

Capítulo

5

Design Web Intuitivo: Potencializando Projetos com Material Design e BeerCSS

Matusalen Costa Alves, Maria Steffany da Silva Viana, Iallen Gábio de Sousa Santos

Abstract

This chapter delves into the development and significance of Human-Computer Interaction (HCI), detailing the evolution of graphical interfaces and the impact of Material Design on creating effective and visually appealing user interfaces. It introduces BeerCSS, a lightweight framework that integrates Material Design principles to streamline interface development. The chapter provides guidance on installing BeerCSS via CDN and NPM, and implementing light and dark themes with JavaScript. Additionally, it illustrates the practical application of these technologies through the creation of a developer portfolio, demonstrating how to build compelling and functional user experiences.

Resumo

Este capítulo explora o desenvolvimento e a importância da Interface Humano-Computador (IHC), abordando a evolução das interfaces gráficas e a aplicação dos princípios do Material Design na criação de interfaces estéticas e funcionais. O estudo destaca o framework BeerCSS, que oferece uma solução leve e eficiente para o desenvolvimento rápido de interfaces ao incorporar os princípios do Material Design. Inclui instruções para a instalação do BeerCSS via CDN e NPM e para a implementação de temas claros e escuros com JavaScript. Além disso, demonstra a aplicação prática dessas tecnologias por meio da criação de um portfólio de desenvolvedor, evidenciando a capacidade de criar experiências de usuário impactantes e funcionais.

5.1. Introdução

Em 1990, o cientista cognitivo Don Norman se juntou à Apple, empresa de tecnologia, e desenvolveu o que é conhecido hoje como User Experience (UX) [Norman 2002]. Na era digital atual, a UX tornou-se um dos aspectos prioritários e críticos no design de

interfaces. A forma como os usuários percebem e interagem com um site ou aplicativo pode determinar não apenas sua satisfação, mas também a eficácia da comunicação e o sucesso geral do produto.

Na maior parte dos casos, os usuários geralmente decidem consumir ou não um produto no prazo de 15 a 20 segundos após o primeiro acesso [Nielsen et al. 2014]. Nesse cenário, as percepções visuais e sensoriais desempenham um papel fundamental, moldando profundamente a forma como os indivíduos respondem a diferentes elementos de design.

As cores, tipografia, disposição de elementos e harmonia visual são essenciais para capturar a atenção do usuário e criar uma experiência mais agradável [Norman 2002]. As cores evocam emoções e estabelecem uma identidade visual; a tipografia contribui para a legibilidade e percepção de credibilidade; a disposição dos elementos afeta a facilidade de navegação e a aparência geral. Esses aspectos são mais do que simples escolhas estéticas, eles impactam diretamente na eficiência da interação do usuário com o produto.

O design tem um impacto significativo na forma de interação e a comunicação visual [Norman 2002]. Além das telas de dispositivos digitais, isto também é percebido em mídias impressas e ambientes físicos, como outdoors e exposições. A maneira como os elementos são dispostos e apresentados pode afetar a forma como as pessoas percebem e entendem a informação.

Com o mercado cada vez mais competitivo, onde a oferta de produtos e serviços é vasta e diversificada, se destacar é um desafio. A capacidade de prender a atenção do usuário e criar uma impressão memorável é essencial para o sucesso. Uma apresentação visual impactante, combinada com uma funcionalidade intuitiva, pode fazer a diferença entre conquistar a lealdade do cliente ou ser esquecido [Eyal and Hoover 2014].

Em ambientes digitais, oferecer interfaces que sejam visualmente atraentes e fáceis de usar é imprescindível, visto que o usuário realiza um julgamento rápido, enquanto em contextos físicos, como publicidade e embalagens, um design inovador pode influenciar a decisão de compra [Krug 2000]. A capacidade de criar experiências que se destacam pelo apelo visual é, portanto, uma vantagem competitiva significativa no mercado atual.

Além disso, o design deve considerar não apenas a aparência, mas também a funcionalidade [Kao and Archer 1997]. A integração harmoniosa desses elementos cria uma experiência aprimorada, que aumenta a satisfação do usuário e a eficácia do produto.

Sob esta óptica, destaca-se a Interação Humano-Computador (IHC), um campo que estuda a interação entre usuários e sistemas computacionais, focando em como os humanos interagem com as tecnologias e como essas interações podem ser aprimoradas [Waldeilson and Paixão 2016]. A IHC abrange aspectos como a usabilidade, acessibilidade e a UX, com o objetivo de criar interfaces mais eficazes. Este campo se preocupa em entender as necessidades dos usuários, suas capacidades cognitivas e comportamentais e como o design pode facilitar a realização de tarefas de maneira satisfatória [Franz and Kauer-Franz 2024]. A aplicação dos princípios de IHC no design de interfaces assegura que as soluções tecnológicas sejam funcionalmente robustas e ofereçam uma experiência produtiva para os usuários.

Neste contexto, o Material Design [Google 2014], criado pela Google [Inc. 2024], surge como uma abordagem inovadora que vai além das interfaces digitais, inspirando-se na física dos materiais e na forma como eles interagem com a luz e o espaço. Lançado em 2014, o Material Design foi desenvolvido para proporcionar uma experiência visual coesa e funcional que se aplica a diversos contextos e mídias. A ideia central é que, assim como os materiais físicos têm propriedades específicas, o design digital pode ser orientado por princípios que priorizam a hierarquia visual e clareza.

O planejamento dessa ideologia de design foi cuidadosamente pensado para atender às necessidades da Interação Humano-Computador e proporcionar uma abordagem estruturada ao design de interfaces. Seus princípios incluem o uso de camadas e sombras para criar profundidade, tipografia legível e uma paleta de cores bem definida para promover a comunicação visual. Esses elementos vão além de escolhas estéticas, são projetados visando garantir uma navegação otimizada para uma experiência de usuário satisfatória. Isto traz maior conforto para o usuário, o que por sua vez, o torna suscetível a consumir mais daquele conteúdo [Dulsrud and Bygstad 2022].

Outro aspecto do Material Design é o foco na acessibilidade. As diretrizes possibilitam a criação de interfaces que possam ser usadas por todos, independentemente de suas capacidades físicas ou cognitivas. Isso se reflete na escolha cuidadosa de cores, na adaptação de tamanhos de texto e no suporte a tecnologias assistivas, garantindo que a navegação seja intuitiva e que a informação esteja sempre ao alcance do usuário.

A consistência na navegação também participa dos padrões. O Material Design estabelece padrões claros para a organização de elementos, como barras de ferramentas, menus e botões. Esses componentes são projetados para oferecer uma experiência uniforme em todas as plataformas, ajudando os usuários a se orientarem e realizarem tarefas sem confusão ou frustração, mesmo quando mudam de dispositivo.

O Material Design adota uma abordagem responsiva, garantindo que as interfaces se adaptem automaticamente a diferentes tamanhos de tela e dispositivos. Isso significa que, seja em um smartphone, tablet ou desktop, o usuário terá uma experiência igualmente consistente. A responsividade não só melhora a estética do design, mas também assegura que a usabilidade seja mantida, independentemente de como ou onde o sistema é acessado [Marcotte 2011].

Neste minicurso, será abordado o Material Design 3.0 (M3), esta é a versão mais atual do sistema de design da Google. O M3 traz inovações para o aprimoramento da UX, além disso, enfatiza a personalização e várias novas adições significativas. Lançado como uma evolução do Material Design original, o 3.0 enfatiza a personalização e a adaptabilidade, permitindo que designers criem interfaces ajustadas às necessidades específicas de seus usuários [Google 2022].

A estilização de páginas da web é geralmente feita através de CSS. Os desenvolvedores podem descrever aspectos estéticos de cada parte da página utilizando esta tecnologia. Entretanto, existem ferramentas de CSS pré-implementado que facilitam o trabalho ao estilos prontos. Essas ferramentas permitem que os desenvolvedores apliquem padrões visuais consistentes de maneira mais rápida e eficiente, sem a necessidade de criar todo o código CSS manualmente, acelerando o processo de desenvolvimento e

garantindo um design mais uniforme. Estas são denominadas frameworks de design.

Para criar aplicativos e páginas na web, os desenvolvedores utilizam diversos frameworks baseados em diferentes tecnologias de design. Exemplos dessas incluem o Human Interface Guidelines da Apple [Apple 2024] e o Fluent Design System da Microsoft [Microsoft 2024]. Cada um desses frameworks oferece um conjunto único de diretrizes e componentes para criar interfaces correspondentes aos critérios de design das marcas.

FRAMEWORK	FOCO	CUSTO	ESTADO DE ATUALIZAÇÃO
Bootstrap for MDB	Material Design for Bootstrap	Pago	Constante
Bootstrap	Bootstrap	Gratuito	Constante
BeerCSS	Material Design 3.0	Gratuito	Constante
Tailwind CSS	Human Interface Guidelines	Pago	Constante
Office UI	Fluent Design Style	Gratuito	Descontinuado
Fluent UI	Fluent Design Style	Gratuito	Constante
Materialize	Material Design 2.0	Gratuito	Em hiato

Tabela 5.1. Comparação entre diferentes frameworks de CSS

A lista dessas ferramentas é extensa e algumas delas são de acesso gratuito e outras pago, como apresentado na Tabela 5.1. Neste minicurso, concentramos nossa atenção no BeerCSS [BeerCSS 2022], devido a sua gratuidade, design atual e integração com o ecossistema da Google, a fim de explorar como essa ferramenta pode nos ajudar a concluir o nosso objetivo de criar de maneira ágil páginas responsivas e visualmente agradáveis.

O restante deste capítulo está organizado da seguinte maneira: a Seção 5.2 explora a história do IHC, UX e interface de usuário (IU), a Seção 5.3 apresenta os princípios fundamentais do Material Design, enquanto a Seção 5.4 o BeerCSS e suas boas práticas, a Seção 5.5 o estudo de caso com um portfólio de desenvolvedor e, por fim, a Seção 5.6 a conclusão deste trabalho.

5.2. Interface Humano-Computador, IU e UX

A Interface Humano-Computador (IHC) surge com campo fundamental da computação devido a necessidade de aprimoramento da interação dos seres humanos com os sistemas computacionais. A IHC é a base sobre a qual se desenvolvem conceitos como interface do usuário (IU) e experiência do usuário (UX) [Franz and Kauer-Franz 2024]. No design de interfaces, IU e UX são termos frequentemente usados para descrever aspectos específicos da interação, mas cada um tem seu próprio foco e importância.

Enquanto a IU se concentra na criação de interfaces visuais eficientes e intuitivas [Krug 2000], a UX abrange uma visão mais ampla, englobando as emoções, percepções e respostas do usuário ao interagir com o produto [Garrett 2010]. Esta seção introduz os principais conceitos de IHC, IU e UX, explorando como esses elementos se inter-relacionam para criar experiências de usuário mais eficazes e satisfatórias.

5.2.1. História e Evolução da Interface Humano-Computador

A Interface Humano-Computador (IHC) começou a se formar como um campo de estudo na década de 1960, quando os primeiros computadores começaram a se tornar mais acessíveis para empresas e instituições de pesquisa. Naquela época, a interação com os

computadores era restrita a comandos de texto complexos, exigindo conhecimentos técnicos aprofundados. A evolução da IHC foi impulsionada pela necessidade de tornar a tecnologia mais acessível para um público mais amplo e a popularização dos computadores pessoais [Raskin 2000].

Nos anos 1970 e 1980, os avanços em hardware e software, como o desenvolvimento das primeiras interfaces gráficas de usuário (GUIs), revolucionaram a forma como os humanos interagem com os computadores. A introdução do mouse, janelas, ícones e menus, liderada por empresas como Xerox e Apple, marcou um ponto de virada ao permitir que os usuários interagissem com computadores de forma mais natural.

Na década de 1990, o surgimento da web trouxe novos desafios e oportunidades para a IHC. A criação de interfaces eficazes para diferentes dispositivos e contextos tornou-se crucial. Com a expansão da internet, surgiram novas plataformas e dispositivos com características e limitações próprias [Raskin 2000]. Isso exigiu que os designers encontrassem soluções adaptáveis para garantir que as interfaces funcionassem bem em diversos cenários. A web não só ampliou o alcance da IHC, mas também trouxe novos elementos e interações que impactaram profundamente o design de interfaces.

Atualmente, a IHC está em constante evolução, incorporando tecnologias inovadoras que transformam a interação com a tecnologia. Realidade virtual aumentada, inteligência artificial e interfaces naturais, como reconhecimento de voz e gestos, moldam o futuro da interação [Zaphiris 2008]. Esses avanços buscam criar experiências eficientes para permitir uma comunicação mais natural entre humanos e sistemas. À medida que essas tecnologias se tornam parte do cotidiano, elas redefinem a experiência do usuário e ampliam as possibilidades para o design de interfaces.

5.2.2. Princípios Fundamentais da IHC

Os princípios fundamentais da IHC guiam o design de interfaces que promovem uma interação fluida entre o usuário e a tecnologia [Zaphiris 2008]. Esses princípios são desenvolvidos com base no entendimento das capacidades cognitivas e comportamentais dos usuários, com o objetivo de criar sistemas que não apenas atendam às suas necessidades, mas também ofereçam uma experiência de fácil compreensão. Entre esses princípios, destacam-se a simplicidade, consistência, visibilidade e feedback. Estes aspectos visam garantir que as interfaces sejam acessíveis e fáceis de navegar, independentemente do nível de habilidade do usuário.

A usabilidade é um dos pilares centrais da IHC, ela consiste na criação de interfaces que sejam eficazes, eficientes e satisfatórias [Krug 2000]. Um sistema com alta usabilidade permite que os usuários realizem suas tarefas com facilidade, sem a necessidade de treinamento extenso ou frustração. Para alcançar isso, o design deve ser intuitivo, permitindo que o usuário aprenda rapidamente a utilizar o sistema e se sinta confortável ao interagir com ele. A simplicidade na apresentação das informações e a redução de complexidade são estratégias fundamentais para melhorar a usabilidade.

O reaproveitamento de componentes de interfaces é uma prática crucial para garantir a consistência nos sistemas. Um exemplo clássico é o botão de fechar representado na Figura 5.1, que geralmente é vermelho. Essa escolha não é acidental; culturalmente,

o vermelho é associado a ações de interrupção ou alerta, como parar, cancelar ou fechar um processo [Garrett 2010]. Ao utilizar consistentemente um botão vermelho para interromper uma ação, os usuários rapidamente associam essa cor com a função, reduzindo o tempo necessário para entender e operar o sistema.



Figura 5.1. Botão Fechar do Windows [Microsoft 2024] em destaque

Essa reutilização de componentes familiares, como ícones, cores e padrões de interação, contribui para uma curva de aprendizado mais suave, já que os usuários podem transferir conhecimentos de uma parte do sistema para outra. Além disso, a padronização desses elementos ajuda a minimizar erros, pois os usuários sabem o que esperar ao interagir com diferentes partes da interface, o que aumenta a satisfação geral com o sistema.

A acessibilidade é outro princípio importante para a IHC, garantindo que os sistemas sejam utilizáveis por todos, independentemente de suas limitações físicas ou cognitivas. Isso inclui a adaptação de interfaces para pessoas com deficiências, como a utilização de textos alternativos para imagens, a implementação de teclas de atalho, e a compatibilidade com tecnologias assistivas. Um design acessível cumpre normas de inclusão e também amplia o alcance do produto, permitindo que mais pessoas possam interagir com ele de maneira eficaz [Johnson 2014].

O feedback ao usuário se concentra em fornecer respostas claras e imediatas às ações do usuário. Isso pode ser feito por meio de mensagens de confirmação, indicações visuais ou sonoras e outros sinais que informam o usuário sobre o estado atual do sistema ou o sucesso de suas ações. O feedback adequado ajuda a evitar confusões, assim, permite que o usuário compreenda facilmente o impacto de suas interações e continue o uso de maneira mais confiável.

5.2.3. Interface do Usuário (IU)

A IU é um dos componentes mais críticos de qualquer sistema, pois é o ponto de interação direta entre o usuário e a tecnologia [Norman 2002]. O design visual da IU é fundamental para gerar uma experiência adequada. Elementos como cores, tipografia, e layout desempenham um papel crucial na comunicação eficaz das funções e na estética geral da interface. Um design visual bem executado melhora a percepção de profissionalismo e facilita a compreensão e o uso do sistema. Por outro lado, um design visual fraco pode proporcionar frustrações e falhas.

A navegação é um aspecto que impacta diretamente a facilidade com que os usuários podem encontrar e acessar diferentes partes do sistema. Uma navegação lógica é essencial para que os usuários possam realizar suas tarefas sem fracasso. Isso inclui a organização clara de menus, botões e links, bem como a implementação de convenções de design reconhecíveis que ajudam os usuários a se orientarem rapidamente.

A consistência é um princípio chave no design da UI herdado diretamente da IHC. Ela visa garantir que todos os elementos da interface mantenham um padrão uniforme. Essa consistência ajuda a criar uma experiência previsível, assim, pode reduzir a carga cognitiva do usuário e facilitar a aprendizagem do uso daquele sistema. Manter uma estética e um comportamento consistentes em diferentes seções e plataformas do sistema contribui com feedbacks indispensáveis como uma navegação fluida e uma interação eficiente.

5.2.4. Experiência do Usuário (UX)

A UX abrange todos os aspectos da interação do usuário com um produto ou sistema, indo além da simples interface visual [Norman 2002]. UX é um campo multidimensional que engloba fatores como satisfação, eficiência e a emoção experimentada durante o uso de produtos digitais. A satisfação do usuário é fundamental, pois está diretamente relacionada à percepção de valor e ao prazer obtido ao usar o sistema. Um design que proporciona uma experiência satisfatória tende a aumentar a lealdade e a recomendação do usuário.

A eficiência é outro componente crucial da UX. Sistemas bem projetados permitem que os usuários realizem suas tarefas com o mínimo de esforço e tempo possível. Isso envolve a otimização de fluxos de trabalho, a redução de etapas desnecessárias e a garantia de que as funções mais usadas sejam facilmente acessíveis.

Além disso, a emoção desempenha um papel significativo na experiência do usuário. As emoções que um usuário experimenta ao interagir com um sistema podem influenciar a forma como ele percebe e se relaciona com o produto. Um design que evoca emoções positivas como prazer e satisfação, pode criar uma conexão mais profunda e duradoura com o usuário. A integração de elementos que proporcionam prazer estético e interações agradáveis é essencial para uma experiência do usuário bem-sucedida.

5.3. Material Design

5.3.1. Conceito e Fundamentos do Material Design

O Material Design, fornece um conjunto de diretrizes que auxilia o usuário a criar interfaces que sejam esteticamente agradáveis, funcionais, interativas e de fácil compreensão. Ele também traz uma inspiração no mundo físico, demonstrando texturas, elevações e diferenças tonais para demonstrar uma certa hierarquia dentro dos elementos.

5.3.2. Princípios de Design: Cores, Tipografia e Formas

Chamadas de tríade do design, cor, tipografia e forma são pilares fundamentais do design. A cor é a linguagem emotiva onde traz significados psicológicos, culturais, harmonia e contraste. A tipografia deve trazer hierarquia, legibilidade e assim ser a voz do design. A forma é a estrutura visual, onde se encaixam as formas geométricas e orgânicas, além dos elementos básicos como ponto, linha, plano e volume. O Material Design se utiliza desses pilares para a criação de boas *interfaces*.

O Material Design enfatiza a necessidade de um contraste entre as cores e ele realiza isso automaticamente, tanto no tema claro como no tema escuro integrado.

O Material Design 3.0 prevê dois sistemas de cores que podem ser utilizados, o sistema de cores estático e o sistema de cores dinâmico. No sistema estático o desenvolvedor escolhe manualmente as cores e criar seu próprio contraste. Já no sistema de cores dinâmico, a paleta de cores é gerada a partir de uma cor básica, isto pode ser utilizado para criar sistemas personalizáveis pelo usuário mantendo a consistência de cores e o contraste do Material Design.

Em ambos os sistemas, o sistema enquadra as cores da paleta em rótulos denominados 'cores-chave complementares', estas são: *Primary*, *Secondary*, *Tertiary*, *Neutral* e *Neutral variant*; A partir destas cores deve ser criada uma escala tonal para cada uma das cores-chave a fim de deixar o layout acessível. O Material Design recomenda um contraste mínimo de 3:1. Além das cores-chave também é apresentada a *Surface* que trás a função de ser a cor utilizada em fundos e grandes áreas com baixa enfase.

A cor-chave *Primary* deve ser utilizada nos componentes mais importantes da UI portanto, deve ter uma cor mais viva e chamativa. Os botões de alta enfase e estados ativos tem a mais alta hierarquia visual dentro da página, portanto devem ter a cor mais chamativa. A cor *Secondary* é utilizada nos componentes menos proeminentes da página, como botões tonais, ele tem uma cor mais neutra e se assemelha mais a cor de *Surface*. A *Secondary* é baseada na cor *Primary* mas com uma alteração no croma neutralizando a mesma. Já a *Tertiary* é usada para trazer contraste e equilíbrio com as cores *Primary* e *Secondary*. Ela é utilizada para trazer atenção a determinado elemento como um campo de entrada. As funções da cor de erro são estáticas mesmo num sistema de cores dinâmicas, ela não muda em nenhuma ocasião caso haja uma alteração das cores do layout.

Para cada cor-chave complementar são geradas outras cores que recebem nomes conforme suas funções dentro do sistema. Existe um prefixo *On* que significa que essa cor será aplicada sobre outra e o sufixo *Container* que significa que a cor será aplicada em um campo de espaço delimitado. Com essas nomeações e funções temos quatro aplicações nessas cores-chaves para cada uma delas: *Primary*, *On Primary*, *Primary Container* e *On Primary Container* como pode ser observado na Figura 1.2 . Há além disso o conceito de *Variant* que é usado para nomear uma opção com menos enfase em comparação a opção não variante.

Primary	Secondary	Tertiary	Error
On Primary	On Secondary	On Tertiary	On Error
Primary Container	Secondary Container	Tertiary Container	Error Container
On Primary Container	On Secondary Container	On Tertiary Container	On Error Container

Figura 5.2. Exemplo de token de cores

Além da cor para o conforto óptico, é recomendado que se alinhe também o tipo de texto, que o Material Design nos trás cinco tipos que são nomeados como, display, headline, title, body e label, a fonte utilizada na escrita, que podem tanto ser criada exclusivamente para determinado fim, como pode ser usada uma já existente que se encaixe nas necessidades e o uso de tokens para uma padronização e melhor coesão entre os assuntos dentro do layout.



Figura 5.3. Exemplo dos tipos de texto

Botões, menus, inputs, cards, etc. São o que o sistema nomeia de *Containers*, portanto todos os elementos de design interagem mutuamente, existem formas no estilo da fonte utilizada assim como tem uma cor vinda da paleta de cores-chave.

5.3.3. Movimento e Interatividade

Elevações e animações fazem com que a UI fique mais interativa e de fácil entendimento e usabilidade ao usuário. As elevações, por exemplo, permitem enfatizar a hierarquia dentro de um layout. Neste caso a elevação acontece no eixo z que é medida em *density-independent pixels* (dps), onde se tem a limitação de 12dps e cinco níveis de descanso, as elevações podem ser classificadas em três tipos de representação: através de alteração de cor entre as superfícies em diferentes alturas, através do sombreado da superfície mais alta e sobreposição de telas.

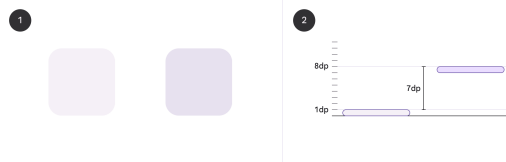


Figura 5.4. Representação de elevação no eixo Z

A versão 3 do Material Design nos trás os *easing tokens*, exemplificados na tabela, agrupados em dois conjuntos, o enfatizado e o padrão, onde o enfatizado é recomendado para a maioria das transições, já o padrão são transições pequenas e focadas e para o sistema iOS e Web é o único porque os mesmos não conseguem suportar o conjunto enfatizado, essas transições devem ser rápidas.

Por inspiração do mundo físico onde as coisas não começam e terminam mecanicamente, os *easing tokens* buscam deixar o movimento mais natural, acelerando e desacelerando assim suavizando o movimento e trazendo o especto de naturalidade.

5.3.4. Componentes

Os componentes do Material Design são os blocos interativos que estruturam a interface do usuário, organizando elementos visuais e ações de forma lógica e intuitiva. Eles estão divididos em categorias funcionais que servem tanto para guiar o design quanto para

EASING	DURAÇÃO	TIPO DE TRANSIÇÃO
Enfaticado	500ms	Iniciar e terminar na tela
Desaceleração enfatizada	400ms	Entrar na tela
Aceleração enfatizada	200ms	Sair da tela
Padrão	300ms	Iniciar e terminar na tela
Desaceleração padrão	250ms	Entrar na tela
Aceleração padrão	200ms	Sair da tela

padronizar a experiência do usuário em diferentes dispositivos. Esses componentes se destacam por sua flexibilidade e integração, facilitando a criação de layouts consistentes, fluídos e que seguem os princípios de uma boa experiência visual.

Entre os principais componentes de ação, encontram-se os botões, ícones de ação e botões flutuantes, que permitem aos usuários realizar funções primárias e secundárias com facilidade. A simplicidade de um botão comum ou a visibilidade de um botão flutuante exemplificam a clareza que esses elementos trazem para a interface, tornando a navegação mais direta.

Elementos de contenção, como cartões e diálogos, são usados para agrupar conteúdo relevante e ações associadas, ajudando a organizar visualmente a interface. Esses componentes permitem que as informações sejam apresentadas de forma acessível, facilitando a compreensão e a interação do usuário com o sistema.

Na comunicação, indicadores de progresso e snackbars desempenham papéis fundamentais ao fornecer feedback imediato sobre o estado de um processo ou tarefa. Esses elementos mantêm o usuário informado de maneira discreta e eficiente, sem sobrecarregar a interface.

A navegação é guiada por barras de ferramentas, menus e trilhos, que garantem que os usuários possam se mover entre diferentes seções do sistema sem dificuldades. A padronização desses componentes facilita a compreensão da interface, mesmo quando o usuário alterna entre diferentes dispositivos ou plataformas.

Os componentes de seleção e entrada de texto, como campos de texto, caixas de seleção e botões de rádio, permitem que o usuário insira informações e faça escolhas dentro do sistema de forma clara e direta. A simplicidade e a consistência desses elementos garantem uma interação mais eficiente e agradável.

Esses componentes, baseados em princípios de interação intuitiva e design coeso, garantem que as interfaces construídas com o Material Design sejam eficientes, responsivas e adequadas às expectativas do usuário moderno.

5.3.5. Acessibilidade e Responsividade

Além do conforto óptico de um bom design, o Material Design tem o objetivo de se adaptar a diferentes tamanhos de tela. Isso acontece pela flexibilidade de seus componentes que, em sua última versão, foram projetados para se adaptar automaticamente, se expandindo ou contraindo, a diferentes tamanhos de tela. O sistema de grade e layout que permite uma reorganização inteligente e automática, e esses componentes são intuitivos em relação as respostas das interações com o usuário. O texto também se adapta a dife-

rentes telas por meio da mudança de tamanho e espessura das letras para que se mantenha o conforto óptico e legibilidade.

Neste contexto, o usuário tem uma fácil navegação em diferentes tipos de tela, pois a interface se adapta de forma a prover uma experiência familiar e intuitiva. Além disso, visando a acessibilidade que gera o contraste de cores certo assim como a responsividade na interação com os componentes do layout.

5.4. BeerCSS

O BeerCSS é um projeto de código aberto com licença MIT em desenvolvimento contínuo por parte do patrocinadores e apoiadores no *GitHub* e *Open Collective*. O *framework* visa combinar simplicidade e eficiência em um pacote que atende às necessidades tanto de desenvolvedores iniciantes quanto experientes.

O encorajamento ao uso dessa ferramenta se deriva de diversas das suas qualidades, entre elas, destacam-se o amadurecimento, a adoção do Material Design em HTML semântico, o seu tamanho pelo menos dez vezes menor que os outros frameworks de CSS baseados no Material Design e a adaptabilidade a utilizar qualquer framework JavaScript (JS). Além disso, não existe a necessidade de compilação, configuração e dependências, uma vez que essa ferramenta é autosuficiente.

5.4.1. Introdução ao BeerCSS: Conceito e Filosofia

O lema do BeerCSS [BeerCSS 2022] é: "Build material design in record time without stress for devs". O BeerCSS é um framework de design leve, pautado pela filosofia minimalista. Criado para gerar interfaces elegantes e funcionais com agilidade, ele permite que desenvolvedores, tanto iniciantes quanto veteranos, trabalhem de forma eficiente sem enfrentar complicações desnecessárias.

O BeerCSS adota uma abordagem em que cada componente é pensado para ser direto e eficaz. O framework elimina a sobrecarga de estilos excessivos, concentrando-se em funcionalidades essenciais. Com isso, o desenvolvedor pode manter o foco na construção de interfaces nítidas e intuitivas sem se preocupar com configurações complexas ou dependências externas.

Sua modularidade destaca-se pela reutilização de componentes alinhada aos princípios do Material Design 3.0 (M3). Ao contrário de frameworks mais robustos que oferecem uma vasta gama de opções e personalizações, o BeerCSS concentra-se em um conjunto enxuto de ferramentas garantindo eficiência no processo de desenvolvimento. Isso favorece a adaptação das interfaces a diversos cenários, isto garante consistência visual clara e coerente.

Em projetos que exigem rapidez como, *Minimum Viable Product* (MVPs) ou sites temporários, o BeerCSS oferece uma solução ágil. Sua arquitetura leve permite ao desenvolvedor entregar interfaces modernas sem necessidade de ajustes extensivos ou longas compilações. Assim, o BeerCSS se consolida como uma ferramenta eficaz para criar projetos funcionais em tempo reduzido, sem sacrificar a qualidade da experiência do usuário.

5.4.2. Settings, Elements e Helpers

Inspirado pela "Lei de Pureza da Cerveja da Alemanha" (*Reinheitsgebot*), que limitava a produção de cerveja a três ingredientes essenciais — água, malte de cevada e lúpulo, o BeerCSS segue um princípio semelhante ao organizar sua documentação em três pilares principais: Settings, Elements e Helpers. Esses pilares simplificam o desenvolvimento, isto proporciona uma experiência ágil e eficiente, sem a complexidade de outras abordagens como *Block, Element, Modifier* (BEM) ou *Scalable and Modular Architecture for CSS* (SMACSS).

As *Settings* controlam aspectos globais da interface e definem as variáveis principais que afetam o comportamento e a aparência do projeto. Entre as funcionalidades mais importantes, estão a escolha entre os temas claro ou escuro e a personalização de cores e elevações dos componentes.

Abaixo estão as principais classes e variáveis de *Settings*:

- **Temas:** O desenvolvedor pode definir o tema padrão como claro ou escuro usando as classes **‘light‘** ou **‘dark‘** diretamente no elemento **‘<body>‘**, como sugere o Código 5.1.

```
<body class="light">...</body>
<body class="dark">...</body>
```

Código 5.1. Definindo o tema claro ou escuro

- **Cores Dinâmicas:** Utilizando **‘material-dynamic-colors‘**, é possível alterar o tema em tempo real com base em cores específicas ou imagens, criando uma experiência altamente personalizável, que, apesar da sua funcionalidade, não será profundamente abordado nesse capítulo. O Código 5.2 demonstra isto.

```
let theme = await ui("theme", "#ffd700");
```

Código 5.2. Alterando o tema dinamicamente com uma cor hexadecimal

- **Variáveis Globais:** O BeerCSS permite a customização completa das cores, animações e fontes usando variáveis CSS. Essas variáveis, como **‘—primary‘**, **—background**, **—shadow**, entre outras, ajustam a paleta de cores e a tipografia, garantindo consistência em todo o projeto. O Código 5.3 exibe um exemplo de como se parece uma paleta, como sugere o *Settings* [BeerCSS 2024], gerada através do Material Theme Builder [Material 2024].

```
:root,
body.light {
  --primary: #6750a4;
  --on-primary: #ffffff;
  --background: #ffffbf;
  --on-background: #1c1b1e;
}
body.dark {
  --primary: #cfbcff;
  --on-primary: #381e72;
  --background: #1c1b1e;
  --on-background: #e6e1e6;
```

```
}
```

Código 5.3. Customizando o tema padrão

Os *Elements* são os componentes básicos do BeerCSS, como botões, cards, grids e ícones. Esses componentes seguem os princípios do Material Design, garantindo uma estética moderna e funcionalidade robusta.

- **Botões:** Permitem aos usuários executar ações com um único toque. Podem ser customizados com classes como **‘primary’**, **‘secondary’**, entre outras, para ajustar a cor e a aparência do botão.

- **Cards:** Superfícies que exibem conteúdo e ações relacionadas a um único tópico. São ideais para apresentar informações de forma clara e hierárquica.

Os *Helpers* adicionam flexibilidade e personalização adicional aos *Elements*. Eles são usados para ajustar o alinhamento, margens, sombras, cores e muito mais, sem a necessidade de criar CSS personalizado.

- **Alinhamento:** Helpers como **left-align**, **right-align**, **center-align** controlam o posicionamento de elementos dentro de seus contêineres.

- **Margens e Espaçamentos:** Classes como **small-margin**, **large-margin**, **no-margin** permitem ajustes rápidos e precisos no espaçamento entre os elementos.

- **Cores e Tipografia:** Helpers para cores (**amber-text**, **blue-border**) e tipografia (**bold**, **italic**, **uppercase**) permitem uma rápida customização da aparência textual e cromática.

Esses três pilares — **Settings**, **Elements** e **Helpers** — trabalham juntos para criar uma experiência de desenvolvimento eficiente e flexível, simplificando tanto o processo de design quanto a implementação de interfaces modernas e responsivas.

5.4.3. Layouts e Grid

O sistema de grid do BeerCSS é uma ferramenta essencial para a construção de layouts responsivos e flexíveis. Baseado em uma estrutura de colunas e linhas, esse sistema permite que os desenvolvedores criem layouts que se adaptam dinamicamente a diferentes tamanhos de tela e dispositivos. A abordagem modular do grid simplifica a tarefa de projetar interfaces que funcionam bem tanto em desktops quanto em dispositivos móveis, assim, pode garantir que a experiência do usuário seja consistente em qualquer tela.

O sistema de grid do BeerCSS divide o espaço em 12 colunas, permitindo que o conteúdo ocupe diferentes larguras dependendo do tamanho da tela. Helpers como **s1 — s12**, **m1 — m12** e **l1 — l12** controlam a quantidade de colunas ocupadas em dispositivos pequenos, médios e grandes.

```
<div class="grid">
  <div class="blue s12 m6 l3">
    <h3>PRIMEIRO</h3>
  </div>
  <div class="red s12 m6 l3">
    <h3>SEGUNDO</h3>
  </div>
```

```

<div class="yellow s12 m6 l3">
  <h3>TERCEIRO</h3>
</div>
<div class="purple s12 m6 l3">
  <h3>QUARTO</h3>
</div>
</div>

```

Código 5.4. Exemplo de Grid no BeerCSS

No exemplo apresentado no Código 5.4, cada div ocupa 12 partes em telas pequenas, 6 partes em telas médias e 3 partes em telas grandes. Na Figura 5.5 é possível visualizar como diferentes telas apresentam este conteúdo com disposição diferente dos elementos.

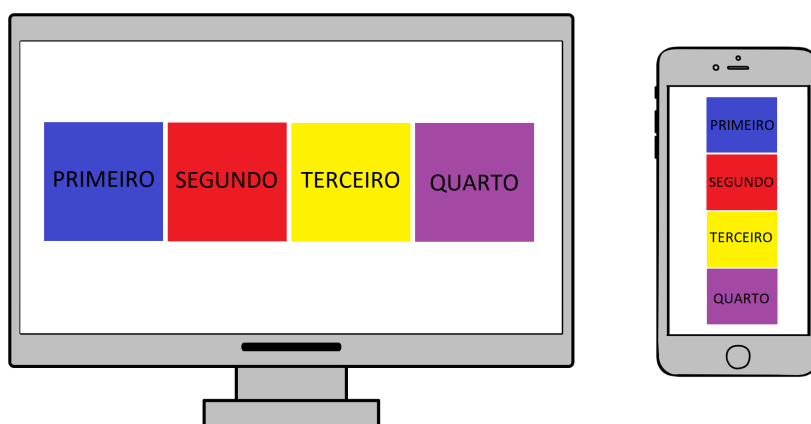


Figura 5.5. Visualização do resultado do código em desktops e dispositivos móveis

O espaço total da tela pode ser dividido como um contêiner para alinhar todo o conteúdo dentro do mesmo. Isto reaproveita o mesmo grid para várias seções ou áreas da mesma página, de forma que otimiza o uso dessa ferramenta.

Os layouts podem ser flexíveis ou fixos, utilizando as classes **absolute** para elementos relativos ao contêiner ou **fixed** para elementos fixos na tela. No Código 5.5 é apresentado o layout absoluto, enquanto no 5.6 um layout fixo.

O comportamento esperado é de que o Código 5.5 gere um contêiner de tamanho pequeno que ocupe toda a área da esquerda até a direita, assim, fazendo jus às suas características **'absolute left bottom right'**. Enquanto o do Código 5.6 deve criar um elemento que permanecerá no canto inferior da janela do navegador, cobrindo toda a largura da página, mesmo que o usuário role-a, graças às classes **'fixed left bottom right'**.

```

<article class="small">
  <div class="absolute left bottom right">
    <h5>Fundo do container</h5>
  </div>

```

```
</article>
```

Código 5.5. Exemplo de Layout Absoluto no BeerCSS

```
<div class="fixed left bottom right">  
  <h5>Fundo do documento</h5>  
</div>
```

Código 5.6. Exemplo de Layout Fixo no BeerCSS

5.4.4. Temas e Personalização

O BeerCSS oferece uma vasta gama de opções de personalização que permitem aos desenvolvedores ajustar tanto a aparência quanto o estilo das interfaces com facilidade. O framework se destaca por sua flexibilidade, permitindo a adaptação rápida de cores, tipografia e outros elementos visuais para que o design se alinhe perfeitamente com a identidade visual de um projeto ou marca.

O sistema de temas do BeerCSS é uma das suas principais características, permitindo a geração automática de paletas de cores completas, como demonstrado no Código 5.7. Isso oferece flexibilidade aos desenvolvedores, possibilitando a criação de funcionalidades de personalização para os usuários de maneira dinâmica. Alternativamente, o desenvolvedor pode optar por utilizar temas estáticos que requerem o trabalho manual de especificar cada cor da paleta do Material Design. Essa abordagem possibilita uma personalização visual sem a necessidade de reescrever o código.

```
let theme = await ui("theme", "#ffd700"); // Alterando pelo amarelo
```

Código 5.7. Alterando o tema dinamicamente com uma cor hexadecimal

Além de temas, o BeerCSS também simplifica a personalização de cores através de variáveis CSS, conforme o exemplo do Código 5.8. Essas variáveis permitem que a paleta de cores da interface seja rapidamente ajustada, facilitando a adaptação do design a preferências estéticas ou necessidades de branding.

```
{  
  --primary: #6750a4;  
  --on-primary: #ffffff;  
  --background: #ffffbf;  
  --on-background: #1c1b1e;  
}
```

Código 5.8. Exemplo de customização de cores no BeerCSS

O uso de variáveis CSS como ‘**—primary**’ e ‘**–background**’ facilita a personalização de cores, permitindo mudanças globais na aparência da interface com apenas algumas linhas de código.

Além das opções básicas de temas e cores, o BeerCSS também oferece uma ampla variedade de classes utilitárias, como as classes utilizadas na div do Código 5.9. Essas classes permitem ajustes rápidos e precisos em aspectos como margens, paddings e alinhamentos, sem a necessidade de escrever CSS adicional. Isso acelera o desenvolvimento e mantém o código mais limpo.

```
<div class="padding small-margin center-align">
  <p>Texto centralizado com margem pequena e padding.</p>
</div>
```

Código 5.9. Exemplo de uso de classes utilitárias no BeerCSS

Por fim, para a importação das fontes, baseado na Seção , deve ser feita como o conteúdo do Código 5.10 apresenta. O Material Theme Builder fará a importação dessa fonte logo após a escolha da paleta. Isso deve ser inserido dentro do arquivo CSS.

```
@import "https://fonts.googleapis.com/css2?family=Roboto:wght@400;500;700&display=swap";
:root {
  --font: "Roboto Flex";
  --font-icon: none;
}
```

Código 5.10. Exemplo de uso de classes utilitárias no BeerCSS

5.4.5. Instalação e Configuração do Ambiente

Nesta seção, abordaremos os preparativos necessários para a criação de um projeto utilizando o BeerCSS. A configuração inicial inclui a instalação do ambiente de desenvolvimento e a adição do framework ao projeto.

O primeiro passo envolve a instalação do BeerCSS no seu projeto. Existem duas maneiras principais de fazer isso: via CDN (Content Delivery Network) ou utilizando o NPM (Node Package Manager). Cada método possui suas vantagens, e você pode escolher de acordo com suas necessidades.

5.4.5.1. Utilizando CDN para Adicionar o BeerCSS

O método CDN oferece uma maneira rápida e simples de adicionar o BeerCSS ao seu projeto. Para utilizar este método, basta incluir as seguintes linhas de código no ‘<head>’ do seu arquivo HTML.

```
<head>
  <link href="https://cdn.jsdelivr.net/npm/beercss@3.6.13/dist/cdn/beer.min.css" rel="stylesheet"/>
  <script type="module" src="https://cdn.jsdelivr.net/npm/beercss@3.6.13/dist/cdn/beer.min.js"></script>
  <script type="module" src="https://cdn.jsdelivr.net/npm/material-dynamic-colors@1.1.2/dist/cdn/material-dynamic-colors.min.js"></script>
</head>
```

Código 5.11. Código de importação via CDN para BeerCSS

Essa abordagem carrega o BeerCSS diretamente da web, permitindo que você comece a utilizá-lo sem precisar instalar nada localmente. A vantagem do CDN é a simplicidade, especialmente útil para projetos menores ou protótipos rápidos, além de ser mais

simples para desenvolvedores iniciantes que não possuem experiência com o Node.js e o NPM.

5.4.5.2. Utilizando NPM para Adicionar o BeerCSS

Caso prefira um controle maior sobre o ambiente de desenvolvimento, a instalação via NPM pode ser a melhor opção. O NPM permite gerenciar dependências localmente e é amplamente utilizado em projetos de maior escala.

Para adicionar o BeerCSS ao seu projeto com NPM, execute os seguintes comandos no terminal:

```
npm i beercss  
npm i material-dynamic-colors
```

Código 5.12. Instalando o BeerCSS via NPM

Após a instalação, você pode importar o BeerCSS e utilizá-lo em seu projeto JavaScript da seguinte maneira:

```
import "beercss";  
import "material-dynamic-colors";
```

Código 5.13. Importando o BeerCSS no JavaScript

Essa abordagem permite que você tenha uma maior flexibilidade e controle sobre a versão e as atualizações do framework.

5.4.5.3. Testando a Configuração

Após a instalação do BeerCSS, é importante garantir que tudo está funcionando conforme o esperado. Para isso, vamos criar um exemplo simples que usa um botão estilizado pelo BeerCSS. O Código 5.14 mostrará um botão "Clique aqui" que utilizará o estilo padrão do framework.

```
<button class="primary">Clique aqui</button>
```

Código 5.14. Testando a configuração do BeerCSS

Ao executar este projeto, você deve ver um botão estilizado pelo BeerCSS. Se o botão aparecer corretamente, isso confirma que a instalação e configuração do BeerCSS foram bem-sucedidas.

5.4.6. Principais Componentes

O BeerCSS oferece uma série de componentes prontos para uso que facilitam a criação de interfaces consistentes e eficientes já apresentados anteriormente. Nesta seção, vamos explorar três dos componentes mais utilizados: *Botões*, *Cartões* e *Navbar*.

5.4.6.1. Botões

Os botões são elementos essenciais em qualquer interface. O BeerCSS simplifica sua criação ao fornecer classes que aplicam automaticamente estilos baseados no Material Design. O código desse componente corresponde ao conteúdo do Código 5.15 e a sua aparência na Figura 5.6.

```
<button class="primary">Clique aqui</button>
<button>
  <i>home</i> <span>Clique aqui</span>
</button>
<button>
   <span>Clique aqui</span>
</button>
```

Código 5.15. Botão primário no BeerCSS



Figura 5.6. Botões implementados em uma página web

5.4.6.2. Cartões

Os cartões são componentes versáteis usados para agrupar informações e ações relacionadas. Eles são frequentemente utilizados em interfaces modernas para exibir conteúdo de forma organizada. O Código 5.16 demonstra o elemento, a Figura 5.7 a sua visualização.

```
<article class="border">
  <h5>Titulo do cartao</h5>
  <p>Conteudo do cartao</p>
</article>
```

Código 5.16. Exemplo de cartão no BeerCSS

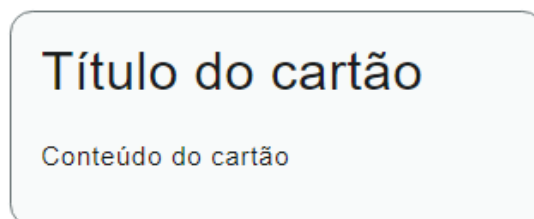


Figura 5.7. Cartão implementado em uma página web

5.4.6.3. Navbar

A barra de navegação (navbar) é um componente importante em muitas interfaces, proporcionando acesso fácil e rápido a diferentes seções de um site ou aplicativo. As linhas

do Código 5.17 exemplificam sua implementação e a Figura 5.17 o seu formato.

```
<nav>
  <button>Botao</button>
  <a class="chip">Chip</a>
  <i>home</i>
  
</nav>
```

Código 5.17. Exemplo de Navbar no BeerCSS

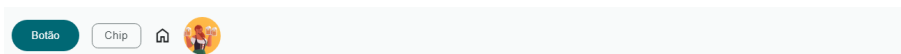


Figura 5.8. Navbar implementada em uma página web

Esse exemplo de navbar inclui um título de página centralizado e ícones de navegação em ambos os lados. A barra de navegação segue os padrões de design responsivo do BeerCSS.

5.5. Estudo de Caso: Aplicando Material Design com BeerCSS

Agora, nesta seção, como estudo de caso, será apresentado uma webpage baseada em um portfólio de desenvolvedor. A página segue as boas práticas da Otimização para Motores de Busca, ou Search Engine Optimization (SEO). Essa métrica serve para aumentar a visibilidade orgânica [Clarke 2023].

5.5.1. Implementação Prática: Criação de um Portfólio

Um portfólio de desenvolvedor deve ter seções definidas e organizadas, essas são: uma área de apresentação para caracterizá-lo ao leitor, uma área para suas produções bibliográficas e outra para os seus projetos pessoais, assim como um meio de contato.

Para implementação do Material Design utilizaremos alguns dos principais componentes do BeerCSS demonstrados na Seção 5.4. Para a página inteira, o grid, que viabiliza responsividade em diferentes tamanhos de tela. O header dessa página deve conter botões que visam facilitar a navegação pelo portfólio, nas informações sobre o desenvolvedor textos que o apresentem para o leitor. Assim como elementos textuais na parte que irá conter as suas produções bibliográficas

Já na área de tecnologias dominadas, deve ser utilizado elementos textuais e ícones gerados a partir de imagens. Na área de projetos, serão utilizados cartões, enquanto na área de contato mais conteúdo sobre o desenvolvedor e uma textarea para enviar mensagem. Por fim, um footer para sinalizar o fim da página.

Com o ambiente de desenvolvimento previamente configurado e a paleta de cores escolhida através do Material Theme Builder, deve-se inserir a linha de código apresentada no Código 5.18 no ‘<head>’ do HTML. Isso importará o arquivo CSS com as cores e fontes.

```
<head>
  <link rel="stylesheet" href="css/style.css">
```

```
</head>
```

Código 5.18. Importando o arquivo CSS para o HTML

Feito isso, o JS deverá ser importado de maneira semelhante no final da página, mas no ‘<body>’, como sugere o Código 5.19. Com isso as partes interativas da página, como o botão de alternar o tema, funcionarão corretamente.

```
<body>
  <script src="js/main.js"></script>
</body>
```

Código 5.19. Importando o arquivo JS para o HTML

Logo em seguida, escolha o tema padrão da página ao inserir ‘light’ ou ‘dark’ como classe no ‘<body>’, exatamente como no Código 5.20. Esta classe fará com que a página inicie sempre com o tema claro ou escuro.

```
<body class="light">
</body>
```

Código 5.20. Escolha do tema claro como padrão do portfólio

Agora, com todas as dependências ajustadas, o header é o primeiro componente a ser inserido na página. Esse componente serve para organizar e exibir conteúdo da página, geralmente relacionado à navegação. Neste deve haver uma navbar, o nome do desenvolvedor em escala reduzida, botões de navegação e um botão com ícone para alternar entre os temas. O Código 5.21 apresenta como ele é o estilo dela.

```
<header class="fixed max primary grid no-space small-padding no-margin"
  >
  <div class="s0 m2 l3"></div>
  <nav class="s12 m8 l6 grid tiny-space">
    <span class="bold s1 m3 l2 primary-container-text">Rafael Alves</
      span>
    <div class="s0 m0 l1"></div>
    <button class="s2 m2 l2 border primary-container small-round bold
      ">
      <a href="#artigos">Artigos</a>
    </button>
    <button class="s1 m1 l1 transparent square bold absolute middle
      right" id="theme-toggle">
      <i>dark_mode</i>
    </button>
  </nav>
  <div class="s0 m2 l3"></div>
</header>
```

Código 5.21. Exemplo de header no portfólio

As classes dos componentes que o BeerCSS dispõe são autoexplicativas.

• **Header:** A classe ‘fixed’ é referente ao documento, o que faz com que esse header fique sempre aparente na tela, enquanto ‘max’ faz esse elemento ocupar o máximo de

espaço possível da esquerda até a direita. Quanto à **‘primary’**, refere-se à cor, como exemplificado na Seção 5.4.2. O **‘grid no-space’** cria um grid sem espaço entre os elementos. Por fim, **‘small-padding’** faz o header assumir um tamanho menor entre o conteúdo e sua borda.

- **Nav:** Sua classe **‘grid tiny-space’** transforma esse elemento em um grid com um espaço pequeno entre cada elemento.

- **Button:** Cada botão tem uma disposição diferente na tela, dependendo do seu conteúdo interno. Baseado nisso, define-se o espaço que ele ocupa e se ajusta o grid. A classe **border** reforça as bordas do botão, **‘primary-container’** é a sua cor. O **‘small-round’** define o seu arredondamento e **‘bold’** que o seu conteúdo será negrito. A tag **‘<a>’** conecta o botão até a parte referida da página, a fim de facilitar a navegação.

- **Toggle Theme:** O último botão é o responsável por alternar entre os temas, trata-se de um ícone. Sua aparência é transparente devido a sua classe **‘transparent’** e seu contorno é quadrado, derivado de **‘square’**. O restante das suas classes seguem a mesma lógica dos outros itens que já foram explanados durante os últimos parágrafos.

A página é separada em seções e todas tem o mesmo espaçamento entre si por uma **‘<div>’** com a classe **‘large-space’**. Isto organiza visualmente o conteúdo, de forma que facilita a leitura. Os espaços variam de **‘tiny-space’** até **‘large-space’** e sempre serão utilizados para separar o conteúdo.

Como citado anteriormente, a próxima parte do portfólio é a apresentação do desenvolvedor. Neste exemplo utilizaremos informações fictícias. Assim, ele deve ser apresentado por meio de títulos, como o Código 5.22 demonstra. Todas as seções do texto têm três grid de área livre nas laterais e seis de conteúdo ao centro.

```
<h1 class="bold">Oi, sou o Rafael</h1>
<h6 class="">Desenvolvedor de software com foco em projetos open source
,
pesquisador e graduando Ciencia da Computacao pela UFPI.</h6>
<div class="small-space"></div>
<h6 class="">Neste site voce pode conferir todos os artigos tecnicos
que escrevi, ver alguns projetos pessoais que desenvolvi e as
ferramentas que eu utilizo.</h6>

```

Código 5.22. Exemplo de uma apresentação no portfólio

O sistema de grid que o BeerCSS oferece permite subdividi-lo inúmeras vezes, ou seja, permite que uma porção do espaço dividido em doze possa ser dividido em doze novamente. O grid e as suas funcionalidades tem uma explicação mais aprofundada na Seção 5.4.3.

As imagens podem ter diferentes tamanhos e formatos a depender da sua função. Neste caso, é a foto de perfil do desenvolvedor dessa página, então, tem dimensões pequena e responsivas, visível o suficiente em qualquer dispositivo. A classe **‘center’** e **‘middle’** alinha essa imagem ao centro da **‘<div>’** que ela ocupa.

Logo após, devemos criar a seção das principais produções bibliográficas, como

no Código 5.23. Este é o título e um botão que leva o usuário para um repositório com todos os artigos disponíveis. Ele será criado dentro de um grid porque os elementos textuais (títulos e datas) estarão alinhados com o início do título e o fim do botão.

```
<div class="grid no-space" id="artigos">
  <nav class="s10 m8 l6">
    <h3 class="bold">Artigos</h3>
    <div class="max"></div>
    <button class="border small-round tertiary-border">
      <span class="tertiary-text bold">Ver todos</span>
    </button>
  </nav>
</div>
```

Código 5.23. Exemplo de um título de seção do portfólio

Todos os títulos de seções seguirão o mesmo padrão do Código 5.23. Este contém uma `<nav>`, onde o título é inserido através de um `'h3'` e a div com classe `'max'` ocupa todo o espaço vazio disponível naquela mesma linha. O cor do conteúdo dentro do botão é definido pela classe `'tertiary-text'`.

Depois do título, uma div com `'medium-space'` e, por fim, o título do trabalho, a data de lançamento e uma linha para dividir os seus subsequentes, assim como o Código 5.24 apresenta. Nesta exposição, recomenda-se até cinco títulos.

```
<h6 class="s8 m8 l8">BeerCSS: Interfaces Web Responsivas</h6>
<p class="right-align tertiary-text monospace s2 m3 l2">Agosto 2024</p>
<hr class="small s10 m8 l6">
```

Código 5.24. Exemplo do artigo no portfólio

Nessa seção as classes novas são `'tertiary-text'`, para colorir e dar destaque ao trecho e `'monospace'` para aplicar um estilo de fontes, ambas para identificação visual.

A próxima seção exige um título nos mesmos conformes da área de *Artigos* com o nome *Habilidades*, sem o botão ao lado. Por isso, o Código 5.25 nos apresenta somente o modelo de como cada habilidade deve ser disposta.

```

<h6 class="bold">Typescript</h6>
<p class="large-text top-align">Superconjunto de Javascript, com
  tipagem estatica e robustez.</p>
```

Código 5.25. Exemplo da habilidade no portfólio

Essa `<div>` é um elemento composto pelo ícone da tecnologia, o seu nome e uma legenda que explica para quem o desenvolvedor a utiliza. As classes da imagem são divididas entre `'left middle'`, para posicionamento dentro da sua `<div>`, `'round'` para arredondamento dos seus cantos e `'extra'` para torná-la um ícone. Nesta seção é recomendado um número par dessas tecnologias, caso contrário, criar uma `<div>` com a classe `'s12 m6 l6'`.

Após isso, a seção *Projetos* segue a mesma lógica de títulos de seção com um botão, como a de *Artigos*. Dentro de *Projetos*, serão inseridos os cartões apresentados na Seção 5.4.6.2, como exibido no Código 5.26.

```

<article class="medium-width border surface-container s12 m6 l4">
  <time class="">2020</time>
  <h5 class="bold">TakeNote</h5>
  <p class="">Um aplicativo open source de notas para paginas web.</p>
  <nav>
    <button class="border small-round tertiary-border">
      <span class="tertiary-text bold">Documentacao</span>
    </button>
    <button class="border small-round tertiary-border">
      <span class="tertiary-text bold">Demo</span>
    </button>
  </nav>
</article>

```

Código 5.26. Exemplo de projeto na portfólio

Depois de implementar todas as seções de conteúdo na página, o desenvolvedor deve colocar um campo para que o leitor, caso interessado, entre em contato. O Código 5.27 demonstra como criar essa área.

```

<h6 class=" s12 m12 l12">Se quiser entrar em contato, me envie uma
  mensagem na caixa de texto abaixo.</h6>
<div class="field border extra">
  <input type="text">
</div>
<button class="border small-round tertiary-border">
  <span class="tertiary-text bold">Enviar</span>
</button>

```

Código 5.27. Exemplo da área de contato no portfólio

Dessa forma, finaliza-se os elementos do portfólio e as suas disposições.

5.5.2. Habilitando os Temas

Nesta seção será exibido o código JavaScript desse projeto. O Código 5.28 contém as linhas que serão inseridas neste arquivo.

```

const toggleTheme = async () => {
  const currentMode = ui("mode");
  if (currentMode === "light") {
    ui("mode", "dark");
  } else {
    ui("mode", "light");
  }
};
document.getElementById('theme-toggle').addEventListener('click',
  toggleTheme);
document.getElementById("theme-toggle").addEventListener("click",
  function() {
    const icon = this.querySelector("i");
    if (icon.textContent === "light_mode") {
      icon.textContent = "dark_mode";
    } else {
      icon.textContent = "light_mode";
    }
  });

```

```
}  
});
```

Código 5.28. Conteúdo do arquivo JavaScript do projeto

5.6. Conclusão

Este capítulo explorou a evolução das interfaces, destacando a Interface Humano-Computador (IHC) e seu papel fundamental na melhoria da interação entre usuários e sistemas computacionais. A importância do Material Design foi enfatizada, revelando como suas diretrizes ajudam na criação de interfaces visuais que combinam estética e funcionalidade.

O BeerCSS foi apresentado como uma solução prática e eficiente, alinhada com os princípios do Material Design. Seu design leve permite a construção rápida de interfaces, e a configuração via CDN e NPM, além da implementação de temas claros e escuros, mostra sua flexibilidade. Essas ferramentas e técnicas ajudam desenvolvedores a criar experiências de usuário adaptáveis e atraentes, respondendo às demandas modernas de design e usabilidade. Por fim, demonstra a criação de um portfólio de desenvolvedor a partir dessa ferramenta.

O portfólio construído neste processo está disponível em [ERCEMAPI 2024] e o código-fonte pode ser acessado no repositório do GitHub em [Alves 2024].

Referências

- [Alves 2024] Alves, M. C. (2024). Código do beercss. Acesso em: agosto 26, 2024.
- [Apple 2024] Apple (2024). Human interface guidelines. Acesso em: 16 de agosto de 2024.
- [BeerCSS 2022] BeerCSS (2022). Beercss: A lightweight css framework. Disponível em: <https://www.beercss.com/>. Acessado em: 31 de maio de 2024.
- [BeerCSS 2024] BeerCSS (2024). Beercss settings documentation. <https://github.com/beercss/beercss/blob/main/docs/SETTINGS.md>. Acesso em: 24 ago. 2024.
- [Clarke 2023] Clarke, A. (2023). *SEO 2023: Learn Search Engine Optimization with Smart Internet Marketing Strategies*. Simple Effectiveness LLC. Focado em estratégias atualizadas de SEO para 2023.
- [Dulsrud and Bygstad 2022] Dulsrud, A. and Bygstad, B. (2022). Digital ecosystems as social institutions: exploring the role of consumption through four research streams of digital ecosystems. *Consumption and Society*, 1(1):99 – 119.
- [ERCEMAPI 2024] ERCEMAPI (2024). Portfólio ercemapi. Acesso em: agosto 26, 2024.
- [Eyal and Hoover 2014] Eyal, N. and Hoover, R. (2014). *Hooked: How to Build Habit-forming Products*. Penguin Business. Portfolio Penguin.

- [Franz and Kauer-Franz 2024] Franz, B. and Kauer-Franz, M. (2024). *Usability and User Experience Design: The Comprehensive Guide to Data-Driven UX Design*. Rheinwerk Computing. Rheinwerk Publishing.
- [Garrett 2010] Garrett, J. J. (2010). *The Elements of User Experience: User-Centered Design for the Web and Beyond*. Voices That Matter. Pearson Education, 2nd edition.
- [Google 2014] Google (2014). Material design 1.0. Disponível em: <https://material.io/design>. Acessado em: 16 de agosto de 2024.
- [Google 2022] Google (2022). Material design 3.0. Disponível em: <https://material.io/design>. Acessado em: 16 de agosto de 2024.
- [Inc. 2024] Inc., G. (2024). Google. Acesso em: 15 de agosto de 2024.
- [Johnson 2014] Johnson, J. (2014). *Designing with the Mind in Mind: Simple Guide to Understanding User Interface Design Guidelines*. Elsevier, London, 2 edition.
- [Kao and Archer 1997] Kao, D. and Archer, N. P. (1997). Abstraction in conceptual model design. *International Journal of Human-Computer Studies*, 46(1):125–150.
- [Krug 2000] Krug, S. (2000). *Don't Make Me Think!: A Common Sense Approach to Web Usability*. Circle.com Library. New Riders.
- [Marcotte 2011] Marcotte, E. (2011). *Responsive Web Design*. Book Apart. A Book Apart.
- [Material 2024] Material (2024). Material theme builder. <https://material-foundation.github.io/material-theme-builder/>. Acessado em: agosto 20, 2024.
- [Microsoft 2024] Microsoft (2024). Fluent design system. Acesso em: 16 de agosto de 2024. O Fluent Design System foi introduzido em 2017.
- [Nielsen et al. 2014] Nielsen, J., Pernice, K., and Whitenton, K. (2014). *How People Read on the Web: The Eyetracking Evidence*. Nielsen Norman Group, New York.
- [Norman 2002] Norman, D. A. (2002). *The Design of Everyday Things*. Basic Books, New York.
- [Raskin 2000] Raskin, J. (2000). *The Humane Interface: New Directions for Designing Interactive Systems*. ACM Press Series. Addison-Wesley Professional, illustrated edition.
- [Waldeilson and Paixão 2016] Waldeilson, A. L. Z. and Paixão (2016). Influência da estética visual na usabilidade e experiência do usuário em interfaces humano-computador. *Ergodesign HCI*, 4(Especial):93–101.
- [Zaphiris 2008] Zaphiris, P., editor (2008). *Human Computer Interaction: Concepts, Methodologies, Tools, and Applications, Volume 3*. IGI Global, illustrated edition.