o resultado do processo da EAP.

Com isso, chega-se ao marco do fim da etapa de concepção e de planejamento, com o trabalho do projeto estando decomposto em um nível de detalhe que deixa o Time do projeto confortável com a sua execução e o controle decorrente. Passemos à execução.

#### 1.6.4. Executando e Controlando com o Kanban

Os Pacotes de Trabalho da EAP irão compor o *backlog* do projeto, sendo dispostos na coluna A FAZER do Quadro Kanban. Caso você tenha múltiplas equipes atuando em paralelo, recomendamos a criação de um Quadro Kanban para cada uma.

À medida que os Pacotes de Trabalho são priorizados pelo Time, eles são movidos para a coluna FAZENDO do Quadro Kanban, permanecendo nela enquanto o

trabalho é executado. Fundamentados no conceito do WIP, de que quanto menor o limite de WIP, mais rápido o trabalho fluirá por meio do sistema, no Modelo CEK recomendamos que o WIP seja limitado a um Pacote de Trabalho por equipe ou responsável.

Conforme os trabalhos em andamento são concluídos, os respectivos cartões dos Pacotes de Trabalho devem ser movidos para a coluna FEITO. Esse movimento “abre espaço” para o início de novos trabalhos (a movimentação de novos cartões de Pacotes de Trabalho da coluna A FAZER para a coluna FAZENDO) e habilita o cálculo do *Lead Time*, métrica de eficiência do sistema Kanban.

O monitoramento do *Lead Time* apoiará a melhoria contínua do Modelo CEK como um todo, pois pode evidenciar ineficiências na decomposição, na qualidade das estimativas ou na performance do Time durante a execução.

A Figura 12 exemplifica a aplicação do Quadro Kanban, a partir dos Pacotes de Trabalho derivados da EAP do projeto Solução para Serviço de Moto Táxi (os pacotes de trabalho estão limitados a alguns, apenas por questões de espaço para exibição).

A FAZER	FAZENDO	FEITO
<div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">1.1.1. Cadastrar passageiro</div> <div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">1.1.2. Cadastrar meio de pagamento</div> <div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">1.1.3. Consultar histórico de corridas</div> <div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">1.1.4. Solicitar corrida</div> <div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">1.1.5. Avaliar motorista</div> <div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">1.2.1. Cadastrar mototaxista</div> <div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">1.2.2. Aceitar uma corrida</div> <div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">1.2.3. Iniciar corrida</div> <div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">1.2.4. Encerrar corrida</div>	<div style="border: 1px solid black; padding: 2px; margin-bottom: 10px;">1.5.1. Desenvolvimento Android</div> <div style="border: 1px solid black; padding: 2px;">1.5.2. Arquitetura Microserviços</div>	<div style="border: 1px solid black; padding: 2px; margin-bottom: 10px;">1.6.1. Celular para testes</div> <div style="border: 1px solid black; padding: 2px;">1.6.2. Estação de desenvolvimento</div>

**Figura 12. Quadro Kanban para o projeto Moto Táxi**

Diferentemente das dinâmicas do Canvas e da EAP, com ritos centrados em tempo de duração, o controle da execução com o Quadro Kanban tende a acompanhar o período de duração de todo o projeto, não sendo recomendável nenhuma limitação de janela de tempo.

## 1.7. Conclusões

Neste capítulo foram apresentados elementos de três ferramentas de gestão consolidadas no mercado e na academia, agrupados na proposta identificada como Modelo CEK, fundamentado na abordagem por projetos aplicada à engenharia de software, em um método híbrido que combinou elementos das abordagens tradicional e ágil, buscando mitigar os problemas que historicamente impactam o desenvolvimento de software (complexidade crescente, falha na definição de objetivos e requisitos e na comunicação da equipe com os clientes).

Além da fundamentação teórica, procuramos apresentar um exemplo prático da adoção do Modelo CEK, aplicado a um projeto de uma Solução para Serviço de Moto

Táxi, com a elaboração do Canvas, da decomposição da EAP e a proposição do Quadro Kanban de acompanhamento da execução.

Conforme apresentado, os benefícios da adoção deste modelo estão na riqueza do processo de transformação de ideias em projetos (na concepção habilitada pelo Canvas), no planejamento e nos refinamentos (decorrentes da aplicação da EAP) e no controle dos projetos (com uso do Quadro Kanban), calcados nos princípios do trabalho colaborativo, ágil, evitando desperdícios e buscando a efetividade nos projetos da Engenharia de Software.

## Referências

Agarwal, A. (2018). *The Basics Of Kanban: A Popular Lean Framework*.

Archibald, R. D. and Archibald, S. (2016). *Leading and Managing Innovation: What Every Executive Team Must Know about Project, Program, and Portfolio Management*. Auerbach Publications.

Campbell, A. (2018). *Kanban A complete Step-by-Step Guide*.

Cient, I. I. I. E., Educa, S. I. O. D. E. and Educa, U. (2011). JUST IN TIME: UMA DAS FERRAMENTAS DE OTIMIZAÇÃO DA PRODUÇÃO RESUMO. n. 2001.

Daychoum, M. (2018). *40+ 20 Ferramentas e técnicas de gerenciamento*. Brasport.

De Sousa Neto, M. V. (2014). *Gerenciamento de projetos: project model canvas (PMC)*. Brasport.

Fernandes Vendruscolo, B. M. (2011). A INFLUÊNCIA DE FERRAMENTAS DE GESTÃO ESTRATÉGICA E DE STAKEHOLDERS NO DESEMPENHO DE ORGANIZAÇÕES DO RAMO PARTICULAR DE SAÚDE DO DISTRITO FEDERAL. [http://bdm.unb.br/bitstream/10483/2030/1/2011\\_BrunoMiguelFernandesVendruscolo.pdf](http://bdm.unb.br/bitstream/10483/2030/1/2011_BrunoMiguelFernandesVendruscolo.pdf), [accessed on Aug 7].

Heller, J. R. and Johnson, H. L. (2010). *What are parents Really saying when they talk with their children about sexuality?* v. 5

Júnior, J. F. (2013). *Project model canvas*. Elsevier Brasil.

Khaled Yacoub, M., Abdel Athim Mostafa, M. and Bahaa Farid, A. (2016). A New Approach for Distributed Software Engineering Teams Based on Kanban Method for Reducing Dependency. *Journal of Software*, v. 11, n. 12, p. 1231–1241.

Klipp, P. (2015). Getting started with Kanban. *Methods in Ecology and Evolution*, v. 2, n. 1, p. 1–10.

Larman, C. and Basili, V. R. (2003). Iterative and incremental development: A brief history. *Computer*

Leandro, W. and Vieira, H. (2017). ESTUDO CANVAS DE PROJETOS NO MUNDO: DIFERENÇAS E SIMILARIDADES. [https://docs.wixstatic.com/ugd/b7403b\\_fdae618ef852438ebd9387e82547a920.pdf](https://docs.wixstatic.com/ugd/b7403b_fdae618ef852438ebd9387e82547a920.pdf)

Leandro, W. and Vieira, H. (2018). *Canvas de Projeto - Canvas Project*

*Design*. 1a. ed. São Paulo: Riemma.

Lopes, T. O. and Frota, C. D. (2015). Aplicação dos conceitos do lean manufacturing para melhoria do processo de produção em uma empresa de eletrodomésticos : um estudo de caso. *XXXV Encontro Nacional de Engenharia de Produção*,

Lüftenegger, E., Grefen, P. and Weisleder, C. (2012). The service dominant strategy canvas: Towards networked business models. In *IFIP Advances in Information and Communication Technology*.

Osterwalder, A., Bernarda, G. and Pigneur, Y. (2019). *Value Proposition Design: Como construir propostas de valor inovadoras*. Alta Books Editora.

Osterwalder, A. and Pigneur, Y. (2010). *Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers*.

PMI (2017). *Um Guia do Conhecimento em Gerenciamento de Projetos (Guia PMBOK)*.

Pressman, R. S. and Maxim, B. R. (2016). *Engenharia de Software*.

ProjectBuilder (2017). Tudo o que você precisa saber sobre o PMCanvas. <https://www.projectbuilder.com.br/blog/tudo-o-que-voce-precisa-saber-sobre-o-pm-canvas/>.

Rigby, D. K. and Others (2009). Ferramentas de gestão: um guia para executivos. *São Paulo, Bain & Company*,

Robiolo, G. (2014). Do Agile Methods Increase Productivity and Quality. *American Journal of Software Engineering and Applications*,

Royce, W. W. (1987). Managing the development of large software systems: concepts and techniques. *ICSE '87 Proceedings of the 9th international conference on Software Engineering*,

Sharma, M. and Kotwal, E. (2016). A Concept Note on Execution and Impact of Agile Software Project Management Methods in Midsized it Product Development Companies. *We'Ken- International Journal of Basic and Applied Sciences*,

Shneiderman, B. and Plaisant, C. (2006). Designing the User Interface : Strategies for Effective Human-Computer Interaction (5th Ed.). *Proceedings of the 2006 AVI workshop on BEyond time and errors: novel evaluation methods for information visualization - BELIV '06*,

Silva, S. V., SILVA, L. B. Da, Sales, M. S., Fernandes, F. A. and Sales, P. P. R. (2015). Uma ferramenta para auxiliar a utilização do project model canvas. *Rio de Janeiro*,

Sommerville, I. (2019). *Engenharia de Software*. v. 10

Vargas, R. V. (2018). *Gerenciamento de Projetos 9a edição: estabelecendo diferenciais competitivos*. Brasport.

Vargas, R. V. (2005). Gerenciamento de Projetos, estabelecendo diferenciais competitivos. *Promon Business & Technology Review*,

## Chapter

# 2

## **SCRATCH FOR ARDUINO: Desenvolvendo o pensamento computacional com auxílio da robótica educacional**

Alan Gomes Silva, Ramásio Ferreira de Melo, Rogério Pereira de Sousa, Charles França de Sousa, Elielton Pereira de Santana, Karoline Araújo Nascimento, Jaciara Costa Naziozeno.

### *Abstract*

*The aim of this short course is to make a didactic model with Arduino and related for the development of computational thinking and introductory learning to computer programming. The course will be offered in 4 hours for 15 participants and divided into teams. It is expected that the activity will promote enthusiasm in the participants and can contribute to the significant development of Arduino and programming logic concepts and skills, in order to introduce them into computational thinking with the aid of educational robotics allowing greater concreteness and autonomy of students, preparing them for the challenges of learning and living in today's society.*

### *Resumo*

*O objetivo deste minicurso é a confecção de um modelo didático com Arduino e afins para o desenvolvimento do pensamento computacional e aprendizado introdutório a programação de computadores. O curso será ofertado em 4 horas de duração destinado a 15 participantes sendo divididos em equipes. Espera-se que a atividade promova entusiasmo nos participantes e possa contribuir para o desenvolvimento significativo de conceitos e habilidades de lógica de programação e Arduino, de forma a introduzi-los no pensamento computacional com auxílio da robótica educacional permitindo maior concretude e autonomia dos alunos, preparando-os para os desafios de aprender e viver na sociedade atual.*

## 1.1. Introdução

Papert Seymour considerado uma referência sobre o uso de computadores na Educação desenvolveu a Linguagem Logo que possibilitou a aplicação da Informática na Educação, defendendo sobre como uma ideia pode habitar uma mente jovem e as influências do computador nessa perspectiva (PAPERT, 1980).

O uso das Tecnologias Digitais de Informação e Comunicação (TDIC) na Educação Básica é abordado de diversas maneiras com o intuito de instigar o letramento digital, inclusão da disciplina no currículo e permitir explorar conceitos do Pensamento Computacional (PC) em jogos e robótica de maneira transversal (VALENTE, 2016).

O PC é um modo de pensar nas formulações de problemas e suas soluções, que devem ser representadas de forma generalizada e executadas por agentes de processamentos distintos (CUNY; SNYDER; WING, 2010). O PC tem como base a busca por maneiras alternativas para resolver problemas reais, compreender o comportamento humano e projetar sistemas embasando-se nos conceitos da Ciência da Computação (WING, 2006).

São princípios norteadores do PC definidos pelas organizações International Society for Technology in Education (ISTE) e America Computer Science Teachers Association (CSTA): coleta de dados, análise de dados, representação de dados, decomposição de problema, abstração, algoritmos, automação, paralelização e simulação (VALENTE, 2016).

Além disso, integrar o PC e algoritmos nas escolas pode contribuir para que os alunos adquiram habilidades e competências fundamentais para o convívio em uma sociedade tecnológica e globalizada (Brasil, 2012). Ainda pouco explorado na educação básica pública brasileira, o PC pode ser uma excelente abordagem para desenvolvimento lógico matemático.

Geralmente, as ações de integração das tecnologias nas escolas são voltadas para informática básica e utilização dos softwares de escritório. Os conhecimentos adquiridos por meio destes tipos de integração são defasados para a sociedade contemporânea, de modo a limitar o crescimento cognitivo do indivíduo com base na teoria do construcionismo de Papert (VALENTE, 2016; ARANTES; RIBEIRO, 2017).

Há trabalhos sobre o uso da ferramenta Scratch para o desenvolvimento do PC dentre eles estão (França; Amaral, 2013; Ramos; Teixeira, 2015; Rodriguez et al., 2015), sua utilização proporciona fácil manuseio por pessoas iniciantes na programação de maneira a substituir as linhas de código por blocos, porém, o uso do Scratch for Arduino (S4A) vai além, por permitir a prototipagem e desenvolvimento de projetos na plataforma Arduino a partir da interação do aluno em tempo real com dispositivo físico (KALIL et al., 2013).

O Scratch For Arduino (S4A) foi desenvolvido no Citilab pelo Edutec Research Group, com a ajuda do grupo de programação Smalltalk.cat e de colaboradores como Jorge Gómez e atualmente está na versão 1.3 (S4A, 2019). Oriundo do Scratch, o S4A é um ambiente para se programar de forma simplificada a plataforma Arduino. Sua interface é similar à do Scratch, exceto por adicionar blocos de ações para gerenciamento de sensores e atuadores (Figura 1).

O S4A viabiliza a interação com a plataforma Arduino e permite maior usabilidade, substituindo sua IDE padrão baseada em Processing. Arduino é uma plataforma de prototipação, sendo uma placa física de distribuição livre. Permite conectar diversos sensores e atuadores como por exemplo, sensor de temperatura, luminosidade, umidade, distância e outros como LED, motores, Bluetooth, etc. (KALIL et al., 2013; ARDUINO, 2019).

O objetivo deste minicurso é a confecção de um modelo didático com Arduino e afins para o desenvolvimento do PC e aprendizado introdutório a programação de computadores.

Este capítulo está estruturado da seguinte forma: na Seção 2 os trabalhos que apresentam-se similares a este; na Seção 3 aborda sua relevância, na Seção 4 relata-se de forma detalhada as ações para a realização do minicurso; a Seção 5 traz os resultados esperados de forma a esclarecer o que os benefícios do minicurso; e por fim, os agradecimentos.



Figura 1: Comparativo entre códigos para piscar um LED, no Arduino e no S4A.  
Fonte: Página do arduinoendo 2019.



Figure 2: Interface completa, simples e intuitiva.

## 1.2. Trabalhos Relacionados

Serão abordados nesta seção, trabalhos e autores que de alguma forma contribuíram para o amadurecimento desta proposta de minicurso elencando os casos de sucesso e insucesso de experiências semelhantes.

Kalil et al. (2013), relatam uma experiência sobre a robótica educacional no ensino médio analisando quais ferramentas se apresentam mais adequadas para desenvolver a lógica e compreensão do mundo real: Arduino, S4A e Lego. Eles ressaltam a importância do ensino e aprendizagem de programação para que a sociedade esteja mais inserida ao universo da computação. Conclusivamente, apresentam o S4A como ferramenta mais adequada, devido sua interface, sintaxe, opção de escolha de idioma e baixo custo.

Sobreira, Takinami e Santos (2013) abordam sobre a relevância de trabalhar com tecnologias no Ensino Fundamental, objetivando formar indivíduos participativos e atuantes no século XXI, dentre os vários meios e ferramentas utilizáveis, o S4A é apontado como o aspecto pertinente até mesmo por ser inclusivo possuindo vários meios sensoriais, como: música, animação, cores, imagens, etc. Os autores concretizam que a linguagem de programação scratch mostrou resultados pragmáticos por aprimorar habilidades em contato com as múltiplas linguagens. Dessa forma, favorecem a criticidade por parte dos alunos, além de criatividade, curiosidade e reflexão de forma construtiva resolvendo problemas que exijam soluções e posturas autônomas para seu desenvolvimento.

Zanetti e Oliveira (2015) propõem uma prática pedagógica com a inserção da robótica educacional e do PC para introdução do ensino de programação em um curso técnico de informática integrado ao ensino médio. Foi trabalhado com três conceitos de programação: entradas e saídas, estrutura condicional e de repetição, para desviar o robô de um obstáculo utilizando o S4A e componentes do Arduino. As resultâncias do trabalho mostram que a prática abordada torna um ambiente de aprendizagem menos abstrato para o aluno, que ele venha a ser mais ativo na construção do conhecimento e do raciocínio lógico.

Albuquerque et al. (2016) relatam um experimento com alunos do curso técnico em mecatrônica integrado ao ensino médio. Utilizou-se para o ensino de programação a plataforma Arduino e o PC, de modo que eles construíssem em duplas um projeto de eletrônica do seu cotidiano. Constatou-se que os alunos não tiveram grandes dificuldades em trabalhar com o Arduino, a não ser, devido ao idioma inglês, destaca-se que a metodologia do experimento é mais apropriada do que a metodologia tradicional por apresentar uma aprendizagem menos abstrata, e foi possível desenvolver o PC por relacionar um problema da vida real sendo resolvido pelos alunos por intermédio dos conceitos de programação e do Arduino.

Pereira e Franco (2018) associaram o Scratch ao Arduino para desenvolver o PC com alunos do 7º, 8º e 9º ano do ensino fundamental. Eles abordaram conteúdos de introdução a programação como: hardware e software, lógica e algoritmo, utilizam o S4A para desenvolver projetos que foram apresentados em uma feira de ciências pelos alunos. Os resultados elencados demonstram que foi possível relacionar os conhecimentos abstratos aos concretos, com a verificação dos códigos agindo sobre o hardware e os desafios impostos aos alunos induzindo-os a realizarem buscas de soluções, assim, influenciando no modo de aprendizagem dos alunos e tornando PC mais aparente.

### **1.3. Justificativa**

Inserir o PC na escola possibilita que o aluno desenvolva habilidades que contribuam para o melhoramento do raciocínio lógico e matemático, que permitem a solução de problemas

de forma transversal nas mais diversas áreas do conhecimento (SCAICO, 2013; BRASIL, 2012), contribuindo para a formação de alunos mais críticos, não apenas meros consumidores de tecnologias, mas sim, produtores de soluções tecnológicas para problemas reais do cotidiano (SANTOS, 2016; SCOLARI, 2007).

Por meio das TDIC é possível realizar o desenvolvimento de conceitos relacionados ao PC, como exemplo, apresentar problemas e situações, analisando as contribuições da tecnologia e as possibilidades que ela nos permite (VALENTE, 2016).

Com isso, a utilização do software S4A possibilita boa interação e é de fácil manuseio para pessoas iniciantes na programação, de maneira a substituir as linhas de código por blocos, onde são organizados de maneira a formar instruções para o Arduino, ao invés de fazer a utilização da sua IDE convencional (KALIL et al., 2013). O S4A permite ainda que o aluno possa visualizar o funcionamento de seu código agindo no dispositivo físico, que no caso seria o Arduino.

Falar em PC na educação, imagina-se logo em computadores nas escolas. O desenvolvimento do PC ultrapassa esta ideia do uso prescrito do computador, especialmente pela utilização das técnicas de computação desplugada.

Santos, et al. (2016) afirma que com essas técnicas é possível estimular o raciocínio e o PC e, que além disso, contribui para a criação de ferramentas que possibilitem uma melhor resolução de problemas. Sendo assim, escolas que não possuem este recurso, também podem desenvolvê-lo a partir de atividades que irão refletir diretamente na maneira de resolver problemas.

A lógica de programação possibilita ao aluno aprimorar suas habilidades, ao estruturar um problema o indivíduo estará segmentando-o para resolvê-lo de forma simplória (BARBOSA, 2017).

Dessa forma, o indivíduo fica a par de conceitos que subsidiam a resolução de problemas e adquire habilidades para estabelecer variáveis, aplicar estrutura condicional de repetição e lógica de programação em que representam uma forma introdutória fundamental na estruturação de instruções para resolução de problemas, sendo estes conhecimentos essenciais para o desenvolvimento do PC (ZANETTI; OLIVEIRA, 2015).

Scolari (2007) coloca que inserir conceitos de Computação pode desenvolver o raciocínio lógico e matemático permitindo tornar os alunos mais críticos nas diferentes disciplinas, e Santos, et al. (2016) afirma ainda, que os indivíduos tendem a ser mais que apenas meros consumidores de tecnologias, mas sim, produtores.

O interesse em realizar este curso surgiu a partir de experiências tidas em trabalhos já realizados nesta área, tendo este, bons resultados, sendo apresentado em escolas e na semana tecnológica em praça pública.

#### **1.4. Metodologia de Execução**

O curso será ofertado a quinze (15) participantes, com faixa etária não determinada. Para os participantes, é necessário que possuam conhecimentos em Computação, noções de lógica de programação, algoritmos e/ou áreas afins. Sua duração será de 4 horas com os seguintes conteúdos programáticos:

Apresentação do PC;

- Apresentação das ferramentas;
- Introdução a programação em blocos;
- Desenvolvimento dos princípios do PC;

### 1.4.1. Discriminação das fases do curso

O curso de S4A será aplicado e distribuído nas seguintes fases:

**1º fase:** apresentação do PC, da plataforma S4A e do Arduino. Esta fase será um momento dialogado por intermédio de recursos audiovisuais para exposição dos conceitos, relevância e princípios básicos do PC, que são: definição, organização, abstração e solução do problema.

**2º fase:** apresentação de mini-projetos utilizando a plataforma de programação em blocos S4A para facilitar o entendimento dos alunos sobre lógica de programação e explorar o funcionamento da plataforma de prototipagem Arduino no gerenciamento dos sensores/atuadores.

- Botão e LED: Um simples exemplo mostrando como acionar um LED apertando um botão.

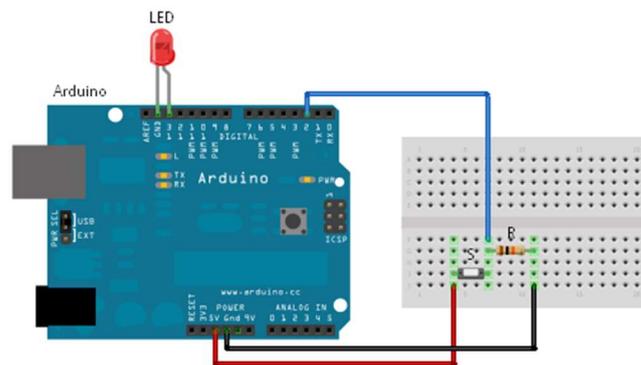


Figura 3: Modelagem da prototipação botão e LED. Fonte: S4A 2019.

- LED RGB: Utilizando um LED RGB.

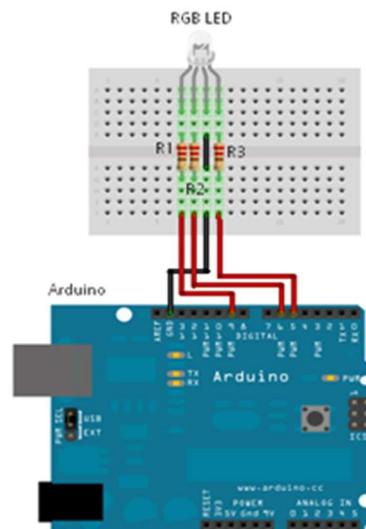


Figura 4: Modelagem da prototipação LED RGB. Fonte: S4A 2019.

- Botão, LED e mantendo estado. Um LED que muda de estado quando aperta um botão.

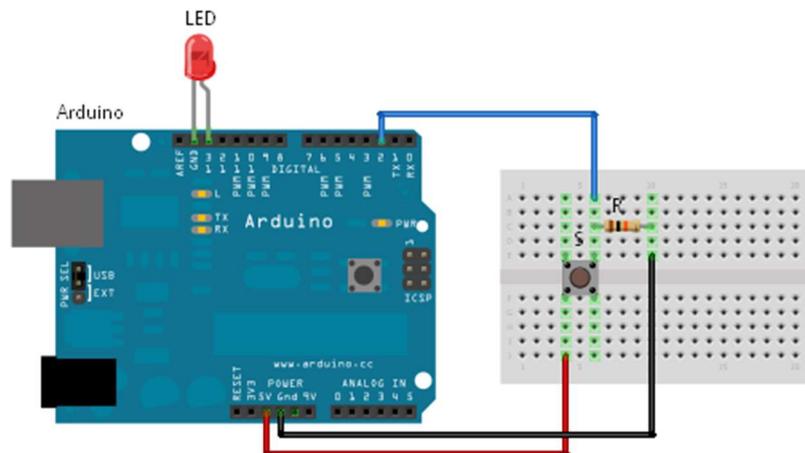


Figura 5: Modelagem da prototipagem botão e LED. Fonte: S4A 2019.

- Fotorresistor e LED. Utilizando a entrada de um sensor fotorresistor para acionar um LED.

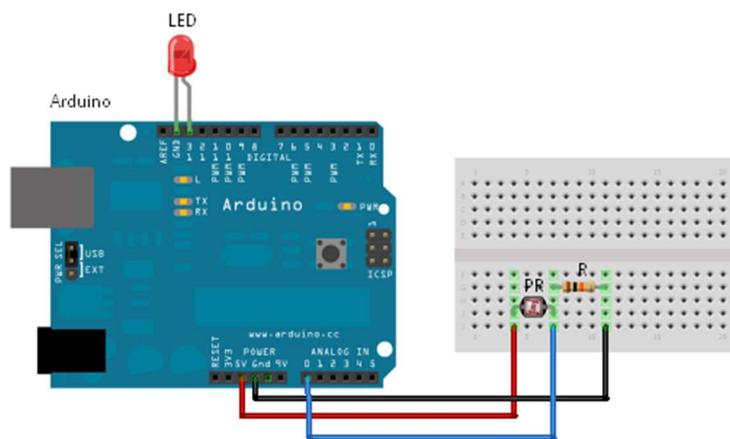


Figura 6: Modelagem da prototipagem fotorresistor e LED. Fonte: S4A 2019.

**3º fase:** O modelo didático selecionado consiste em um protótipo para gerenciar o estacionamento. Será fragmentado em 4 partes e distribuídas em: (a) carro, (b) semáforo, (c) controle de acesso e (d) controle de vagas (figuras 7 e 8). Os participantes serão divididos em grupos.

Cada grupo ficará responsável pela construção de uma das partes destacadas anteriormente seguindo os princípios do PC que estão descritos abaixo:

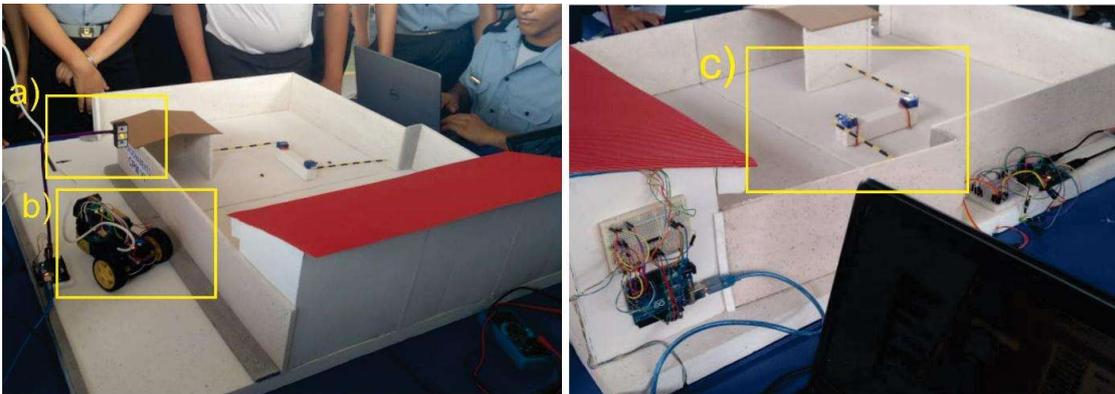
(I) definição: Segue-se o problema geral: Como seria possível minimizar a onerosidade de um estacionamento em pleno século XXI?

- Como guiar um carro por meio computacional?
- Como permitir um melhor fluxo no trânsito próximo ao estacionamento?
- Como controlar o acesso ao estacionamento de forma autônoma?
- Como fazer um controle de vagas indicando qual vaga está disponível?

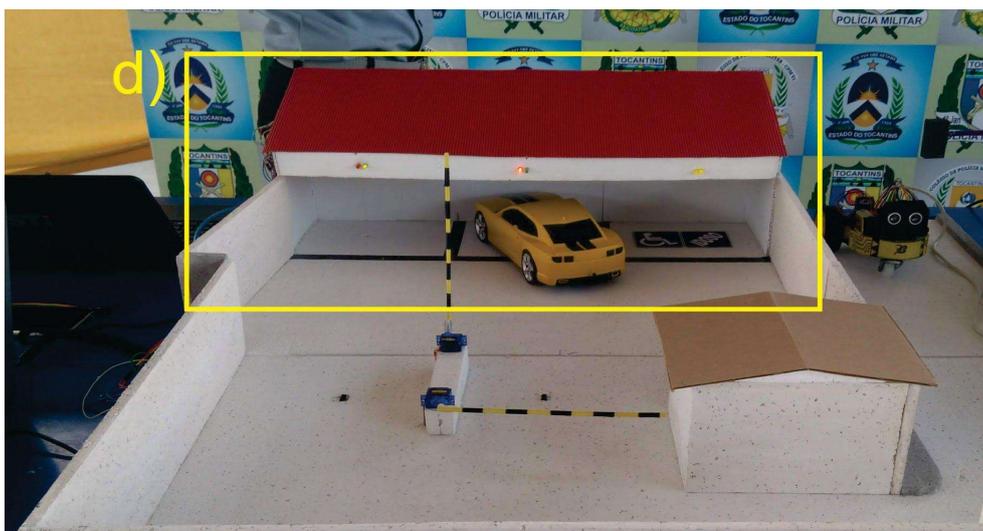
(II) organização: Visualizar e delimitar a problemática de modo a saber quais recursos e tempo levará para desenvolvimento da solução.

(III) abstração: fragmentação de um problema maior em partes menores, permitindo assim, a definição de uma sequência de passos lógicos para se chegar a solução.

(IV) solução do problema: destacado nas figuras a seguir (figuras 7 e 8).



**Figura 7: Apresentação dos protótipos em trabalhos passados.**



**Figura 8: Apresentação dos protótipos em trabalhos passados.**

**4º fase:** caracteriza-se pelo desenvolvimento do modelo didático com base no que foi definido na fase anterior. (a) carro, (b) semáforo, (c) controle de acesso e (d) controle de vagas. Os participantes deverão desenvolver a lógica de programação para o funcionamento do estacionamento automatizado, aplicando-se, o que foi apresentado no minicurso. Como foi dividido as partes precisarão fazer um trabalho colaborativo, proporcionando o conhecimento das partes aos colegas por meio da socialização e discussão do projeto.

A seguir, na figura 9 apresenta-se o código para condução do carro no estacionamento.

```

quando tecla seta para baixo pressionada
  repita 1
    analog 9 value 30
    espere 1 segundos
    analog 9 value 70
    espere 1 segundos
    analog 9 value 0
  fim

quando tecla b pressionada
  analog 9 value 40
  espere 1 segundos
  analog 9 value 0

quando tecla espaço pressionada
  digital 13 off
  digital 12 off
  analog 6 value 0
  digital 11 off
  digital 10 off
  analog 5 value 0
  diga PARE por 1 segundos

quando tecla seta para a esquerda pressionada
  digital 12 off
  digital 13 on
  analog 6 value 110
  digital 10 on
  digital 11 off
  analog 5 value 110
  diga ESQUERDA por 1 segundos

quando tecla seta para a direita pressionada
  digital 10 off
  digital 11 on
  analog 6 value 110
  digital 12 on
  digital 13 off
  analog 5 value 110
  diga DIREITA por 1 segundos

quando tecla seta acima pressionada
  digital 13 off
  digital 12 on
  analog 6 value 110
  digital 10 on
  digital 11 off
  analog 5 value 110
  diga FRENTE por 1 segundos

quando tecla seta para baixo pressionada
  digital 12 off
  digital 13 on
  analog 6 value 110
  digital 11 on
  digital 10 off
  analog 5 value 110
  diga RÉ por 1 segundos
  
```

Figura 9: controle do carro.

Na figura 10 é apresentado o código para o semáforo.

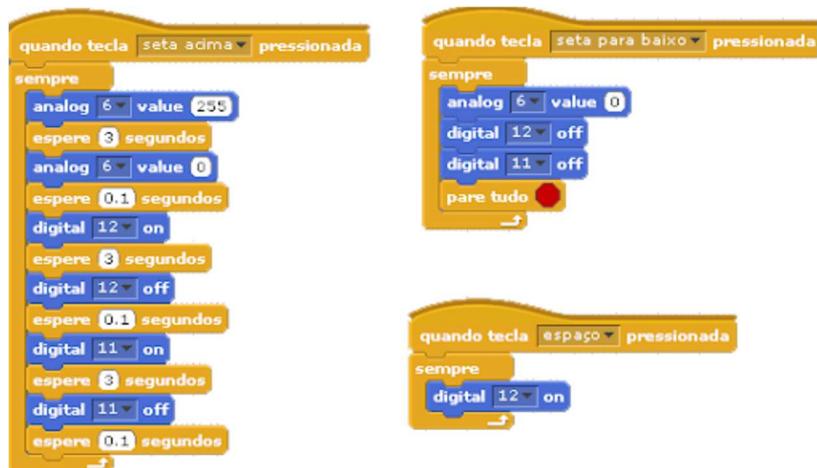


Figura 10: semáforo.

A figura 11 apresenta-se o código para o controle de acesso ao estacionamento

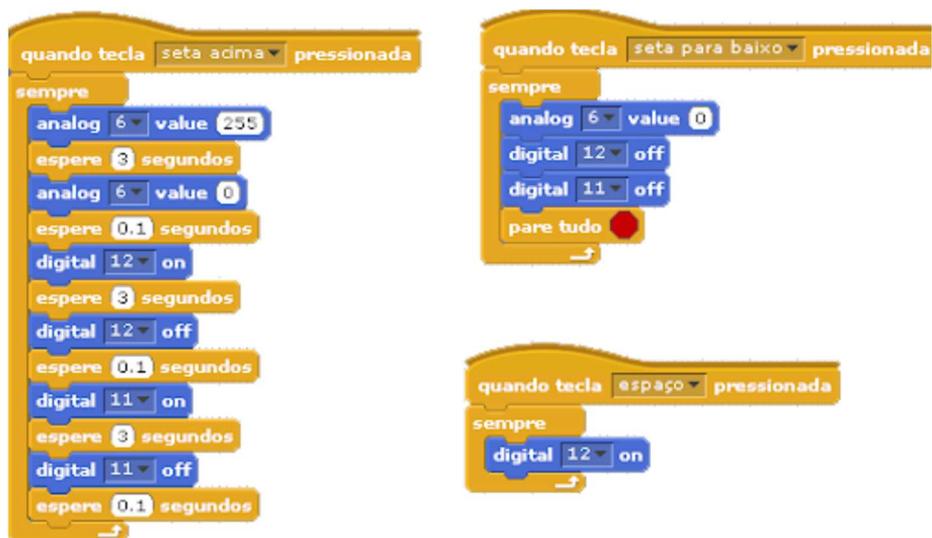


Figura 11: controle de acesso.

A figura 12 mostra o código para o controle das vagas no estacionamento.

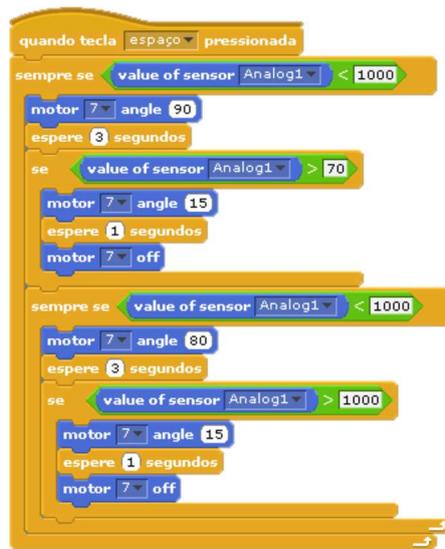


Figura 12: controle de vagas

## 1.4.2. Especificações dos materiais

Neste capítulo será tratado a fundamentação teórica que será utilizada no processo de desenvolvimento do minicurso a ser desenvolvido. Desta forma, faz-se necessário o estudo de alguns conceitos, como sensores e ferramentas, tais como Arduino uno, sensor LDR, LED, sensor ultrassônico, motores DC, ponte h, motores servos, sensores de linha.

### 1.4.2.1. Arduino

O Arduino é uma plataforma *open source* de prototipagem com hardware e software flexível. Vale ressaltar que o Arduino possui um código fonte aberto, dispõe um próprio ambiente de desenvolvimento que utiliza a linguagem C. É uma poderosa ferramenta podendo ser utilizada em inúmeros projetos e/ou diversas áreas, tal pelo motivo do seu custo ser bem baixo comparado com outras placas de prototipagem, atualmente uma unidade está custando no valor de 30,00 reais.

No minicurso, o Arduino uno r3 será utilizado em várias etapas, tais como, controle de vagas no estacionamento, controle de acesso de entrada e de saída. O r3 irá controlar motores servos que serão acionados por meio de sensor de presença e um simulador de semáforo, assim como o carrinho, que ao detectar algum objeto, automaticamente aciona o *buzzer*, fazendo um pequeno som.

Uma placa Arduino Uno conta com 01 microcontrolador ATmega328P (Fabricado pela empresa Atmel), 14 entradas e saídas digitais, 6 entradas analógicas, 1 oscilador de cristal de 16 MHz e pinos de alimentação com 3,3 V 5 V Terra (GND). (DE LIMA, 2013, p.3).

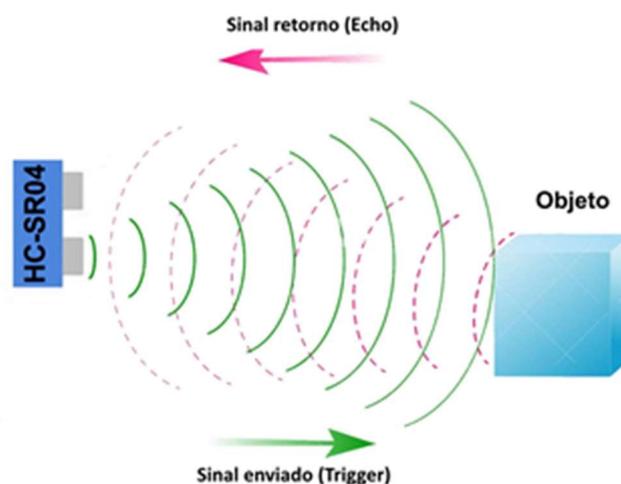


**Figura 13: Arduino uno R3.**  
**Fonte: arduoeleetro 2019.**

#### 1.4.2.2. Ultrassônico

Ultrassônico é um sensor aplicado na área da robótica, utilizado para, por exemplo: desviar um robô de obstáculos e acionamento de alarmes. Este sensor pode ser controlado pela microcontroladora Arduino, *Raspberry*, etc. O ultrassônico faz leituras de 2 centímetros a 4 metros, com precisão de 3 mm de margens de erro. No minicurso, ele será utilizado no carrinho para desvio de obstáculo e seu custo é de R\$10,00 a unidade.

O sensor ultrassônico é composto por um emissor e um receptor ultrassônico. O seu funcionamento acontece quando o emissor emite o sinal ultrassônico no momento em que encontra algum objeto e retorna esta informação ao sensor. Com base no tempo que o sinal ultrassônico emitido leva para retornar ao sensor, o mesmo já efetiva o cálculo da distância do sensor com o objeto (FELIPEFLOP, 2019).

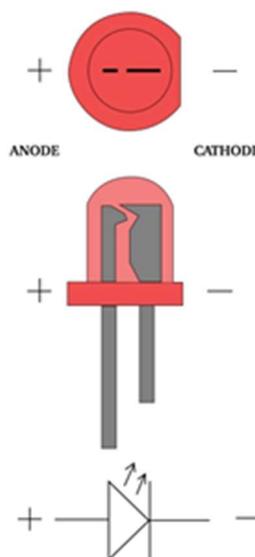


**Figura 14: Sensor Ultrassônico.**  
**Fonte: filipeflop 2019.**

#### 1.4.2.3. Led

Led é um diodo de emissão de luz, caracterizado pelo componente que conduz corrente elétrica, que ao passar pelo seu cristal, caracterizam-se as cores vermelha, verde, amarelo, etc.

Este LED possui polaridades e deve ser ligado corretamente com seu terminal maior anodo sendo positivo e seu terminal menor catodo, sendo o negativo, como mostra na figura abaixo. Este diodo de emissão de luz será utilizado na vaga de estacionamento para informar se a vaga está livre ou não, no semáforo nas cores verde, amarelo e vermelho. Uma unidade custa em média R\$2,00 (FELIPEFLOP, 2019).

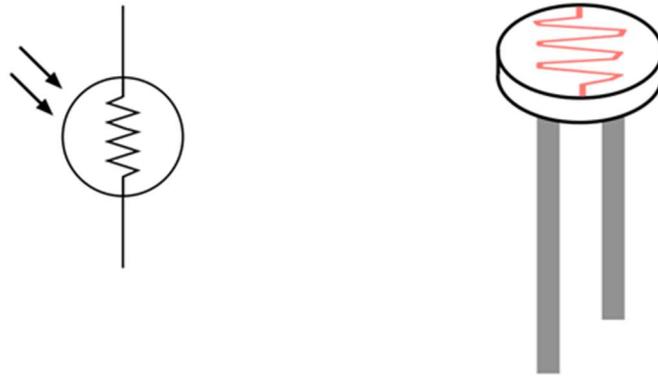


**Figura 15: LED.**  
**Fonte: filipeflop 2019.**

#### **1.4.2.4. LDR**

O LDR é um resistor dependente de luz. Neste resistor, quanto maior a luminosidade, menor será sua resistência. O LDR não tem polaridade, isto é, pode ser ligado em positivo ou negativo e contém de um semicondutor de alta resistência. Para detectar mudanças, este resistor utiliza 10ms e seu custo equivale a menos de R\$1,00 (FELIPEFLOP, 2019).

No minicurso, o sensor LDR será utilizado no controle de vagas, para detectar quando um veículo estiver em uma determinada vaga, informando através de um LED – diodo de emissão de luz, se a vaga está ocupada ou não.



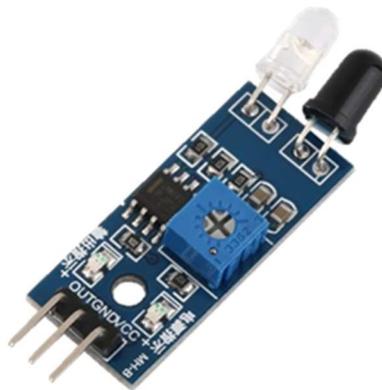
**Figura 16: LDR.**

**Fonte: blogs.plymouth.ac.uk 2019.**

#### **1.4.2.5. TCRT5000**

O sensor TCRT5000 conhecido como sensor de seguidor de linha, muito utilizado na área da robótica industrial e educacional, é composto de dois diodos de emissão de luz, sendo um emissor e outro receptor de sinal infravermelho. O seu funcionamento acontece quando o emissor emite luz infravermelho e, se este não retornar em intensidade suficiente, o mesmo tem seu funcionamento interrompido.

No minicurso, ele será utilizado no controle de acesso à garagem. Quando o veículo chegar próximo a cancela, o sensor TCRT5000 irá identificá-lo, enviando um sinal para Arduino, que irá acionar o motor servo para abertura do controle de acesso a garagem. Este sensor possui polaridades como mostra na figura abaixo, e seu custo é, em média R\$6,00 (FELIPEFLOP, 2019).



**Figura 17: Sensor TCRT5000.**

**Fonte: Baú da eletrônica 2019.**

#### **1.4.2.6. Rodas**

A roda boba, também conhecida como rodízio giratório de 30 mm, será utilizada no minicurso descrito neste trabalho, especificamente no carrinho que é controlado pelo computador. Bem como esta roda boba, existe a roda de 68mm de coloração amarela, também muito utilizada em projetos, como de carrinho autônomo e não autônomo. No minicurso será utilizado o carrinho, que será controlado pelo computador, como mostra na imagem abaixo, sendo do lado esquerdo a roda boba e do lado direito, a roda 68mm, cada uma com o mesmo valor de R\$6,00 (FELIPEFLOP, 2019).

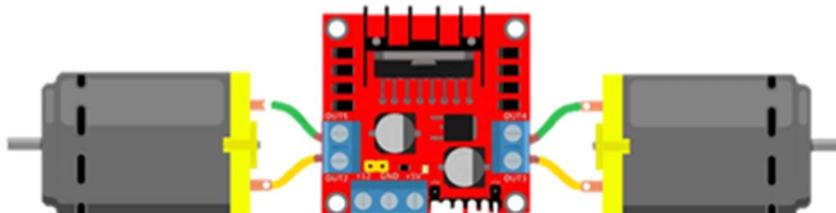


**Figura 18: Roda Boba, Roda de 68mm.**  
**Fonte: Eletrogate 2019.**

#### 1.4.2.7. Motor DC

O Motor DC 6V é diferente de outros motores, tais como o motor de passo 6v, pois este possui uma caixa de redução e um eixo duplo, podendo ser utilizado nos dois lados. O motor DC 6V possui polaridades e quando ligado em polaridades diferentes, o mesmo muda a rotação de horário para anti-horário do motor.

A utilização do motor DC 6V é simples e facilita os projetos robóticos e/ou de engenharia, pelo seu baixo custo, equivalente a R\$15,00. Este motor será utilizado no carrinho que é controlado pelo computador, o mesmo, controlado pela ponte h, como mostra na figura abaixo.



**Figura 19: Motor DC 6V (Ponte h).**

#### 1.4.2.8. Ponte H

A ponte h é um circuito que controla a saída de corrente podendo inverter as polaridades, uma vez que, sendo alimentado inverte o sentido de funcionamento, como por exemplo: motor DC 6V são motores que pode ser ligado em ambas polaridades, ou seja quando a ponte h inverte a saída ele também irá inverter a sua rotação de horário para anti-horário (FELIPEFLOP, 2019). É de suma importância ressaltar que o mesmo poder ser utilizado em projetos com válvula solenoide, rele, motor DC é motor de passo, como mostra a próxima figura.

No minicurso, a ponte h será utilizada no carrinho controlado pelo computador, o mesmo será utilizado para controlar a saída de corrente para os motores DC 6V com caixa de redução.



**Figura 20: Ponte H.**  
**Fonte: Filipeflop.**

#### **1.4.2.9. Servomotor**

Micro Servo 9g SG90 é um motor de fácil uso, pois o mesmo tem seu custo benefício barato, onde uma unidade custa em média, R\$16,00. Este micro faz um giro de até 160° graus, possui polaridade e não pode ser ligado em polaridades diferentes (FELIPEFLOP, 2019). No minicurso, ele será utilizado no controle de acesso. Quando um veículo chegar próximo ao controle, o sensor óptico reflexivo irá identificá-lo e enviará a informação para o Arduino, que irá acionar os motores micro servo 9g SG90 como mostra na próxima figura.

De acordo com Almir (2013),

“Servomotor é uma máquina, mecânica ou eletromecânica, que apresenta movimento proporcional a um comando, em vez de girar ou se mover livremente sem um controle mais efetivo de posição como a maioria dos motores; servomotores são dispositivos de malha fechada, ou seja: recebem um sinal de controle; verificam a posição atual; atuam no sistema indo para a posição desejada.”



**Figura 21: Micro Servo 9g SG90.**  
**Fonte Filipeflop 2016.**

## 1.5. Resultados Esperados

Espera-se que a atividade promova entusiasmo nos participantes e possa contribuir para o desenvolvimento significativo de conceitos e habilidades de lógica de programação e Arduino, de forma a introduzi-los no PC com auxílio da robótica educacional permitindo maior concretude e autonomia no aprendizado dos alunos, preparando-os para os desafios de aprender e viver na sociedade atual.

Espera-se que o curso venha contribuir para uma aprendizagem significativa de conceitos, desenvolver habilidades de lógica de programação e manuseio da plataforma Arduino, embasando-se em alguns princípios do PC, gerando produtos para educação básica permitindo maior concretude e autonomia no aprendizado dos participantes, preparando-os para os desafios de aprender e viver na sociedade atual.

Que os participantes adquiriram habilidades como: manipular alguns sensores e atuadores (ultrassônico, infravermelho, luminosidade, LEDs, *buzzers* e motores), desenvolver conhecimentos básicos de eletrônica, compreender razoavelmente a lógica de programação no S4A e sua interação com hardware, consigam ser mais autônomos no desenvolvimento das atividades do seu cotidiano.

Sendo assim, capazes de analisar vários problemas do seu cotidiano, buscando possíveis soluções expressadas em modelos didáticos que sirvam tanto como produtos gerados para atender diversas áreas do conhecimento abordadas no ensino, quanto para modelos que influencie a criatividade na construção de outros modelos didáticos, gerando assim um processo de aprendizagem continuada.

## 1.6. Agradecimentos

Os agradecimentos aos professores Rogério Pereira de Sousa e Ramásio Ferreira de Melo, pelas contribuições e apoio para o desenvolvimento do minicurso Scratch For Arduino, primordial para o desenvolvimento deste trabalho. Ao Instituto Federal de Educação, Ciência e Tecnologia do Tocantins - Campus Araguatins, por proporcionar o desenvolvimento pessoal e profissional, por proporcionar a pesquisa, a extensão e o desenvolvimento do PC na Educação Básica.

## 1.7. Referências

ALBUQUERQUE, David et al. Uma Experiência do Uso Do Hardware Livre Arduino no Ensino De Programação De Computadores. In: Anais do Workshop de Informática

- na Escola. 2016. p. 51. <<http://br-ie.org/pub/index.php/wie/article/download/6602/4513>>. Acesso em: 29 maio 2019.
- ALMIR.Como Funcionam os Servos Motores de Modelismo. O Mecatronico, 2013. Disponível em: <<https://omecatronico.com.br/blog/como-funcionam-os-servos-motores-de-modelismo/>>. Acesso em: 20 jul. 2019.
- ARANTES, Flávia Linhalis; RIBEIRO, Paula Eduarda Justino. Desenvolvimento do PC com Valores da Ética Hacker. Informática na educação: teoria & prática, v. 20, n. 2 mai/ago. Disponível em: <<https://seer.ufrgs.br/InfEducTeoriaPratica/article/view/68953/43628>>. Acesso em: 29 maio 2019.
- ARDUINO. Disponível em: <[www.arduino.cc/](http://www.arduino.cc/)>. Acesso em: 27 jan. 2019.
- BARBOSA, Débora Nice Ferrari et al. ENSINANDO LÓGICA COM AS TECNOLOGIAS DA INFORMAÇÃO: DESENVOLVENDO O RACIOCÍNIO LÓGICO E O PENSAMENTO COMPUTACIONAL. CATAVENTOS-Revista de Extensão da Universidade de Cruz Alta, v. 9, n. 1, p. 54-72, 2017. Disponível em: <<http://www.revistaeletronica.unicruz.edu.br/index.php/Cataventos/article/download/5334/1127>>. Acesso em: 27 maio 2019.
- BRASIL. Ministério da educação. Conselho Nacional de Educação. Parecer CNE/CES 136/2012, 2012. Disponível em: <[http://portal.mec.gov.br/index.php?option=com\\_docman&view=download&alias=11205-pces136-11-pdf&category\\_slug=julho-2012-pdf&Itemid=30192](http://portal.mec.gov.br/index.php?option=com_docman&view=download&alias=11205-pces136-11-pdf&category_slug=julho-2012-pdf&Itemid=30192)>. Acesso em: 28 jul. 2019.
- DE LIMA, Gustavo Fernandes; Glauco George C.; SALAZAR, Andrés Ortiz. CONTROLE DE TEMPERATURA DE UMA TOCHA DE PLASMA TÉRMICO COM ACOPLAMENTO INDUTIVO UTILIZANDO A PLACA ARDUINO. 2013. Disponível em: <<http://www.sbai2013.ufc.br/pdfs/5130.pdf>>. Acesso em: 24 jun. 2019.
- CUNY, J.; SNYDER, L.; WING, Jeannette M. Demystifying computational thinking for non-computer scientists. Unpublished manuscript in progress, 2010. Disponível em: <<http://www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf>>. Acesso em: 20 maio 2019.
- FRANÇA, Rozelma Soares de; AMARAL, Haroldo José Costa do. Proposta metodológica de ensino e avaliação para o desenvolvimento do PC com o uso do scratch. In: Anais do Workshop de Informática na Escola. 2013. p.179. Disponível em: <<http://www.br-ie.org/pub/index.php/wie/article/download/2646/2300>>. Acesso em: 30 maio 2019.
- KALIL, Fahad. et al. Promovendo a robótica educacional para estudantes do ensino médio público do Brasil. Nuevas Ideas en Informática Educativa, TISE, p. 739-742, 2013. Disponível em: <<http://www.tise.cl/volumen9/TISE2013/739-742.pdf>>. Acesso em: 22 maio 2019.
- PAPERT, Seymour. Mindstorms: Children, computers, and powerful ideas. Basic Books, Inc., 1980.

- PEREIRA, Amanda C.; FRANCO, Matheus E. Desenvolvendo o pensamento computacional no ensino fundamental com Arduino e Scratch. In: 5º Encontro Nacional de Computação dos Institutos Federais (ENCompIF 2018). SBC, 2018. Disponível em: <<https://sol.sbc.org.br/index.php/encompif/article/view/3561>> Acesso em: 15 jun. 2019.
- RAMOS, Fellipe; TEIXEIRA, Lilian da Silva. Significação da aprendizagem através do PC no ensino médio: uma experiência com scratch. In: Anais do Workshop de Informática na Escola. 2015. p. 217. Disponível em: <<http://www.br-ie.org/pub/index.php/wie/article/download/5024/3434>>. Acesso em: 12 maio 2019.
- RODRIGUEZ, Carla et al. PC: transformando ideias em jogos digitais usando o Scratch. Anais do Workshop de Informática na Escola. 2015. p. 62. Disponível em: <<http://br-ie.org/pub/index.php/wie/article/view/4992>>. Acesso em: 30 maio 2019.
- S4A. Disponível em: <<http://s4a.cat/>>. Acesso em: 21 maio. 2019.
- FELIPEFLOP. Disponível em: <<https://www.filipeflop.com/>>. Acesso em: 24 ago. 2019.
- SANTOS, Elisângela Ribas dos et al. Estímulo ao PC a partir da Computação Desplugada: uma proposta para Educação Infantil. 2016. Disponível em: <<http://dehesa.unex.es/handle/10662/5939>>. Acesso em: 28 maio 2019.
- SCAICO, Pasqueline Dantas et al. Ensino de programação no ensino médio: Uma abordagem orientada ao design com a linguagem scratch. Revista Brasileira de Informática na Educação, v. 21, n. 02, p. 92, 2013. Disponível em: <<http://www.academia.edu/download/43035180/2364-4111-1-PB.pdf>>. Acesso em: 28 maio 2019.
- SCOLARI, T. Angélica; BERNARDI, Giliane; CORDENONSI, Z. Andre. O Desenvolvimento do Raciocínio Lógico através de Objetos de Aprendizagem. Santa Maria-RS, 2007. Disponível em: <<http://www.cinted.ufrgs.br/ciclo10/artigos/4eGiliane.pdf>>. Acesso em: 28 maio 2019.
- SOBREIRA, Elaine Silva Rocha; TAKINAMI, Olga Kikue; DOS SANTOS, Verônica Gomes. Programando, Criando e Inovando com o Scratch: em busca da formação do cidadão do século XXI. Jornada de Atualização em Informática na Educação, v. 1, n. 1, 2013. Disponível em: <<https://br-ie.org/pub/index.php/pie/article/view/2592/2248>> Acesso em: 28 maio 2019.
- VALENTE, José Armando. Integração do PC no currículo a Educação Básica: diferentes estratégias usadas e questões de formação de professores e avaliação do aluno. Revista e-Curriculum, v. 14, n. 3, p. 864-897, 2016. Disponível em: <<http://www.redalyc.org/pdf/766/76647706006.pdf>>. Acesso em: 23 maio 2019.
- WING, Jeannette M. Computational thinking. Communications of the ACM, v. 49, n. 3. p. 33-35, 2006. Disponível em: <<https://www.cs.cmu.edu/~15110-s13/Wing06-ct.pdf>>. Acesso em: 28 maio. 2019.
- ZANETTI, Humberto; OLIVEIRA, Claudio. Práticas de ensino de Programação de Computadores com Robótica Pedagógica e aplicação de Pensamento Computacional. In: Anais dos Workshops do Congresso Brasileiro de Informática na Educação. 2015.

p. 1236. Disponível em: <<https://br-ie.org/pub/index.php/wcbie/article/view/6268/4389>>. Acesso em: 19 jul. 2019.

## Capítulo

# 3

## Introdução ao V-REP: Uma Plataforma Virtual para Simulação de Robôs

Luís B. P. Nascimento, Diego S. Pereira, Vitor G. Santos,  
Daniel H. S. Fernandes, Pablo J. Alsina

### *Abstract*

*Practical experiments in robotics are usually difficult task to perform, since many robots are reasonably expensive devices. Thus, a plausible solution is to use simulator to test if the developed system works satisfactorily. In this sense, the Virtual Robot Experimentation Platform (V-REP) is an interesting tool that enables to model robots and perform simulations in a simple and intuitive way, focused in robotics areas such as path planning, mapping, controls, among others. Thus, this chapter presents an introductory course about V-REP, focusing on basic concepts of the plataform and some simulation examples.*

### *Resumo*

*A realização de experimentos práticos na robótica nem sempre é trivial devido aos altos custos nessa área, porém, é comum a utilização de simuladores para possibilitar a realização de experimentos. A Plataforma Virtual para Experimentação Robótica (Virtual Robot Experimentation Platform - V-REP) é uma interessante ferramenta que viabiliza a modelagem de robôs, assim como a realização de experimentos simulados de maneira simples e didática, podendo atuar em áreas da robótica como planejamento de caminho, mapeamento, controle, dentre outras. Dessa forma, este capítulo apresenta uma visão introdutória sobre a plataforma V-REP apresentando conceitos básicos da ferramenta, assim como alguns exemplos de simulações.*

### **3.1. Introdução**

O aumento exponencial do poder de processamento dos computadores combinado com o vasto crescimento de pesquisas na área de robótica não só estimulou, como também tornou viável a utilização de simuladores capazes de reproduzir com qualidade cenários

2D/3D. Além disso, os simuladores tornam possível a realização de experimentos em tempo real permitindo ao pesquisador avançar mais rapidamente com seus resultados e, inicialmente, ter independência da utilização do robô real (*hardware*) que, em muitos casos, é um dos componentes mais onerosos do cenário em estudo.

Atualmente, existem diversas plataformas de simulação robótica disponíveis no mercado, tais como o Open HRP, Gazebo, Webots, V-REP, entre outras [Rohmer et al. 2013]. Contudo, a Plataforma Virtual de Experimentação Robótica (*Virtual Robot Experimentation Platform*), tratada na literatura apenas como V-REP<sup>1</sup>, destaca-se por se tratar de uma ferramenta fácil e didática, o que atrai usuários com pouco ou sem conhecimentos na área de robótica.

A plataforma V-REP busca atender todos os requisitos de uma estrutura de simulação versátil e escalável que, além de oferecer abordagens tradicionais, já encontradas em outros simuladores, possui itens adicionais, pois com uma arquitetura de controle distribuída, cada objeto/modelo V-REP pode ser controlado individualmente através de um *script*, um *plugin*, uma API cliente remota, entre outras soluções [Rohmer et al. 2013].

É possível encontrar diversos trabalhos que fizeram uso desta plataforma, entre eles o trabalho de [Obdržálek 2017] apresenta uma estratégia para utilizar um sistema multi-agente formado por Veículos Aéreos Não Tripulados (VANTs), enquanto o trabalho de [Knoll et al. 2017] propõe um esquema de navegação para robôs terrestres usando um conjunto de sensores de rádio frequência. Em [Tanberk and Tükel 2017], o simulador é utilizado para efetuar testes com um robô capaz de interpretar dados oriundos de um sensor Kinect na tomada de decisão para movimentação de peças em um jogo de xadrez. Finalmente, o trabalho de [Lima et al. 2018] apresenta a utilização da V-REP para auxiliar no processo de aprendizagem de robótica como uma ferramenta educacional. Isso é possível graças à sua versão educacional (V-REP PRO EDU) disponibilizada gratuitamente.

Diante disso, percebe-se a versatilidade do simulador em atuar com diferentes plataformas robóticas e em diversas aplicações, isso permite aos pesquisadores, alunos e professores validarem suas ideias, seja para a obtenção de resultados iniciais em pesquisas utilizando ambientes simulados, ou como ferramenta educacional para o ensino-aprendizagem de conceitos dentro da robótica. Nesse sentido, esse capítulo apresenta uma visão introdutória da plataforma V-REP a fim de destacar algumas características e funcionalidades interessantes, além de descrever como prototipar um robô terrestres utilizando os próprios objetos disponíveis na plataforma.

### 3.2. Virtual Robot Experimentation Platform (V-REP)

Desenvolvido pela Coppelia Robotics, o simulador robótico V-REP possui um ambiente de desenvolvimento integrado e baseia-se em uma arquitetura distribuída onde cada objeto/modelo pode ser controlado individualmente através de um *script*, um *plugin*, um nó ROS (*Robot Operating System*), um cliente de uma API (*Application Programming Interface*) remota ou uma solução personalizada. Tais itens tornam este simulador versátil

---

<sup>1</sup>Site oficial do V-REP: <http://www.coppeliarobotics.com/>

e ideal para diversas aplicações. Os controladores podem ser escritos em C/C++, Python, Java, Lua, Matlab ou Octave. A versão atual da V-REP é a 3.6.2, disponibilizada em 25 de junho de 2019 [Robotics 2019].

### 3.2.1. Onde encontrar o simulador V-REP?

A plataforma de simulação V-REP está disponível para diversos sistemas operacionais. O software está disponível para download através do link (<http://www.coppeliarobotics.com/downloads.html>). A Figura 3.1 mostra as versões do software disponíveis para download.

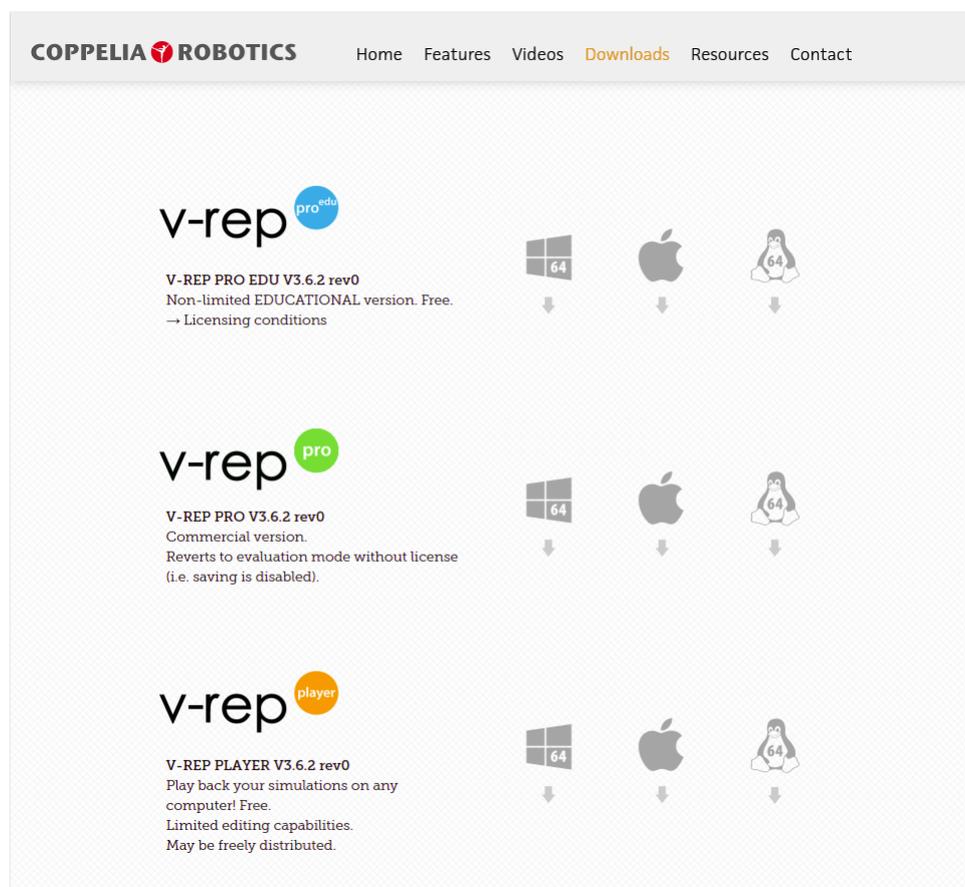


Figura 3.1. Página de download do V-REP.

No site do V-REP estão disponíveis três versões do software (Pro EDU, Pro e Player) para Windows, Mac e Sistemas Linux. A primeira (Pro EDU) é uma versão gratuita, disponível para fins educacionais (essa é a versão que utilizaremos em todos os exemplos). A segunda versão mostrada é a versão Pro, disponível para fins comerciais, sendo necessária a aquisição de uma licença de software. A terceira versão (V-REP Player) é uma ferramenta utilizada para reproduzir e interagir com simulações.

### 3.2.2. Interface do Usuário

A interface de usuário do simulador V-REP é composta por um conjunto de elementos. Os principais elementos são: o terminal, a janela de aplicação e *dialogs* (caixas de diálogo).

O primeiro deles, o terminal, é ocultado quando iniciado em ambiente Windows. Nas demais plataformas, Linux e Mac, é possível acompanhar a inicialização da plataforma de simulação, o carregamento de plugins, dentre outras informações, inclusive mostra se o sistema foi inicializado de forma correta. Este capítulo não irá apresentar maiores detalhes para interação com o terminal.

A janela de aplicação ou janela principal (interface gráfica) do simulador V-REP agrupa a grande maioria das funcionalidades do simulador e, através dela, é possível exibir, editar, simular e interagir com uma cena. O usuário também pode editar e interagir com uma cena ajustando as configurações ou parâmetros presentes em uma caixa de diálogo. Cada caixa de diálogo agrupa um conjunto de funções relacionadas ou funções que se aplicam a um mesmo objeto de destino. O conteúdo de um diálogo pode ser sensível ao contexto (por exemplo, dependente do estado de seleção do objeto)[Robotics 2019]. A Figura 3.2 retrata o ambiente da janela principal do simulador V-REP.

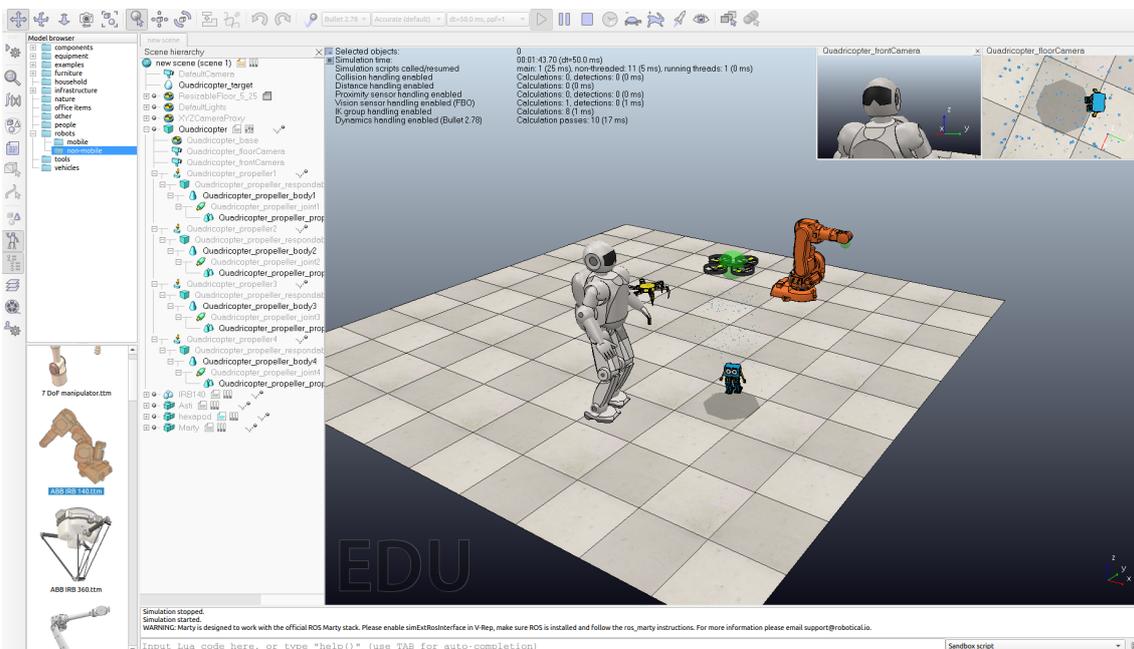
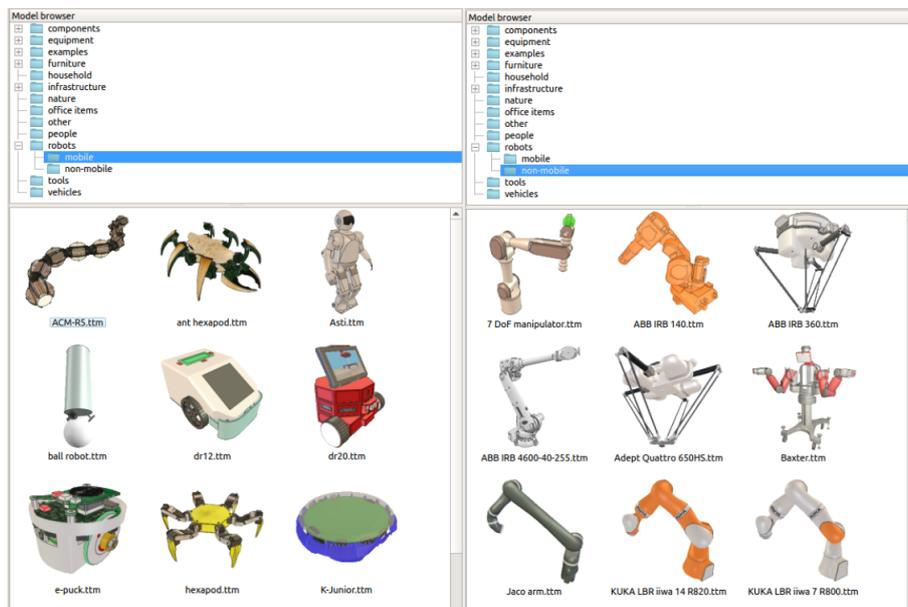


Figura 3.2. Exemplo da interface V-REP com diversos tipos de robôs.

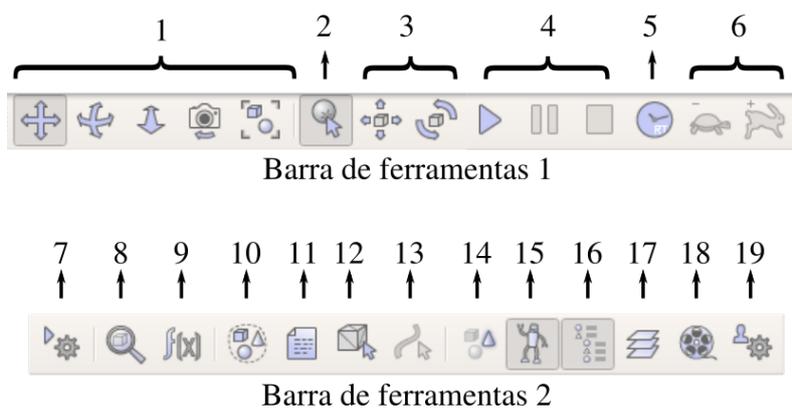
De forma geral, o simulador V-REP possui uma interface gráfica agradável e intuitiva formada por menus, barras de ferramentas e *dialogs* que permitem ao usuário inserir, remover, movimentar, editar objetos de forma simples, fazendo uso apenas do *mouse*. No canto esquerdo da imagem é possível observar o navegador de modelos (*Model Browser*). A partir desse navegador é possível encontrar diversos modelos de robôs já prontos para ser utilizados em simulações. No V-REP, os robôs disponíveis estão divididos em móveis e não-móveis, sendo possível adicioná-los à cena arrastando e soltando (*drag-and-drop*). O navegador de modelos com alguns robôs pode ser observado na Figura 3.7.

Como comentado anteriormente, outros elementos importantes que facilitam a criação de simulações e que estão presentes na interface principal do V-REP são



**Figura 3.3. Exemplo da interface V-REP com diversos tipos de robôs.**

as barras de ferramentas. Nelas é possível encontrar botões com funções úteis que tornam o manuseio do simulador mais simples, já que as funcionalidades presentes nas barras estão contidas dentro de menus internos, o que tornaria mais difícil tarefas básicas que necessitem ser realizadas, tais como, mudança na angulação da câmera ou reposicionamento de objetos dentro de cena. A Figura 3.4 mostra duas barras de ferramentas contendo alguns botões com as funcionalidades mais úteis.



**Figura 3.4. Algumas funções encontradas nas barras de ferramentas do V-REP.**

A Figura 3.4 mostra duas barras de ferramentas contendo algumas funções importantes para a criação de simulações no V-REP. Na barra de ferramentas 1 é possível encontrar algumas funções diretamente relacionadas às simulações, como seguem:

1. Navegação da câmera
2. Seleção por clique

3. Manipulação de objetos (translação e rotação)
4. Botões de simulação (iniciar, pausar e parar)
5. Simulação em tempo real
6. Controle de velocidade das simulações.

A barra de ferramentas 2 apresenta algumas funcionalidades mais avançadas, assim como algumas configurações, como podem ser listadas a seguir:

7. Configurações da simulação
8. Propriedade do objeto selecionado
9. Módulo de cálculos (ex: distâncias)
10. Coleções
11. Scripts
12. Editor de formas
13. Editor de caminhos selecionados
14. Objetos Selecionados
15. Navegador de Modelos
16. Hierarquia de cena
17. Camadas
18. Gravador de vídeos
19. Configurações do usuário

É possível ver com detalhes a função de cada item no Manual do Usuário disponível na página oficial do V-REP (<http://www.coppeliarobotics.com/helpFiles/>). Além disso, alguns robôs, tais como o IRB140 e o Asti, possuem uma interface própria que permite ao usuários enviar informações aos robôs e interagir com o cenário em tempo de execução durante uma simulação.

Conforme mostrado, uma simulação V-REP pode ser iniciada, pausada ou encerrada através de botões na barra de menu. Internamente, o simulador gerencia estados intermediários adicionais necessários para informar a *scripts* ou programas o que ocorrerá a seguir, contudo, isto é transparente para o usuário. Vale destacar que existe uma preocupação do simulador em manter o tempo de simulação sincronizado com o tempo real, entretanto, em alguns casos isto não é possível devido a complexidade da simulação, dos recursos computacionais disponíveis e das próprias configurações atribuídas pelo usuário. É possível observar na Figura 3.5 que são exibidas diversas informações durante o processo de simulação. Elas referem-se as próprias configurações atribuídas a cena, como também aos objetos presentes nela.

Selected objects:	0
Simulation time:	00:01:43.70 (dt=50.0 ms)
Simulation scripts called/resumed	main: 1 (25 ms), non-threaded: 0 (0 ms)
Collision handling enabled	Calculations: 0, detections: 0 (0 ms)
Distance handling enabled	Calculations: 0 (0 ms)
Proximity sensor handling enabled	Calculations: 0, detections: 0 (0 ms)
Vision sensor handling enabled (FBO)	Calculations: 1, detections: 0 (1 ms)
IK group handling enabled	Calculations: 8 (1 ms)
Dynamics handling enabled (Bullet 2.78)	Calculation passes: 10 (17 ms)

Figura 3.5. Informações da Cena em execução no V-REP.

### 3.2.3. Cenas e Modelos

Cenas e modelos são os principais elementos de simulação V-REP. Um modelo é um sub-elemento de uma cena e uma cena pode conter vários modelos. Um modelo só pode ser executado quando incluído em uma cena, o Asti, robô humanoide de cor branca apresentado anteriormente, é um exemplo de modelo. A plataforma V-REP permite que pesquisadores desenvolvam modelos e importem para seu ambiente de simulação.

Todos os itens que compõe uma cena estão organizados de forma hierárquica, compondo a hierarquia de cena (*scene hierarchy*). A hierarquia da cena, apresentada na Figura 3.6, exibe o conteúdo de uma cena, ou seja, todos os objetos presentes na respectiva cena. Essa hierarquia é feita em uma estrutura de árvore e representa a relação pai-filho entre todos os objetos. No exemplo a seguir, a cena com nome *new scene* possui um modelo Quadricóptero adicionado. É possível observar na Figura que ele está em um nível inferior ao objeto *new scene*, logo é chamado nó filho. De forma análoga, os objetos *Quadricopter\_floorCamera* e *Quadricopter\_frontCamera* são nós filhos do nó *Quadricopter*, que é seu respectivo nó pai.

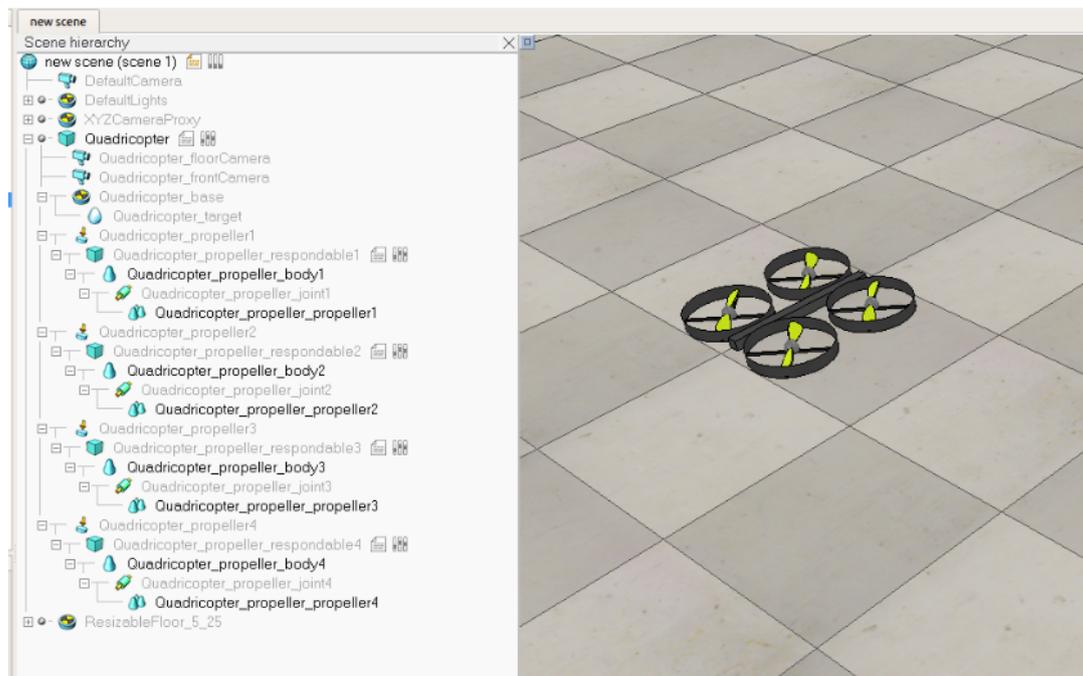


Figura 3.6. Exemplo de Hierarquia de Cena no V-REP.

### 3.2.4. Objetos de Cena

No simulador V-REP existem diversos elementos dos quais é possível construir uma cena de simulação, são os chamados Objetos de Cena (*scene objects*). Esses objetos são visíveis e possuem uma representação em três dimensões. Além disso, possuem papéis fundamentais para garantir a qualidade da simulação. Os principais objetos são apresentados a seguir:

- *Shapes*: São as formas básicas encontradas no V-REP, como por exemplo, cubo, cilindros, esferas, etc.
- *Joints*: São objetos utilizadas para a modelagem de robôs para prover movimento, ex: junta rotacional, junta prismática, etc.
- *Graphs*: São gráficos utilizados para visualizar e gravar dados durante as simulações.
- *Sensores*: Objetos utilizados para percepção do ambiente (Sensores de proximidade, Sensores de Visão e Sensores de Força)
- *Path*: Esse objeto define caminhos ou trajetórias no espaço.

Para a adição de objetos ao cenário, basta ir ao menu Add da barra de menus e escolher qual objeto, ou simplesmente, clicar com o botão esquerdo do mouse sob o cenário e selecionar Add. A Figura 3.7 mostra alguns objetos adicionados à cena: Três formas primitivas (Esfera, Cubo e um Cilindro) e um objeto *Graph*.

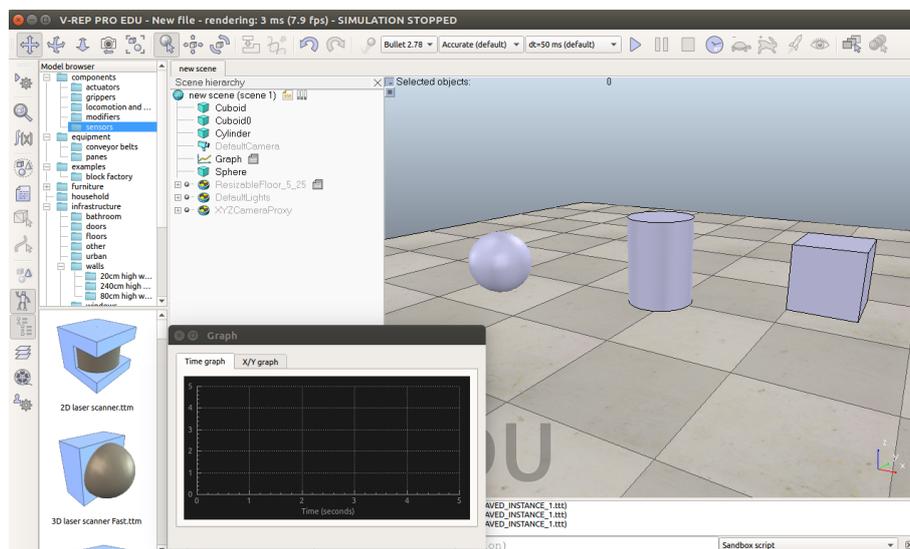


Figura 3.7. Objetos em cena. Algumas formas primitivas adicionadas ao cenário.

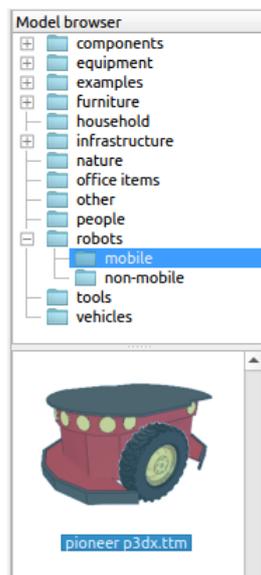
Os objetos do tipo *shapes* são formas que possibilitam a modelagem de outros elementos na cena, como por exemplo, paredes, obstáculos, partes de robôs, e etc. Na Seção 3.5 serão apresentados alguns passos para a construção de um simples robô utilizando alguns objetos de cena básicos.

### 3.2.5. Exercício 01

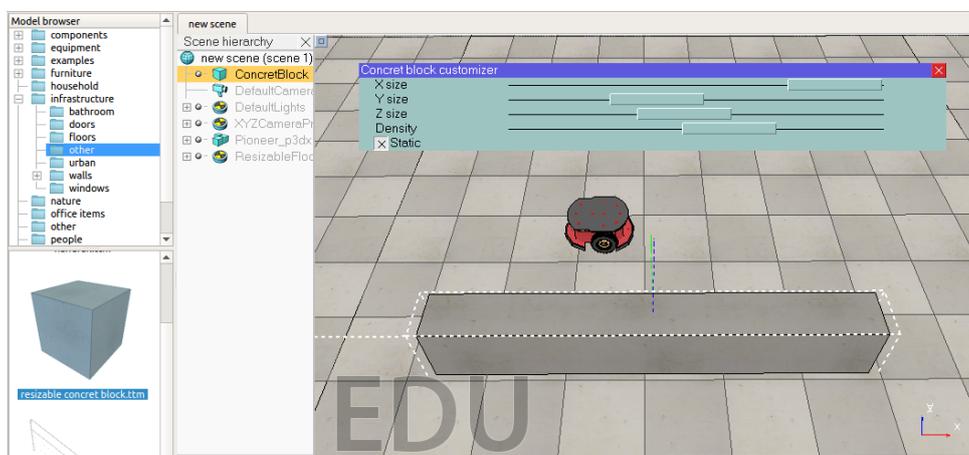
O cenário a seguir tem como objetivo apresentar uma primeira simulação com uso da plataforma V-REP. Para tal, será feito uso de um robô modelo Pioneer 3DX, já disponibilizado pelo V-REP, sem a necessidade de nenhuma configuração adicional. Será construída uma barreira impedindo que o robô saia do cenário. Dessa maneira, os seguintes passos devem resultar no cenário descrito.

**PASSO 01:** Criar uma nova *cena*;

**PASSO 02:** Inserir um robô *Pioneer 3DX* ao cenário criado. ;



**PASSO 03:** Inserir quatro *resizable concret blocks* e redimensiona-los por meio do customizador do bloco, de tal maneira que o espaço de navegação que o robô está inserido seja limitado;



**Figura 3.8.** Selecionando o *resizable concret block*.

**PASSO 04:** Finalmente, execute a simulação e observe o comportamento do

robô. A Figura 3.9 retrata o robô P3DX já inserido no espaço de trabalho adequadamente limitado pelos blocos para restringir sua área de circulação.

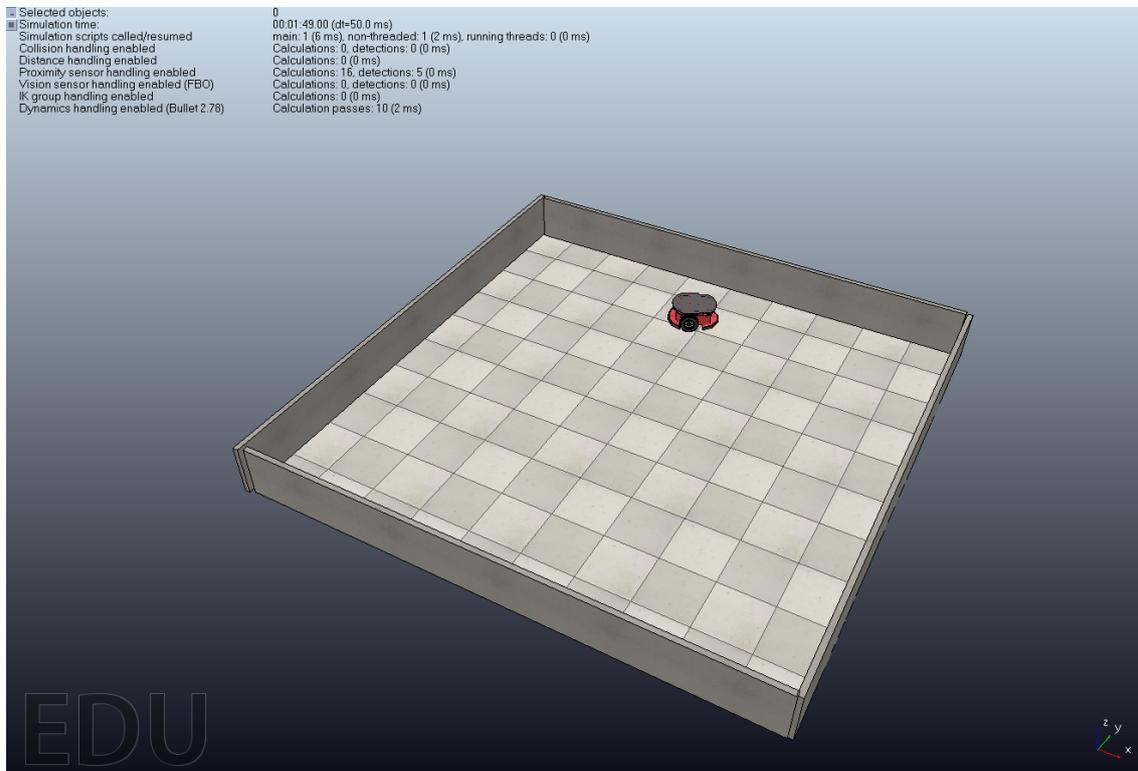


Figura 3.9. Exercício 01 em execução.

**QUESTIONAMENTO 01:** Qual a estratégia utilizada pelo robô para evitar colisões? Onde é possível encontrá-la?

### 3.3. Scripts

Um script é considerado uma entidade utilizada para controlar um modelo. O V-REP possui um interpretador de scripts integrado onde, por padrão, a linguagem de programação suportada é Lua<sup>2</sup>. A linguagem Lua permite programação procedural, programação orientada a objetos, programação funcional, programação orientada a dados e descrição de dados. Ela é considerada uma linguagem ideal para configuração, automação e rápida prototipagem, amplamente utilizada em aplicações industriais, como o Adobe's Photoshop Lightroom, e jogos, por exemplo, World of Warcraft e Angry Birds.

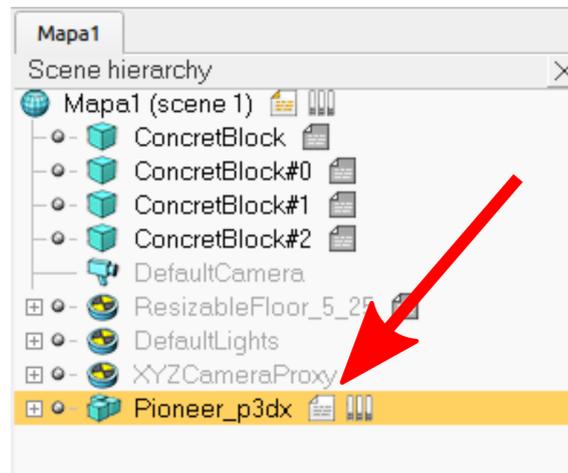
#### 3.3.1. Exercício 02 - Codificando um robô com linguagem Lua

O objetivo desse cenário é realizar um exemplo prático para demonstrar como os objetos do ambiente de simulação V-REP são controlados por um script escrito na linguagem de programação Lua. O ponto de partida desse exercício é a conclusão do Exercício 01.

**PASSO 01:** Abra o arquivo referente ao Exercício 01;

<sup>2</sup><https://www.lua.org/home.html>

**PASSO 02:** Vá até a Hierarquia de Cena, selecione o robô Pioneer P3DX e dê dois cliques no ícone script, localizado ao lado direito do nome Pioneer\_p3dx, conforme Figura 3.10;



**Figura 3.10. Ícone Script vinculado ao modelo Pioneer P3Dx.**

**PASSO 03:** Apague todo o código fonte encontrado na janela e insira o código Lua apresentado seguir. Tenha atenção na indentação e na escrita do script;

```
1 if (sim_call_type==sim.syscb_init) then
2     motorLeft=sim.getObjectHandle("Pioneer_p3dx_leftMotor")
3     motorRight=sim.getObjectHandle("Pioneer_p3dx_rightMotor")
4 end
5
6 if (sim_call_type==sim.syscb_actuation) then
7     sim.setJointTargetVelocity(motorLeft,1)
8     sim.setJointTargetVelocity(motorRight,2)
9 end
```

**Script 3.1. Acionando os motores do robô Pioneer em Lua**

**PASSO 04:** Confira se seu script está conforme a imagem 3.11. Em caso positivo, execute a simulação e observe o comportamento do robô.

**QUESTIONAMENTO 02:** Como é possível justificar o comportamento do robô tomando como base no código fonte utilizado para controlá-lo? Qual a solução para o movimento que o robô executa ser no sentido horário?

### 3.4. Cliente de API Remota

A plataforma de simulação V-REP trata-se de uma ferramenta altamente flexível, sendo possível personalizar vários aspectos de uma simulação. Um dos itens que está diretamente relacionado a tal característica é a utilização de APIs. Por meio de determinadas APIs, é possível controlar e manipular aspectos da cena ou objetos a partir de diferentes maneiras, tais como, por meio de *scripts* ou de *plugins*. Uma interessante API que pode ser utilizada pelo simulador V-REP é a API Remota, que torna possível

```
1 if (sim_call_type==sim.syscb_init) then
2     motorLeft=sim.getObjectHandle("Pioneer_p3dx_leftMotor")
3     motorRight=sim.getObjectHandle("Pioneer_p3dx_rightMotor")
4 end
5
6 if (sim_call_type==sim.syscb_actuation) then
7     sim.setJointTargetVelocity(motorLeft,1)
8     sim.setJointTargetVelocity(motorRight,2)
9 end
```

Figura 3.11. Script Lua concluído.

a comunicação do V-REP em tempo de execução, com diferentes tecnologias externas, além de tornar possível controle de robôs de maneira remota.

A API remota é composta por cerca de 100 funções específicas e uma função genérica que pode ser chamada a partir da aplicação cliente desenvolvida na linguagem escolhida pelo desenvolvedor. Essas funções remotas interagem com o V-REP através de soquetes de comunicação. Para cada cliente que realiza comunicação com o servidor V-REP, um identificador é atribuído. Todas as funções das APIs suportadas pelo V-REP estão disponíveis *online*, como por exemplo, as funções para utilização em conjunto com o Python estão disponíveis no link <sup>3</sup>.

A Figura 3.12 ilustra um cenário típico de comunicação via API remota utilizando linguagens de programação distintas. Neste caso, a partir do momento que o servidor V-REP tem sua simulação iniciada um ponto de comunicação é criado através do endereço ip:A.A.A.A e porta 19999, enquanto os clientes, normalmente, fazem uso de configurações dinâmicas, das quais o ip é atribuído através de um serviço de rede a qual o cliente está conectado e a porta pelo sistema operacional nativo.

### 3.4.1. Exercício 03 - Utilizando uma API Remota

Com base no cenário dos exercícios anteriores, faremos uma pequena modificação com o objetivo de controlar o robô de maneira remota através de um script em linguagem Python. Para possibilitar essa comunicação, é necessário adicionar uma linha de código ao *script* principal para inicializar o servidor V-REP e permitir a chamada de clientes remotos. A linha de código que segue abaixo habilita o servidor no V-REP e deve ser adicionada na função `sysCall_init()`, como mostra a Figura 3.13.

```
1 simRemoteApi.start(19999)
```

Além de modificar o *script* principal da cena, uma modificação no *script* do robô deve ser realizada para impedir que o código original seja executado ao invés do código remoto. Dessa forma, sugere-se apagar todo o conteúdo do *script* atribuído ao robô. Uma outra forma é acessar o botão `Scripts` na barra de ferramentas e desabilitar o *script* do robô.

<sup>3</sup><http://www.coppeliarobotics.com/helpFiles/en/remoteApiFunctionsPython.htm>

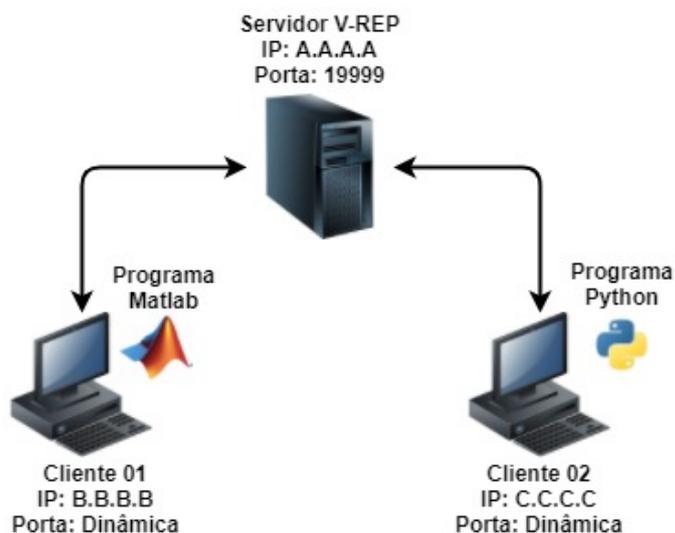


Figura 3.12. Comunicação típica utilizando API remota e V-REP.

```

Main script (customized)
10
11 function sysCall_init()
12     sim.handleSimulationStart()
13     sim.openModule(sim.handle_all)
14     sim.handleGraph(sim.handle_all_except_explicit, 0)
15     simRemoteApi.start(19999)
16 end
17

```

Figura 3.13. Arquivo com o script em Python e arquivos da API Remota.

É importante ressaltar que cada cliente deve fazer da API para determinada tecnologia (Matlab, Python, Java, etc) e para seu sistema operacional (arquivos remoteAPI.so, remoteApi.dll e remoteApi.dylib para Linux, Windows e Mac, respectivamente). Isso é feito por meio da adição de arquivos de configuração dentro do mesmo diretório do programa responsável em fazer as chamadas as servidor V-REP. As APIs para cada tecnologia estão disponíveis dentro do diretório de instalação do V-REP, na pasta /programming/remoteApiBlidings/. Nosso exemplo foi realizado no Linux e o nome do arquivo do nosso script é testePython.py. A Figura 3.14 mostra os arquivos necessários para essa simulação.



Figura 3.14. Arquivo com o script em Python e arquivos da API Remota.

O script abaixo deve ser adicionado ao arquivo *testePython.py* para inicializar a conexão com o servidor e acionar os motores. O valor da variável *clientID* indica se a comunicação com o servidor está de fato ocorrendo. A função *simxGetObjectHandle()*

obtem uma referênciade um objeto dentro do simulador, por isso é importante que o segundo parâmetro represente o nome exato do objeto referente ao simulador. O terceiro parâmetro está relacionado ao modo como a troca de mensagens ente cliente e servidor ocorre, então é recomendado seguir estritamente o parâmetro indicado na documentação. Por fim, a função `simxSetJointTargetVelocity()` deve indicar a velocidade e a ser aplicada e roda que receberá o sinal.

```
1 import vrep
2 import time
3
4 clientID = vrep.simxStart('127.0.0.1',19999,True,True,5000,5)
5
6 if (clientID == 0):
7     print('Conectado!')
8
9 returnC, LwMotor = vrep.simxGetObjectHandle(clientID,'
10     Pioneer_p3dx_leftMotor',vrep.simx_opmode_oneshot_wait)
11 returnC, RwMotor = vrep.simxGetObjectHandle(clientID,'
12     Pioneer_p3dx_rightMotor',vrep.simx_opmode_oneshot_wait)
13 vrep.simxSetJointTargetVelocity(clientID,LwMotor,1,vrep.
14     simx_opmode_streaming)
15 vrep.simxSetJointTargetVelocity(clientID,RwMotor,1,vrep.
16     simx_opmode_streaming)
17
18 time.sleep(1)
```

### Script 3.2. Acionando os motores do robô Pioneer em Python

No instante em que o código do cliente é executado o cenário passa fazer uso das chamadas remotas e atender as requisições que lhe são feitas, sendo assim, o robô deve acionar as rodas e seguir em linha reta.

## 3.5. Construindo meu primeiro robô

O último conteúdo a ser apresentado neste capítulo será um breve exemplo para a modelagem um robô móvel simples. Para isso, inicialmente deve-se criar uma nova cena. Alguns passos devem ser seguidos para a modelagem desse robô como podem ser observados nos itens abaixo:

### • Passo 1 - Criação do Corpo do Robô

1. Add (botão direto do mouse) → Primitive Shape → Cuboid
2. Dimensão: (x = 0.4, y = 0.2, z = 0.1)
3. Nos botões localizados na barra de ferramentas (item 3 na Figura 3.4):
  - Posição: (x = 0, y = 0, z = 0.1)
  - Orientação: (Alpha = 0, Gamma = 0, Beta = 0)
4. Renomeie o *Cuboid* como *Carro*
5. Em Scene Object Properties (item 8 na Figura 3.4) → Common, Marcar as opções Collidable, Measurable, Detectable e Renderable.

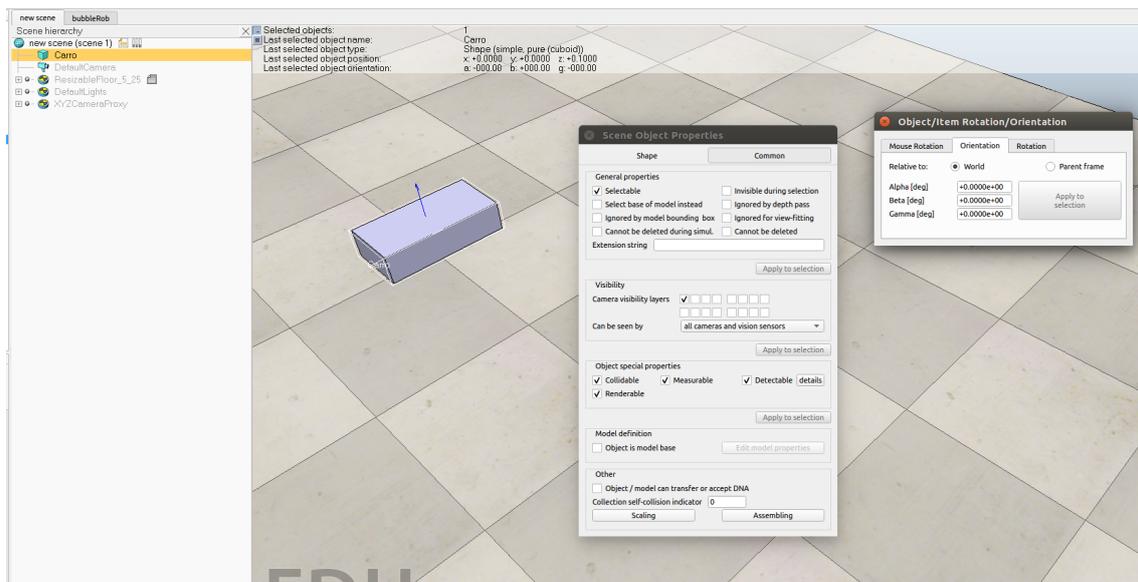


Figura 3.15. Resultado referente ao Passo 1.

### ● Passo 2 - Criação da Roda Direita

1. Add → Primitive Shape → Cylinder
2. Dimensão: (x = 0.125, z = 0.01)
3. Posição: (x = -0.1, y = -0.105, z = 0.0625)
4. Orientação: (Alpha = -90, Gamma = 0, Beta = 0)
5. Renomeie Cylinder como Roda Direita
6. Em Scene Object Properties → Common, Marcar as opções Collidable, Measurable, Detectable e Renderable

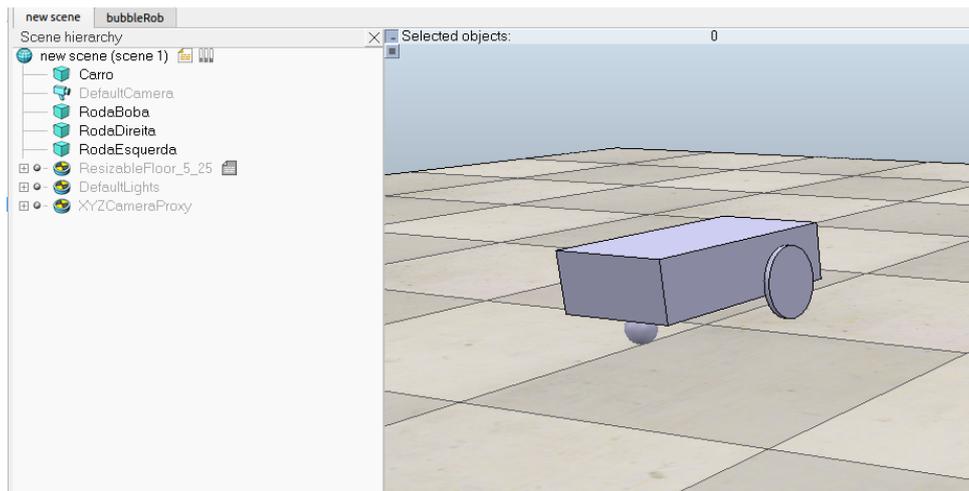
### ● Passo 3 - Criação da Roda Esquerda

1. Add → Primitive Shape → Cylinder
2. Dimensão: (x = 0.125, z = 0.01)
3. Posição: (x = -0.1, y = 0.105, z = 0.0625)
4. Orientação: (Alpha = -90, Gamma = 0, Beta = 0)
5. Renomeie Cylinder como Roda Esquerda
6. Em Scene Object Properties → Common, Marcar as opções Collidable, Measurable, Detectable e Renderable

### ● Passo 4 - Criação da Roda Boba

1. Add → Primitive Shape → Sphere
2. Dimensão: (x = 0.05)

3. Posição: ( $x = 0.15$ ,  $y = 0$ ,  $z = 0.025$ )
4. Orientação: ( $\text{Alpha} = 0$ ,  $\text{Gamma} = 0$ ,  $\text{Beta} = 0$ )
5. Renomeie Sphere como Roda Boba
6. Em Scene Object Properties → Common, Marcar as opções Collidable, Measurable, Detectable e Renderable



**Figura 3.16. Resultado referente ao Passos 2 a 4.**

- **Passo 5 - Adicionando o conector da roda boba**

1. Add → Force Sensor
2. Posição: ( $x = 0.15$ ,  $y = 0$ ,  $z = 0.05$ )
3. Renomeie para Conector

- **Passo 6 - Criação Motor Direito**

1. Add → Joint → Revolute
2. Posição: ( $x = -0.1$ ,  $y = -0.105$ ,  $z = 0.0625$ )
3. Orientação: ( $\text{Alpha} = -90$ ,  $\text{Gamma} = 0$ ,  $\text{Beta} = 0$ )
4. Renomeie Joint como MotorDireito

- **Passo 7 - Criação Motor Esquerdo**

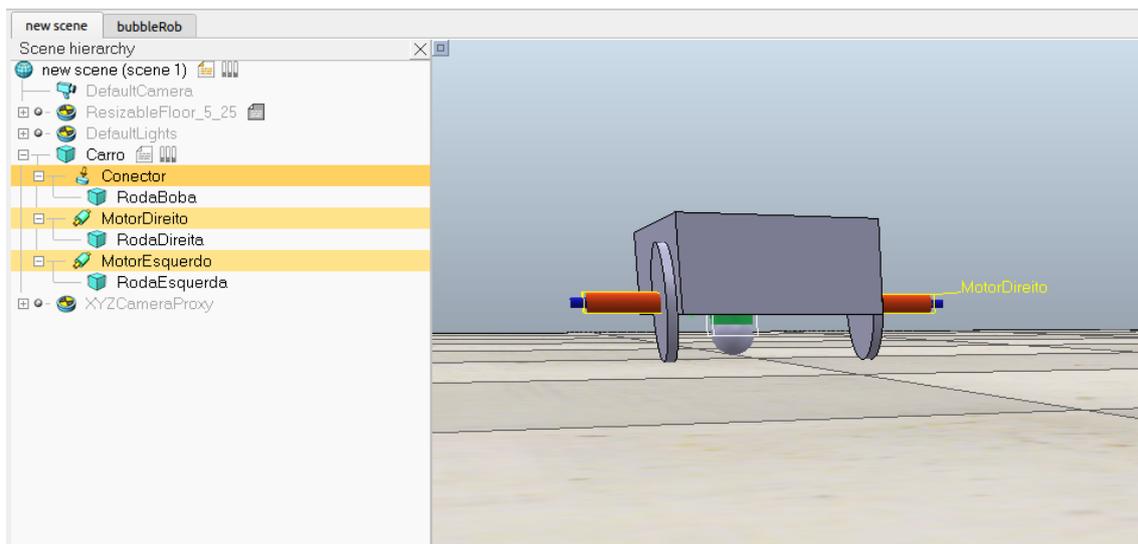
1. Add → Joint → Revolute
2. Posição: ( $x = -0.1$ ,  $y = 0.105$ ,  $z = 0.0625$ )
3. Orientação: ( $\text{Alpha} = -90$ ,  $\text{Gamma} = 0$ ,  $\text{Beta} = 0$ )
4. Renomeie Joint como MotorEsquerdo

- **Passo 8 - Habilitar motores**

1. Selecione os dois motores
2. Em Scene Object Properties → *Show dynamic properties dialog*, selecione as opções *Motor Enabled* e *Lock motor when target velocity is zero*

- **Passo 9 - Hierarquizar os componentes**

1. Ponha o Conector, MotorDireito e MotorEsquerdo como filhas do Carro
2. Ponha RodaDireita como filha do MotorDireito
3. Ponha RodaEsquerda como filha do MotorEsquerdo
4. Ponha RodaBoba como filha do Conector



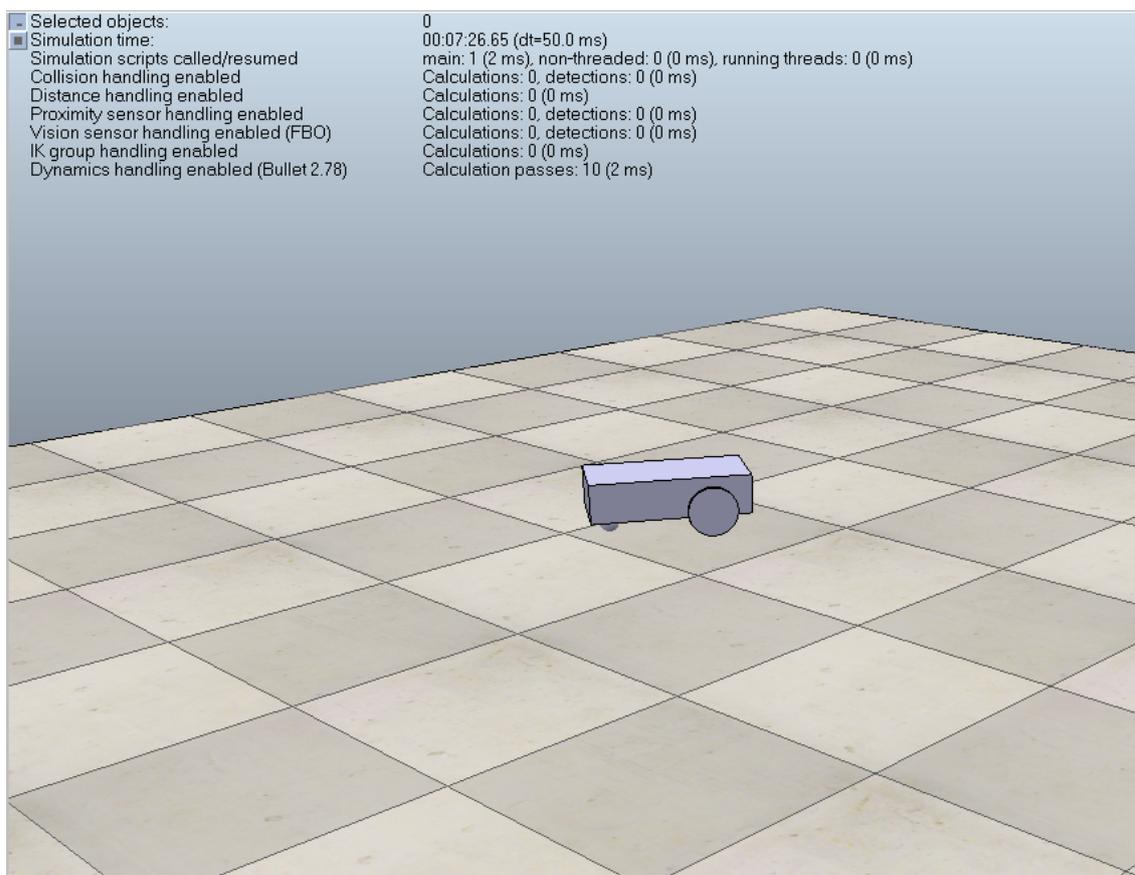
**Figura 3.17. Resultado referente ao Passo 9.**

- **Passo 10 - Escondendo as juntas**

1. Selecione o Conector, o MotorEsquerdo e o MotorDireito.
2. Scene Object Properties → Common
3. Em *Camera visible layers*, desmarque todos as caixas
4. Apply to selection

- **Passo 11 - Movimentando o carro**

1. Com o mouse sobre o objeto Carro na hierarquia de cena, clique com o botão direito Add → Associated schild Script → Non threaded
2. Adicione o código conforme explicado na Seção 3.3
3. Execute a simulação



**Figura 3.18. Robô montado executando.**

### 3.6. Conclusão

Nesse capítulo foi apresentado um conteúdo introdutório sobre a plataforma de simulação de robôs V-REP. O software de simulação foi descrito, assim como suas principais ferramentas, demonstrando como realizar simulações mais simples e até mesmo criando seu ambiente próprio modelo de simulação.

A API remota apresentada é uma interessante característica dessa plataforma pois, mesmo sem conhecimento na linguagem Lua, o usuário poderá programar os robôs utilizando outras tecnologias que mais favorecer o seu conhecimento técnico.

Devido a facilidade do software, qualquer pessoa, mesmo sem um conhecimento aprofundado em robótica, poderá realizar simulações. Além disso, para alunos de robótica, profissionais e pesquisadores da área que necessitam realizar testes experimentais e não possuem uma bancada completa com robôs e sensores, a plataforma ajuda a realizar os experimentos simulados a fim de validar sua metodologia num estágio inicial.

Finalmente, V-REP é uma plataforma para simulação de robôs completa, podendo ser utilizada, inclusive, como ferramenta de base para disciplinas de Introdução à robótica ou Sistemas robóticos autônomos, por exemplo, por ser uma ferramenta ideal para colocar em prática os conceitos abordados sobre robótica.

### 3.7. Currículo dos autores

**Luís Bruno Pereira do Nascimento** possui Bacharelado em Ciência da Computação pela Universidade Estadual do Piauí (UESPI) e Mestrado em Engenharia Elétrica e de Computação pela Universidade Federal do Ceará (UFC). Atualmente é aluno de doutorado pelo Programa de Engenharia Elétrica e de Computação (PPgEEC) pela Universidade Federal do Rio Grande do Norte (UFRN). Suas áreas de interesse incluem robótica autônoma, otimização e aprendizado de máquina.

**Diego da Silva Pereira** possui graduação em Redes de Computadores pelo Instituto Federal do Rio Grande do Norte (IFRN), Mestrado em Ciência da Computação pela Universidade Estadual do Rio Grande do Norte (UERN) e está em doutoramento no Programa de Pós-graduação em Engenharia Elétrica e de Computação (PPgEEC) pela UFRN. Atualmente é professor no IFRN com pesquisas na área de redes de comunicação sem fio aplicadas à sistemas robóticos autônomos.

**Vitor Gaboardi dos Santos** possui graduação em Engenharia Elétrica na Universidade Federal do Rio Grande do Norte (UFRN) com período sanduíche na University of Kansas (USA). Atualmente é aluno de Mestrado no Programa de Pós-graduação em Engenharia Mecatrônica pela UFRN com pesquisas relacionadas a visão computacional aplicada à robótica assistiva.

**Daniel Henrique Silva Fernandes** possui Bacharelado em Engenharia Mecatrônica e Bacharelado em Ciências e Tecnologia pela Universidade Federal do Rio Grande do Norte (UFRN). Atualmente é aluno de Mestrado no Programa de Pós-graduação em Engenharia Mecatrônica pela UFRN com pesquisas relacionadas a robótica assistiva e visão computacional.

**Pablo J. Alsina** possui graduação em Engenharia Elétrica (1987), mestrado na área de controle de motores de indução (1991) e doutorado em Engenharia Elétrica com tema em controle de manipuladores robóticos (1996), pela Universidade Federal da Paraíba. Atualmente é professor titular do Departamento de Engenharia de Computação e Automação (DCA). É professor nos Programas de Pós-Graduação em Engenharia Mecatrônica (PPGEM) e em Engenharia Elétrica e de Computação (PPgEEC) da Universidade Federal do Rio Grande do Norte (UFRN), onde é chefe do Laboratório de Robótica. Desenvolve pesquisas em robótica, nas áreas de controle, planejamento e percepção robótica, com aplicações em robótica assistiva, robótica móvel e Veículos Aéreos Não Tripulados.

### Referências

[Knoll et al. 2017] Knoll, J., Hevrdejs, K., Malinowski, A., and Miah, S. (2017). Virtual robot experiments for navigation in structured environments. In *Industrial Electronics (ISIE), 2017 IEEE 26th International Symposium on*, pages 1173–1178. IEEE.

[Lima et al. 2018] Lima, A. T., de Oliveira, I. F., Rodrigues, G. B., Domingues, J. D., Rocha, F. A. S., and Freitas, G. M. (2018). Utilização do robot operating system (ros) em conjunto com o simulador v-rep no ensino de robótica. In *Anais do XXII CBA, João Pessoa*.

- [Obdržálek 2017] Obdržálek, Z. (2017). Mobile agents in multi-agent uav/ugv system. In *Military Technologies (ICMT), 2017 International Conference on*, pages 753–759. IEEE.
- [Robotics 2019] Robotics, C. (2019). V-rep: Virtual robot experimentation platform.
- [Rohmer et al. 2013] Rohmer, E., Singh, S. P., and Freese, M. (2013). V-rep: A versatile and scalable robot simulation framework. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 1321–1326. IEEE.
- [Tanberk and Tükel 2017] Tanberk, S. and Tükel, D. B. (2017). Kinect controlled chess playing robot. In *Smart Technologies, IEEE EUROCON 2017-17th International Conference on*, pages 594–598. IEEE.

## Capítulo

# 4

## Planejamento de Caminho para Sistemas Robóticos Autônomos

Luís B. P. Nascimento, Vitor G. Santos, Diego S. Pereira,  
Daniel H. S. Fernandes e Pablo J. Alsina

### *Abstract*

*Path Planning is one of the most fundamental processes in autonomous robotics. It is a computational method that calculates a set of poses that a robot must perform in order to fulfil a specific task without colliding with obstacles. Currently, many path planners, libraries and virtual platforms has been developed focused in this area due to its applicability in robotics, games, virtual reality, bioinformatics, among others. In this chapter we present introductory aspects of path planning, presenting the main techniques, libraries and applications, focusing in autonomous robotics.*

### *Resumo*

*O Planejamento de Caminho é um dos processos fundamentais da robótica autônoma dado que planejadores são métodos computacionais capazes de calcular um conjunto de poses que o robô deve assumir para para realizar uma determinada tarefa, evitando colisões. Atualmente tem surgido diversos planejadores, bibliotecas e plataformas para trabalhar nessa área devido a sua aplicabilidade não só na robótica, mas também para jogos/realidade virtual, bioinformática, entre outros. Neste capítulo, aspectos introdutórios sobre planejamento de caminho são abordados, apresentando as principais técnicas, bibliotecas, e aplicações, com destaque na robótica autônoma.*

### **4.1. Introdução**

É indiscutível que o ramo da robótica vem se expandindo cada vez mais, já que dispositivos robóticos vêm sendo aplicados na execução de tarefas anteriormente realizadas pelo homem. Em fábricas de produção, por exemplo, atividades que exigem demasiada força física e um tempo hábil para serem concluídas, estão sendo realizadas com maior frequência por robôs, uma vez que máquinas possuem características como repetibilidade,

velocidade e segurança na execução de tarefas. Em outras aplicações em que dispositivos interagem diretamente com o ser humano, como em procedimentos cirúrgicos, órteses ativas e exoesqueletos, por exemplo, a precisão dos movimentos é algo imprescindível, sendo assim, a robótica é introduzida a fim de auxiliar na realização de movimentos humanos e na redução de falhas.

Dispositivos robóticos necessitam de uma programação prévia para realizar determinadas tarefas, e quando programados corretamente, realizam tarefas complexas com sucesso. Anteriormente, os robôs eram limitados a movimenta-se apenas de acordo com as situações para as quais foram designados. Atualmente, são capazes de realizar movimentos de maneira autônoma, com um mínimo de intervenção humana. O robô autônomo necessita reagir de forma coerente à mudanças dinâmicas no ambiente no qual está inserido, como por exemplo, desviar mediante a presença de obstáculos. Nesse contexto, tomando como base os problemas fundamentais da robótica (Percepção, Planejamento e Ação), um robô autônomo deve ser habilitado a fazer aquisição de informações do ambiente para estimar sua posição e se mover em segurança de uma posição para outra, sem perder sua localização ou sem colidir com possíveis obstáculos [Murphy 2000].



Figura 4.1: Fluxograma com os problemas fundamentais da robótica autônoma.

No sistema de percepção, o robô faz uso de sensores para coletar informações do ambiente no qual está inserido. Na Figura 4.2 um sistema de percepção cria um mapa 2D do ambiente na qual o robô está inserido através de sensores, ao mesmo tempo que se localiza, técnica conhecida como SLAM (Simultaneous Localization and Mapping).

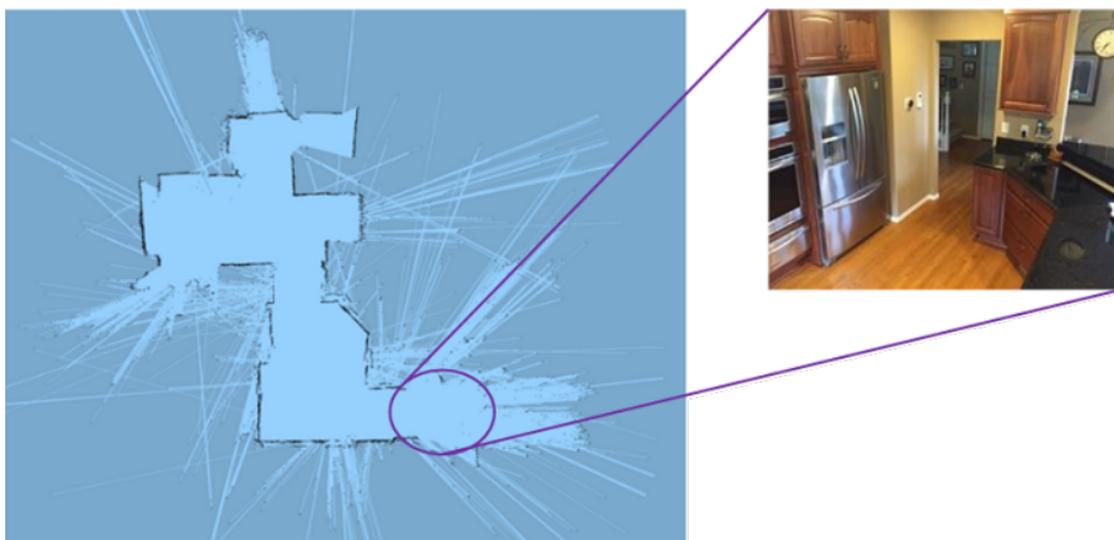


Figura 4.2: Mapeamento e localização simultânea. Na esquerda, um mapeamento em 2D de um ambiente. Na direita, o ambiente na qual o robô está inserido. Ilustração adaptada de [Sakemoto 2017]

O Planejamento de caminho (*Path planning* ou *motion planning*) é um dos principais problemas da robótica autônoma pois define uma sequência de ações a serem realizadas para que um robô, partindo de seu estado atual, possa alcançar o objetivo final, evitando colisões ao longo do percurso [Siegwart and Nourbakhsh 2004, Latombe 1991]. Um exemplo do processo de planejamento de caminho para um robô móvel é ilustrado na Figura 4.3.

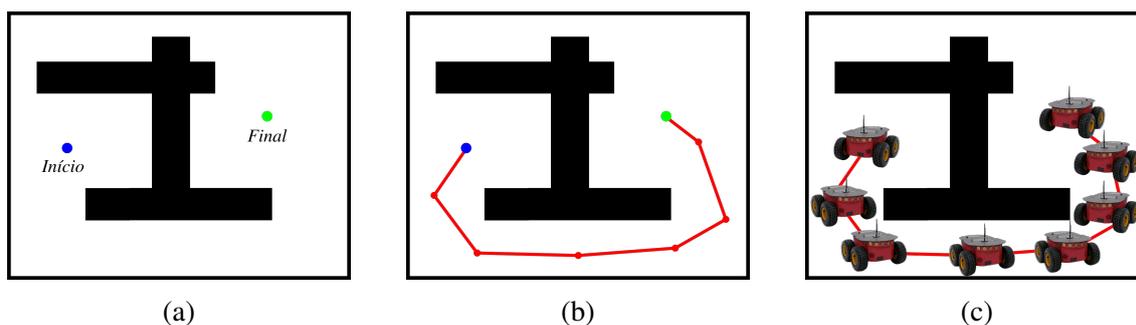


Figura 4.3: Ilustração do processo de planejamento de caminho. (a) Representação do mapa do ambiente. (b) Caminho calculado entre o ponto inicial e o ponto final. (c) Robô executando a ação planejada.

O planejamento de caminho é realizado por meio de métodos computacionais capazes de calcular um conjunto de poses e orientações que o robô deve assumir para realizar uma determinada tarefa. Na Figura 4.3 o robô deve sair de um ponto inicial (ponto azul) até um ponto final (ponto verde), evitando colisões. Na Figura 4.3(a) o mapa do ambiente é representado. Em Figura 4.3(b) mostra um caminho em vermelho que descreve as poses e orientações que o robô móvel deve assumir para sair do ponto inicial e chegar até o final. Na Figura 4.3(c), de maneira ilustrativa, o robô realiza as ações planejadas.

Planejar um caminho é uma etapa essencial na automatização de movimentos para todos os tipos de robôs, seja na transposição de obstáculos ou subir uma escada para um robô humanoide, definir uma missão para um veículo aéreo autônomo ou até mesmo uma tarefa para um robô manipulador industrial. Além disso, a cada momento vêm surgindo novos métodos, bibliotecas e plataformas para trabalhar com o planejamento devido a sua aplicabilidade não só na robótica, mas também para jogos [Edelkamp and Plaku 2014], bioinformática [Al-Bluwi et al. 2012], entre muitos outros. Nesse sentido, este capítulo apresenta uma visão introdutória sobre o problema do planejamento de caminho mostrando sua importância na robótica autônoma.

As demais seções desse documento estão descritas como segue: Na Seção 4.2 é apresentado um breve histórico sobre o planejamento de caminho, onde o importante conceito de Espaço de Configurações é introduzido e importantes definições são realizadas. Na Seção 4.3 os métodos clássicos de planejamento são apresentados. A Seção 4.4 apresenta o que são métodos baseados em amostragem, ilustrando alguns algoritmos. Na Seção 4.5 são apresentadas algumas ferramentas interessantes sobre planejamento de caminho, como OMPL e MoveIt. Finalmente, as conclusões são apresentadas na Seção 4.6, seguida dos currículos resumidos dos autores e das referências.

## 4.2. Histórico e Espaço de Configurações

Segundo [LaValle 2006], a primeira ideia de algoritmo de planejamento foi provavelmente introduzida por [Nilsson 1969], entretanto, apenas no final dos anos de 1970 as pesquisas envolvendo planejamento de caminho ganharam maior força, devido ao advento do conceito de espaço de configurações, o *C-space*, uma representação que abstrai e simplifica problemas de planejamento. O *C-space* é uma representação matemática que mapeia um problema que está especificado no espaço de trabalho (*workspace*), incluindo as características físicas do robô, para um novo espaço, onde os obstáculos possuem uma representação equivalente e o robô é representado por um ponto, com forma e dimensão irrelevante. Essa ideia de representação do robô foi inicialmente introduzida por [Udupa 1977] e melhor explorada e popularizada por [Lozano-Pérez and Wesley 1979] e [Lozano-Perez 1983]. Uma ilustração do processo de mapeamento do espaço de trabalho para o espaço de configurações é mostrado na Figura 4.4.

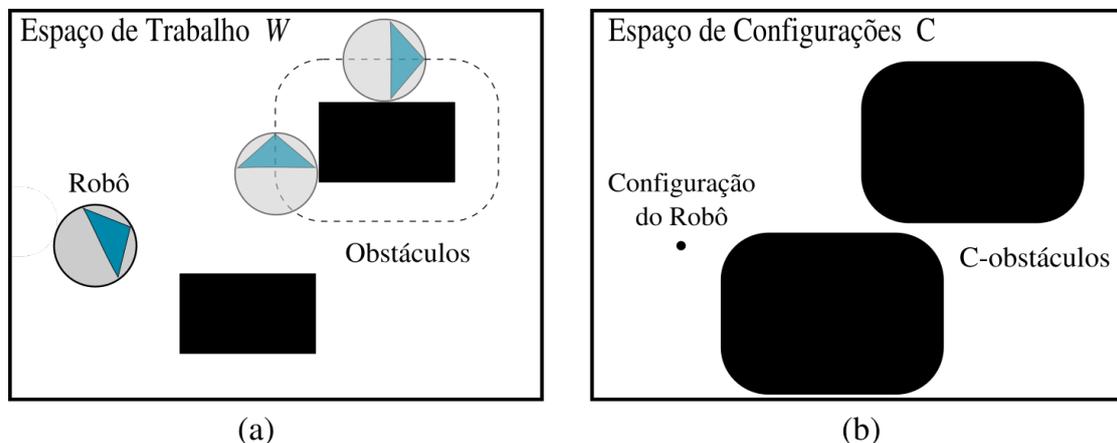


Figura 4.4: Ilustração do processo computação do espaço de configurações para um robô circular. (a) Representação do robô no espaço de trabalho. (b) Representação do robô e dos obstáculos no espaço de configuração.

A representação do mundo através do *C-space* [Lozano-Pérez and Wesley 1979] facilita o planejamento dois motivos fundamentais: O processo de chegar a colisão entre o robô e um obstáculo ocorre ao chegar se um ponto está localizada numa determinada região. Além disso, possibilita que problemas relacionados a geometria e cinemática do robô sejam resolvidos pelo mesmo algoritmo.

Na Figura 4.5 pode ser observada mais uma ilustração que aborda o espaço de configurações, agora para um robô manipulador, com duas juntas rotacionais ( $\theta_1$  e  $\theta_2$ ).

Em 4(a) robô no espaço de trabalho realiza uma tarefa de sair de uma posição e assumir outra e a na outra imagem 4(b) o mesmo cenário é utilizado, porém o robô é representado de maneira pontual (sem forma ou dimensões relevantes), mapeado para o *C-space* e realizando a mesma tarefa. A simplificação proporcionada pelo *C-space* é facilmente perceptível, principalmente quando observa-se que o planejamento no espaço de trabalho  $\mathcal{W}$  envolve toda a análise da cinemática para obter uma sequência movimentos que permita uma trajetória segura partindo de sua pose inicial até sua posição final, diferentemente do caminho resultante no espaço de configuração, representado simplesmente

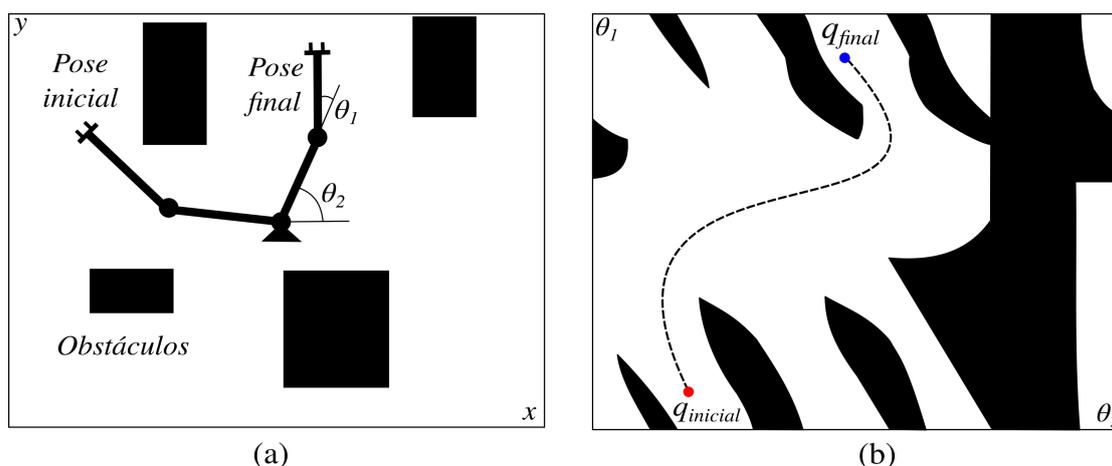


Figura 4.5: (a) Robô manipulador com duas juntas rotacionais representado no Espaço de trabalho  $\mathcal{W}$  com duas poses diferentes. Em (b) representação do caminho que o robô deve assumir para sair de  $q_{inicial}$  para  $q_{final}$ , mapeado no espaço de configuração  $C_{space}$ . Ilustração adaptada de Siegwart and Nourbakhsh (2004)

por uma sequência de pontos entre as configurações  $q_{inicial}$  até  $q_{final}$  através da região  $C_{livre}$ .

A fim de facilitar o entendimento dos conceitos abordados nesse trabalho, serão apresentadas formalmente as definições associadas à representação de um problema de planejamento no espaço de configuração. As seguintes definições foram retiradas da tese de doutorado de [Molina 2014], que apresentou uma extensa fundamentação teórica sobre planejamento:

**Definição 1 (Configuração  $q$  de um robô)** A configuração  $q$  de um robô é um conjunto de  $n$  parâmetros que especifica completamente a postura de um robô,  $\mathcal{A}$ , em seu espaço de trabalho,  $\mathcal{W}$ .

**Definição 2 (O espaço de configurações  $\mathcal{C}$ , ou  $C_{space}$ )** O espaço de configurações  $\mathcal{C}$ , ou  $C_{space}$ , é o espaço  $n$ -dimensional  $\mathcal{C}$  que contém todas as possíveis configurações  $q$  de um robô  $\mathcal{A}$  em seu espaço de trabalho  $\mathcal{W}$ .

**Definição 3 (Caminho  $\mathcal{T}$ )** Dada uma configuração inicial do robô  $\mathcal{A}$ ,  $q_{inicial}$ , e uma configuração final desse mesmo robô,  $q_{final}$ , um caminho no espaço de configurações é o mapeamento definido por:

$$\mathcal{T} : [0, 1] \rightarrow C_{space} \mid \mathcal{T}(0) = q_{inicial} \text{ e } \mathcal{T}(1) = q_{final}. \quad (1)$$

**Definição 4 (Obstáculo no  $C_{space}$  ou  $C_{obs}$ )** Seja  $\mathcal{B}$  o conjunto de todas as regiões ocupadas em  $\mathcal{W}$  e  $\mathcal{A}(q)$  a parte de  $\mathcal{W}$  ocupada pelo robô  $\mathcal{A}$  em uma dada configuração  $q$ . Então, os obstáculos no  $C_{space}$  são definidos como:

$$C_{obs} = \{q \in C_{space} \mid \mathcal{A}(q) \cap \mathcal{B} \neq \emptyset\}. \quad (2)$$

**Definição 5 (Região navegável do  $C_{space}$  ou  $C_{livre}$ )** A região navegável do  $C_{space}$ , ou  $C_{livre}$ , é o subconjunto do espaço de configurações que não pertence ao  $C_{obs}$ , ou seja:

$$C_{livre} = C_{space} \setminus C_{obs}. \quad (3)$$

**Definição 6 (Caminho  $\mathcal{T}$  livre de obstáculos)** Dada uma configuração inicial do robô  $\mathcal{A}$ ,  $q_{inicial}$ , e uma configuração final desse mesmo robô,  $q_{final}$ , um caminho livre de obstáculos no espaço de configurações é o mapeamento definido por:

$$\mathcal{T} : [0, 1] \rightarrow C_{livre} \mid \mathcal{T}(0) = q_{inicial} \text{ e } \mathcal{T}(1) = q_{final}. \quad (4)$$

A partir dessas definições, já será possível introduzir alguns métodos de planejamento de caminho presentes na literatura. Iniciaremos falando dos chamados Métodos Clássicos de planejamento, e depois comentaremos sobre alguns métodos baseados em amostragem.

### 4.3. Métodos clássicos de planejamento

Uma classificação apresentada em 1991 por Latombe [Latombe 1991], e ainda muito aceita nos dias atuais, divide as técnicas de planejamento quanto à forma de representar e interpretar o espaço de configuração do robô. A chamada *taxonomia clássica* agrupa os métodos de planejamento em três grandes categorias:

1. Métodos baseados em **Mapa de Rotas** (*Roadmaps*)
2. Métodos baseados em **Decomposição de Células** (*Cell Decomposition*)
3. Métodos baseados em **Campo Potencial** (*Potential Field*)

#### 4.3.1. Mapa de Rotas

Os métodos baseados em Mapa de Rotas, considerados uma das primeiras abordagens no planejamento de caminho, objetivam capturar a conectividade entre as diferentes regiões de  $C_{livre}$  por meio de uma rede de curvas 1-dimensional, isto é, um *mapa de rotas*. Com a estrutura mapa de rotas construída, o problema do planejamento resume-se a um processo de busca por um caminho  $\tau$  que conecte  $q_{inicial}$  e  $q_{final}$  entre os demais vértices.

Dentre as diversas abordagens aplicadas na construção de um mapa de rotas, é possível destacar o *grafo de visibilidade* (*visibility graph*) e diagrama de Voronoi (*Voronoi diagram*) como as técnicas mais populares. [Molina 2014] destaca que essas duas técnicas, individualmente, retratam duas das principais preocupações da navegação autônoma: A busca por caminhos mais curtos e a busca por caminhos mais seguros.

##### 4.3.1.1. Grafos de Visibilidade

O método de grafo de visibilidade foi inicialmente introduzido por [Nilsson 1969], sendo uma das primeiras estratégias de planejamento de caminho desenvolvidas. Esse método tem o objetivo de construir um caminho livre de obstáculos através de uma linha poligonal

que conecta o ponto inicial  $q_{inicial}$  e o ponto final  $q_{final}$  através dos vértices dos obstáculos do ambiente representados no  $C_{space}$  [Latombe 2012].

Dessa forma, um grafo de visibilidade pode ser definido como um grafo  $G$  não-dirigido, que possui as seguintes características:

- Os vértices de  $G$  correspondem a  $q_{inicial}$ ,  $q_{final}$  e os vértices de  $C_{obs}$ .
- Dois vértices de  $G$  são conectados por uma aresta se e somente o segmento de reta unindo eles for também uma aresta de  $C_{obs}$ , ou se este segmento estiver totalmente contido no espaço livre  $C_{livre}$ , com a possível exceção de suas extremidades.

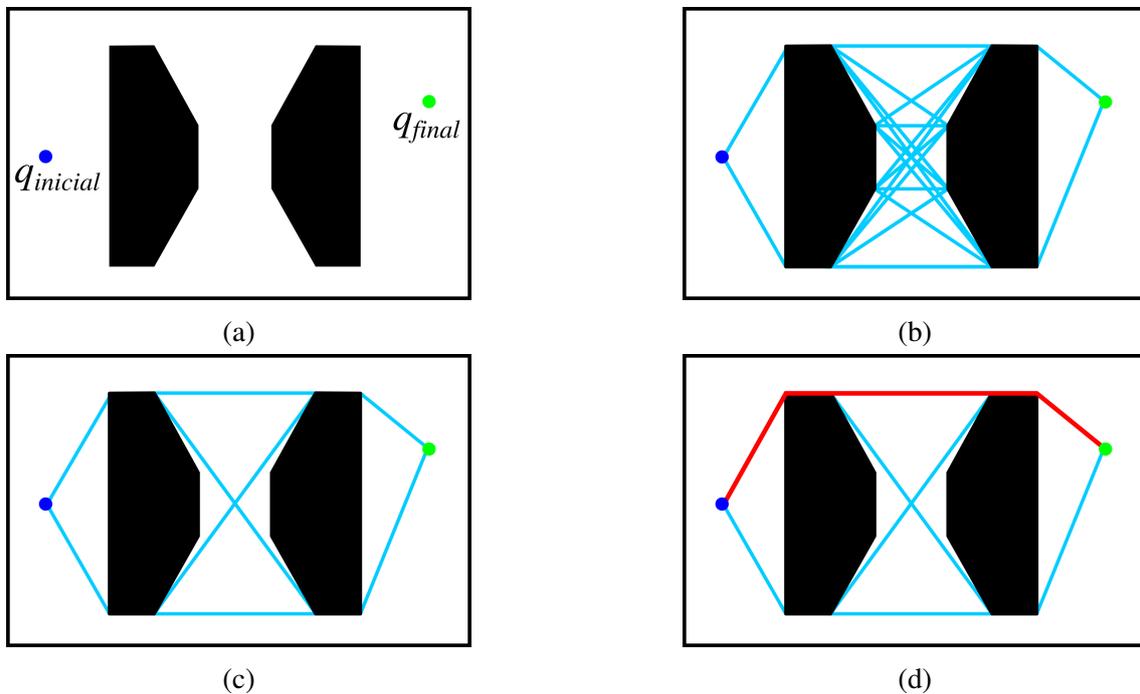


Figura 4.6: Planejamento com Grafo de Visibilidade. (a) Mapa no espaço de configuração. (b) Arestas do Grafo de visibilidade. (c) Arestas do Grafo de visibilidade reduzido. (d) Caminho livre de obstáculos encontrado.

Após a construção do grafo de visibilidade, é necessário realizar um algoritmo de busca para encontrar o caminho a ser percorrido pelo robô. Nessa etapa, geralmente procura-se pelo menor caminho possível de acordo com a distância Euclidiana no  $\mathbf{R}^2$ .

Essa metodologia da construção do grafo pode ter seu desempenho otimizado se percebermos que algumas arestas de  $G$  nunca serão utilizadas, uma vez que elas geram um maior caminho a ser percorrido. Assim, dados dois obstáculos, são úteis apenas as arestas que são tangentes a ambos os obstáculos. Esse grafo pode ser obtido pela eliminação de segmentos não-tangentes, denominamos *grafo de visibilidade reduzido*  $G'$ . Se em tal grafo existir um caminho que une  $q_{inicial}$  e  $q_{final}$ , é possível afirmar que  $G'$  conterá os segmentos de linhas que correspondem ao caminho de menor distância. [Otoni 2000].

### 4.3.1.2. Diagrama de Voronoi

No planejamento baseado no clássico diagrama de Voronoi [Dirichlet 1850], o espaço livre é mapeado por uma rede de arcos 1-dimensional, que visam maximizar a distância entre o robô e a região  $C_{obs}$ . Após o diagrama ser construído, as configurações inicial e final são conectadas aos pontos pertencentes ao mapa de rotas mais próximos à  $q_{inicial}$  e  $q_{final}$ , respectivamente. A Figura 4.7 ilustra um planejamento de caminho com diagrama de Voronoi.

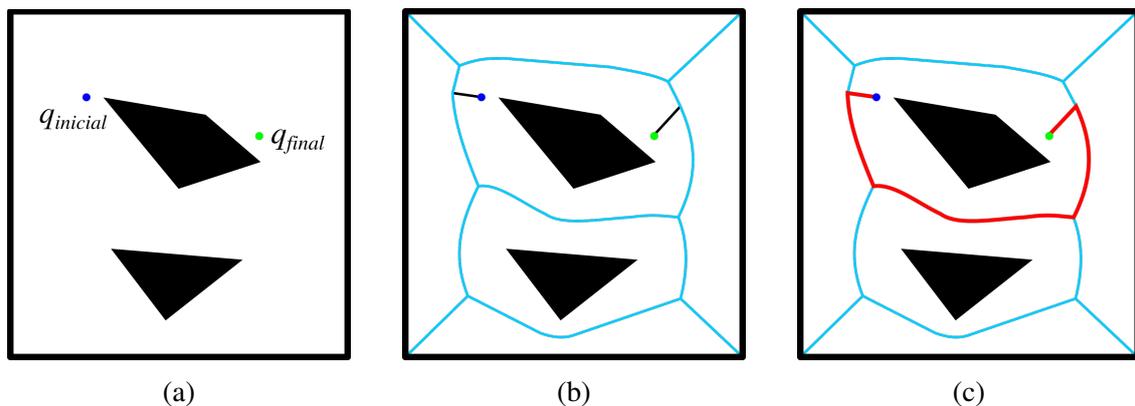


Figura 4.7: Planejamento de caminho baseado em Diagrama de Voronoi. (a) Mapa no espaço de configurações. (b) Diagrama de Voronoi calculado para os vértices do obstáculo. (c) Caminho encontrado. Ilustração adaptada de [Latombe 1991]

Para os métodos basados em mapa de rotas, uma etapa fundamental é a busca pelo menor caminho dentro do grafo, pois após a construção do grafo, o planejamento de caminho resume-se a uma busca por um caminho entre vértices e arcos de um grafo. Por meio de um algoritmo de busca é possível percorrer o grafo de maneira sistemática, explorando os seus vértices (nós) e arcos, a fim de obter o melhor caminho entre os nós representados pela configuração  $q_{inicial}$  e pela configuração  $q_{final}$ .

Conforme [Russell and Peter Norvig 1995], as estratégias de buscas se distinguem pela ordem em que os nós do grafo são examinados, sendo assim, classificados em Algoritmos de Busca não-informada ou busca cega (*Uninformed search*) e Algoritmos de busca informada ou busca heurística (*Informed search*). Nas abordagens de busca não-informadas é possível citar os algoritmos de Busca em Largura ou *Breadth-first* [Lee 1961, Even 1979], Busca em Profundidade ou *Depth-first* [Tarjan 1971], Busca de custo uniforme [Russell and Peter Norvig 1995] e o algoritmo de Dijkstra [Dijkstra 1959].

Nas abordagens de busca informada existe um componente fundamental nos algoritmos, a chamada *função heurística*, denotada por  $h(n)$ , que retorna um custo estimado entre um determinado nó e o nó objetivo, no caso, o nó que representa a configuração final. O algoritmo de Busca pela melhor escolha ou Busca gulosa (*Best-first*) [Pearl 1984] e o algoritmo A\* [Hart et al. 1968, Hart et al. 1972] são exemplos de buscas heurísticas.

### 4.3.2. Decomposição em Células

Nos métodos baseados em decomposição de células, o espaço é decomposto em um conjunto de regiões não sobrepostas, chamadas células. Na estratégia de decomposição exata em células, o espaço de configuração livre é decomposto em um conjunto de regiões convexas, construídas a partir dos limites geométricos da região de obstáculos, considerando que  $C_{obs}$  é uma região poligonal [Latombe 1991]. O método trata cada célula computada como um nó único que será armazenado num grafo, o qual representa a conectividade das regiões livres. O próximo passo é a busca pelas células que levam da configuração inicial até a configuração final. Por meio das células selecionadas, um caminho pode facilmente ser traçado seguindo os nós do grafo correspondentes às células convexas.

A Figura 4.8 ilustra uma decomposição exata de um espaço bidimensional povoado por dois obstáculos poligonais.

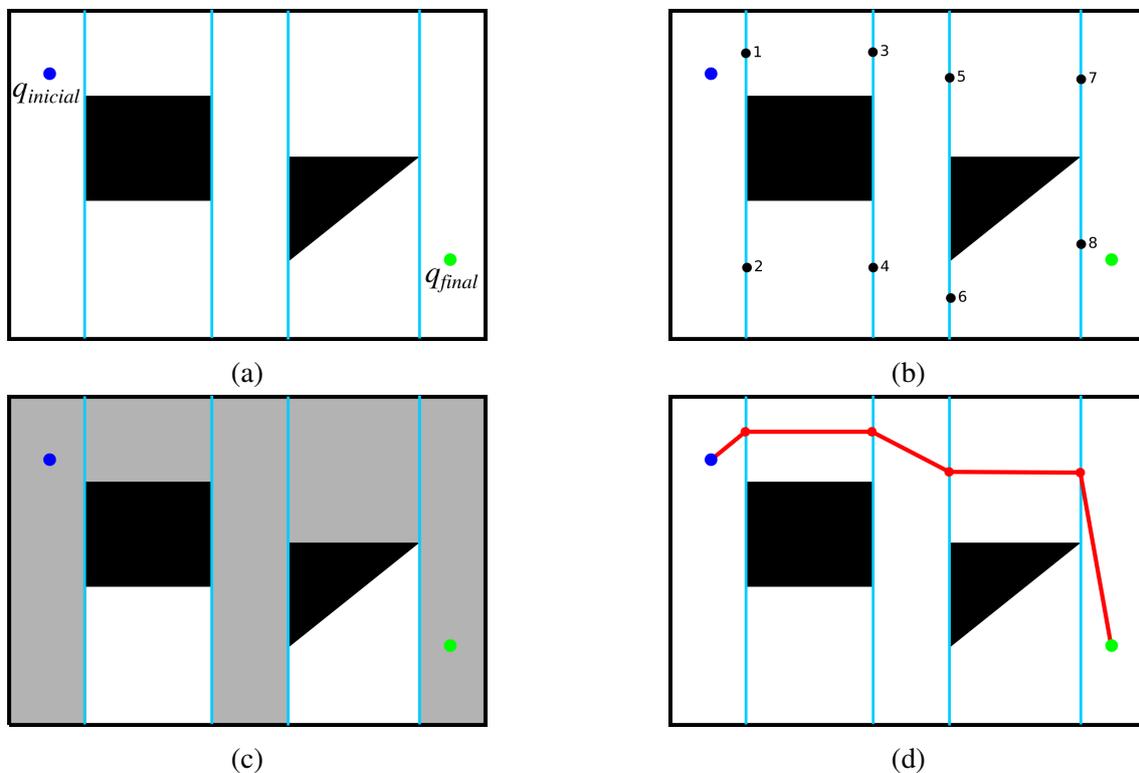


Figura 4.8: Planejamento baseado em Decomposição em Células. (a) Mapa decomposto em regiões convexas. (b) Cada célula é representada por um nó num grafo. (c) Células necessária para encontrar o menor caminho. (d) Caminho encontrado.

De modo similar aos métodos baseados em mapa de rotas, os métodos baseados em decomposição de células têm como principal objetivo a criação de um grafo não dirigido capaz de capturar a conectividade entre as diferentes regiões do espaço livre. Conforme [Molina 2014], a principal diferença entre as duas abordagens é que nos mapas de rotas se preocupa principalmente na forma como as arestas (arcos entre dois nós) são definidas, na decomposição em células a atenção é voltada na forma como os nós (células) que compõem o grafo são definidos, fazendo com que as arestas sejam definidas trivialmente a partir da simples relação de adjacência entre duas células.

### 4.3.3. Campos Potenciais

Nos métodos baseados em Campos Potenciais [Khatib 1986], sugere-se que o robô seja uma partícula movendo-se sob a influência de um campo de potencial artificial, tal que os obstáculos geram um campo potencial de repulsão contra o robô e a configuração final o robô. Dessa forma, a cada configuração, a direção do movimento do robô é determinada pela força resultante proveniente do campo potencial.

A Figura 4.9 um ambiente com três obstáculos mapeado para o espaço de configurações é discretizado em uma grade regular. A cada posição da grade, um potencial é atribuído com base em uma função potencial  $U(q)$ , definida como segue:

$$U(q) = U_{atr}(q) + U_{rep}(q). \quad (5)$$

sendo que  $U_{atr}(q)$  é a função de potencial atrativo associada à configuração final e  $U_{rep}(q)$  é a função de potencial repulsivo, associada à região de obstáculos.

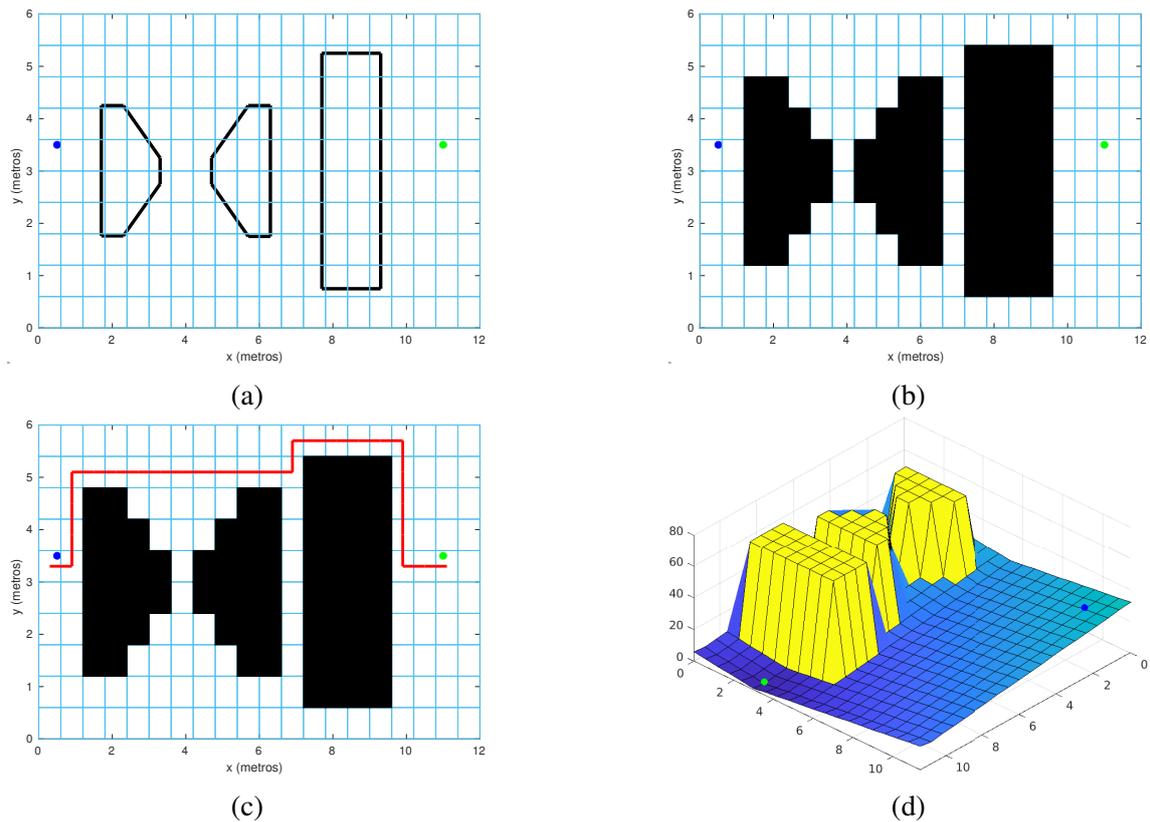


Figura 4.9: Ilustração do método Campos Potenciais. (a) Espaço de configuração com dois obstáculos representado por uma grade. (b) As regiões da grade ocupadas pelos obstáculos são consideradas como obstáculos, expandindo a região  $C_{obs}$ . (c) Caminho livre de obstáculos gerado. (d) Região representando o campo potencial gerado.

O planejamento é realizado de forma implícita onde a configuração do robô é deslocada incrementalmente na direção de uma força artificial  $F(q)$ , sendo que, as ações de controle recebidas pelo robô são geradas a partir dessa força que segue o gradiente

negativo do campo potencial, isto é:

$$F(q) = -\nabla U(q). \quad (6)$$

O método Campo Potencial funciona como um método de otimização baseado no gradiente que busca por um mínimo global, nesse caso, a configuração final. A região mostrada na Figura 4.9(d) ilustra o espaço de busca a ser minimizado.

#### 4.4. Métodos de Planejamento Baseados em Amostragem

Os métodos de Planejamento Baseados em Amostragem (*Sampling-Based Path Planning*) ou Planejamento Probabilístico são estratégias que surgiram inicialmente durante a década de 1990, capazes de resolver eficientemente problemas de planejamento em altas dimensões através de uma representação aproximada do espaço de configuração, o que permite uma redução considerável do custo computacional associado [LaValle 2006].

O planejamento probabilístico tem como principal característica a possibilidade de resolver um problema de planejamento sem a necessidade da computação explícita da região  $C_{obs}$ , diferentemente dos métodos da taxonomia clássica. Nessa estratégia, o espaço de configurações é representado de forma aproximada por meio de amostras aleatórias, normalmente utilizando uma determinada distribuição de probabilidade. Os pontos amostrados são conectados uns aos outros de maneira incremental, resultando, normalmente, num grafo que representa a região livre do  $C_{space}$ . As duas abordagens principais que representam os métodos probabilísticos, são eles o RRT [Lavelle 1998] e o PRM [Kavraki et al. 1996], comentados brevemente nas subseções seguintes:

##### 4.4.1. Árvore Aleatória de Exploração Rápida

A Árvore Aleatória de Exploração Rápida (*Rapidly-Exploring Random Tree - RRT*) é um algoritmo de planejamento de caminho probabilístico introduzido por [Lavelle 1998] como uma eficiente estratégia para realizar buscas em espaços de altas dimensões. De forma geral, o RRT constrói uma árvore que se expande por  $C_{livre}$  partindo da configuração  $q_{inicial}$  em direção à configuração  $q_{final}$ , evitando obstáculos, através da amostragem de nós aleatoriamente pelo espaço livre por meio de uma distribuição polarizada. O funcionamento do RRT básico pode ser observado na Figura 4.10.

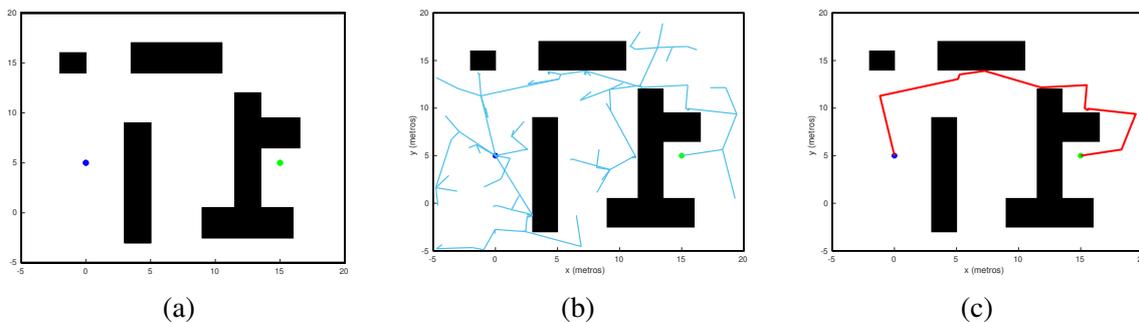


Figura 4.10: Planejamento de caminho com RRT. (a) Mapa representado no espaço de configurações. (b) Árvore aleatória formada. (c) Caminho encontrado a partir da árvore.

Nesse método, uma configuração auxiliar  $q_{aux}$  é amostrada aleatoriamente em  $C_{livre}$ . Considerando que a árvore já tenha alguns nós, é realizada uma busca para encontrar o nó  $q_{prox}$  mais próximo de  $q_{aux}$ . Se for possível conectar  $q_{prox}$  e  $q_{aux}$  através de um segmento de linha,  $q_{aux}$  se torna o novo nó da árvore. Se não for possível, é calculado um segmento de linha até a configuração livre mais próxima de  $q_{aux}$ , onde não haja colisão. Esse procedimento ocorre até que seja possível conectar as configurações inicial e final por meio da árvore gerada.

#### 4.4.2. Mapa de Rotas Probabilístico

O Mapa de Rotas Probabilístico (Probabilistic Roadmap) RPM é uma técnica de planejamento baseado em amostragem com características similares ao RRT. Introduzido por [Kavraki et al. 1996], o PRM visa representar a região  $C_{livre}$  de maneira aproximada através de um grafo unidirecional, enquanto que o RRT constrói uma árvore para o mesmo fim. A Figura 4.11 apresenta um exemplo utilizando o PRM.

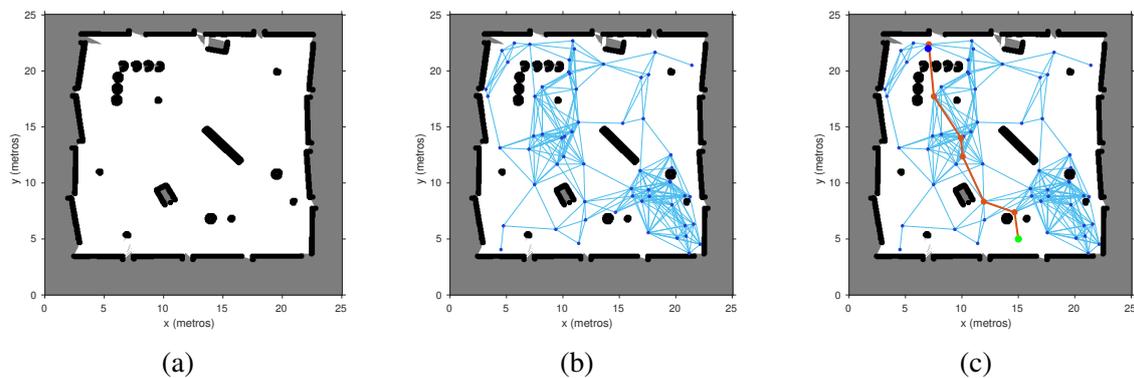


Figura 4.11: Ilustração do processo de planejamento por PFM. (a) Um mapa calculado. (b) Grafo formado no espaço livre. (c) Menor caminho extraído do grafo. Imagens de simulação obtida em [Matlab 2019]

Nesse método, um grafo é formado por meio de algumas configurações amostradas aleatoriamente pelo espaço livre. A partir de nós sementes, um grafo é montado, buscando conectar as sementes. Quanto mais pontos amostrados, maior será a probabilidade de cobertura de  $C_{livre}$  pelo grafo. Por meio de uma estratégia de busca, é possível encontrar um caminho factível pelo grafo.

#### 4.4.3. O Método da Espuma Probabilística

O método da Espuma Probabilística (*Probabilistic Foam Method* - PFM), introduzido por [Silveira and Alsina 2016], é um método de planejamento de caminho baseado em amostragem, desenvolvido com o objetivo de garantir uma região livre de obstáculos para que o robô possa realizar manobras seguras. Nesse método, uma região chamada bolha formada por um conjunto de configurações livres e limitadas por uma superfície definida como uma  $(n-1)$ -esfera. As bolhas se expandem no espaço livre de configurações sobre a superfície de outras bolhas, formando uma estrutura denominada espuma probabilística (organizada como uma árvore de busca) que é capaz de realizar uma cobertura aproximada do espaço livre.

A espuma se propaga no espaço livre a partir da configuração inicial, onde uma bolha é expandida até o obstáculo mais próximo. Um conjunto de bolhas (filhas) são expandidas na superfície da bolha inicial (bolha mãe), posicionadas de maneira aleatória, concluindo a primeira geração. Uma nova bolha mãe é selecionada dentre as filhas expandidas anteriormente, dentro de um ciclo iterativo que termina quando uma bolha engloba a configuração final (Figura 4.12(b)). Dessa maneira, por meio de uma busca partindo da bolha final (bolha que englobou  $q_{final}$ ) até a bolha inicial, seguindo o grau de parentesco das bolhas, é possível encontrar uma estrutura chamada *rosário* que formada um conjunto de bolhas sobrepostas a fim de extrair um caminho livre de obstáculos (Figura 4.12(c)).

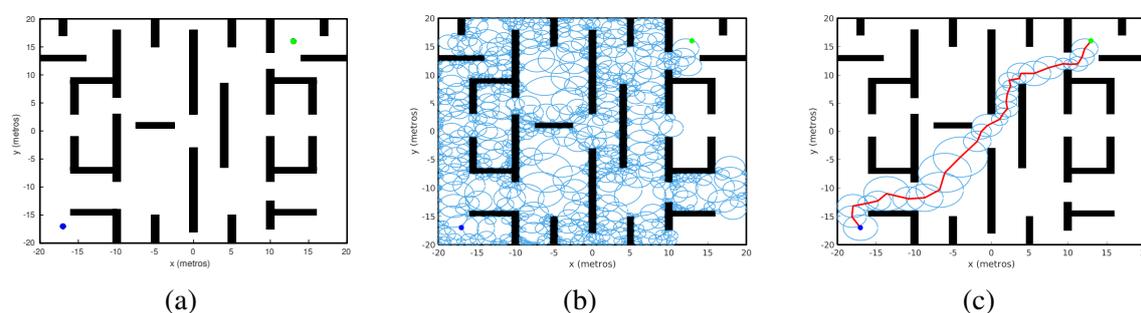


Figura 4.12: Planejamento por PFM. (a) Mapa de um labirinto. (b) Espuma probabilística formada. (c) Rosário extraído da espuma e caminho encontrado.

Esse método é ideal para aplicações em robótica assistiva onde existe a necessidade de garantir um maior grau de segurança nos movimentos. Um exemplo de aplicação pode ser observado em [Nascimento et al. 2018]b onde o PFM foi aplicado no planejamento dos movimentos de uma órtese ativa para os membros inferiores com o objetivo de transpor um obstáculo. Na Figura 4.13(a) é possível observar a região de obstáculo e uma espuma cobrindo o espaço livre. As variáveis de configuração são o ângulo da junta do quadril, o ângulo da junta do joelho e o movimento horizontal do quadril. Na Figura 4.13(b) o rosário é extraído da espuma. Na Figura 4.13(c) um caminho é calculado através do rosário e, finalmente, na Figura 4.13(d) é mostrado o movimento de transposição de obstáculo realizado para o caminho planejado.

Algumas variantes dessa técnica foram desenvolvidas como mostra o trabalho de [Nascimento et al. 2018]a onde o PFM original foi modificado, aplicando uma nova estratégia de propagação da espuma, inspirada no processo de expansão da árvore aleatória do RRT. O principal objetivo dessa estratégia foi a geração de caminhos mais curtos.

## 4.5. Ferramentas para Planejamento

Nesta seção serão apresentadas duas poderosas ferramentas para auxiliar no processo de planejamento de caminho para robôs autônomos. A primeira é uma biblioteca que contém a implementação de vários métodos de planejamento baseados em amostragem que estão no estado da arte. A segunda é uma *framework* que engloba diversas estratégias para auxiliar na programação de robôs autônomos.

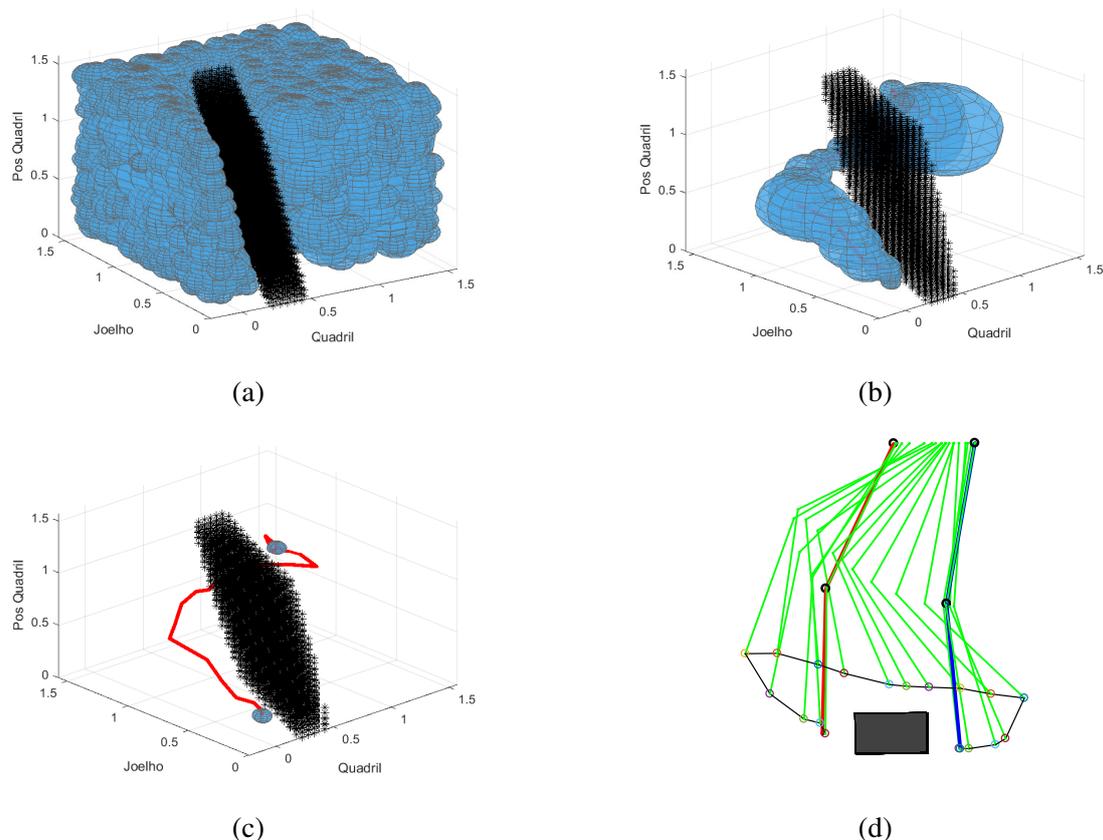


Figura 4.13: Simulação do planejamento de caminho para uma transposição de obstáculos por uma órtese ativa. (a) Espuma probabilística formada. (b) Rosário extraído da espuma. (c) Caminho encontrado. (d) Movimento da perna executado. [Nascimento et al. 2018]b

#### 4.5.1. A biblioteca OMPL

A biblioteca OMPL (*Open Motion Planning Library*) [OMPL 2019] consiste de um conjunto de métodos de planejamento de caminho baseados em amostragem que estão no estado da arte. É possível desenvolver aplicações utilizando métodos de planejamento de caminho como RRT e PRM, por exemplo, podendo ser implementados em linguagens como C++ e Python. Além disso, a biblioteca OMPL conta com classes para trabalhar com análise de colisões.

Além da biblioteca (conjunto das classes), é possível instalar uma aplicação com interface gráfica OMPL.app, onde é possível testar visualmente os métodos de planejamento com ambientes que simulam problemas clássicos de planejamento. A Figura 4.14 mostra a interface da aplicação OMPL.APP.

Nessa aplicação é possível abrir um ambiente para a simulação e um robô para realizar alguma tarefa nesse ambiente. Nas aba *Problem* é possível configurar o problema que você deseja simular. É possível selecionar o tipo de robô, definir a pose inicial e a pose final do robô dentro do ambiente. Na segunda aba *Planner* é possível selecionar qual estratégia de planejamento será utilizada, além de realizar configurações de parâmetros dos planejadores. Apenas com esses ajustes já será possível realizar uma simulação. A

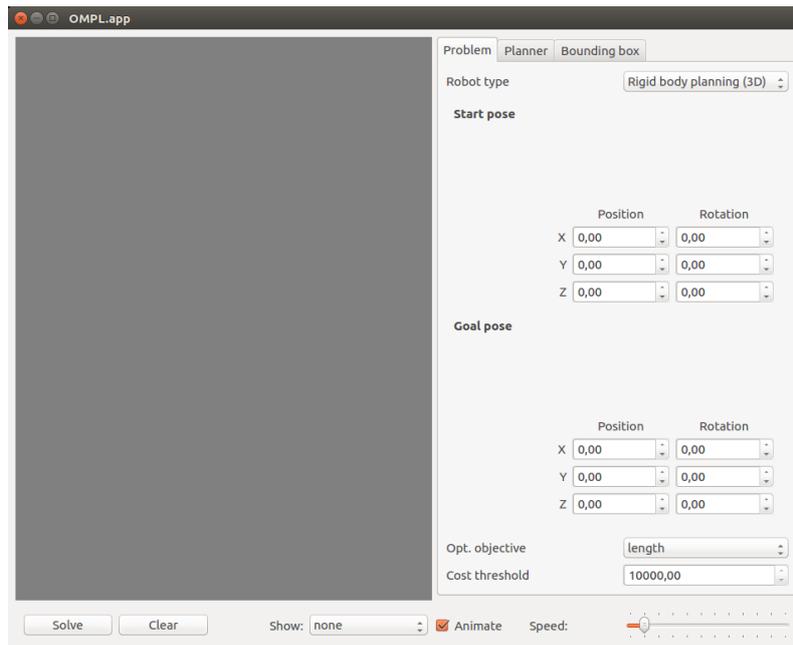


Figura 4.14: Interface gráfica inicial do OMPL.app.

Figuras 4.15-4.17 mostra uma simulação realizada com o OMPL.app para o problema clássico de planeamento do *carregador de piano*.



Figura 4.15: Ambiente de simulação iniciado. As configurações inicial e final são representadas pelos pianos.

Esse ambiente de apartamento está disponível para testes com todos os métodos de planeamento disponíveis na biblioteca. Além disso, é possível implementar um método de planeamento externo utilizando a biblioteca OMPL e testar utilizando a ferramenta OMPL.app com esse ou com outros ambientes.

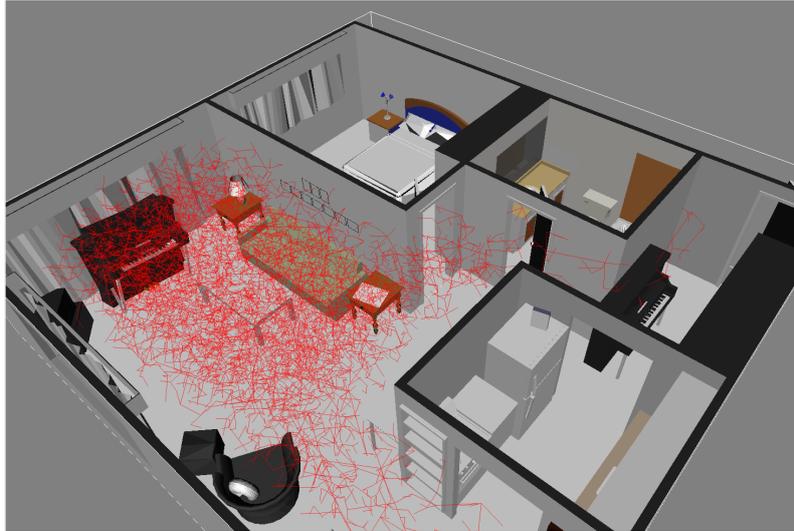


Figura 4.16: Árvore aleatória calculada partindo do piano inicial (ao fundo) para o piano final (à direita).

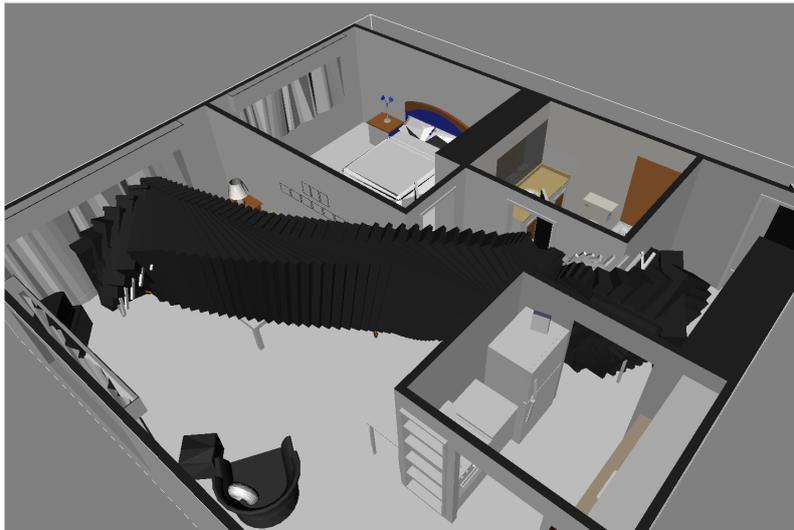


Figura 4.17: Planejamento realizado. Conjunto de configurações necessárias que o piano deve assumir para ser deslocado da configuração inicial até a configuração final.

#### 4.5.2. A *framework* MoveIt!

A *framework* MoveIt [MoveIt 2019] é uma ferramenta poderosa que engloba uma série de bibliotecas e aplicações para realizar projetos simples aos mais complexos de robótica autônoma. Essa *framework* é executada dentro do sistema ROS (*Robot Operating System*), uma completa plataforma para desenvolvimento de robôs, que fornece funcionalidades de um sistema operacional.

A *framework* MoveIt possibilita trabalhar com visão computacional, planejamento de caminho (fazendo uso de bibliotecas como OMPL), sistemas para cálculo de colisão, biblioteca para cálculo de cinemática, controle, entre outros. A Figura 4.18 mostra a

interface do sistema de visualização 3D Rviz, com a simulação de um braço robótico Panda.

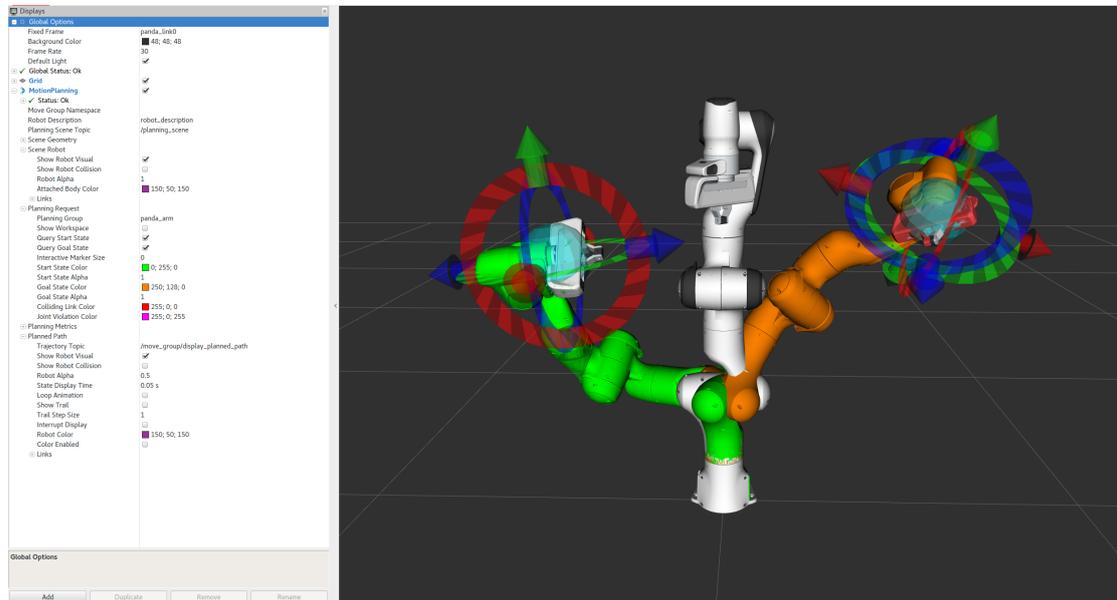


Figura 4.18: Fluxograma com os problemas fundamentais da robótica autônoma.

Na imagem, o braço robótico está sendo preparado para uma tarefa de planejamento. A imagem esverdeada do robô representa a sua configuração inicial e a imagem alaranjada do robô está representando a configuração final definida pelo usuário.

## 4.6. Conclusões

Neste documento foi apresentada uma introdução sobre o problema do planejamento de caminho para sistemas robóticos autônomos. Como mostrado, planejar caminhos livre de obstáculos é uma etapa fundamental no desenvolvimento de sistemas robóticos onde se necessita autonomia na realização de tarefas. Entretanto, é necessária uma análise do problema robótico a fim para definir uma estratégia de planejamento a ser implementada para obtenção de melhores resultados.

Para problemas mais simples, com sistemas robóticos de dois graus de liberdade (2 DoF), por exemplo, os métodos clássicos são capazes de realizar o planejamento de forma ideal. Porém, para problemas mais complexos, os métodos baseados em amostragem são os mais indicados. Além disso, é necessário analisar a natureza do problema, pois se o problema exige caminhos mais seguros, como é o caso de robôs assistivos, é indicado métodos que calculam não só caminhos factíveis, mas que também garantam regiões de livre manobrabilidade.

Finalmente, a fim de facilitar o desenvolvimento e implementação de sistemas robóticos autônomos, algumas bibliotecas e *frameworks* estão disponíveis gratuitamente para uso. Além de facilitar o desenvolvimento, essas ferramentas também ajudam no processo de ensino e aprendizagem de conceitos da robótica autônoma.

#### 4.7. Currículo dos autores

**Luís Bruno Pereira do Nascimento** possui Bacharelado em Ciência da Computação pela Universidade Estadual do Piauí (UESPI) e Mestrado em Engenharia Elétrica e de Computação pela Universidade Federal do Ceará (UFC). Atualmente é aluno de doutorado pelo Programa de Engenharia Elétrica e de Computação (PPgEEC) pela Universidade Federal do Rio Grande do Norte (UFRN). Suas áreas de interesse incluem robótica autônoma, otimização e aprendizado de máquina.

**Diego da Silva Pereira** possui graduação em Redes de Computadores pelo Instituto Federal do Rio Grande do Norte (IFRN), Mestrado em Ciência da Computação pela Universidade Estadual do Rio Grande do Norte (UERN) e está em doutoramento no Programa de Pós-graduação em Engenharia Elétrica e de Computação (PPgEEC) pela UFRN. Atualmente é professor no IFRN com pesquisas na área de redes de comunicação sem fio aplicadas à sistemas robóticos autônomos.

**Vitor Gaboardi dos Santos** possui graduação em Engenharia Elétrica na Universidade Federal do Rio Grande do Norte (UFRN) com período sanduíche na University of Kansas (USA). Atualmente é aluno de Mestrado no Programa de Pós-graduação em Engenharia Mecatrônica pela UFRN com pesquisas relacionadas a visão computacional aplicada à robótica assistiva.

**Daniel Henrique Silva Fernandes** possui Bacharelado em Engenharia Mecatrônica pela Universidade Federal do Rio Grande do Norte (UFRN). Atualmente é aluno de Mestrado no Programa de Pós-graduação em Engenharia Mecatrônica pela UFRN com pesquisas relacionadas a robótica assistiva e visão computacional.

**Pablo J. Alsina** possui graduação em Engenharia Elétrica (1987), mestrado na área de controle de motores de indução (1991) e doutorado em Engenharia Elétrica com tema em controle de manipuladores robóticos (1996), pela Universidade Federal da Paraíba. Atualmente é professor titular do Departamento de Engenharia de Computação e Automação (DCA). É professor nos Programas de Pós-Graduação em Engenharia Mecatrônica (PPGEM) e em Engenharia Elétrica e de Computação (PPgEEC) da Universidade Federal do Rio Grande do Norte (UFRN), onde é chefe do Laboratório de Robótica. Desenvolve pesquisas em robótica, nas áreas de controle, planejamento e percepção robótica, com aplicações em robótica assistiva, robótica móvel e Veículos Aéreos Não Tripulados.

#### Referências

- [Al-Bluwi et al. 2012] Al-Bluwi, I., Siméon, T., and Cortés, J. (2012). Motion planning algorithms for molecular simulations: A survey. *Computer Science Review*, 6(4):125–143.
- [Dijkstra 1959] Dijkstra, E. W. (1959). A Note on Two Problems in Connexion with Graphs. *Numer. Math.*, 1(1):269–271.
- [Dirichlet 1850] Dirichlet, G. L. (1850). Über die Reduction der positiven quadratischen Formen mit drei unbestimmten ganzen Zahlen. *Journal für die reine und angewandte Mathematik*, 40:209–227.

- [Edelkamp and Plaku 2014] Edelkamp, S. and Plaku, E. (2014). Multi-goal motion planning with physics-based game engines. In *2014 IEEE Conference on Computational Intelligence and Games*, pages 1–8.
- [Even 1979] Even, S. (1979). *Graph Algorithms*. W. H. Freeman & Co., New York, NY, USA.
- [Hart et al. 1968] Hart, P. E., Nilsson, N. J., and Raphael, B. (1968). A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107.
- [Hart et al. 1972] Hart, P. E., Nilsson, N. J., and Raphael, B. (1972). Correction to "A Formal Basis for the Heuristic Determination of Minimum Cost Paths". *SIGART Bull.*, (37):28–29.
- [Kavraki et al. 1996] Kavraki, L., Svestka, P., Latombe, J.-C., and Overmars, M. (1996). Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12(4):566–580.
- [Khatib 1986] Khatib, O. (1986). Real-Time Obstacle Avoidance for Manipulators and Mobile Robots. *The International Journal of Robotics Research*, 5(1):90–98.
- [Latombe 1991] Latombe, J.-C. (1991). *Robot Motion Planning*. Springer US, Boston, MA.
- [Latombe 2012] Latombe, J.-C. (2012). *Robot motion planning*, volume 124. Springer Science & Business Media.
- [Lavalle 1998] Lavalle, S. M. (1998). Rapidly-exploring random trees: A new tool for path planning. Technical report, Computer Science Dept., Iowa State University.
- [LaValle 2006] LaValle, S. M. (2006). *Planning Algorithms*. Cambridge University Press, New York, NY, USA.
- [Lee 1961] Lee, C. Y. (1961). An Algorithm for Path Connections and Its Applications. *IRE Transactions on Electronic Computers*, EC-10(3):346–365.
- [Lozano-Perez 1983] Lozano-Perez, T. (1983). Spatial Planning: A Configuration Space Approach. *IEEE Transactions on Computers*, C-32(2):108–120.
- [Lozano-Pérez and Wesley 1979] Lozano-Pérez, T. and Wesley, M. A. (1979). An Algorithm for Planning Collision-free Paths Among Polyhedral Obstacles. *Commun. ACM*, 22(10):560–570.
- [Matlab 2019] Matlab (2019). Path planning in environments of different complexity. *Matlab® Robotics System Toolbox Documentation*. Online em: <https://www.mathworks.com/help/robotics/examples/path-planning-in-environments-of-difference-complexity.html>; acessado em 16-Agosto-2019.

- [Molina 2014] Molina, L. (2014). *Planejamento de Movimento Para Robôs Móveis Baseado na Condição de Horizonte Continuado (CHC)*. Doutorado, Universidade Federal de Campina Grande, Pernambuco, Brasil.
- [MoveIt 2019] MoveIt (2019). MoveIt Motion Planning Framework. Online em: <https://moveit.ros.org>; acessado em 18-Agosto-2019.
- [Murphy 2000] Murphy, R. R. (2000). *Introduction to AI Robotics*. Intelligent robotics and autonomous agents. MIT Press, Cambridge, Mass. OCLC: 247251178.
- [Nascimento et al. 2018] Nascimento, L. B. P., Pereira, D. S., Alsina, P. J., Silva, M. R., Fernandes, D. H. S., Roza, V. C. C., and Sanca, A. S. (2018). Goal-biased probabilistic foam method for robot path planning. pages 199–204, Torres Vedras, Portugal. IEEE.
- [Nascimento et al. 2018] Nascimento, L. B. P., Silva Pereira, D., Sanca Sanca, A., Silva Eugenio, K. J., Silva Fernandes, D. H., Javier Alsina, P., Valério Araujo, M., and Rabello Silva, M. (2018). Safe path planning based on probabilistic foam for a lower limb active orthosis to overcoming an obstacle. In *2018 Latin American Robotic Symposium, 2018 Brazilian Symposium on Robotics (SBR) and 2018 Workshop on Robotics in Education (WRE)*, pages 413–419.
- [Nilsson 1969] Nilsson, N. J. (1969). A Mobile Automaton: An Application of Artificial Intelligence Techniques. Technical report, Sri International Menlo Park Ca Artificial Intelligence Center.
- [OMPL 2019] OMPL (2019). The Open Motion Planning Library. *Kavraki Lab*. Online em: <https://ompl.kavrakilab.org>; acessado em 18-Agosto-2019.
- [Ottoni 2000] Ottoni, G. d. L. (2000). Planejamento de trajetórias para robôs móveis. *Projeto de Graduação em Engenharia de Computação–FURG, Rio Grande*.
- [Pearl 1984] Pearl, J. (1984). *Heuristics: Intelligent Search Strategies for Computer Problem Solving*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.
- [Russell and Peter Norvig 1995] Russell, S. J. and Peter Norvig (1995). *Artificial Intelligence: A Modern Approach*, volume 25. Prentice-Hall, Englewood Cliffs, New Jersey, 1 edition.
- [Sakemoto 2017] Sakemoto, M. (2017). Build an autonomous mobile robot with the Intel® RealSense™ Camera, ROS\*, and SAWR. *Intel® Software*. Online em: <https://software.intel.com/en-us/articles/build-an-autonomous-mobile-robot-with-the-intel-realsense-camera-ros-and-sawr>; acessado em 16-Agosto-2019.
- [Siegwart and Nourbakhsh 2004] Siegwart, R. and Nourbakhsh, I. R. (2004). *Introduction to Autonomous Mobile Robots*. Intelligent robotics and autonomous agents. MIT Press, Cambridge, Mass, 1 edition. OCLC: ocn649700153.
- [Silveira and Alsina 2016] Silveira, Y. S. and Alsina, P. J. (2016). A New Robot Path Planning Method Based on Probabilistic Foam. In *2016 XIII Latin American Robotics Symposium and IV Brazilian Robotics Symposium (LARS/SBR)*, pages 217–222.

- [Tarjan 1971] Tarjan, R. (1971). Depth-first search and linear graph algorithms. In *12th Annual Symposium on Switching and Automata Theory (Swat 1971)*, pages 114–121.
- [Udupa 1977] Udupa, S. M. (1977). *Collision Detection and Avoidance in Computer Controlled Manipulators*. Doutorado, Dept. of Electrical Engineering, California Institute of Technology, California.

## Capítulo

# 5

## Os cinco passos para transformar qualquer site de dados públicos em uma API de dados abertos

Sérgio Souza Costa, Mateus Vitor Duarte Sousa, Micael Lopes da Silva

### *Abstract*

*This text presents, in a practical way, five steps to transform public data sites into open data portals. These steps include (1) modeling and (2) extracting data from a web site; The (3) development and (4) documentation of a REST API; And the (5) deployment in a cloud service.*

### *Resumo*

*Esse texto apresenta de maneira prática, cinco passos para transformar sites com dados públicos em portais de dados abertos. Esses passos incluem a (1) modelagem e (2) extração dos dados de um site web; (3) O desenvolvimento e (4) documentação de uma API REST; E a (5) implantação em um serviço de nuvem.*

### **5.1. Introdução**

No Brasil o acesso à informação sobre os recursos públicos está previsto pela Constituição de 1988, mas foi reforçado através Lei de Acesso à Informação (Lei nº 12.527/2011<sup>1</sup>). Em resposta a esta demanda, instituições públicas têm disponibilizado um grande volume de informações públicas na Web. Contudo, muitos destes sites foram desenvolvidos para serem acessados por pessoas, e não são compreensíveis por algoritmos computacionais, impedindo assim uma busca automatizada das informações. Tornando a extração de uma informação específica impraticável ou até impossível em alguns casos. Quando isso ocorre o interessado precisa entrar em contato com a instituição provedora, para requisitar os dados de interesse. Esse processo pode demandar um tempo maior que o esperado pelo solicitante, além de consumir tempo da instituição detentora dos dados. Dado este cenário, muitas instituições estão migrando para o paradigma de dados abertos.

---

<sup>1</sup>A lei pode ser acessada integralmente em [http://www.planalto.gov.br/ccivil\\_03/\\_ato2011-2014/2011/lei/112527.htm](http://www.planalto.gov.br/ccivil_03/_ato2011-2014/2011/lei/112527.htm)

Nesse paradigma, os dados abertos precisam ser publicados de acordo com alguns princípios. Por exemplo, ser completos, primários, atuais, acessíveis, compreensíveis por máquina, não proprietários e livres de licença [Lessig 2007]. A principal vantagem é possibilitar tanto o desenvolvimento de aplicativos pela comunidade quanto suportar a utilização de algoritmos de análise de dados. Muitas destas análises seriam impraticáveis de serem realizadas sem a utilização de um algoritmo computacional para automatização da recuperação, manipulação e visualização dos dados. Segundo [Vaz 2011], atualmente espera-se que tudo que não seja sigiloso esteja disponível na internet no formato de dados abertos, ou seja, não basta disponibilizar quando solicitado. O paradigma de dados abertos é muito importante para a transparência de instituições públicas, onde elas passam a disponibilizar todos os dados de modo bruto e atualizado ao invés de proverem os dados apenas quando solicitados. Deste modo, a comunidade pode usar estes dados tanto para fiscalizar quanto para gerar soluções úteis a sociedade [Keßler and Kauppinen 2012, Daga et al. 2015, Molloy 2011, Dietrich et al. 2009, Zablith et al. 2011]

Existem alguns portais que facilitam a busca por dados. Por exemplo, no Brasil o portal <http://dados.gov.br/dataset> agrupa diversos conjunto de dados por categorias e instituições brasileiras. Não restrito a dados nacionais, a Google possui um portal específico para a busca de dados abertos: <https://toolbox.google.com/datasetsearch>. Porém, muitos dados ainda são acessíveis apenas através de páginas HTML<sup>2</sup> e requerem a interação direta com uma pessoa.

O objetivo desse texto é apresentar uma solução para expor os dados públicos como dados abertos utilizando para isso a extração de dados. Nesse texto, será apresentado então como realizar a extração dos dados e disponibilizá-los através de uma API implantada na plataforma em um serviço de nuvem. Para isso, esse texto está estrutura da seguinte forma. A Seção 5.2 apresenta os pré-requisitos necessários para o desenvolvimento da API. A Seção 5.3 apresenta os 5 passos propostos: (1) identificação e modelagem dos dados, (2) extração dos dados, (3) desenvolvimento, (4) documentação e (5) implantação. A Seção 5.4 mostra algumas experimentações simples e discute algumas possibilidades de evolução. E por fim, a Seção 5.5 apresenta algumas conclusões finais.

## 5.2. Conhecimento prévios e pré-requisitos

Esse texto tem como objetivo apresentar todos os passos de modo simples e de fácil reprodução. O nível de dificuldade dos passos dependerá dos seguintes conhecimentos prévios. É esperado um conhecimento básico de programação e alguma familiaridade com a linguagem de programação Python, e conhecimento básico em HTML (*HyperText Markup Language*). Além disso, antes de seguir a execução dos cinco passos detalhados na Seção 5.3, será necessário atender os seguintes pré-requisitos:

1. Ter instalado e configurado o interpretador da Linguagem Python e o instalador de pacotes PIP (<https://www.python.org/>).
2. Ter uma conta na Plataforma Heroku (<https://www.heroku.com/>).

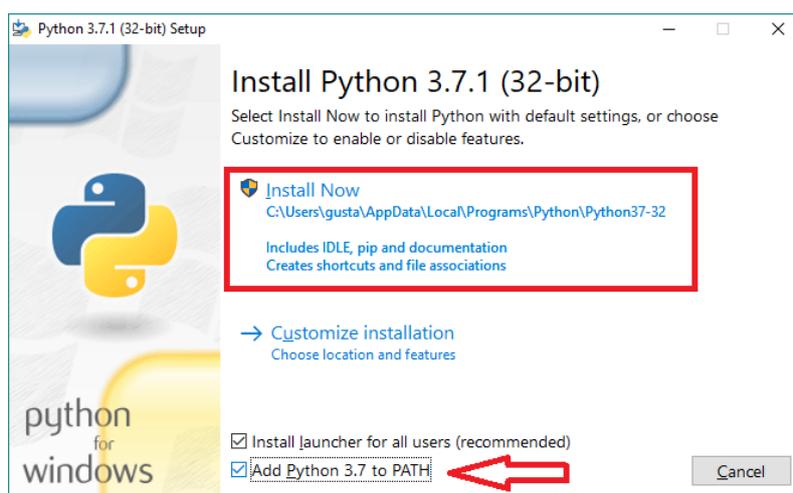
---

<sup>2</sup>HTML do inglês HyperText Markup Language, é uma linguagem de marcação utilizada na construção de páginas na Web com diversos metadados ou *tags* e recursos que são úteis apenas para a visualização através dos navegadores.

3. Ter uma conta no Github (<https://github.com/>).
4. Ter instalado o Git (<https://git-scm.com/>).

A solução apresentada nesse texto usará a linguagem de programação Python. Ela é uma linguagem de alto nível, interpretada, multiparadigma, dinamicamente tipada e com o enfoque na legibilidade de código e facilidade de desenvolvimento. Criada em 1989, por Guido Von Rossum no Instituto de Pesquisa Nacional para Matemática e Ciências da Computação, nos Países Baixos. Ela é mantida pela organização sem fins lucrativos denominada *Python Software Foundation* (<https://www.python.org/psf/>). Atualmente encontra-se na versão 3.7.3 que foi lançada em 25 de março de 2019.

A instalação do Python dependerá do sistema operacional utilizado. No caso do Linux, maioria das distribuições já vem com o Python instalado e configurado para ser executado diretamente pelo terminal. No caso do Windows, é necessário baixar o instalador e prosseguir com a instalação. Durante a instalação, marque a opção que permite ao instalador adicionar o Python na variável de ambiente `PATH`, como na Figura 5.1. Isso permitirá usar o Python diretamente pelo terminal de comandos.



**Figura 5.1. Instalando o Python no Windows. Fonte:** <http://bit.ly/2P0wqNN>

Caso não tenha marcado a seleção como na Figura 5.1, será necessário configurar a variável de ambiente `PATH`. Para isso, será necessário identificar a localização do Python em seu computador e adicioná-la para variável de ambiente. Por exemplo, para um dado computador foi executado o seguinte comando:

```
$ set PATH=C:\Users\sergi\AppData\Local\Programs\Python\Python37-32\Scripts;C:\Users\sergi\AppData\Local\Programs\Python\Python37-32;%PATH%
```

Se está usando o Windows, uma alternativa interessante ao CMD é o terminal `git bash` que é instalado juntamente com o `git`. Caso esteja usando esse terminal, o comando a ser executado seria:

```
PATH=/C/Users/sergi/AppData/Local/Programs/Python/Python37-32:/C/Users/sergi/AppData/Local/Programs/Python/Python37-32/scripts:$PATH
```

Além do Python, será necessário ter uma conta na plataforma Heroku<sup>3</sup> que será usado para a implantação. Ela é uma plataforma como serviço ou PaaS, do inglês *Platform as a Service*, ou seja, ela provê serviços na nuvem que abstraem a infraestrutura de hardware e software que irá hospedar a aplicação. Heroku permite fácil escalabilidade para as aplicações a partir de *dynos*<sup>4</sup>, que são máquinas virtuais que executam um sistema operacional com uma determinada quantidade de memória e processador. Nela as aplicações são executadas em contêineres de modo isolado. Ter uma conta no Heroku é fundamental para a implantação da aplicação, que será o último passo, como detalhado na Seção 5.3.

A plataforma Heroku se integra diretamente com repositórios Github. Então, nesse texto iremos usar tanto o Git quanto o Github para o gerenciamento e a implantação do serviço. O Git é um *software* livre, desenvolvido por Linus Torvalds, para versionamento que foi utilizado durante o desenvolvimento do kernel do linux [Torvalds 2005]. Basicamente o Git monitora todos os arquivos, diretório e subdiretórios dentro do projeto, provendo histórico completo e acompanhamento de revisões sem necessidade de uma rede encarregada pra isso ou uma máquina centralizada [Chacon 2014]. Por outro lado, o Github é uma ferramenta online para armazenamento de projetos gerenciados pelo Git e oferece a opção de hospedagem de projetos *open source* ou privados.

### 5.3. Mapeando os dados públicos para dados abertos em cinco passos

Nesta seção serão apresentados os cinco passos para transformar os dados públicos de um dado site, para o formato de dados abertos. O objetivo dessa seção é ser bem prática e permitir a fácil replicação por alguém com conhecimentos básicos em programação. Para a execução desses passos são usadas diversas tecnologias como mostra a Figura 5.2. Algumas delas já foram destacadas na Seção 5.2 e outras serão apresentadas em cada passo em que for utilizada.

#### 5.3.1. Identificação e modelagem dos dados

Como discutido anteriormente, existem diversos sites com dados relevantes, porém, inacessíveis diretamente por algoritmos e aplicativos. Eles apresentam os dados através de páginas web, porém não disponibilizam o acesso direto aos dados em formatos abertos. Então, a identificação e a modelagem dos dados requer a exploração do site para identificar os dados de interesse.

Neste trabalho será usado o site da transparência do governo do Estado do Maranhão (<http://www.transparencia.ma.gov.br/>). Ele foi lançado em 2015 e apresenta diversas melhorias em relação ao anterior, mas ainda não disponibiliza dados abertos. O acesso a todos os dados requer a interação entre um usuário e um navegador web. Por exemplo, ao acessar o seguinte endereço [---

<sup>3</sup><https://www.heroku.com/about>](http://www.transparencia.</a></p></div><div data-bbox=)

<sup>4</sup><https://devcenter.heroku.com/articles/dynos>



Figura 5.2. Principais tecnologias utilizadas

[ma.gov.br/app/despesas/por-funcao/2019#lista](http://ma.gov.br/app/despesas/por-funcao/2019#lista) o usuário irá visualizar uma página HTML com uma tabela similar a da Figura 5.3, que descreve as despesas de cada função administrativa em 2019.

CÓDIGO	FUNÇÃO	EMPENHADO	LIQUIDADO	PAGO
12	EDUCACAO	1.684.622.426,16	1.390.292.803,31	625.194.524,41
10	SAUDE	1.381.755.880,82	1.153.294.533,10	1.042.897.165,13
09	PREVIDENCIA SOCIAL	1.304.287.119,54	1.275.240.828,87	179.238.233,65
06	SEGURANCA PUBLICA	1.075.738.298,56	1.037.012.947,76	776.244.140,29
28	ENCARGOS ESPECIAIS	902.049.666,77	894.992.848,30	863.561.971,07
02	JUDICIARIA	675.138.826,77	603.905.156,34	556.102.369,34
04	ADMINISTRACAO	529.798.789,14	451.046.595,86	351.247.216,01
03	ESSENCIAL A JUSTICA	417.413.106,68	379.601.831,41	373.304.788,51
01	LEGISLATIVA	328.702.859,43	312.632.771,91	273.589.849,54
14	DIREITOS DA CIDADANIA	272.659.104,45	214.876.489,62	187.946.915,15

Figura 5.3. Despesas por função administrativa em 2019

Observe pela Figura 5.3 que cada linha representa uma função administrativa, com o seu código, nome e valores empenhado, liquidado e pago. No próximo passo, cada um destes dados serão extraídos para depois serem enviados em um formato de dados abertos.

### 5.3.2. Extração dos dados

No segundo passo, os dados selecionados serão extraídos usando uma biblioteca Python denominada *Beautiful Soup*<sup>5</sup>. Essa biblioteca permite automatizar a obtenção de dados de

<sup>5</sup><https://www.crummy.com/software/BeautifulSoup/bs4/doc/> - Mais informações sobre o Beautiful Soup

um *website*, através de um conjunto técnicas denominadas de *Web scraping*. A função de um *web scraper* é verificar certos tipos de informações, extraí-las e agregá-las em novos formatos [Mitchael 2015]. Então, antes de mais nada será necessário já ter instalado e configurado o Python, como discutido na Seção 5.2. Com o Python configurado, já é possível acessá-lo diretamente pelo terminal, incluso o instalador de pacotes `pip`. Com ele instalado, pode-se instalar o `pipenv` através do seguinte comando:

```
$ pip install pipenv
```

O `Pipenv` atua como criador e gerenciador de dependências em ambientes de desenvolvimento e de produção. Em resumo, o `pipenv` integra em um mesmo software as tarefas que eram usualmente desempenhados através do `pip` e do `virtualenv` (<https://virtualenv.pypa.io/>). Ou seja, ele gerencia tanto as dependências quanto se encarrega de criar e carregar um ambiente isolado Python para a aplicação [VanTol 2018].

Com o `pipenv` instalado, crie a pasta para o projeto chamada `transparencia-ma` e depois em um terminal Linux ou Windows (CMD ou Git Bash), execute os seguintes comandos:

```
$ pipenv --three
$ pipenv install requests beautifulsoup4 lxml
```

O primeiro comando irá iniciar um ambiente isolado para a versão 3 do Python, e o segundo irá instalar as três bibliotecas básicas usadas durante esse passo. A biblioteca `requests` é usada para fazer requisições HTTPs, `beautifulsoup4` para a extração de dados e a `lxml` é usada pelo `Beautiful Soup` para analisar as estruturas em HTML ou XML [Richardson 2019].

#### Fica a dica

Use um ambiente de desenvolvimento integrado (IDE) que tenha acesso a terminais de comandos dentro do própria aplicação, como o Visual Studio Code (<https://code.visualstudio.com/>).

Usando uma IDE, ou diretamente pelo terminal, crie um arquivo denominado (`scrapper.py`) com o Código 5.1.

```
1 import requests
2 from bs4 import BeautifulSoup as BS
3
4 url = "http://www.transparencia.ma.gov.br/app/despesas/por-funcao/2019#
   lista"
5 page_response = requests.get(url)
6 page = BS(page_response.text, 'lxml')
7 table = page.find('table')
8 print(table)
```

**Código 5.1. Arquivo `scrapper.py`: Um simples código para testar a instalação**

O Código 5.1 vai ser usado apenas para testar a instalação do ambiente e as bibliotecas. Na linha 5 é feita a requisição da página HTML usando a biblioteca `requests`. Na linha 6, usa-se uma função da biblioteca `Beautiful Soup` para transformar o conteúdo requisitado em um objeto que irá permitir a busca de elementos específicos. Como ocorre na linha 7, que recuperou a primeira tabela (`table`) do resultado da requisição. Para testar esse código, em um terminal execute os seguintes comandos:

```
$ pipenv shell
$ python scrapper.py
```

O primeiro comando ativa o ambiente, enquanto o segundo executa o *script*. A execução dos comandos acima, deverá imprimir na tela a tabela HTML que foi encontrada.

### CheckPoint

Se chegou até aqui, significa que já está com o Python, Pipenv e as bibliotecas funcionando corretamente.

Antes de extrair os dados de interesse é necessário analisar como eles estão estruturados através de *tags* HTML. Usualmente, essa tarefa será executada explorando os recursos presentes nos navegadores web atuais que permitem visualizar o código fonte das páginas. Neste texto, pegou-se uma linha qualquer da tabela impressa pela execução do Código 5.1, tendo como saída o Código 5.2.

```
100 <tr>
101 <td>19</td>
102 <td class="secondLeft">
103 <a href="http://www.transparencia.ma.gov.br/app/despesas/por-funcao
    /2019/funcao/19?">
104     CIENCIA E TECNOLOGIA
105     </a>
106 </td>
107 <td>22.570.913,72</td>
108 <td>17.746.719,47</td>
109 <td>13.433.197,71</td>
110 </tr>
```

### Código 5.2. Trecho de código HTML da tabela de a ser extraída

No Código 5.2, a primeira coluna é o código da função administrativa. A segunda coluna possui o nome da função administrativa e uma URL para mais detalhes sobre as despesas dessa função. As demais colunas apresentam respectivamente os valores empenhado, liquidado e pago. Então, para extrair os dados, apague o código escrito anteriormente e substitua-o pelo Código 5.3.

```
1 import requests
2 from bs4 import BeautifulSoup as BS
3
```

```

4 def despesas_total ():
5     url = "http://www.transparencia.ma.gov.br/app/despesas/por-funcao
        /2019#lista"
6     response = requests.get(url)
7     page = BS(response.text, 'lxml')
8     table = page.find ('table')
9     rows = table.find_all('tr')
10    despesas = []
11    for row in rows[1:]: # testando apenas com 3 linhas
12        cols =row.find_all("td")
13        despesa = {}
14        despesa["nome"] = cols[1].find("a").get_text().strip()
15        despesa["url_detalhe"] = cols[1].find("a").get('href')
16        despesa["empenhado"] = cols[2].get_text().strip()
17        despesa["liquidado"] = cols[3].get_text().strip()
18        despesa["pago"] = cols[4].get_text().strip()
19        despesas.append(despesa)
20
21    return despesas
22
23 # remover o código abaixo
24 print (despesas_total())

```

**Código 5.3. Arquivo scrapper.py: Extraindo os dados de despesas**

O Código 5.3 é responsável por retornar uma lista de dicionários, onde cada dicionário representa uma despesa. Na linha 8, está sendo recuperada a tabela através do método `find` que irá retornar a primeira tabela existente. Nesse caso específico, é exatamente a tabela que possui os dados a serem extraídos. Em alguns casos, pode ser necessário passar algumas informações adicionais para que se possa recuperar a tabela de interesse. Com a tabela recuperada o comando `find_all` irá retornar todas as linhas dessa tabela. Como isso inclui a linha de cabeçalho da tabela a iteração da linha 11 irá iniciar a partir do segundo elemento da lista. Então, para cada linha é recuperada as colunas, e depois é construído um dicionário python `despesa` com as chaves `nome`, `url_detalhe`, `empenhado`, `liquidado` e `pago`. Por fim, esse código já pode ser testado com o comando a seguir.

```

$ python scrapper.py
[{'nome': 'ADMINISTRACAO', 'url_detalhe': 'http://www.transparencia.ma.gov.br/app/despesas/por-funcao/2019/funcao/04?', 'empenhado': '530.070.090,32', 'liquidado': '452.420.550,87', 'pago': '351.633.728,08'}, {'nome': 'AGRICULTURA', 'url_detalhe': 'http://www.transparencia.ma.gov.br/app/despesas/por-funcao/2019/funcao/20?', 'empenhado': '69.816.420,71', 'liquidado': '58.739.278,15', 'pago': '45.119.980,63'}]

```

### CheckPoint

Neste ponto, já está finalizado um código para extração dos dados. A execução desse código deverá produzir uma saída similar a apresentada anteriormente.

Com essa extração inicial já é possível desenvolver a API, como demonstrada na

próxima Seção.

### 5.3.3. Desenvolvimento

No terceiro passo é desenvolvido uma API REST que irá disponibilizar os dados extraídos em um formato acessível para algoritmos computacionais. O REST (*Representational State Transfer*), ou em português Estado Representacional de Transferência, é um estilo arquitetural proposto em [Fielding 2000]. Ele consiste em um conjunto de princípios/regras/*constraints* baseados no protocolo HTTP para a criação de APIs com interfaces bem definidas. Atualmente as diversas linguagens de programação possuem *frameworks* que auxiliam na construção desse tipo de API.

Para esse texto, foi usado o *micro framework* denominado Flask<sup>6</sup>, com a extensão flask-restplus<sup>7</sup>. Uma das principais vantagens do Flask é a possibilidade de desenvolver APIs simples que poderão ser extendidas através de bibliotecas a medida que surgem novas necessidades. A instalação dessa biblioteca é similar as demais:

```
$ pipenv install flask-restplus
```

Após a instalação já é possível escrever uma API bem simples que para retorna os dados extraídos na seção anterior através de uma única rota, como apresentada no Código 5.4.

```
1 from flask import Flask
2 from flask_restplus import Resource, Api, fields
3 from scrapper import despesas_total
4
5 app = Flask(__name__)
6 api = Api(app)
7
8 @api.route('/despesas')
9 class Despesas(Resource):
10     def get(self):
11         return despesas_total()
12
13 if __name__ == '__main__':
14     app.run(debug=True)
```

**Código 5.4. Arquivo app.py: Primeira versão da API**

As rotas são implementadas com o decorator `@api.route`, que recebe como parâmetro a rota, que neste caso é `despesas`. Para essa rota só está implementado o método GET, que irá retornar todos os dados extraídos pela função `despesas_total`. A função retorna os dados como uma lista de dicionários, e será convertida e retornada para o cliente no formato JSON (*JavaScript Object Notation*). Esse um formato de texto de tipo chave-valor muito utilizado para a serialização de dados abertos, por ser auto-

---

<sup>6</sup>Site oficial <http://flask.pocoo.org/>

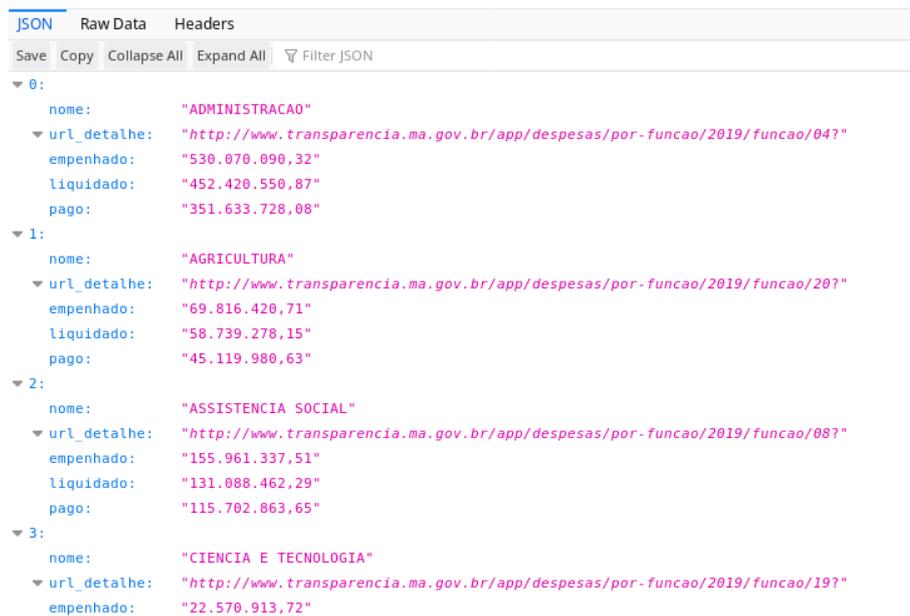
<sup>7</sup>Documentação: <https://flask-restplus.readthedocs.io/en/stable/>

descritivo e de fácil interpretação e geração, tanto por agentes humanos, quanto por agentes computacionais (<https://www.json.org/>).

Para testar a primeira versão do servidor, basta executar o Código 5.4:

```
$ python app.py
* Serving Flask app "app" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production
  deployment.
  Use a production WSGI server instead.
* Debug mode: on
* Restarting with stat
* Debugger is active!
* Debugger PIN: 230-864-203
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

A saída acima mostra que o servidor foi iniciado e esta aguardando requisições através do endereço local (127.0.0.1) e na porta 500. Então, já é possível acessar os dados através de um cliente web, por exemplo, ao entrar com o seguinte endereço <http://localhost:5000/despesas> em um navegador web tem-se uma saída similar a da Figura 5.4.



Index	nome	url_detalhe	empenhado	liquidado	pago
0	ADMINISTRACAO	<a href="http://www.transparencia.ma.gov.br/app/despesas/por-funcao/2019/funcao/041">http://www.transparencia.ma.gov.br/app/despesas/por-funcao/2019/funcao/041</a>	530.070.090,32	452.420.550,87	351.633.728,08
1	AGRICULTURA	<a href="http://www.transparencia.ma.gov.br/app/despesas/por-funcao/2019/funcao/207">http://www.transparencia.ma.gov.br/app/despesas/por-funcao/2019/funcao/207</a>	69.816.420,71	58.739.278,15	45.119.980,63
2	ASSISTENCIA SOCIAL	<a href="http://www.transparencia.ma.gov.br/app/despesas/por-funcao/2019/funcao/087">http://www.transparencia.ma.gov.br/app/despesas/por-funcao/2019/funcao/087</a>	155.961.337,51	131.088.462,29	115.702.863,65
3	CIENCIA E TECNOLOGIA	<a href="http://www.transparencia.ma.gov.br/app/despesas/por-funcao/2019/funcao/197">http://www.transparencia.ma.gov.br/app/despesas/por-funcao/2019/funcao/197</a>	22.570.913,72		

Figura 5.4. Visualizando os dados de despesas, testando o Código 5.4

A API já está funcionando, mas é necessário fazer algumas melhorias para torná-la mais útil, por exemplo, permitindo especificar o ano das despesas. Ao verificar o Código 5.3, é possível identificar pela URL que os dados que estão sendo retornados são referentes apenas ao ano de 2019. Porém, é possível acessar os dados de diferentes anos apenas alterando a URL. Com essa característica pode-se adaptar a função `despesas_total` para retornar os dados de um determinado ano de acordo com o valor passado no parâmetro, como na linha 6 do Código 5.5.

```

4 def despesas_total (ano):
5     url_base = "http://www.transparencia.ma.gov.br/app"
6     url = url_base + "/despesas/por-funcao/"+ano
7     response = requests.get(url)
8     ...

```

**Código 5.5. Arquivo `scraper.py`: Adicionando o ano como parâmetro da função `despesas_total`**

Tendo alterado a função `despesas_total`, deve-se alterar a rota para incluir o ano, e então passá-lo para a função `despesas_total`, como demonstrado no Código 5.6.

```

8 @api.route('/despesas/<string:ano>')
9 class Despesas(Resource):
10     def get(self, ano):
11         return despesas_total(ano)

```

**Código 5.6. Arquivo `app.py`: Adicionando o ano para a rota da API**

Com essas alterações é possível acessar os dados através de uma rota que inclui o ano de referência para os dados. Por exemplo, os dados do ano de 2016 poderão ser acessados através da seguinte URL: <http://localhost:5000/despesas/2016>.

O site da transparência do Governo do Maranhão permite visualizar os detalhes das despesas de uma dada função ou órgão administrativo. Por exemplo, o código da função administrativa **educação** é 12. Então, a URL <http://www.transparencia.ma.gov.br/app/despesas/por-funcao/2018/funcao/12#lista> detalha como a despesa com a educação foi distribuída para cada órgão. Acessando este endereço, percebe-se que a estrutura dos dados é muito similar, permitindo reutilizar o código escrito anteriormente. A principal diferença é na formação da URL que requer o código da função administrativa. Então, pode-se adaptar o código para incluir uma função que irá extrair o total das despesas e outra por função administrativa, Código 5.7.

```

1 import requests
2 from bs4 import BeautifulSoup as BS
3 def despesas_total (ano):
4     url_base = "http://www.transparencia.ma.gov.br/app"
5     url = url_base + "/despesas/por-funcao/"+ano
6     return extrai_despesas (url)
7
8 def despesas_por_funcao (cod, ano):
9     url_base = "http://www.transparencia.ma.gov.br/app"
10    url = url_base + "/despesas/por-funcao/"+ano+"/funcao/"+cod
11    return extrai_despesas (url)

```

**Código 5.7. Arquivo `scraper.py`: Incluindo a extração por função administrativa.**

Ambas funções do Código 5.7 apenas montam a URL e repassam para a função `extrai_despesas`, apresentada pelo Código 5.8.

```

12 def extrai_despesas (url):
13     response = requests.get(url)
14     page = BS(response.text, 'lxml')
15     table = page.find ('table')
16     rows = table.find_all ('tr')
17     despesas = []
18     for row in rows[1:]:
19         cols =row.find_all ("td")
20         despesa = {}
21         despesa["codigo"] = cols[0].get_text().strip()
22         despesa["nome"] = cols[1].find("a").get_text().strip()
23         despesa["url_detalhe"] = cols[1].find("a").get ('href')
24         despesa["empenhado"] = cols[2].get_text().strip()
25         despesa["liquidado"] = cols[3].get_text().strip()
26         despesa["pago"] = cols[4].get_text().strip()
27         despesas.append(despesa)
28     return despesas

```

**Código 5.8. Arquivo `scraper.py`: Detalhe da função `extrai_despesas`.**

Para suportar essa nova rota, será necessário adaptar o arquivo `app.py` para incluí-la. O Código 5.9 demonstra a inclusão dessa nova rota, que inclui o código da função administrativa e o ano de referência.

```

15 @api.route ('/<string:cod_funcao>/<string:ano>')
16 class DespesasPorFuncao (Resource):
17     def get (self, cod_funcao, ano):
18         return despesas.despesas_por_funcao (cod_funcao, ano)

```

**Código 5.9. Arquivo `app.py`: Inclusão de nova rota para retornar as despesas detalhadas por função administrativa**

Com essas alterações, pode-se testar o servidor através de um navegador web ou pelo terminal:

```

$ curl -X GET "http://localhost:5000/despesas/12/2018" -H "accept:
  application/json"
[ {      "codigo": "170203",      "nome": "FUNDACAO NICE LOBAO",
...

```

### CheckPoint

Nesse ponto, já está finalizado o código básico da API. A execução acima deverá retornar os dados em JSON de diferentes órgãos. O mesmo teste poderá ser feito diretamente no navegador, caso não tenha o utilitário `curl`.

#### 5.3.4. Documentação

Uma parte importante em qualquer API é a documentação. Então, no quarto passo será utilizado a biblioteca Swagger<sup>8</sup> para a construção automatizada de documentação. A

<sup>8</sup>Site oficial <https://swagger.io/>

biblioteca `flask-restplus` vem com o suporte para o Swagger e já cria uma documentação básica ao acessar o endereço raiz da API. Porém, através de uma coleção de *decorators* e parâmetros é possível adicionar novas informações ao código, gerando uma documentação mais detalhada como no Código 5.10.

```
1 from flask import Flask
2 from flask_restplus import Resource, Api, fields
3 from scrapper import despesas_total, despesas_por_funcao
4
5 app = Flask(__name__)
6 api = Api(app = app,
7         version = "1.0",
8         title = "Transparência Maranhão",
9         description = "Uma API não oficial com os dados sobre as
10                    receitas e despesas do Governo do Maranhão")
11 ns = api.namespace('despesas', description='Dados de despesas')
12
13 @ns.route('/<string:ano>')
14 class Despesas(Resource):
15     def get(self, ano):
16         return despesas_total(ano)
17 @ns.route('/<string:cod_funcao>/<string:ano>')
18 class DespesasPorFuncao(Resource):
19     def get(self, cod_funcao, ano):
20         return despesas_por_funcao(cod_funcao, ano)
21
22 if __name__ == '__main__':
23     app.run(debug=True)
```

**Código 5.10. Arquivo `app.py`: Adicionando novas informações para o Swagger**

As linhas 7, 8 e 9 adicionaram a versão, o nome e a descrição como informações principais da API. Além disso, uma API pode ter diferentes rotas, por exemplo, poderia ter rotas específicas para despesas e outras para receitas. Essas rotas poderiam estar agrupadas por dois diferentes *namespaces*. Aqui foi então criado na linha 11, um *namespace* para as despesas, incluindo sua descrição, e as linhas 13 e 17 foram adaptadas para usá-lo. O resultado da documentação pode ser observado na Figura 5.5.

Além das informações para a API e *namespace*, é possível adicionar informações diretamente aos métodos e parâmetros, como apresentado no Código 5.11.

```
12 ...
13 @ns.route('/<string:ano>')
14 class Despesas(Resource):
15     @api.doc(responses={ 200: 'OK', 400: 'Despesas não encontradas' },
16             params={ 'ano': 'Ano de referência para as despesas' })
17     def get(self, ano):
18         return despesas_total(ano)
19
20 @ns.route('/<string:cod_funcao>/<string:ano>')
21 class DespesasPorFuncao(Resource):
```

# Transparência Maranhão

1.0

[ Base URL: / ]  
<http://localhost:5000/swagger.json>

Uma API não oficial como os dados sobre as receitas e despesas do Governo do Maranhão

## despesas Dados de despesas

GET /despesas/despesas/{ano}

GET /despesas/despesas/{cod\_funcao}/{ano}

Figura 5.5. Visualização da documentação em Swagger

```
21 @api.doc(responses={ 200: 'OK', 400: 'Despesas não encontradas' },
22         params={ 'ano': 'Ano de referência para as despesas',
23                 'cod_funcao': 'Código da função (educação, saúde ...) de referê
24                   ncia para as despesas' })
24     def get(self, cod_funcao, ano):
25         return despesas_por_funcao(cod_funcao, ano)
26     ...
```

Código 5.11. Arquivo `app.py`: Adicionando novas informações para os métodos HTTP.

No código 5.11 foi utilizado o *decorator* `@api.doc` para passar algumas informações para o método GET, tendo como resultado a Figura 5.6.

Name	Description
<b>cod_funcao</b> * required string (path)	Código da função (educação, saúde ...) de referência para as despesas
<b>ano</b> * required string (path)	Ano de referência para as despesas

Figura 5.6. Documentação para o método GET no Swagger

Por fim, pode-se criar os metadados dos dados providos pela API, que incluem os tipos e as descrições dos dados. Os tipos de dados para os valores liquidados, pagos

e empenhados são números, no entanto, os dados extraídos estão no formato de texto e usando a representação brasileira. Então, a conversão para número deverá considerar essa representação. Existe uma biblioteca denominada *babel* que possui já implementada essa funcionalidade e pode ser instalada com o seguinte comando:

```
$ pipenv install babel
```

Com a biblioteca *babel* instalada, será necessário algumas atualização no arquivo `scraper.py`. Primeiro será necessário importar a função `parse_decimal`.

```
1 from babel.numbers import parse_decimal
```

E depois, aplicá-la para os valores empenhado, liquidado e pago:

```
1 despesa["empenhado"] = parse_decimal(cols[2].get_text().strip(), locale
    = 'pt_BR')
```

Após a alteração do arquivo `scraper.py`, será necessário descrever os metadados através da função `api.model` como no Código 5.12. Além do tipo e descrição, foi incluído um valor de exemplo que será utilizado também para a documentação.

```
22 model = api.model('Dados sobre uma função ou órgão', {
23     'codigo': fields.String(description='Código da função ou órgão',
24                             example="04"),
25     'nome': fields.String(description='Nome da função ou órgão',
26                             example="ADMINISTRACAO"),
27     'url_detalhe': fields.String(description='Endereço para mais
28                             detalhes', example="http://www.transparencia.ma.gov.br/app/despesas/
29                             /por-funcao/2016/funcao/04?"),
30     'empenhado': fields.Float(description='Valor empenhado', example
31                               =821854500.93),
32     'liquidado': fields.Float(description='Valor liquidado', example
33                               =794738131.95),
34     'pago': fields.Float(description='Valor pago', example=775701742.7)
35 },
36 )
```

#### Código 5.12. Arquivo `app.py`: Adicionando o metadado a API

Para associar o metadado aos dados retornados, será usado o *decorator* `@api.marshall_with` em ambas rotas:

```
1 @ns.route('/<string:ano>')
2 class Despesas(Resource):
3     @api.marshall_with(model, mask='*')
4     @api.doc ...
```

Após a atualização da API, pode-se visualizar os metadados como na Figura 5.7.

Code	Description
200	Success
Example Value <b>Model</b>	
<pre> <b>Dados sobre uma função ou órgão</b> {   codigo      string                example: 04                Código da função ou órgão    nome        string                example: ADMINISTRACAO                Nome da função ou órgão    url_detalhe string                example: http://www.transparencia.ma.gov.br/app/despesas/por-funcao/2016/funcao/04?                Endereço para mais detalhes    empenhado   number                example: 821854500.93                Valor empenhado    liquidado   number                example: 794738131.95                Valor liquidado    pago        number                example: 775701742.7                Valor pago }           </pre>	
400	Despesas não encontradas

Figura 5.7. Visualização dos metadados

### 5.3.5. Implantação

O quinto e último passo tem como objetivo implantar a API desenvolvida. Aqui será usada a plataforma Heroku<sup>9</sup>. Então, antes de prosseguir será necessário criar uma conta gratuita nesse portal. Para a implantação será usado também o gerenciador de configuração Git e o repositório Github (<http://github.com/>).

Antes de mais nada, vamos instalar o *gunicorn* (<https://gunicorn.org/>), um servidor WSGI (*Web Server Gateway Interface*) necessário para executar os *scripts* Python do lado do servidor. De modo similar as instalações anteriores, basta executar o seguinte comando:

```
$ pipenv install gunicorn
```

Depois de instalado, será necessário criar um arquivo denominado Procfile<sup>10</sup> que é utilizado pelo Heroku para a inicialização do serviço. Nesse caso ele indicará o WSGI e o nome do aplicativo, que é `app`, e está localizado no módulo (ou arquivo) `app.py`.

```
1 web: gunicorn app:app
```

<sup>9</sup>Site oficial: <https://www.heroku.com/>

<sup>10</sup>Mais informações sobre o Procfile <https://devcenter.heroku.com/articles/procfile>

Como será usado o repositório *Github*, garanta que já tenha uma conta no *github* e o aplicativo *git* instalado e configurado no computador. Com isso, será necessário logar na sua conta, criar um repositório denominado `transparencia-ma` e executar os seguintes comandos porém usando o repositório que foi criado.

```
git init
git add *
git commit -m "first commit"
git remote add origin https://github.com/profsergiocosta/transparencia-
  ma.git
git push -u origin master
```

### Fica a dica

Nesse exemplo está sendo um repositório HTTP, então será necessário entrar com o nome do usuário e senha para enviar os dados para o repositório, ou seja, executar o comando *push*. Uma dica é trabalhar com repositório SSH ao invés de HTTP, caso esteja usando esses procedimento em um computador pessoal.

A Figura 5.8 mostra os três passos para implantar uma aplicação no Heroku. O passo 1 cria uma aplicação Heroku com o nome `transparencia-ma`, o 2 conecta-se a aplicação ao repositório *Github* e o 3 executa a implantação (*deploy*) manual. Na Figura 5.8 (4) apenas mostra as mensagens que são impressas durante a implantação. Além do *deploy* manual, é possível marcar implantação (*deploy*) automática, que irá ocorrer a cada novo *commit* ao repositório github.

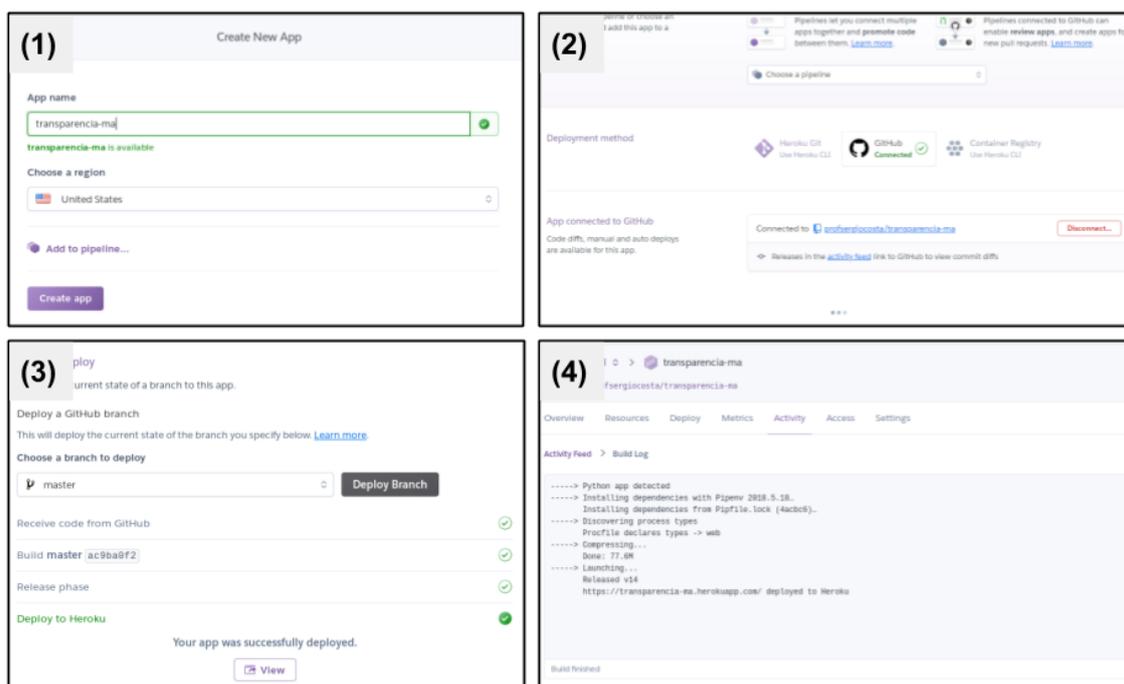


Figura 5.8. Implantação no Heroku em 3 passos

Depois de implantado, é possível testar diretamente em um navegador ou usando o comando `curl`.

```
$ curl -X GET "https://transparencia-ma.herokuapp.com/despesas/2018" -H
  "accept: application/json"
[{"codigo": "04", "nome": "ADMINISTRACAO", "url_detalhe": "http://www.
  transparencia.ma.gov.br/app/despesas/por-funcao/2018/funcao/04?"
  ...
```

Na próxima Seção, serão apresentados outros exemplos de como usar a API desenvolvida.

## 5.4. Experimentando e evoluindo

A disponibilização de dados abertos cria muitas oportunidades de soluções que podem ser desenvolvidas pela comunidade. Nesta Seção, serão apresentados alguns exemplos simples de como uma API pode ser utilizada por algoritmos externos.

### 5.4.1. Um teste simples de análise de dados

O portal da transparência do Maranhão já disponibiliza diversos dados tabelados. Porém, um usuário pode precisar de um dado que não esteja disponível diretamente. Por exemplo, o usuário poderia querer um gráfico que mostrasse a evolução temporal dos gastos por uma dada função administrativa. Sem uma API, seria necessário entrar em diversas páginas e manualmente anotar os gastos, ou salva-los e manipulá-los através de outro software. Contudo, com a API desenvolvida aqui é possível gerar este gráfico escrevendo um pouco de código em linguagens como Python, R ou JavaScript.

Para exemplificar, foi escrito um código Python diretamente no Google Colab <sup>11</sup>. O código completo da Figura 5.9 pode ser acessado em <http://bit.ly/304z5XF>.

---

<sup>11</sup>Google Colab pode ser acessado em: <https://colab.research.google.com/>

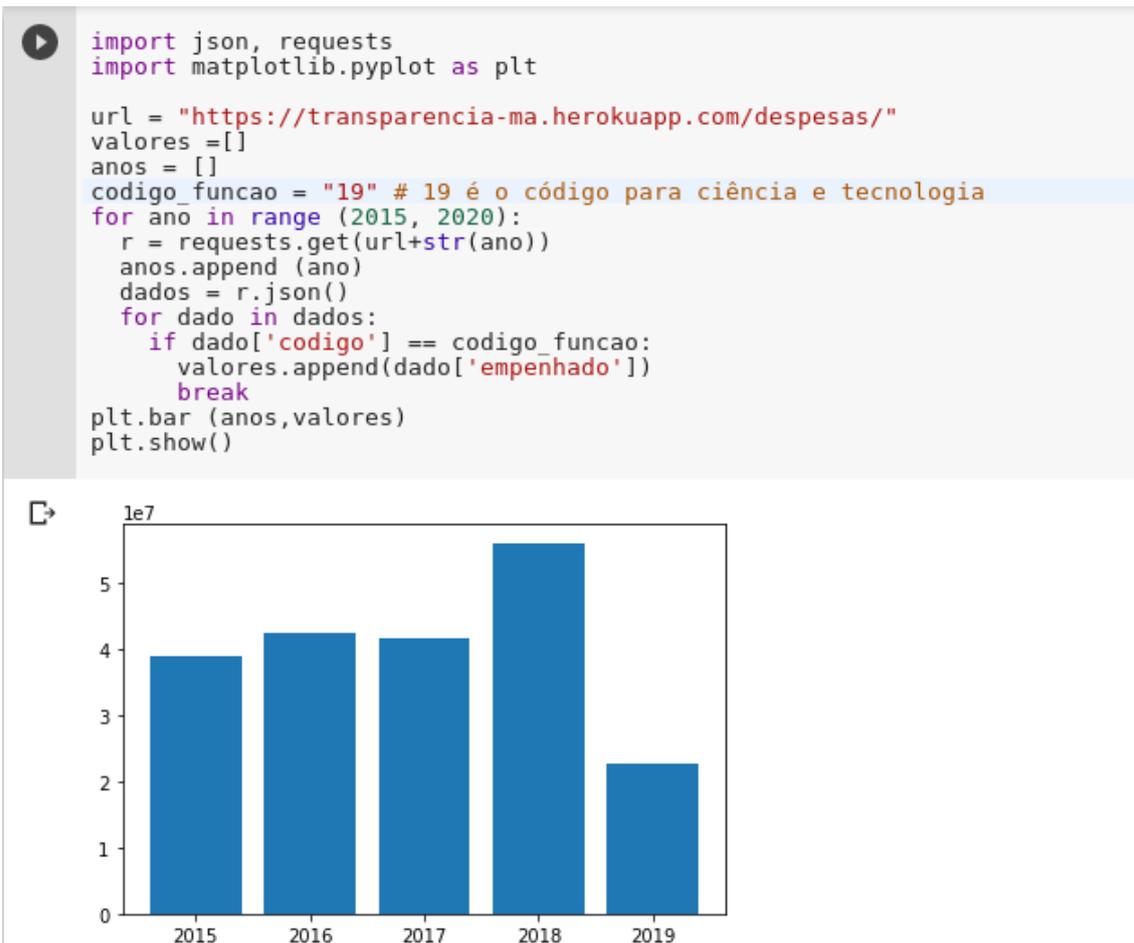


Figura 5.9. Código e gráfico gerado no Google Colab

#### 5.4.2. Um teste simples de aplicação

Além de ser usado para análise, os dados abertos podem ser acessados por aplicativos web e móveis. Nesse caso, uma possibilidade seria desenvolver aplicativos híbridos usando a linguagem JavaScript, através de *frameworks* como o Angular, React e Electron. Como o objetivo é fazer um teste simples, foi escrito um código usando a biblioteca *JQuery* que executa uma requisição e imprime no console o resultado, Código 5.13. O código completo pode ser acessado em <https://repl.it/@SergioSouza1/testeapi>.

```

22 <script>
23 $(document).ready(() =>
24     $("button").click( () => $.ajax({
25         url: "https://transparencia-ma.herokuapp.com/despesas/10/2018",
26         success: (result) => console.log (result) })
27 ))
28 </script>

```

Código 5.13. Testando a API através de requisições assíncronas

Antes de executar o teste, modifique a URL para referenciar o serviço que foi

criado no Heroku. Porém, esse simples teste irá falhar e apresentar seguinte mensagem no navegador:

```
1 Cross-Origin Request Blocked: The Same Origin Policy disallows reading the remote resource at https://transparencia-ma.herokuapp.com/despesas/10/2018. (Reason: CORS header Access-Control-Allow-Origin missing).
```

Isso ocorre porque, por padrão, o JavaScript não permite que aplicações de domínios distintos compartilhem recursos. Porém é possível habilitar o que é conhecido como CORS (que em português seria Compartilhamento de Recursos de Origem Cruzada). Então, para tudo funcionar corretamente, será necessário retornar ao ambiente de desenvolvimento da API.

Primeiro passo será instalar a extensão para dar esse suporte, que é feito pela biblioteca `flask_cors`:

```
pipenv install flask_cors
```

Depois no arquivo `app.py`, será adicionado um novo `import`:

```
1 from flask_cors import CORS
```

E então, será executado a função `CORS` ao aplicativo:

```
1 CORS(app)
```

O código completo pode ser acessado em: <http://bit.ly/2PcfHXZ>.

Atualizado esse código, deve-se enviar as modificações para o repositório *GitHub* e, por fim, executar a implantação novamente no Heroku. Com o serviço implantado e atualizado, basta executar o código novamente para ser impresso os dados em JSON no console. Lembrando, que esse é um exemplo extremamente simples, porém a sua execução mostra que a API está pronta para ser usada por aplicações em JavaScript.

### 5.4.3. Evolução

O site do governo do Maranhão possui diversos outros dados que poderão ser extraídos. Por exemplo, ao acessar um dado órgão, pela seguinte <http://www.transparencia.ma.gov.br/app/despesas/por-funcao/2019/funcao/12/orgao/240201> é possível ter acesso aos nomes, CPF ou CNPJ dos credores, e também os valores empenhado, liquidado e pago, como na Figura 5.10.

CREDOR	EMPENHADO	LIQUIDADADO	PAGO
A. L. T TRINDADE CPF/CNPJ: 30.865.611/0001-63	134.999,70	0,00	0,00
A.E. MENDES CPF/CNPJ: 41.472.655/0001-40	13.650,00	0,00	0,00
ABILIO NEIVA MOREIRA CPF/CNPJ: 00.002.513/2433-82	459,00	459,00	459,00

**Figura 5.10. Credores de um dado órgão público**

Observa-se que a tabela é muito similar a que foi extraída na Seção 5.3, com a diferença que cada credor possui um CPF ou CNPJ em vez de um código. Então o Código 5.14 será também muito similar ao escrito anteriormente na Seção 5.3

```

30
31 def despesas_por_orgao (orgao, funcao, ano):
32     url = "http://www.transparencia.ma.gov.br/app/despesas/por-funcao/"
33     +ano+"/funcao/"+funcao+"/orgao/"+orgao+"?#lista"
34     response = requests.get(url)
35     page = BS(response.text, 'lxml')
36     table = page.find('table')
37     rows = table.find_all('tr')
38     despesas = []
39     for row in rows[1:]:
40         cols = row.find_all("td")
41         despesa = {}
42         despesa["nome"] = cols[0].find("a").get_text().strip()
43         despesa["url_detalhe"] = cols[0].find("a").get('href')
44         despesa["cpf/cnpj"] = cols[0].find("small").get_text().strip().
45         replace("CPF/CNPJ: ", "")
46         despesa["empenhado"] = parse_decimal(cols[1].get_text().strip
47         (), locale='pt_BR')
48         despesa["liquidadado"] = parse_decimal(cols[2].get_text().strip
49         (), locale='pt_BR')
50         despesa["pago"] = parse_decimal (cols[3].get_text().strip(),
51         locale='pt_BR')
52         despesas.append(despesa)
53     return despesas

```

**Código 5.14. Extraindo os dados por órgão público**

Após criar a função do Código 5.14, é possível criar uma nova rota que inclui o código do órgão administrativo, por exemplo: <https://transparencia-ma.herokuapp.com/despesas/240206/12/2018>. Então, será necessário atualizar a rota, fazer um novo *commit*, enviar as atualizações e executar um novo *deploy*.

O processo de transformação de dados públicos em dados abertos requer muita exploração do site de dados públicos. Por exemplo, aqui seria possível extrair ainda diversas outras informações como:

- Ainda sobre despesas, é possível extrair informações detalhadas por credor.
- Além dos dados de despesas, pode-se extrair os dados de receitas de modo muito similar.
- Dados sobre servidores públicos, como salário e diárias.
- Transferência de valores, por exemplo, repasse a municípios.

Além desses dados é possível evoluir bastante essa API, permitindo extrair cada vez mais informações relevantes.

## 5.5. Conclusões

A web continua a crescer e evoluir desde o primeiro site desenvolvido e publicado por Tim Berners-Lee em 06 de agosto de 1991<sup>12</sup>. Atualmente, com o seu grande volume de dados, a web demanda ferramentas automatizáveis para a extração e estruturação das informações. Foi neste contexto, que surgiu o paradigma de dados abertos que hoje é muito demandado pela sociedade.

A disponibilização de dados abertos por órgãos públicos pode fomentar o desenvolvimento de soluções bem como a participação das pessoas no controle e na compreensão da gestão dos recursos públicos. Nesse texto, foi apresentada uma solução parcial que permitiu disponibilizar dados que já estavam públicos como dados abertos, através do desenvolvimento de uma API. Essa solução é interessante, pois atualmente existem diversos portais com muitos dados públicos mas que não os disponibilizam em formato acessível por algoritmos. Muitos desses portais não têm previsão para o lançamento de uma API de dados abertos. Vale ressaltar que a solução proposta aqui tem a desvantagem de ter um custo computacional adicional, pois requer carregar uma página web e extrair as informações antes de enviar os dados. Uma possibilidade para contornar o problema deste curso, é salvar os dados extraídos das páginas em um banco de dados.

Acredita-se que os passos detalhados aqui, irão permitir a qualquer programador transformar qualquer site de dados públicos em uma API de dados abertos, sendo apenas necessário um pouco de experiência na linguagem Python. Uma possibilidade seria continuar a extrair mais informações do site da transparência do governo do Maranhão, ou de outros sites que tenham interesse.

Por fim, as tecnologias utilizadas nesse texto não são exclusivas para a criação de portais de dados abertos. Na verdade, o estilo arquitetura REST é muito utilizado por diversas soluções. Principalmente para a comunicação de serviços e aplicativos móveis. Essa abordagem permite que um mesmo serviço seja consumido por diferentes clientes, e as redes sociais são excelentes exemplos de sucesso dessa abordagem.

## Referências

[Chacon 2014] Chacon, S. (2014). *Git Pro*. Apress, 2nd edition.

---

<sup>12</sup>Esse site ainda pode ser acessado em <http://info.cern.ch/hypertext/WWW/TheProject.html>

- [Daga et al. 2015] Daga, E., d'Aquin, M., Adamou, A., and Brown, S. (2015). The open university linked data - data.open.ac.uk. *Semantic Web Journal*, 7(2):183–191.
- [Dietrich et al. 2009] Dietrich, D., Gray, J., McNamara, T., Poikola, A., Pollock, P., Tait, J., and Zijlstra, T. (2009). Open data handbook.
- [Fielding 2000] Fielding, R. T. (2000). Rest: architectural styles and the design of network-based software architectures. *Doctoral dissertation, University of California*.
- [Keßler and Kauppinen 2012] Keßler, C. and Kauppinen, T. (2012). Linked open data university of münster—infrastructure and applications. In *Extended Semantic Web Conference*, pages 447–451. Springer.
- [Lessig 2007] Lessig, L. (2007). 8 principles of open government data.
- [Mitcheal 2015] Mitcheal, R. (2015). *Web Scraping with Python - Collecting data from the modern web*. O'Reilly, 1nd edition.
- [Molloy 2011] Molloy, J. C. (2011). The open knowledge foundation: open data means better science. *PLoS Biol*, 9(12):e1001195.
- [Richardson 2019] Richardson, L. (2019). Beautiful soup documentation.
- [Torvalds 2005] Torvalds, L. (2005). Kernel scm saga..
- [VanTol 2018] VanTol, A. (2018). Pipenv: A guide to the new python packaging tool.
- [Vaz 2011] Vaz, J. C.; Ribeiro, M. M. M. R. E. (2011). Dados governamentais abertos e seus impactos sobre os conceitos. *Cadernos PPG-AU/UFBA*, pages 45–62.
- [Zablith et al. 2011] Zablith, F., Fernandez, M., and Rowe, M. (2011). The ou linked open data: production and consumption. In *Extended Semantic Web Conference*, pages 35–49. Springer.