



Universidade de Brasília

Instituto de Ciências Exatas  
Departamento de Ciência da Computação

## **Proposta de aplicação web colaborativa para integração e acolhimento de estudantes**

Henrique Lopes Curzio

Monografia apresentada como requisito parcial  
para conclusão do Curso de Engenharia da Computação

Orientador  
Prof. Dr. ...

Brasília  
2023



Universidade de Brasília

Instituto de Ciências Exatas  
Departamento de Ciência da Computação

## **Proposta de aplicação web colaborativa para integração e acolhimento de estudantes**

Henrique Lopes Curzio

Monografia apresentada como requisito parcial  
para conclusão do Curso de Engenharia da Computação

Prof. Dr. ... (Orientador)  
CIC/UnB

Prof. Dr. Donald Knuth    Dr. Leslie Lamport  
Stanford University      Microsoft Research

Prof.a Dr.a Ada Lovelace  
Coordenadora do Curso de Engenharia da Computação

Brasília, 20 de março de 2023

# Dedicatória

Dedico este trabalho aos meus pais.

# Agradecimentos

Agradeço primeiramente a Deus. Em segundo lugar, às pessoas que investiram caro na minha formação e tornaram este projeto possível; meus pais, tia e avó. Em terceiro, agradeço aos que acreditaram em mim, me apoiaram e me incentivaram a continuar; irmãos, familiares e líderes. E, finalmente, aos tutores que me capacitaram com conhecimento e habilidades necessárias para cumprir este trabalho e lidar com desafios futuros; meus pastores e professores escolares, universitários e orientadores.

# Resumo

O *resumo* é um texto inaugural para quem quer conhecer o trabalho, deve conter uma breve descrição de todo o trabalho (apenas um parágrafo). Portanto, só deve ser escrito após o texto estar pronto. Não é uma coletânea de frases recortadas do trabalho, mas uma apresentação concisa dos pontos relevantes, de modo que o leitor tenha uma ideia completa do que lhe espera. Uma sugestão é que seja composto por quatro pontos: 1) o que está sendo proposto, 2) qual o mérito da proposta, 3) como a proposta foi avaliada/validada, 4) quais as possibilidades para trabalhos futuros. É seguido de (geralmente) três palavras-chave que devem indicar claramente a que se refere o seu trabalho. Por exemplo: *Este trabalho apresenta informações úteis a produção de trabalhos científicos para descrever e exemplificar como utilizar a classe L<sup>A</sup>T<sub>E</sub>X do Departamento de Ciência da Computação da Universidade de Brasília para gerar documentos. A classe UnB-CIC define um padrão de formato para textos do CIC, facilitando a geração de textos e permitindo que os autores foquem apenas no conteúdo. O formato foi aprovado pelos professores do Departamento e utilizado para gerar este documento. Melhorias futuras incluem manutenção contínua da classe e aprimoramento do texto explicativo.*

**Palavras-chave:** LaTeX, metodologia científica, trabalho de conclusão de curso

# Abstract

O *abstract* é o resumo feito na língua Inglesa. Embora o conteúdo apresentado deva ser o mesmo, este texto não deve ser a tradução literal de cada palavra ou frase do resumo, muito menos feito em um tradutor automático. É uma língua diferente e o texto deveria ser escrito de acordo com suas nuances (aproveite para ler [http://dx.doi.org/10.6061/2Fclinics%2F2014\(03\)01](http://dx.doi.org/10.6061/2Fclinics%2F2014(03)01)). Por exemplo:

**Keywords:** LaTeX, scientific method, thesis

# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Motivação e contexto . . . . .	1
1.2	Objetivos . . . . .	2
1.3	Requisitos . . . . .	4
1.3.1	Modelo entidade relacionamento . . . . .	5
1.4	Soluções existentes . . . . .	5
1.5	Organização do trabalho . . . . .	7
<b>2</b>	<b>Referencial teórico</b>	<b>8</b>
2.1	Aplicativo Web . . . . .	8
2.1.1	<i>Web standards</i> : HTML, CSS e JavaScript . . . . .	9
2.1.2	<i>Single Page Application (SPA)</i> . . . . .	11
2.1.3	<i>Web development tools</i> . . . . .	13
2.1.4	Arquitetura . . . . .	14
2.2	Pilha de tecnologias . . . . .	15
2.2.1	Node.js . . . . .	15
2.2.2	Bootstrap . . . . .	17
2.2.3	SQLite . . . . .	19
2.2.4	Sistema de controle de versões . . . . .	20
2.2.5	Visual Studio Code . . . . .	23
2.2.6	Outras tecnologias auxiliares . . . . .	25
<b>3</b>	<b>Metodologia</b>	<b>27</b>
3.1	Engenharia de requisitos . . . . .	27
3.2	Rapid application development (RAD) . . . . .	27
3.3	Continuous Integration/Continuous Deployment (CI/CD) . . . . .	28
3.4	Unified Modeling Language (UML) . . . . .	28
3.4.1	Diagrama de caso de uso . . . . .	28
3.4.2	Diagrama de classe . . . . .	28

3.4.3	Diagramas de atividades . . . . .	28
3.4.4	Diagramas de estado . . . . .	28
3.4.5	Diagramas de sequência . . . . .	28
3.5	Modelo Entidade-Relacionamento (MER) . . . . .	29
3.6	Ambiente de implantação (Deployment environment) . . . . .	29
3.7	CRUD . . . . .	30
3.8	Protocolos . . . . .	31
3.9	Algoritmos e funções . . . . .	31
<b>4</b>	<b>Desenvolvimento</b>	<b>32</b>
4.1	Autenticação . . . . .	32
<b>5</b>	<b>Resultados</b>	<b>34</b>
5.1	. . . . .	34
<b>6</b>	<b>Conclusões</b>	<b>35</b>
6.1	Considerações finais . . . . .	35
6.2	Trabalhos futuros . . . . .	35
	<b>Referências</b>	<b>36</b>



# Lista de Figuras

1.1	Diagrama de modelo entidade relacionamento. . . . .	6
2.1	Os usuários de aplicativos gastam 77% de seu tempo em seus 3 principais aplicativos. . . . .	9
2.2	<i>The 2017 U.S. Mobile App Report.</i> . . . .	9
2.3	Especificação HTML na pilha de especificações da plataforma web. . . . .	10
2.4	Single Page Application (SPA) vs multiple pages sites. . . . .	12
2.5	<i>Browser Market Share Worldwide.</i> . . . .	14
2.6	Arquitetura de aplicações Web. . . . .	14
2.7	Arquitetura Node.js. . . . .	15
2.8	<i>Stack Overflow Developer Survey 2022 - Web frameworks and technologies.</i> . . . .	17
2.9	<i>Bootstrap breakpoints.</i> . . . .	18
2.10	<i>RDBMS vs SQLite.</i> . . . .	20
2.11	<i>Stack Overflow Developer Survey 2022 - Version control systems.</i> . . . .	21
2.12	<i>Fluxo de dados Git.</i> . . . .	22
2.13	<i>Stack Overflow Developer Survey 2022 - Version control platforms.</i> . . . .	23
2.14	<i>Stack Overflow Developer Survey 2022 - Integrated development environment (IDE).</i> . . . .	24
2.15	Interface de usuário do MailDev. . . . .	26
3.1	Fases na abordagem de James Martin para RAD. . . . .	27
3.2	Diagrama de sequência da solicitação de cadastro. . . . .	29
3.3	CRUD na interface de usuário (UI). . . . .	30
3.4	Mapeamento comum entre as operações de CRUD e HTTP. . . . .	31
4.1	Cadastro. . . . .	33

# Lista de Abreviaturas e Siglas

**ACID** atomicity, consistency, isolation, durability.

**API** application programming interface.

**CI/CD** Continuous Integration/Continuous Deployment.

**CRUD** create, read, update, and delete.

**CSS** Cascading Style Sheets.

**ENEM** Exame Nacional do Ensino Médio.

**HTML** HyperText Markup Language.

**HTTP** Hypertext Transfer Protocol.

**IDE** Integrated development environment.

**JIT** just-in-time.

**npm** Node Package Manager.

**PNAD Contínua** Pesquisa Nacional por Amostra de Domicílios Contínua.

**RAD** Rapid application development.

**RDBMS** Relational Database Management System.

**SGBDR** Sistema de Gerenciamento de Banco de Dados Relacional.

**SIGAA** Sistema Integrado de Gestão de Atividades Acadêmicas.

**SPA** Single Page Application.

**UI** user interface.

**UnB** Universidade de Brasília.

**VCS** Version Control System.

**W3C** World Wide Web Consortium.

# Capítulo 1

## Introdução

### 1.1 Motivação e contexto

A Universidade de Brasília (UnB) disponibiliza de vários recursos para estudantes universitários aprenderem sobre o funcionamento da graduação e obterem informações para guiá-los ao longo de sua trajetória na universidade. Estes recursos variam desde ferramentas, plataformas e manuais que facilitam a gestão das atividades acadêmicas e ensinam sobre os procedimentos a serem seguidos pelos alunos, principalmente os recém admitidos na instituição. Entre eles, a Página de Boas Vindas<sup>1</sup> e o Manual de Graduação<sup>2</sup>, que apresentam informações sobre registro acadêmico, matrícula e vários outros serviços da universidade.

Para produzir estes materiais, a instituição utiliza de alguns recursos para obter as informações que precisam ser disponibilizadas aos estudantes. Entre eles está a própria experiência adquirida pelos docentes, coordenadores e administradores ao longo dos períodos de aulas anteriores, diante das solicitações e dúvidas dos alunos. Outra forma é por meio de formulários encontrados, por exemplo, na hora da matrícula no Sistema Integrado de Gestão de Atividades Acadêmicas (SIGAA) utilizado pela universidade. Além disso, cada departamento ou instituto possui suas próprias páginas de informações dedicada aos alunos dos seus cursos específicos. Assim, a secretaria disponibiliza, em sua própria página web, informações particulares e específicas daquele departamento. Por fim, a universidade disponibiliza também o recurso da ouvidoria como forma de promover mudanças e garantir a satisfação de todos os membros da instituição.

A maior parte destes recursos, no entanto, têm como expositor dos dados os docentes e servidores integrantes da instituição, mas não os próprios alunos. Por mais que apresentem, nos portais, informações de excelente qualidade obtidas do parecer dos estudantes,

---

<sup>1</sup><https://www.boasvindas.unb.br/>

<sup>2</sup>[https://deg.unb.br/images/DEG/estudante/manual\\_para\\_estudantes\\_2022.pdf](https://deg.unb.br/images/DEG/estudante/manual_para_estudantes_2022.pdf)

ainda existe uma carência de compartilhamento de informações entre os próprios alunos. Isto é perceptível pela existência de grupos nas redes sociais e aplicativos de mensagens destinados a assuntos da universidade com uma grande quantidade de discentes participantes. Esses grupos, no entanto, apesar de suprirem essa necessidade de compartilhar informações entre os próprios estudantes, não apresentam uma interface adequada para a organização de informações por meio, por exemplo, de filtragem de categorias.

Existem também alguns recursos disponibilizados pela instituição para compartilhamento de informações entre os próprios estudantes de forma geral, como a criação de fóruns e grupos nas ferramentas Microsoft Teams<sup>3</sup> e Moodle<sup>4</sup>. Estes podem ser destinados a todos os estudantes da instituição ou de um departamento específico. Porém eles também têm o mesmo problema de organização que as redes sociais e os grupos de mensagem apresentam. Além disso, eles concorrem em interface e experiência do usuário com os grandes criadores mundiais de software de redes sociais e serviço de mensagens. Com isso, a interação entre os alunos é bem menor nas redes da própria instituição.

## 1.2 Objetivos

Este trabalho tem como objetivo elaborar uma proposta de solução de um software colaborativo para integração de estudantes, por meio de um protótipo de projeto web que permite os alunos se cadastrarem e compartilharem informações entre eles. As categorias de informações são pré-cadastradas na plataforma, mas podem ser adicionadas ou modificadas pelo desenvolvedor conforme a demanda de utilização da ferramenta.

O cadastro dos alunos é feito por meio do e-mail estudantil oferecido pela universidade. O envio de um link de confirmação por e-mail para ativação da conta garante que somente os discentes da instituição podem se cadastrar. Além disso, o código-fonte é código aberto, ou seja, disponibilizado gratuitamente para consulta, examinação, modificação e redistribuição. Assim, outras instituições podem rodar o *software* em seus próprios servidores e fazer modificações para se adequar às suas necessidades específicas como, por exemplo, o domínio de e-mail permitido para cadastro.

Este projeto tem como proposta inicial uma solução com apenas dois tipos de usuários: estudante e administrador. O estudante é qualquer usuário com um e-mail institucional fornecido pela UnB e que pode realizar o cadastro na plataforma. O administrador é um ou mais usuários pré-cadastrados no sistema e que tem acesso a visualização, edição e exclusão de todos os dados disponíveis criados pelos outros usuários, os estudantes.

---

<sup>3</sup><https://teams.microsoft.com/>

<sup>4</sup><https://aprender3.unb.br/>

Além disso, ele tem permissão para gerir as próprias contas dos usuários, podendo criar, modificar e deletá-las.

Este sistema visa ser parte de uma estrutura oferecida pela universidade que busca estabelecer medidas institucionais de acolhimento estudantil e de promoção do sucesso acadêmico de forma a aumentar a inclusão. Esta necessidade existe principalmente entre os novos estudantes universitários. As funcionalidades do sistema são separadas por seções que são acessadas por meio de itens na barra de menus superior. As seções fornecidas pela proposta inicial são: ferramentas, locais e cuidados. Estas seções, no entanto, podem ser aumentadas e modificadas em trabalhos futuros.

## **Ferramentas**

A ideia desta página é disponibilizar uma lista ferramentas que possam ser úteis para o dia a dia dos estudantes. Elas podem ser recursos físicos, como materiais escolares, ou recursos digitais. Hoje em dia cada vez mais a internet é utilizada pelos estudantes, usando desde os computadores aos *smartphones*. Considerando este cenário, esta coletânea de ferramentas é uma medida de inclusão digital para os estudantes com acesso tardio ao cenário da internet, através da colaboração de outros alunos com mais experiências e que gostariam de compartilhar conhecimento.

Esta seção se destina à sugestão, por parte dos alunos, principalmente de ferramentas externas à universidade. Isto porque a maioria das ferramentas utilizadas internamente, como Moodle e Teams já são recomendadas pelos manuais institucionais e professores.

## **Locais**

Este setor da plataforma destina-se à integração principalmente de alunos que são de outras regiões do país ou, até mesmo, estrangeiros e que ainda não têm muita familiaridade com a cidade. No caso da UnB, muitos destes estudantes são ingressantes por meio da prova do Exame Nacional do Ensino Médio (ENEM), da Mobilidade Acadêmica (Convênio Interinstitucional Internacional e Aluno Visitante) ou de outras maneiras. Estas formas de ingresso facilitaram a entrada de alunos de outras regiões do Brasil e de fora do país na instituição. Desta forma, a plataforma incentiva a colaboração entre estes estudantes que não são do local com os que já habitam próximo há mais tempo. Assim, esta medida facilita o acolhimento no ambiente universitário.

O objetivo da sessão de locais não se destina, no entanto, a fornecer sugestões de locais da própria universidade. Muitas instituições já disponibilizam um mapa da maioria dos departamentos e espaços comunitários. A UnB fornece também um mapeamento facilitado de salas por meio de uma numeração por metragem.

## Cuidados

A página de cuidados é uma seção de sugestões dos próprios alunos como medida de prevenção de acidentes e problemas do cotidiano. Ela se destina a fornecer informações de quaisquer problemas relacionados ao dia a dia da universidade que possam causar problemas aos estudantes. Coisas como falta de acessibilidade em locais específicos, linhas de ônibus desativadas, mudanças de locais de acesso, etc. Esta solução pode facilitar, por exemplo, o deslocamento de estudantes com dificuldades de acessibilidade por meio da busca prévia de vias alternativas. Pode evitar também deslocamentos desnecessários ou até mesmo acidentes mais graves.

Não objetiva ser uma página para reclamações ou sugestões. A UnB, assim como a maioria das universidades, já dispõe de recursos de ouvidoria, cujo o foco é a resolução de problemas. A ferramenta do projeto, no entanto, foca na informação e prevenção. A ideia é que, enquanto o problema não é resolvido, ele pode ser contornado.

## 1.3 Requisitos

### Ferramentas

A página contém os seguintes requisitos:

- Listagem inicial de todas as ferramentas com filtragem por categorias;
- Modal para inclusão de novas ferramentas;
- Modal para edição de ferramentas anteriormente criadas;
- Botão para exclusão da ferramenta;
- Somente o usuário que inseriu a ferramenta pode editar e deletar ela (com exceção do administrador)

O item *ferramenta* deve conter os seguintes campos: ID, nome, descrição, link, imagem, ID do usuário que inseriu, data de criação.

### Locais

A página contém os seguintes requisitos:

- Listagem inicial de todos os locais com filtragem por tipo;
- Modal para inclusão de novos locais;
- Modal para edição de locais anteriormente criados;

- Botão para exclusão da local;
- Somente o usuário que inseriu o locais pode editar e deletar ele (com exceção do administrador)

O item *local* deve conter os seguintes campos: ID, nome, descrição, tipo, foto, site, localização, ID do usuário que inseriu, data de criação.

## Cuidados

A página contém os seguintes requisitos:

- Listagem inicial de todas as sugestões de cuidados com filtragem por principal público alvo;
- Modal para inclusão de novas sugestões de cuidados;
- Modal para edição de sugestões de cuidados anteriormente criadas;
- Botão para exclusão da sugestões de cuidados;
- Somente o usuário que inseriu a sugestões de cuidados pode editar e deletar ela (com exceção do administrador)

O item *sugestões de cuidados* deve conter os seguintes campos: ID, nome, descrição, principal público alvo, grau de risco, foto, localização, ID do usuário que inseriu, data de criação.

### 1.3.1 Modelo entidade relacionamento

Figura 1.1

## 1.4 Soluções existentes

### Guia do aluno UnB: aplicativo georreferenciado de apoio a atividades acadêmicas

A monografia "Guia do Aluno UnB: Aplicativo Georreferenciado de Apoio a Atividades Acadêmicas"[1] apresentada como requisito parcial para conclusão do Curso de Licenciatura em Computação da Universidade de Brasília, elaborada pelos alunos Frederico Fernandes de Faria e Jean Victor Ribeiro Vieira propõe um aplicativo composto por um conjunto de ferramentas destinadas a todos os usuários da Universidade, vinculados ou visitantes, como software de apoio a atividades acadêmicas.[1]



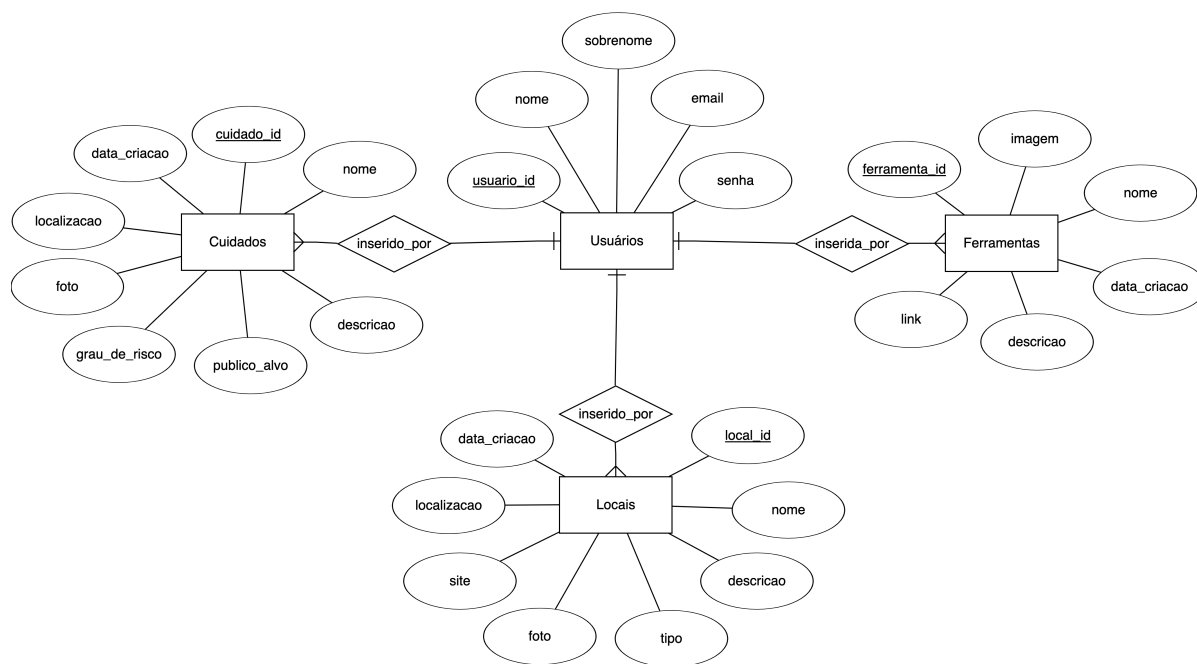


Figura 1.1: Diagrama de modelo entidade relacionamento.

- Busca das instalações de todos os campi da Universidade de Brasília;
- Mostra a localização do usuário no mapa, e indica o melhor caminho para um determinado ponto de origem;
- Busca matérias que estão no Matricula Web e indica qual o local da sala em que a disciplina está sendo ministrada.
- Gerencia as faltas e as tarefas das matérias;
- Disponibiliza o acervo de documentos e telefones úteis da UnB;
- Acessa os sites Matrícula Web e o portal da UnB.

### Scholar: Desenvolvimento de um aplicativo móvel genérico de apoio acadêmico a estudantes em universidades

O trabalho de conclusão de curso "Scholar: Desenvolvimento de um aplicativo móvel genérico de apoio acadêmico a estudantes em universidades" [2] apresentado como parte dos requisitos para obtenção do grau de Bacharel em Sistemas de informação na Universidade Federal de Santa Catarina, elaborado pelos alunos Angelo Manoel de Matos Leal e Adson Pereira Leal propõe um aplicativo [2]

- Grade de horário e local da aulas;

- Cardápio do Restaurante Universitário;
- Histórico escolar;
- Avaliações adicionadas individualmente ou por colegas;
- Controle de faltas;
- Calendário de provas;
- Atestado de matrícula

## **1.5 Organização do trabalho**

# Capítulo 2

## Referencial teórico

### 2.1 Aplicativo Web

Um aplicativo web (ou *web app*) é um software de aplicativo acessado por meio de um navegador da Web.[3] Enquanto aplicativos nativos precisam ser desenvolvidos individualmente para cada plataforma, os aplicativos web estão no extremo oposto. O código não é escrito visando a plataforma, mas sim qualquer navegador rodando sobre ela.[4] Existem também algumas ferramentas intermediárias como compilador cruzado (*cross compiled*), por exemplo o Flutter<sup>1</sup> e *web native apps*, por exemplo o React Native<sup>2</sup>. No entanto, algumas estatísticas mostradas a seguir sugerem que a escolha do aplicativo web é mais adequada para este projeto.

De acordo com o *The 2017 U.S. Mobile App Report*[5], nos Estados Unidos, os usuários de *smartphones* gastam quase 80% do uso total de aplicativos em seus três aplicativos favoritos e impressionantes 96% em seus 10 principais aplicativos pessoais conforme ilustrado na Figura 2.1. Este cenário, por mais que não represente uma amostra de estudantes da universidade e nem mesmo da população brasileira, representa uma tendência dos usuários de utilizarem apenas poucos aplicativos nos celulares. Esta mesma pesquisa afirma que a fidelidade de um usuário a um aplicativo móvel é muito mais relevante do que a sites móveis.

No entanto, a mesma pesquisa afirma que, apesar dos usuários gastarem 16 vezes mais tempo nos principais aplicativos do que nos principais sites, os sites tendem a atrair públicos maiores, ou um maior número de visitantes por mês conforme ilustrado na Figura 2.2.

Considerando estas informações, foi tomada a decisão pelo desenvolvimento de um aplicativo web já que a plataforma para integração de estudantes não visa ser um aplicativo de uso recorrente como os principais aplicativos utilizados pelo usuário mostrado na

---

<sup>1</sup><https://flutter.dev/>

<sup>2</sup><https://reactnative.dev/>

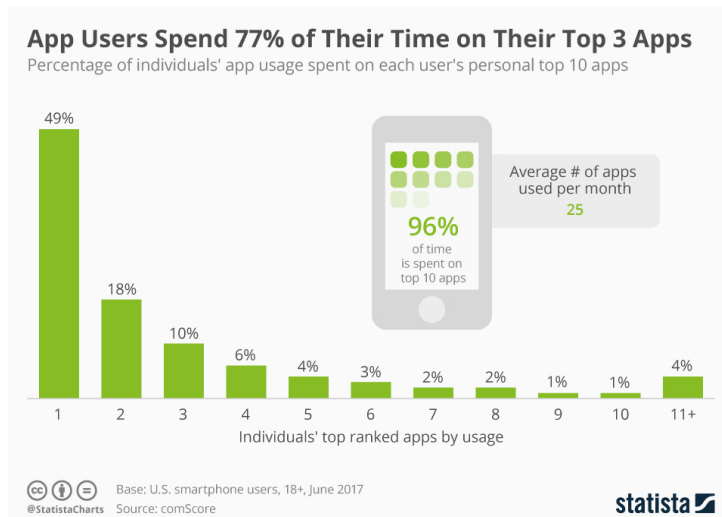


Figura 2.1: Os usuários de aplicativos gastam 77% de seu tempo em seus 3 principais aplicativos (Fonte: [6]).

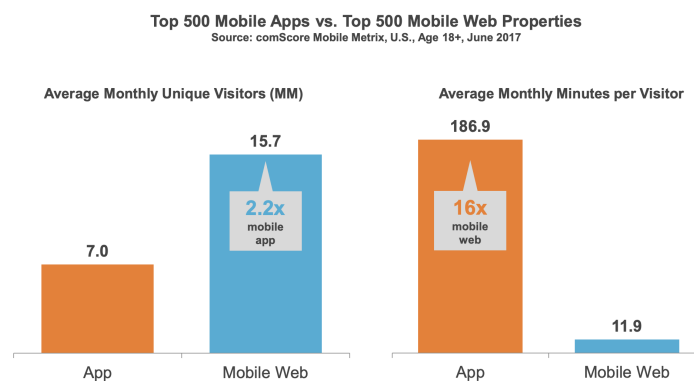


Figura 2.2: *The 2017 U.S. Mobile App Report* (Fonte: [5]).

pesquisa. Além do mais, considerando a resistência existente por parte dos usuários para instalação de novas aplicações nos seus dispositivos, a solução proposta visa não causar resistência de uso para os usuários. Por fim, o desenvolvimento para uma única plataforma, o navegador, facilita a manutenção do software e provê mais agilidade na entrega do produto.

### 2.1.1 *Web standards*: HTML, CSS e JavaScript

Considerando a escolha pela plataforma web, conseqüentemente é definida a escolha pelas ferramentas base para desenvolvimento web: HTML, CSS e JavaScript.

*Web Standards*, traduzido como "Normas para Web", são padrões formais e não proprietários e outras especificações técnicas que definem e descrevem aspectos da *World*

*Wide Web* definidos pela World Wide Web Consortium (W3C). O termo tem sido associado com mais frequência à tendência de endossar um conjunto de práticas recomendadas padronizadas para a construção de sites da Web. Quando um site ou página da web é descrito como em conformidade com os padrões da web, isso geralmente significa que o site ou a página tem HTML, CSS e JavaScript.<sup>3</sup>

## HTML

A especificação oficial do HyperText Markup Language (HTML)<sup>4</sup> o define como "linguagem de marcação central da *World Wide Web*. Originalmente, o HTML foi projetado principalmente como uma linguagem para descrever semanticamente documentos científicos. A sua concepção geral, no entanto, permitiu que fosse adaptada, ao longo dos anos seguintes, para descrever uma série de outros tipos de documentos e mesmo aplicações"[7]. Além disso, a especificação definiu sua posição na pilha de especificações da plataforma web em relação a outras especificações conforme mostrado na Figura 2.3.



Figura 2.3: Especificação HTML na pilha de especificações da plataforma web (Fonte: [7]).

## CSS

A especificação oficial do Cascading Style Sheets (CSS)<sup>5</sup> o define como "uma linguagem para escrever *style sheet* (folhas de estilo)". Este termo é usado para expressar a apresentação de documentos estruturados, ou seja, em que algum método de marcação é usado

<sup>3</sup><https://www.webstandards.org/>

<sup>4</sup><https://html.spec.whatwg.org/>

<sup>5</sup><https://www.w3.org/TR/CSS/>

para identificar o todo e as partes do documento como tendo vários significados além de sua formatação. A especificação também afirma que o CSS é "projetado para descrever a renderização de documentos estruturados (como HTML e XML) em uma variedade de mídias. CSS é usado para descrever a apresentação de um documento de origem e geralmente não altera a semântica subjacente expressa por sua linguagem de documento." [8]

## JavaScript

JavaScript é uma linguagem compilada de alto nível, geralmente compilação *just-in-time* (*JIT*) também conhecida como tradução dinâmica ou em tempo de execução, que está em conformidade com o padrão ECMAScript.[9] Este é um padrão destinado a garantir a interoperabilidade de páginas da Web em diferentes navegadores. É padronizado pela Ecma International no documento ECMA-262.<sup>6</sup> O ECMAScript é comumente usado para scripts do lado do cliente na World Wide Web e está sendo cada vez mais usado para escrever aplicativos e serviços do lado do servidor usando Node.js e outros ambientes de tempo de execução. O JavaScript é a linguagem de script dominante do lado do cliente da Web, com 98% de todos os sites usando-o para essa finalidade.[10]

### 2.1.2 *Single Page Application (SPA)*

Os *Single Page Application (SPA)* ou aplicativos de página única são aplicativos ou sites que interage com o usuário reescrevendo a página dinamicamente com novos dados do servidor, em vez do método padrão de um navegador da Web que carrega novas páginas inteiras com o objetivo de fazer transições mais rápidas.[11]

Os SPAs têm como vantagem a performance após o carregamento inicial e a experiência do usuário intuitiva e de baixa fricção com sensação de coesão e imersão, construindo uma familiaridade com a interface rapidamente. Além disso, a página pode ser armazenada no cache do navegador e pode funcionar offline a partir dos dados já recebidos. Também promove agilidade de desenvolvimento já que, por serem baseadas em *frameworks* muito populares, as SPAs não apresentam grandes desafios de *debugging*. [12] Todas essas vantagens são essenciais para o projeto em questão, já que os estudantes precisam construir uma familiaridade com a plataforma para se sentirem participantes ativos, ou seja, criadores de conteúdo e fornecedores de informações da universidade, não apenas leitores.

Conforme mostrado na Figura 2.4, uma página inicial pode fazer parte de um site de várias páginas ou de uma única página. Desta forma, é possível perceber que as SPAs estimulam o reaproveitamento de código e mantém a consistência de itens repetidos entre páginas, por exemplo, o logo e o menu.

---

<sup>6</sup><https://www.ecma-international.org/publications-and-standards/standards/ecma-262/>

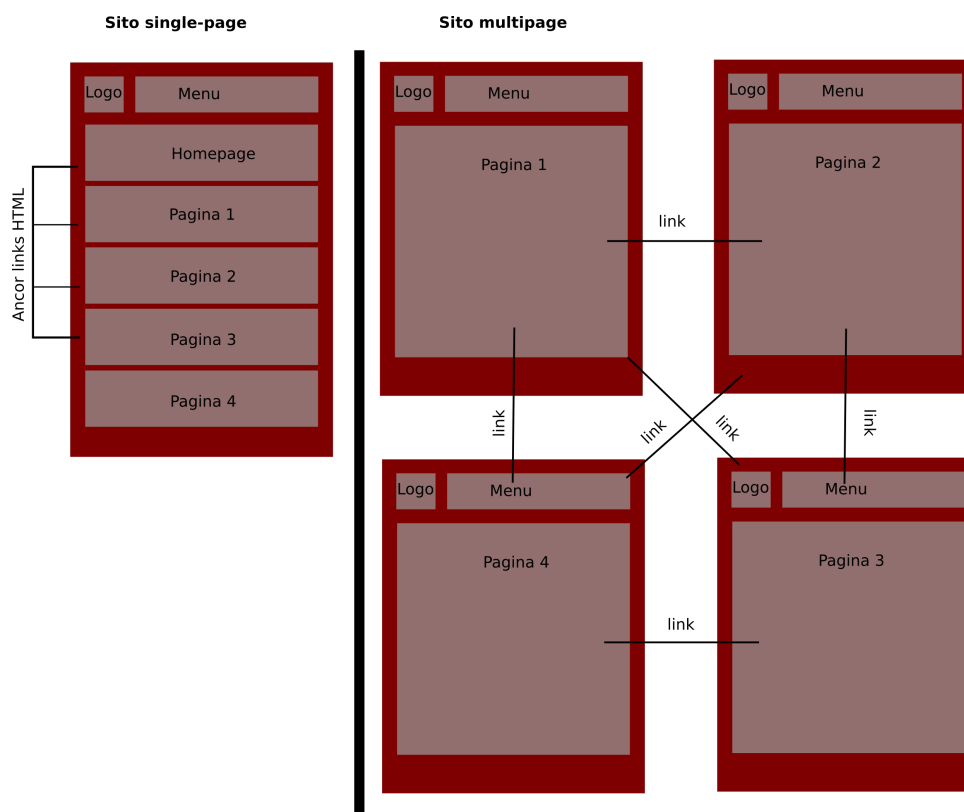


Figura 2.4: Single Page Application (SPA) vs multiple pages sites (Fonte: [13]).

Os mesmos fatores que tornam as SPAs uma escolha inteligente para muitas aplicações, também podem representar obstáculos.[12] Uma das desvantagens é com relação a mecanismos de busca que têm maiores dificuldades em conseguir detectar ou indexar este conteúdo, pois atualizam a mesma página para exibir diversas informações. Já existem tecnologia para gerenciamento de rotas que resolvem este problema, mas para o projeto em questão este problema não se aplica já que é uma ferramenta interna, ou seja, todo o conteúdo produzido pelos alunos só está disponível para usuários autenticados. Então as SPAs continuam sendo vantajosas.

Outra desvantagem é com relação ao tempo de carregamento. As SPAs costumam ter um carregamento inicial ligeiramente mais lento do que páginas estáticas. Além da diferença no tamanho do arquivo, a quantidade de conteúdo a ser renderizado pode significar um tempo de carregamento maior, mesmo quando a página está armazenada no cache. Como alternativa para este problema, o projeto deve ser focado na recuperação de conteúdo por meio de paginação.[14] Assim, a diminuição na recuperação de dados do banco de dados pode compensar na rapidez do carregamento dos arquivos estáticos.

### 2.1.3 *Web development tools*

As *Web development tools*, em português ferramentas de desenvolvimento da Web, também chamadas de *devtools* ou inspecionador de elemento, permitem que os desenvolvedores da Web testem e depurem seus códigos-fonte. São ferramentas usadas para testar a interface do usuário de um site ou aplicativo da web.[15]

As ferramentas de desenvolvimento da Web vêm como recursos integrados nos navegadores da Web. Os navegadores mais populares, como Google Chrome, Firefox, Internet Explorer, Safari, Microsoft Edge e Opera possuem estas ferramentas integradas. Em todos os navegadores, a funcionalidade básica das ferramentas do desenvolvedor permanece a mesma, apenas as convenções de nomenclatura e as terminologias mudam.[16]

Algumas funcionalidades, baseadas no Google Chrome, são:

- Elementos: permite ver o HTML e CSS da página da Web inspecionada no momento;
- Console: usado para depurar o JavaScript presente no código-fonte. Atua como a janela de depuração que permite lidar com o JavaScript que não está funcionando conforme o esperado;
- Fontes: permite definir pontos de interrupção e observar o valor das variáveis;
- Rede: usado para garantir que todos os recursos que estão sendo baixados ou carregados estão sendo executados conforme o esperado;
- Desempenho: é registrado em tempo de execução e informa como a página funciona quando está em execução, em vez de quando é carregada
- Aplicativo: utilizado para acompanhar e gerenciar *cookies*, sessões, cache e armazenamento local.

Todas essas funcionalidades são essenciais durante o processo de desenvolvimento do projeto. Considerando que os usuários finais da aplicação podem acessá-la de diferentes navegadores web, é adequado fazer testes em mais de um deles durante a criação do software. Para este projeto, dois deles são escolhidos, O Google Chrome e o Safari. Esta decisão é tomada com base na liderança deles na participação de mercado. A Figura 2.5 destaca uma pesquisa feita pela statCounter com a participação de mercado dos principais navegadores web. Nela é possível notar uma disparidade da primeira e segunda colocações para as demais.



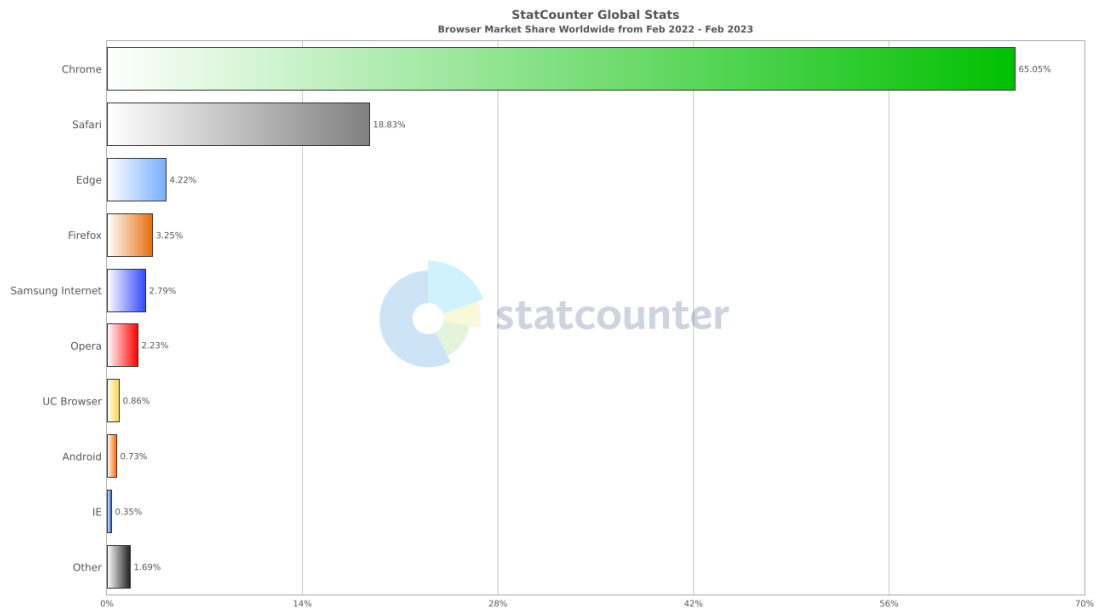


Figura 2.5: *Browser Market Share Worldwide* (Fonte: [17]).

## 2.1.4 Arquitetura

A arquitetura de uma aplicação web é composta dos elementos mostrados na Figura 2.6.[18] As seções seguintes detalha a escolha das ferramentas e recursos utilizados no projeto para cada uma dos elementos da arquitetura web.

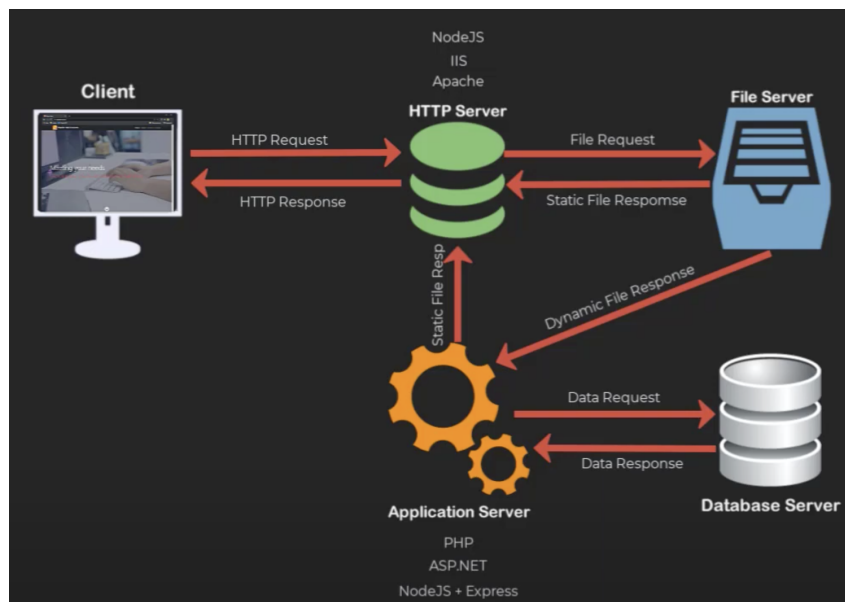


Figura 2.6: *Arquitetura de aplicações Web* (Fonte: [18]).

## 2.2 Pilha de tecnologias

### 2.2.1 Node.js

O Node.js foi criado como uma alternativa ao Apache HTTP Server buscando resolver algumas limitações dele, como o suporte relativamente baixo a múltiplas conexões. Este problema é resolvido com uma abordagem focada em eventos e execução assíncrona, sendo uma opção com menos consumo de recursos do que os servidores “tradicionais”. [19]

O Node.js é um ambiente de tempo de execução de JavaScript de back-end, executado no V8 *JavaScript Engine* e executa o código fora de um navegador da web. [20] O livro *Node.js Design Patterns* destaca os principais componentes da arquitetura do Node.js conforme mostrado na Figura 2.7.

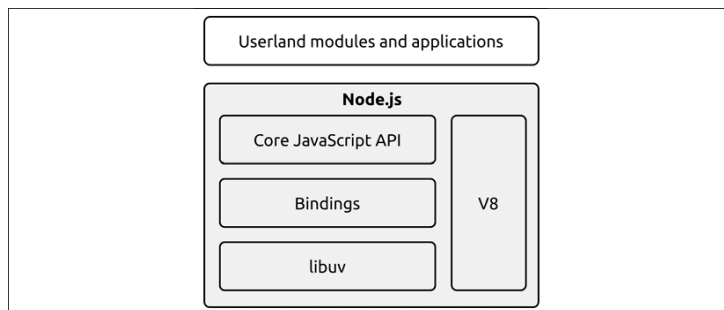


Figura 2.7: Arquitetura Node.js (Fonte: [20]).

#### V8

O V8 é um interpretador JavaScript, originalmente desenvolvido pelo Google para o navegador Chrome.

#### libuv

O *libuv* tem como objetivo de tornar o Node.js compatível com todos os principais sistemas operacionais e normalizar o comportamento não bloqueador dos diferentes tipos de recursos. *libuv* representa o mecanismo de E/S de baixo nível do Node.js e, segundo os autores do livro *Node.js Design Patterns*, é provavelmente o componente mais importante no qual o Node.js é construído.

#### Bindings

Os *bindings* são um conjunto de ligações responsáveis por agrupar e expor o *libuv* e outras funcionalidades de baixo nível para JavaScript.

## Core JavaScript API

O *Core JavaScript API* é uma biblioteca JavaScript básica que implementa a API Node.js de alto nível.

## npm

O Node Package Manager (npm) é o gerenciador de pacotes padrão do Node.js. Ele dá acesso a um enorme repositório de bibliotecas e módulos que podem ser adicionadas ao projeto, contendo ferramentas e soluções para os mais diversos problemas e requisitos.[19] O npm também pode instalar todas as dependências de um projeto definidas no arquivo package.json, o que facilita a distribuição do software.

## Stack Overflow Developer Survey 2022

O site *Stack Overflow*, um portal focado em perguntas e respostas para profissionais e entusiastas na área de programação de computadores, realizou em 2022 uma Pesquisa do Desenvolvedor com 73.268 desenvolvedores de software de 180 países ao redor do mundo.[21] Na seção *Web frameworks and technologies* foi feita uma pergunta que pode ser traduzida como:

*Em quais frameworks e tecnologias da web você fez um trabalho de desenvolvimento extensivo no ano passado e em quais você deseja trabalhar no próximo ano? (Se ambos trabalharam com a estrutura e desejam continuar a fazê-lo, marque as duas caixas nessa linha.)*

Esta questão obteve 49.625 respostas. A partir dos resultados, conforme mostrado na Figura 2.8, é perceptível uma relevância notável do Node.js na comunidade de desenvolvedores. Esta grande participação de mercado representa, conseqüentemente, a disponibilidade de um grande número de ferramentas e recursos desenvolvidos sobre a plataforma e uma evolução ativa e contínua da tecnologia.

## Decisão

Considerando os fatores acima de arquitetura e relevância do Node.js, esta plataforma foi adotada no projeto também por adotar uma mesma linguagem de desenvolvimento, tanto no *front-end* como no *back-end*, o JavaScript. Desta forma, existe a possibilidade de manter o ecossistema de aplicações e toda a base de código em uma só linguagem de programação.

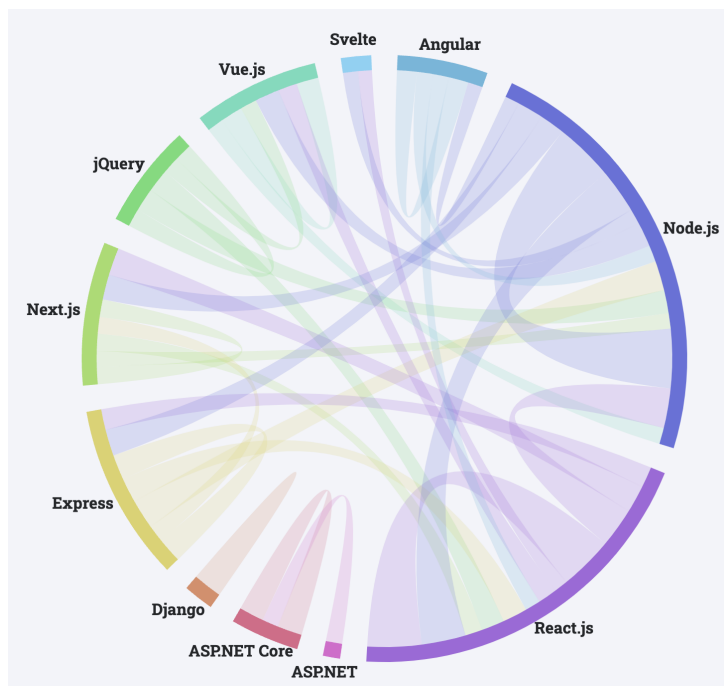


Figura 2.8: *Stack Overflow Developer Survey 2022 - Web frameworks and technologies* (Fonte: [21]).

## 2.2.2 Bootstrap

O Bootstrap<sup>7</sup> é uma estrutura CSS gratuita e de código aberto voltada para o desenvolvimento web front-end responsivo e móvel. Ele contém HTML, CSS e modelos de design baseados em JavaScript para tipografia, formulários, botões, navegação e outros componentes de interface.[22]

Em março de 2023, o Bootstrap é o 16º projeto mais estrelado no GitHub , com mais de 162.000 estrelas.[23] De acordo com a W3Techs, o Bootstrap é usado por 79,0% de todos os sites cujo framework CSS é conhecido. Isso representa 19,1% de todos os sites.[24]

Esta grande relevância do Bootstrap representa uma abundância de recursos da comunidade desenvolvido sobre a ferramenta, uma grande quantidade de modelos de *design* de páginas para inspiração e uma evolução ativa e contínua da tecnologia. Além disso, a ferramenta facilita o desenvolvimento de aplicações responsivas, o que aumenta a acessibilidade ao software desenvolvido, já que telefone celular segue como principal equipamento para acesso à internet no Brasil. Segundo a Pesquisa Nacional por Amostra de Domicílios Contínua (PNAD Contínua) ano de 2021, o aparelho é utilizado em 99,5% dos domicílios com acesso à internet.[25]

---

<sup>7</sup><https://getbootstrap.com>






<b>xs</b>	Extra small <576px	 portrait mobile
<b>sm</b>	Small ≥576px	 landscape mobile
<b>md</b>	Medium ≥768px	 portrait tablets <i>navbar collapse</i>
<b>lg</b>	Large ≥992px	 landscape tablets
<b>xl</b>	Extra large ≥1200px	 laptops, desktops, TVs

Figura 2.9: Bootstrap *breakpoints* (Fonte: [26]).

Bootstrap é usado para fazer o layout de sites usando itens como contêineres, linhas e colunas. Ele também permite que você adicione facilmente coisas como cartões e *accordions*. Um tema com cores e fontes globais pode ser configurado facilmente. Ele também possui muitas classes CSS predefinidas para estilizar facilmente. Dois dos principais recursos do Bootstrap são os *grid systems* e os *breakpoints*.

O *grid system* é uma grade *flexbox* móvel (um modelo de layout da Web CSS3) para criar layouts de todas as formas e tamanhos, graças a um sistema de doze colunas, seis níveis responsivos padrão e dezenas de classes predefinidas. O sistema de grade do Bootstrap usa uma série de contêineres, linhas e colunas para fazer o layout e alinhar o conteúdo. É construído com *flexbox* e é totalmente responsivo.[27] Já os *breakpoints* são larguras personalizáveis que determinam como seu layout responsivo se comporta em dispositivos ou tamanhos de viewport no Bootstrap.[28] A Figura 2.9 apresenta classes de grade predefinidas para criar rapidamente layouts de grade para diferentes tipos de dispositivos.[26]

### 2.2.3 SQLite

O site oficial do SQLite<sup>8</sup> o define como "uma biblioteca de linguagem C que implementa um mecanismo de banco de dados SQL leve, rápido, independente, de alta confiabilidade e completo. O SQLite é o mecanismo de banco de dados mais usado no mundo.[29] O SQLite está embutido em todos os telefones celulares e na maioria dos computadores e vem embutido em inúmeros outros aplicativos que as pessoas usam todos os dias. O formato de arquivo SQLite é estável, multiplataforma e compatível com versões anteriores e os desenvolvedores se comprometem a mantê-lo assim até o ano de 2050. Arquivos de banco de dados SQLite são comumente usados como contêineres para transferir conteúdo entre sistemas e como um formato de arquivamento de longo prazo para dados. Existem mais de 1 trilhão de bancos de dados SQLite em uso ativo.[29]"

Normalmente, um Relational Database Management System (RDBMS), em português Sistema de Gerenciamento de Banco de Dados Relacional (SGBDR), como MySQL, PostgreSQL, etc., requer um processo de servidor separado para operar. As aplicações que desejam acessar o servidor de banco de dados utilizam o protocolo TCP/IP para enviar e receber solicitações.[30] Já o SQLite é um bancos de dados independente, contido em arquivos. Como ele está contidos em arquivos, não há servidor necessário para manter os dados. Isso também significa que o banco de dados é específico do projeto, pois o arquivo é acessado no diretório do projeto. A Figura 2.10 ilustra a arquitetura cliente/servidor de um RDBMS (o mesmo se aplica para bancos não-relacionais em geral) em comparação à arquitetura sem servidor do SQLite.

Outras características do SQLite incluem ser independente de plataforma, requer suporte mínimo do sistema operacional, configuração zero por causa da arquitetura sem servidor, transacional, ou seja, todas as consultas e alterações são atômicas, consistentes, isoladas e duráveis (ACID).[30]

Assim SQLite é uma opção simples e compacta adequada para aplicativos que não exigem grandes sistemas de banco de dados. Exatamente o caso do projeto de integração de estudantes. Como os usuários são alunos específicos de uma instituição, então a quantidade de usuários é controlada. Além do mais, a estrutura do SQLite, mesmo sendo mais leve, ainda é capaz de suportar um aumento de escalabilidade bem acima do esperado pela plataforma a ser desenvolvida.

### Knex.js

O knex.js<sup>9</sup> é um *query builder* por meio do javascript que unifica a forma de fazer *queries* para os bancos de dados SQL, não dependendo de um banco específico e podendo trocar

---

<sup>8</sup><https://sqlite.org>

<sup>9</sup><https://knexjs.org>

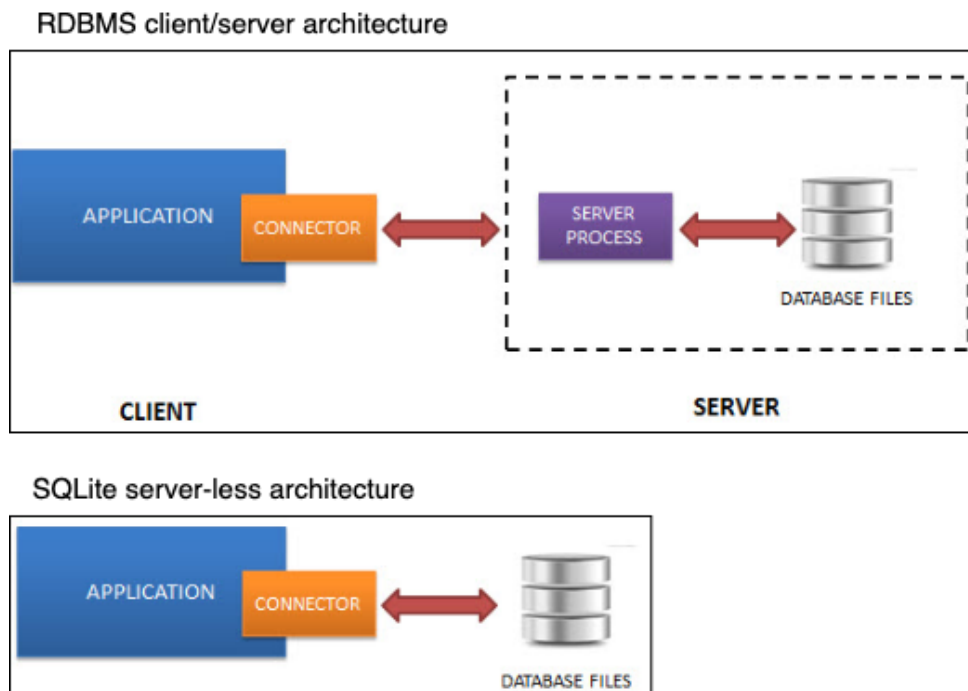


Figura 2.10: *RDBMS vs SQLite* (Fonte: [30]).

de banco a qualquer hora. Sendo assim, a aplicação ainda funcionará da mesma forma sem alterar o código. O ambiente em que ele é mais utilizado é o Node.js. O knex.js tem suporte completo ao Postgres, Sql Server, Mysql, MariaDB, Sqlite3, Oracle e Amazon Redshift.[31] Além disso ele disponibiliza de *migrations* (migrações) e *seeders* (propagadores) que referem-se, respectivamente, ao gerenciamento de alterações controladas por versão, incrementais e reversíveis em esquemas de banco de dados relacionais[32]; e ao preenchimento um banco de dados com um conjunto inicial de dados.[33]

Estas funcionalidades providas pelo Knex.js são essenciais para facilitar uma possível mudança futura no banco de dados utilizado pelo projeto de integração de estudantes. Esta alteração não é o propósito deste trabalho em específico, mas o projeto se torna, assim, mais escalável e manutenível.

## 2.2.4 Sistema de controle de versões

O *Version Control System (VCS)*, ou no português sistema de controle de versões, é uma categoria de ferramentas de software que ajudam a registrar as alterações feitas nos arquivos, mantendo um registro das modificações feitas no código.[34] Ele acompanha cada modificação no código em um tipo especial de banco de dados. Se um erro for cometido, os desenvolvedores podem voltar no tempo e comparar as versões anteriores do

código para ajudar a corrigir o erro, minimizando a interrupção de todos os membros da equipe.[35]

As três ferramentas de controle de versão mais conhecidas são Git, Subversion (SVN) e Mercurial.[36]. A pesquisa *Stack Overflow Developer Survey 2022* realizou, na seção *Version control systems*, a seguinte pergunta, que pode ser traduzida como: *Quais são os principais sistemas de controle de versão que você usa? Selecione tudo que se aplica*.[21]

Esta questão obteve 71.379 respostas. A partir dos resultados, conforme mostrado na Figura 2.11, é perceptível uma disparidade no número de utilização da ferramenta Git como sistema de controle de versões. Em decorrência desta ampla utilização e de seus recursos que serão abordados a seguir, esta ferramenta é escolha feita para utilização no projeto.

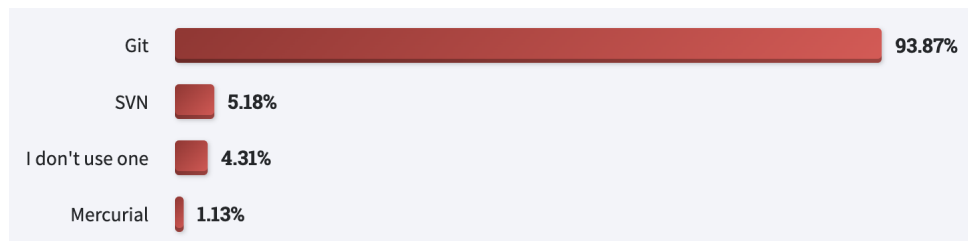


Figura 2.11: *Stack Overflow Developer Survey 2022 - Version control systems* (Fonte: [21]).

## Git

O Git<sup>10</sup> é um sistema de controle de versão distribuído que rastreia alterações em qualquer conjunto de arquivos de computador, geralmente usado para coordenar o trabalho entre programadores que desenvolvem código-fonte de forma colaborativa durante o desenvolvimento de software. Seus objetivos incluem velocidade, integridade de dados e suporte para fluxos de trabalho não lineares distribuídos.[37]

A Figura 2.12 mostra as principais partes que são importantes ao usar o Git. Como ele é um sistema distribuído, tem-se um repositório local (*local repository*) que é onde todas as alterações são registradas. Todas as suas alterações são primeiro confirmadas em seu repositório local e devem, então, ser explicitamente enviadas para um repositório remoto (*remote repository*). Os arquivos com seu código estão em um diretório de trabalho (*working directory*). Entre seu diretório de trabalho e o repositório local há uma área de preparação (*staging area*) que reúne todas as alterações antes de serem confirmadas no repositório local.[38]

---

<sup>10</sup><https://git-scm.com>



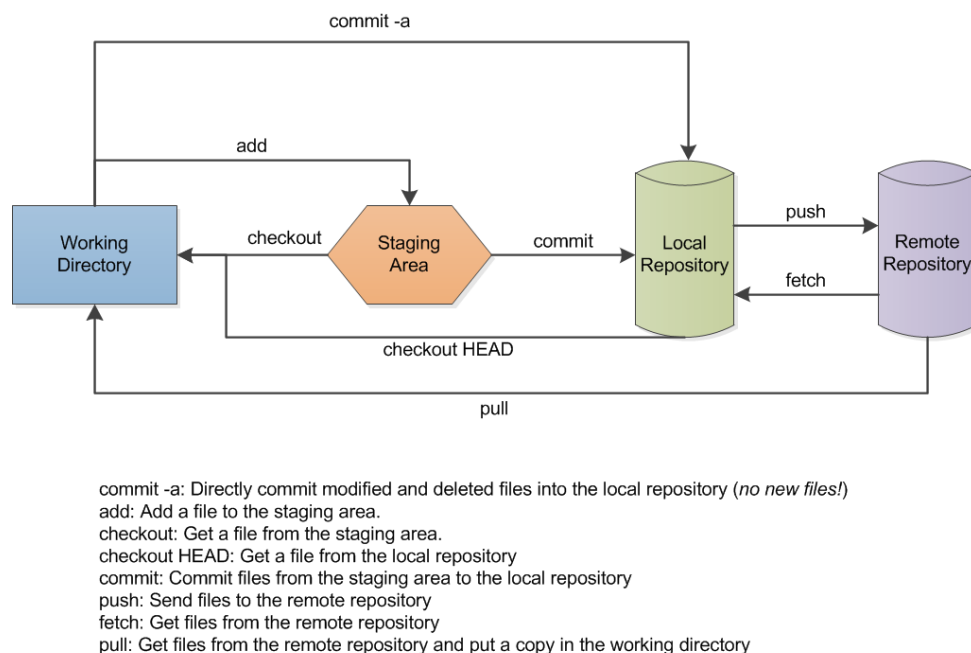


Figura 2.12: *Fluxo de dados Git* (Fonte: [39]).

Ainda na Figura 2.12, é possível ver os principais fluxos de dados Git e suas nomenclaturas. As alterações são adicionadas (*add*) à *staging area*. O *commit* descreve o processo de adicionar arquivos ao repositório local a partir da área de teste, enquanto um *push* envia todas as alterações para o repositório remoto. O *fetch* significa obter todas as alterações do repositório remoto para o repositório local. Um *pull* copia diretamente o repositório remoto para o repositório local. O *checkout* reverte as alterações em seu repositório local e restaura o estado dos arquivos da *staging area* ou do repositório local.[38]

## GitHub

O GitHub<sup>11</sup> é um serviço de hospedagem de projetos de desenvolvimento de software na internet. Ele fornece o controle de versão distribuída do Git, controle de acesso, rastreamento de bugs, solicitações de recursos de software, gerenciamento de tarefas, integração contínua e wikis para cada projeto.[40]

Para o projeto de integração de estudantes da universidade, o GitHub é usado como uma hospedagem para o projeto, já que ele oferece hospedagem gratuita principalmente para projetos *open-source* tornados público. Este é o caso da aplicação deste trabalho. Além de hospedar o código fonte, ele permite expor todo o processo de desenvolvimento

<sup>11</sup><https://github.com>

por meio dos *commits* permitindo novos desenvolvedores entenderem o processo em que foi desenvolvido. Como o projeto proposto neste documento é apenas um modelo inicial que pode ter seu desenvolvimento continuado, o GitHub auxilia o desenvolvimento em equipe por meio de recursos como *forks* e *branches* do Git

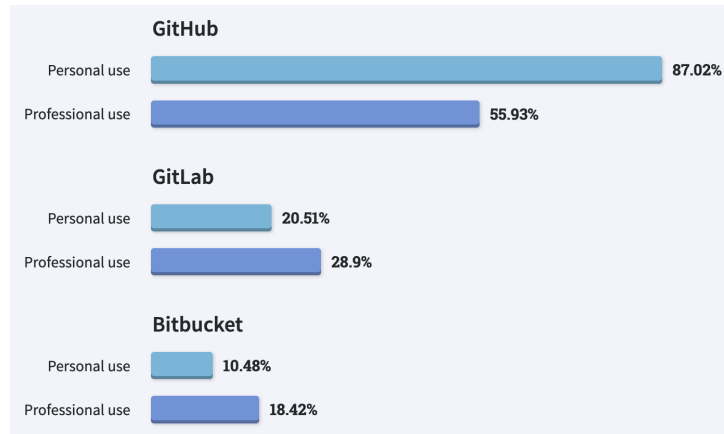


Figura 2.13: *Stack Overflow Developer Survey 2022 - Version control platforms* (Fonte: [21]).

A pesquisa *Stack Overflow Developer Survey 2022* realizou, na seção *Version control platforms*, a seguinte pergunta, que pode ser traduzida como: *Qual serviço de hospedagem de controle de versão você está usando?*[21]

Esta questão obteve 67.035 respostas. A partir dos resultados, conforme mostrado na Figura 2.13, é possível notar uma utilização maior do GitHub em relação às outras alternativas. Em decorrência desta maior utilização e de seus recursos apresentados acima, esta ferramenta foi escolhida para fazer o controle de versões do projeto de integração de estudantes.

## 2.2.5 Visual Studio Code

Visual Studio Code<sup>12</sup>, também conhecido como VS Code, é um ambiente de desenvolvimento integrado, do inglês *Integrated development environment (IDE)* desenvolvido pela Microsoft. A Microsoft lançou a maior parte do código-fonte no GitHub sob a licença permissiva MIT, enquanto os lançamentos da Microsoft são *freeware* proprietário.[41]

A pesquisa *Stack Overflow Developer Survey 2022* realizou, na seção *Integrated development environment*, a seguinte pergunta que pode ser traduzida como:

*Quais ambientes de desenvolvimento você usou regularmente no ano passado e com quais deseja trabalhar no próximo ano? Por favor, verifique tudo o que se aplica.*[21]

<sup>12</sup><https://code.visualstudio.com/>

Esta questão obteve 71.010 respostas. A partir dos resultados, conforme mostrado na Figura 2.14, é possível notar um número de utilização disparada do Visual Studio Code em relação às outras IDEs.

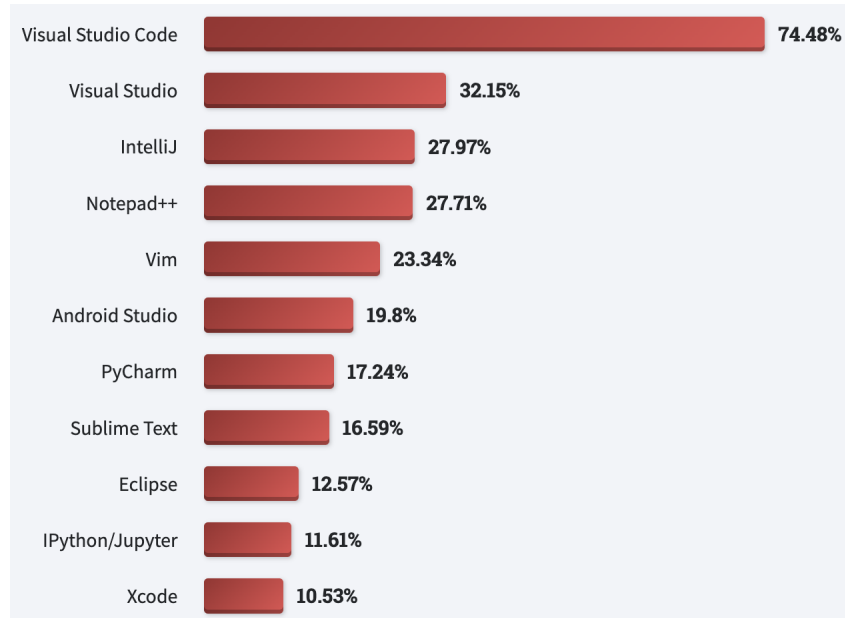


Figura 2.14: *Stack Overflow Developer Survey 2022 - Integrated development environment (IDE)* (Fonte: [21]).

O Visual Studio Code pode ser estendido por meio de extensões, disponíveis por meio de um repositório central. Isso inclui adições ao editor e suporte a idiomas. Um recurso notável é a capacidade de criar extensões que adicionam suporte para novos idiomas, temas, depuradores, depuradores de *time traveling*, realizar análise de código estático e adicionar *linters* de código usando o *Language Server Protocol*.<sup>[42]</sup>

Ao contrário de muitos outros editores de código, o Visual Studio Code possui um depurador embutido, tornando o fluxo de desenvolvimento mais fluido e mantém uma visão única com código e depurador. Isso torna o processo de rastreamento de *bugs* e execução de código muito mais fácil e rápido. Não é necessário ter várias telas para executar os diferentes consoles e reorganizá-los cada vez que precisar minimizar algo. Ele está embutido no design e na configuração do espaço de trabalho desejado.<sup>[43]</sup>

## Wappler

O Wappler<sup>13</sup> é um ambiente de desenvolvimento proprietário multiplataforma (Windows, macOS e Linux), executado em cima do Visual Studio Code e com várias ferramentas

---

<sup>13</sup><https://wappler.io>

visuais para auxiliar no desenvolvimento. O código-fonte gerado visualmente pode ser modificado através da visualização de código, dentro da própria plataforma e em outras IDEs ou editores de código.[44] A aplicação criada pode ser executada em *localhost* ou publicada em qualquer provedor de hospedagem. Ele é um produto da empresa DMX-zone<sup>14</sup>, que desenvolveia extensões para a IDE proprietária Adobe Dreamweaver e passou a desenvolver também para o VS Code, combinando todas as extensões em um só produto comercial e focando os esforços da empresa nele, o Wappler.[45]

Os recursos suportados pelas ferramentas visuais incluem:

- *Application Server*: NodeJS, PHP, ASP.NET e Classic ASP;
- *Frameworks*: Bootstrap, Framework7, jQuery;
- *Bancos de dados*: SQLite, MySQL, MariaDB, MS SQL Server, Postgres e CouchDB;
- *Ambientes de desenvolvimento*: Servidor local, Firebase Hosting, Heroku e Docker;
- *HTTP Server*: NodeJS, Apache e IIS;
- *Deploys*: FTP, SFTP e FTPS;
- *Git*: GitHub, Gitlab e BitBucket.

Para o projeto em questão, o processo de escolha para cada ferramenta foi descrito nas seções anteriores ao longo deste documento. As mesmas ferramentas se encontram destacadas na listagem acima (sublinhadas).

## 2.2.6 Outras tecnologias auxiliares

### Docker

O Docker<sup>15</sup> é uma plataforma de código aberto que permite aos desenvolvedores criar, implantar, executar, atualizar e gerenciar *contêineres*, componentes executáveis padronizados que combinam o código-fonte do aplicativo com as bibliotecas e dependências do sistema operacional necessárias para executar esse código em qualquer ambiente.[46]

De acordo com as decisões de ferramentas utilizadas no projeto, o Docker não é necessário para gerenciar as bibliotecas e dependências. Isto porque foi optado por configurar um ambiente de desenvolvimento leve com NodeJS e SQLite. Desta forma o NodeJS já tem o npm para gerenciar as dependências e o SQLite é simplesmente um arquivo independente. Porém o Docker ainda é necessário para utilização de ferramentas de desenvolvimento disponíveis como imagens de contêineres criadas por terceiros. Muitas delas disponíveis no Docker Hub, o maior repositório mundial de imagens de contêineres.

---

<sup>14</sup><https://www.dmxzone.com/>

<sup>15</sup><https://www.docker.com>

## MailDev

MailDev<sup>16</sup> é um Servidor SMTP local e interface web para visualização e teste de e-mails durante o desenvolvimento. Ele dispõe de uma imagem Docker para utilização.[47] Esta ferramenta é usada para testar o envio de e-mails no projeto durante, por exemplo, o cadastro de usuários. Desta forma, evita o envio desnecessário de mensagens para caixas de e-mail reais. Além disso, permite o teste de e-mail de forma local, sem a necessidade de conexão com um provedor de hospedagem SMTP. Este recurso é também conhecido como "*Fake SMTP*". A Figura 2.15 mostra a interface da ferramenta.

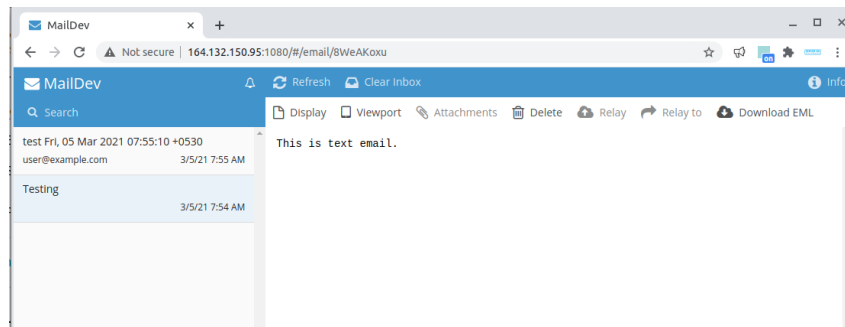


Figura 2.15: Interface de usuário do MailDev.

## JSON Viewer

O JSON Viewer<sup>17</sup>

## Railway

Railway<sup>18</sup> é uma plataforma de implantação na qual é possível provisionar a infraestrutura, desenvolver com essa infraestrutura localmente e, em seguida, implantar na nuvem.

## Postman

---

<sup>16</sup><https://maildev.github.io/maildev/>

<sup>17</sup><https://github.com/tulios/json-viewer>

<sup>18</sup><https://railway.app/>

# Capítulo 3

## Metodologia

### 3.1 Engenharia de requisitos

.

### 3.2 Rapid application development (RAD)

O Desenvolvimento Rápido de Aplicação ou, em inglês Rapid application development (RAD), é um termo geral para abordagens de desenvolvimento de software que colocam menos ênfase no planejamento e mais ênfase em um processo adaptativo. É especialmente adequado para (embora não se limite a) desenvolver software orientado por requisitos de interface do usuário. Os construtores de interface gráfica do usuário geralmente são chamados de ferramentas de desenvolvimento rápido de aplicativos.[48]

Figura 3.1

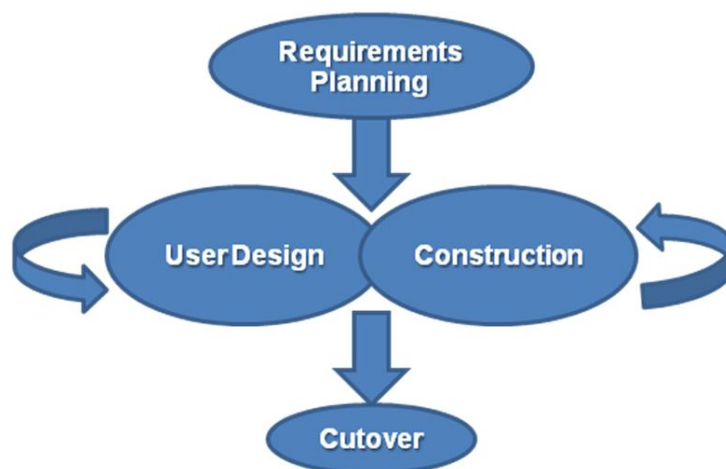


Figura 3.1: Fases na abordagem de James Martin para RAD (Fonte: [48]).

### 3.3 Continuous Integration/Continuous Deployment (CI/CD)

*Continuous Integration/Continuous Deployment (CI/CD)* são as práticas combinadas de integração contínua e entrega contínua. A primeira é a prática de mesclar todas as cópias de trabalho dos desenvolvedores em uma linha principal compartilhada várias vezes ao dia. A segunda é uma abordagem de engenharia de software na qual as equipes produzem software em ciclos curtos, garantindo que o software possa ser lançado de forma confiável a qualquer momento e, seguindo um pipeline através de um "ambiente de produção", sem fazê-lo manualmente.[49]

O GitHub permite desenvolver *workflows* CI/CD utilizando o recurso "Actions".[50] Além disso, o Railway permite configurar *triggers* do GitHub para que o aplicativo possa ser implantado automaticamente quando enviado para uma ramificação selecionada no repositório conectado.[51] Isto permite que automatizar várias partes do fluxo de trabalho de desenvolvimento.

### 3.4 Unified Modeling Language (UML)

.

#### 3.4.1 Diagrama de caso de uso

.

#### 3.4.2 Diagrama de classe

.

#### 3.4.3 Diagramas de atividades

.

#### 3.4.4 Diagramas de estado

.

#### 3.4.5 Diagramas de sequência

.

## Solicitação de cadastro

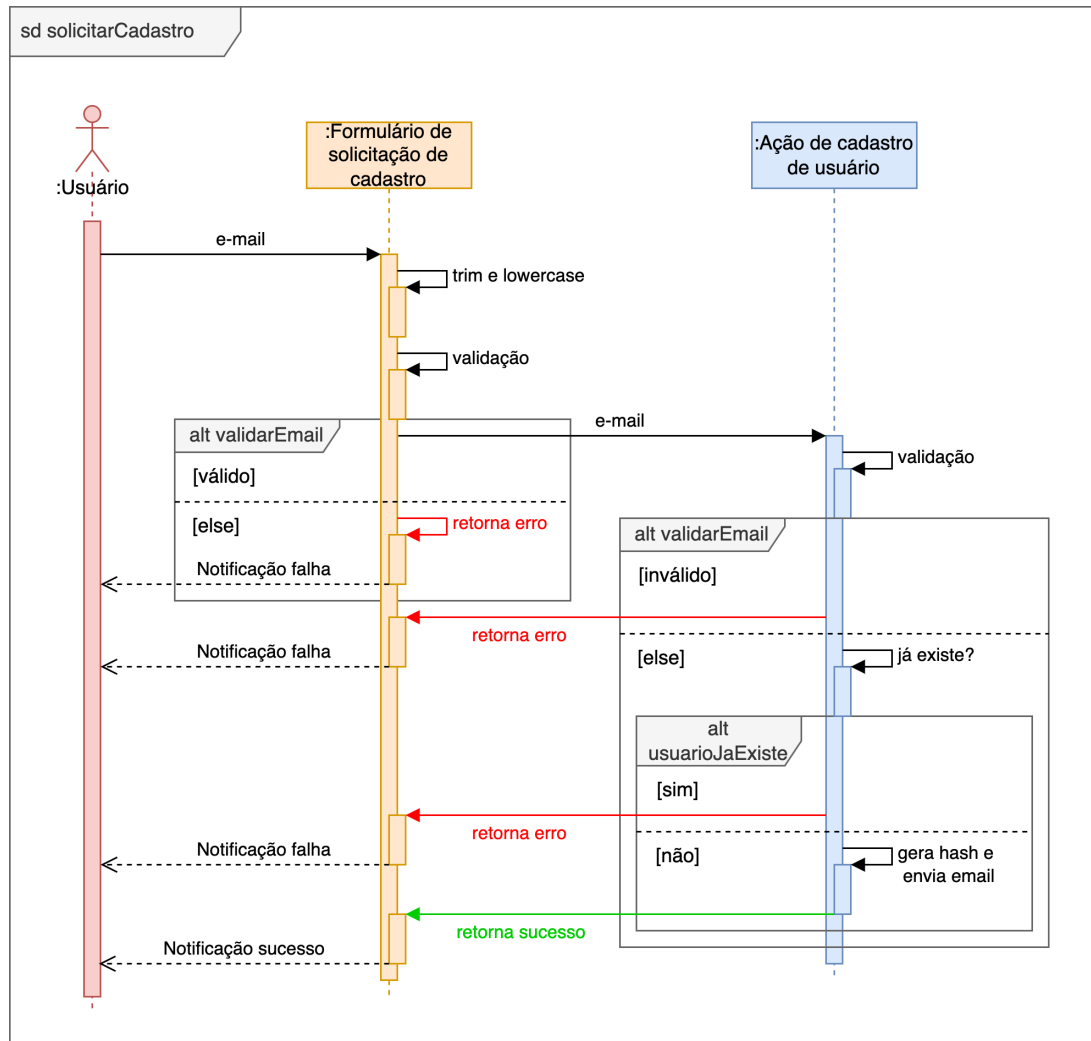


Figura 3.2: Diagrama de sequência da solicitação de cadastro.

## 3.5 Modelo Entidade-Relacionamento (MER)

## 3.6 Ambiente de implantação (Deployment environment)



Desenvolvimento (development)

.

Teste (testing)

.

Homologação (staging)

.ngrok

Produção (production)

.Railway

## 3.7 CRUD

O termo create, read, update, and delete (CRUD) refere-se às quatro operações básicas que um aplicativo de software deve ser capaz de executar: criar, ler, atualizar e excluir. Nesses aplicativos, os usuários devem ser capazes de criar dados, ter acesso aos dados na interface de usuário (UI) por meio da leitura, atualizar ou editar os dados e excluir os dados.[52] A Figura 3.3 mostra uma representação, na interface de usuário (UI) destas operações.

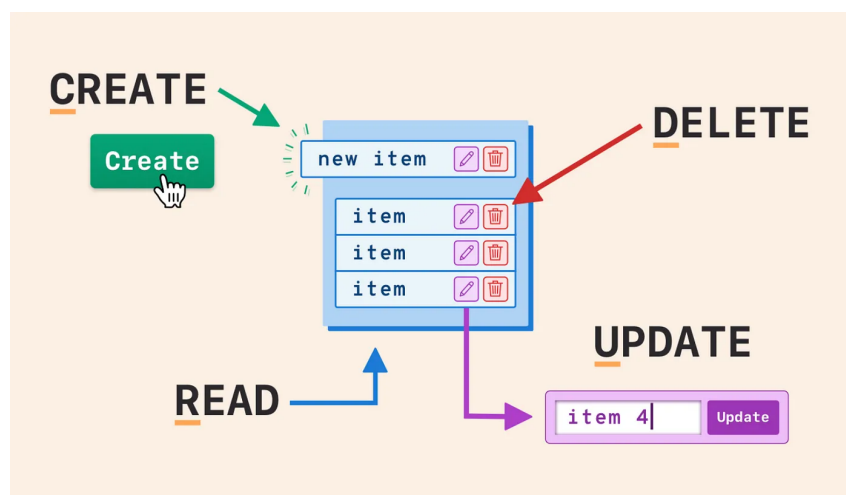


Figura 3.3: CRUD na interface de usuário (UI) (Fonte: [53]).

Em aplicativos completos, os aplicativos CRUD consistem em 3 partes: uma application programming interface (API) ou servidor, um banco de dados e uma interface de

usuário (UI). A API contém o código e os métodos, o banco de dados armazena e ajuda o usuário a recuperar as informações, enquanto a interface do usuário ajuda os usuários a interagir com o aplicativo.[52]

Essas quatro funções principais são usadas para interagir com aplicativos de banco de dados e são um lembrete de quais funções de manipulação de dados são necessárias para que um aplicativo pareça completo. Ao trabalhar com serviços da web, o CRUD corresponde aos métodos HTTP, que comunicam a um servidor da web como você deseja interagir com um site.[53] A Figura 3.4 mostra um mapeamento comum entre as operações de CRUD e HTTP.

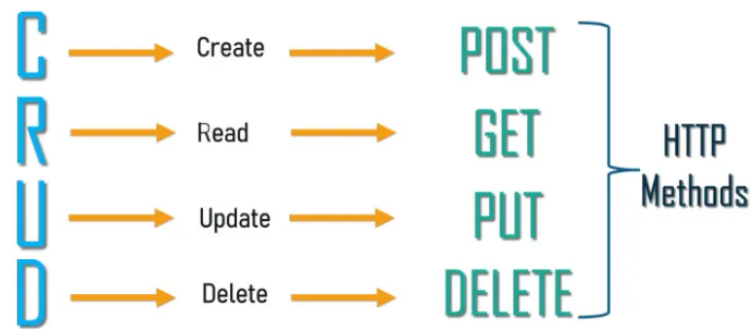


Figura 3.4: Mapeamento comum entre as operações de CRUD e HTTP (Fonte: [53]).

## 3.8 Protocolos

HTTP

SMTP

## 3.9 Algoritmos e funções

Argon2

Regular expression

# Capítulo 4

## Desenvolvimento

### 4.1 Autenticação

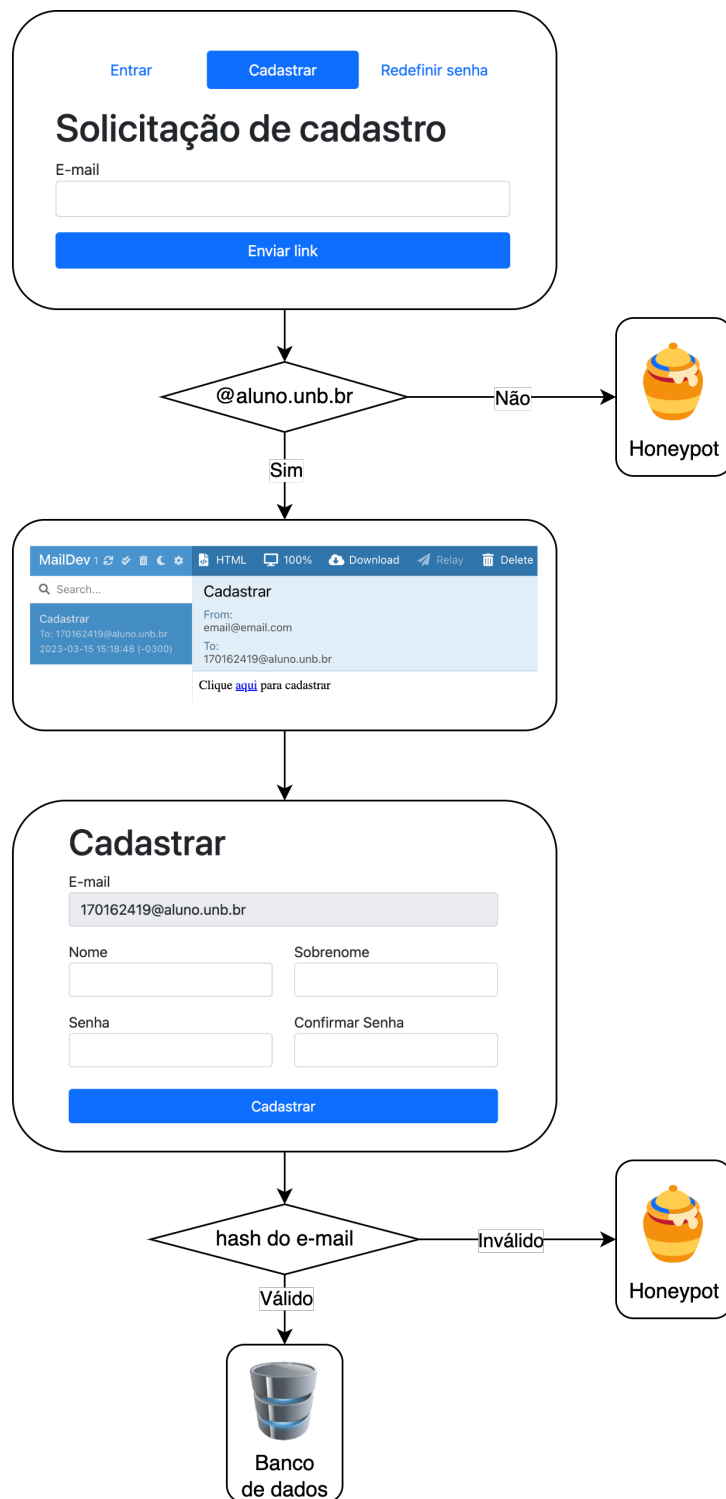


Figura 4.1: Cadastro.

# Capítulo 5

## Resultados

### 5.1

# Capítulo 6

## Conclusões

### 6.1 Considerações finais

### 6.2 Trabalhos futuros

# Referências

- [1] *Faria, frederico fernandes de; vieira, jean victor ribeiro. guia do aluno unb: aplicativo georreferenciado de apoio a atividades acadêmicas. 2017. il., 77 f. monografia (licenciatura em ciência da computação) universidade de brásilia, brásilia, 2017.* 5
- [2] Leal, Angelo Manoel de Matos; Leal, Adson Pereira: *Scholar: desenvolvimento de um aplicativo móvel genérico de apoio acadêmico a estudantes em universidades.* TCC(graduação) - Universidade Federal de Santa Catarina. Centro Tecnológico. Sistemas de Informação., 2019. 6
- [3] *What is a web application?* <https://www.stackpath.com/edge-academy/what-is-a-web-application/>, acesso em 2023-03-13. 8
- [4] *A deeply detailed but never definitive guide to mobile development architecture.* <https://www.freecodecamp.org/news/a-deeply-detailed-but-never-definitive-guide-to-mobile-development-architecture-6> acesso em 2023-03-13. 8
- [5] *The 2017 u.s. mobile app report.* <https://www.comscore.com/Insights/Presentations-and-Whitepapers/2017/The-2017-US-Mobile-App-Report>, acesso em 2023-03-13. 8, 9
- [6] *Os usuários de aplicativos gastam 77% de seu tempo em seus 3 principais aplicativos.* <https://www.statista.com/chart/3835/top-10-app-usage/>, acesso em 2023-03-13. 9
- [7] *Html living standard.* <https://html.spec.whatwg.org>, acesso em 2023-03-13. 10
- [8] *Css snapshot 2023.* <https://www.w3.org/TR/CSS/>, acesso em 2023-03-13. 11
- [9] *Ecma-262 ecma-script® 2022 language specification.* <https://www.ecma-international.org/publications-and-standards/standards/ecma-262/>, acesso em 2023-03-13. 11
- [10] *Usage statistics of javascript as client-side programming language on websites.* <https://w3techs.com/technologies/details/cp-javascript/>, acesso em 2023-03-13. 11
- [11] *Single-page application.* [https://en.wikipedia.org/wiki/Single-page\\_application](https://en.wikipedia.org/wiki/Single-page_application), acesso em 2023-03-14. 11

- [12] *Spa (single page application): O que é, vantagens e desvantagens.* <https://www.iugu.com/iugu4devs/blog/single-page-applications>, acesso em 2023-03-14. 11, 12
- [13] *Single vs multi site.* <https://commons.wikimedia.org/wiki/File:Single-vs-multi-site.png>, acesso em 2023-03-14. 12
- [14] *Bootstrap pagination.* <https://getbootstrap.com/docs/4.0/components/pagination/>, acesso em 2023-03-14. 12
- [15] *Web development tools.* [https://en.wikipedia.org/wiki/Web\\_development\\_tools](https://en.wikipedia.org/wiki/Web_development_tools), acesso em 2023-03-15. 13
- [16] *Browser developer tools.* <https://www.geeksforgeeks.org/browser-developer-tools/>, acesso em 2023-03-15. 13
- [17] *Browser market share worldwide.* <https://gs.statcounter.com/browser-market-share>, acesso em 2023-03-15. 14
- [18] *Web architecture with node.js.* <https://www.youtube.com/watch?v=6dmX2qCAeP0>, acesso em 2023-03-15. 14
- [19] *O que é node.js? [guia para iniciantes].* <https://tecnoblog.net/responde/o-que-e-node-js-guia-para-iniciantes/>, acesso em 2023-03-14. 15, 16
- [20] Mario Casciaro, Luciano Mammino: *Node.js Design Patterns*. Packt Publishing, 2020. 15
- [21] *Stack overflow developer survey 2022.* <https://survey.stackoverflow.co/2022/>, acesso em 2023-03-14. 16, 17, 21, 23, 24
- [22] *Bootstrap (framework front-end).* [https://en.wikipedia.org/wiki/Bootstrap\\_\(front-end\\_framework\)](https://en.wikipedia.org/wiki/Bootstrap_(front-end_framework)), acesso em 2023-03-14. 17
- [23] *Most starred github projects.* <https://github.com/search?q=stars:%3E100000&type=repositories>, acesso em 2023-03-14. 17
- [24] *Usage statistics and market share of bootstrap for websites.* <https://w3techs.com/technologies/details/cs-bootstrap>, acesso em 2023-03-14. 17
- [25] *Celular segue como aparelho mais utilizado para acesso à internet no brasil.* <https://www.gov.br/mcom/pt-br/noticias/2022/setembro/celular-segue-como-aparelho-mais-utilizado-para-acesso-a-internet-no-brasil>, acesso em 2023-03-14. 17
- [26] *Bootstrap grid system.* <https://www.beta-labs.in/2020/11/bootstrap-grid-system.html>, acesso em 2023-03-15. 18
- [27] *Bootstrap grid system.* <https://getbootstrap.com/docs/5.0/layout/grid/>, acesso em 2023-03-15. 18



- [28] *Bootstrap breakpoints.* <https://getbootstrap.com/docs/5.0/layout/breakpoints/>, acesso em 2023-03-15. 18
- [29] *Most widely deployed and used database engine.* <https://sqlite.org/mostdeployed.html>, acesso em 2023-03-14. 19
- [30] *What is sqlite.* <https://www.sqlitetutorial.net/what-is-sqlite/>, acesso em 2023-03-14. 19, 20
- [31] *Como e porque usar o knex.js.* <https://medium.com/como-e-poque-usar-o-knex-js/sobre-o-knex-js-613fdf170727>, acesso em 2023-03-14. 20
- [32] *Schema migration.* [https://en.wikipedia.org/wiki/Schema\\_migration](https://en.wikipedia.org/wiki/Schema_migration), acesso em 2023-03-14. 20
- [33] *Database seeding.* [https://en.wikipedia.org/wiki/Database\\_seeding](https://en.wikipedia.org/wiki/Database_seeding), acesso em 2023-03-14. 20
- [34] *Version control systems.* <https://www.geeksforgeeks.org/version-control-systems/>, acesso em 2023-03-15. 20
- [35] *What is version control?* <https://www.atlassian.com/git/tutorials/what-is-version-control>, acesso em 2023-03-15. 21
- [36] *What is version control?* <https://about.gitlab.com/topics/version-control/>, acesso em 2023-03-15. 21
- [37] *Git.* <https://en.wikipedia.org/wiki/Git>, acesso em 2023-03-14. 21
- [38] *Git and subversion.* [https://en.wikibooks.org/wiki/Game\\_Creation\\_with\\_XNA/Programming/Git\\_and\\_Subversion](https://en.wikibooks.org/wiki/Game_Creation_with_XNA/Programming/Git_and_Subversion), acesso em 2023-03-14. 21, 22
- [39] *Git data flow.* [https://commons.wikimedia.org/wiki/File:Git\\_data\\_flow.png](https://commons.wikimedia.org/wiki/File:Git_data_flow.png), acesso em 2023-03-14. 22
- [40] *Github.* <https://en.wikipedia.org/wiki/GitHub>, acesso em 2023-03-14. 22
- [41] *microsoft/vscode.* <https://github.com/microsoft/vscode>, acesso em 2023-03-14. 23
- [42] *Visual studio code.* [https://en.wikipedia.org/wiki/Visual\\_Studio\\_Code](https://en.wikipedia.org/wiki/Visual_Studio_Code), acesso em 2023-03-14. 24
- [43] *Is visual studio code really the best code editor?* <https://www.tabnine.com/blog/visual-studio-code-really-the-best-code-editor/>, acesso em 2023-03-14. 24
- [44] *Wappler - the alternative.* <https://www.youtube.com/watch?v=cXdzCHFgZvw>, acesso em 2023-03-14. 25
- [45] *About wappler.* <https://wappler.io/about/>, acesso em 2023-03-14. 25

- [46] *What is docker?* <https://www.ibm.com/topics/docker>, acesso em 2023-03-15. 25
- [47] *Dockerhub maildev/maildev*. <https://hub.docker.com/r/maildev/maildev>, acesso em 2023-03-15. 26
- [48] *Rapid application development*. [https://en.wikipedia.org/wiki/Rapid\\_application\\_development](https://en.wikipedia.org/wiki/Rapid_application_development), acesso em 2023-03-23. 27
- [49] *What is ci/cd? continuous integration and continuous delivery explained*. <https://www.infoworld.com/article/3271126/what-is-cicd-continuous-integration-and-continuous-delivery-explained.html>, acesso em 2023-03-23. 28
- [50] *Github actions 101: Develop a ci/cd workflow*. <https://lo-victoria.com/github-actions-101-develop-a-cicd-workflow>, acesso em 2023-03-24. 28
- [51] *Using github actions with railway*. <https://blog.railway.app/p/github-actions>, acesso em 2023-03-24. 28
- [52] *Crud operations – what is crud?* <https://www.freecodecamp.org/news/crud-operations-explained/>, acesso em 2023-03-27. 30, 31
- [53] *Crud operations explained*. <https://medium.com/geekculture/crud-operations-explained-2a44096e9c88>, acesso em 2023-03-27. 30, 31