

INSTITUTO FEDERAL DE EDUCAÇÃO CIÊNCIA E TECNOLOGIA DE MINAS  
GERAIS - *CAMPUS* SÃO JOÃO EVANGELISTA BACHARELADO EM SISTEMAS DE  
INFORMAÇÃO

Enzo Alves Dupin

**PROPOSTA DE UM SISTEMA INTEGRADO DE AGENDAMENTO E GESTÃO DE  
REUNIÕES PARA SERVIDORES DO IFMG**

São João Evangelista - MG

2025

ENZO ALVES DUPIN

**PROPOSTA DE UM SISTEMA INTEGRADO DE AGENDAMENTO E GESTÃO DE  
REUNIÕES PARA SERVIDORES DO IFMG**

Trabalho de conclusão de curso apresentado ao Curso de Bacharelado em Sistemas de Informação do Instituto Federal de Educação Ciência e Tecnologia de Minas Gerais - *Campus* São João Evangelista para a obtenção do título de Bacharel em Sistemas de Informação.

Orientador: Prof. Dr. Wesley Gomes de Almeida

São João Evangelista – MG

2025

---

D934p Dupin, Enzo Alves.

Proposta de um sistema integrado de agendamento e gestão de reuniões para servidores do IFMG/ Enzo Alves Dupin– 2025.

56f.: il.

Orientador: Dr. Wesley Gomes de Almeida.

Trabalho de Conclusão de Curso (bacharelado em Sistemas de Informação) – Instituto Federal Minas Gerais. *Campus* São João Evangelista, 2025.

1. Suporte. 2. Agendamento. 3. Integração. 4. Reunião. I. Dupin, Enzo Alves. II. Instituto Federal de Minas Gerais *Campus* SJE. III. Título.

CDD 005.1

---


Catálogo: Esther Soares Cunha - CRB-6/4333

Enzo Alves Dupin

**PROPOSTA DE UM SISTEMA INTEGRADO DE AGENDAMENTO E GESTÃO DE  
REUNIÕES PARA SERVIDORES DO IFMG**


Trabalho de conclusão de curso apresentado ao Curso de Bacharelado em Sistemas de Informação do Instituto Federal de Educação Ciência e Tecnologia de Minas Gerais - *Campus* São João Evangelista para a obtenção do título de Bacharel em Sistemas de Informação.

Aprovado em: 07 / 02 / 2025 pela banca examinadora:

Documento assinado digitalmente  
 **WESLEY GOMES DE ALMEIDA**  
Data: 26/02/2025 18:31:28-0300  
Verifique em <https://validar.iti.gov.br>


---

Prof. Dr. Wesley Gomes de Almeida (Orientador)

Documento assinado digitalmente  
 **FABIO RODRIGUES MARTINS**  
Data: 26/02/2025 18:49:39-0300  
Verifique em <https://validar.iti.gov.br>

---

Prof. Dr. Fábio Rodrigues Martins

Documento assinado digitalmente  
 **JOAO ANTONIO GOMES DE ALMEIDA JUNIOR**  
Data: 27/02/2025 18:12:13-0300  
Verifique em <https://validar.iti.gov.br>

---

Analista de T.I. João Antonio Gomes de Almeida Junior

Dedico este trabalho aos meus amados pais e amigos, por serem os maiores incentivadores e fontes inesgotáveis de apoio.

## **AGRADECIMENTOS**

Gostaria de expressar meus mais sinceros agradecimentos, primeiramente a Deus e a todos que, de alguma forma, contribuíram para a realização deste trabalho. Agradeço ao meu orientador, cujas orientações e apoio foram fundamentais para o desenvolvimento deste projeto, aos meus familiares, pela paciência e suporte incondicional, que me motivaram a seguir em frente nos momentos mais desafiadores. Por fim, agradeço aos colegas e amigos que, direta ou indiretamente, colaboraram com sugestões valiosas e apoio moral ao longo de toda a jornada. A todos, o meu profundo agradecimento.

"O conhecimento é o único bem que cresce à medida que é compartilhado."

Sócrates

## RESUMO

Este trabalho visou o desenvolvimento de um sistema web para otimizar o agendamento de reuniões entre servidores no Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais - Campus São João Evangelista (IFMG-SJE). O agendamento é feito de forma manual, deixando brechas para o surgimento de desafios como desorganização, atrasos e conflitos de horários. O novo sistema proposto teve como objetivo melhorar a comunicação e a gestão de reuniões, proporcionando um canal que facilitava a marcação de encontros, reduzindo conflitos de agenda e permitindo melhor aproveitamento de tempo. A solução proposta permite o acesso remoto e a qualquer momento, utilizando dispositivos móveis ou *desktop*, trazendo maior eficiência e organização ao ambiente acadêmico. Para a realização deste projeto, foram levantados os requisitos do sistema, desenvolvidos os protótipos, implementados e documentados sua utilização. A fim de obter ao final, um sistema funcional, seguro e simples, alcançando então os objetivos estipulados.

**Palavras-chave:** Sistema. Suporte. Agendamento. Integração. Reunião.

## **ABSTRACT**

This work aimed at the development of a web system to optimize the scheduling of meetings among staff members at the Federal Institute of Education, Science, and Technology of Minas Gerais - São João Evangelista Campus (IFMG-SJE). Scheduling was done manually, leaving room for challenges such as disorganization, delays, and scheduling conflicts. The proposed system aimed to improve communication and meeting management by providing a channel that facilitated scheduling, reduced agenda conflicts, and allowed better time management. The proposed solution enabled remote access at any time using mobile or desktop devices, bringing greater efficiency and organization to the academic environment. For the execution of this project, system requirements were gathered, prototypes were developed, implemented, and their use was documented to achieve a functional, secure, and simple system, thus meeting the stipulated objectives.

**Keywords:** System. Support. Scheduling. Integration. Meeting.

## LISTA DE ILUSTRAÇÕES

Figura 1 – Pesquisa de <i>software</i> usados pelos usuários para prototipagem de UI.....	22
Figura 2 – Passos para desenvolver para <i>desktop</i> e <i>web</i> .....	26
Figura 3 – Arquitetura do MVC .....	30
Figura 4 – Protótipo de Tela de <i>Login</i> .....	38
Figura 5 – Protótipo de Tela de Cadastro de Usuário .....	38
Figura 6 – Protótipo de Tela de Horários.....	39
Figura 7 – Protótipo de Tela de Cadastro de Reunião.....	40
Figura 8 – Protótipo de Tela de Reuniões.....	41
Figura 9 – Protótipo de Tela de Convites .....	41
Figura 10 – Tela de <i>Login</i> .....	42
Figura 11 – Tela de Horários .....	43
Figura 12 – Tela de Cadastro de Horário .....	43
Figura 13 – Tela de Cadastro de Reunião .....	44
Figura 14 – Tela de Reuniões .....	45
Figura 15 – Tela de Encerramento de Reunião .....	45
Figura 16 – Tela de Relatório de Reunião .....	46
Figura 17 – Tela de Convites .....	46
Figura 18 – Tela de Responder Convite.....	47
Figura 19 – Tela de Respostas de Convites.....	47
Figura 20 – Diagrama de Caso de Uso .....	51
Figura 21 – Diagrama de Atividade .....	51

## **LISTA DE TABELAS**

Tabela 1 – Requisitos Funcionais.....	49
Tabela 2 – Requisitos Não Funcionais .....	50

## LISTA DE ABREVIATURAS E SIGLAS

ANSI	<i>American National Standards Institute</i>
API	<i>Application Programming Interface</i>
CERN	<i>Conseil Européen pour la Recherche Nucléaire</i>
CMS	<i>Content Management System</i>
CSS	<i>Cascading Style Sheets</i>
Dr.	Doutor
DBMS	<i>Data Base Management System</i>
DCI	<i>Data-Context-Interaction</i>
DVCS	<i>Distributed Version Control System</i>
Go	Golang
GPS	<i>Global Positioning System</i>
GUI	<i>Graphical User Interface</i>
HD	<i>Hard Disk Drive</i>
HTML	<i>HyperText Markup Language</i>
HTTP	<i>Hypertext Transfer Protocol</i>
Inc.	<i>Incorporation ou Incorporated</i>
IDE	<i>Integrated Development Environment</i>
IFMG-SJE	<i>Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais Campus São João Evangelista</i>
ISO	<i>International Organization for Standardization</i>
JVM	<i>Java Virtual Machine</i>
MG	Minas Gerais
MVC	<i>Model-View-Controller</i>
MVCU	<i>Model-View-Controller-User</i>

NF-e	Nota Fiscal Eletrônica
NFC-e	Nota Fiscal de Consumidor Eletrônica
Prof.	Professor
PHP	<i>Hypertext Preprocessor</i>
RDBMS	<i>Relational Data Base Management System</i>
SEO	<i>Search Engine Optimization</i>
SQL	<i>Structured Query Language</i>
UI	<i>User Interface</i>
URI	<i>Uniform Resource Identifier</i>
UX	<i>User Experience</i>
VBA	<i>Visual Basic for Application</i>
VCS	<i>Version Control System</i>
W3C	<i>World Wide Web Consortium</i>
WWW	<i>World Wide Web</i>

## SUMÁRIO

1	<b>INTRODUÇÃO.....</b>	<b>15</b>
2	<b>FUNDAMENTAÇÃO TEÓRICA .....</b>	<b>18</b>
2.1	<b>Funcionamento das Reuniões.....</b>	<b>18</b>
2.2	<b>Fluxo Atual das Reuniões .....</b>	<b>18</b>
2.3	<b>O Que é um Sistema Web.....</b>	<b>19</b>
2.4	<i>Integrated Development Environment (IDE) .....</i>	<i>20</i>
2.4.1	<i>Visual Studio Code.....</i>	<i>20</i>
2.5	<b>Figma .....</b>	<b>21</b>
2.6	<b>Sistema de Controle de Versão, Git e GitHub.....</b>	<b>22</b>
2.7	<b>Desenvolvimento para Desktop.....</b>	<b>23</b>
2.7.1	<i>Vantagens e desvantagens do desenvolvimento desktop .....</i>	<i>24</i>
2.7.2	<i>Linguagens de programação para desenvolvimento desktop.....</i>	<i>25</i>
2.8	<b>Desenvolvimento para web .....</b>	<b>26</b>
2.8.1	<i>Vantagens e desvantagens do desenvolvimento web .....</i>	<i>26</i>
2.8.2	<i>Tecnologias base no desenvolvimento web.....</i>	<i>27</i>
2.8.3	<i>Linguagens de programação para desenvolvimento web .....</i>	<i>28</i>
2.8.4	<i>PHP para desenvolvimento web.....</i>	<i>28</i>
2.8.5	<i>Frameworks e bibliotecas para desenvolvimento web.....</i>	<i>31</i>
2.9	<b>Banco de Dados .....</b>	<b>32</b>
2.9.1	<i>SQL e MySql .....</i>	<i>33</i>
2.10	<b>Integração de APIs.....</b>	<b>33</b>
2.11	<b>Trabalhos Relacionados.....</b>	<b>34</b>
3	<b>METODOLOGIA.....</b>	<b>36</b>
3.1	<b>Natureza da Pesquisa.....</b>	<b>36</b>
3.2	<b>Ferramentas Empregadas .....</b>	<b>36</b>
3.3	<b>Prototipagem .....</b>	<b>37</b>
4	<b>RESULTADOS E DISCUSSÕES .....</b>	<b>42</b>

<b>4.1</b>	<b>Sistema Desenvolvido.....</b>	<b>42</b>
<b>4.2</b>	<b>Métodos e Procedimentos .....</b>	<b>48</b>
<b>5</b>	<b>CONSIDERAÇÕES FINAIS.....</b>	<b>52</b>
	<b>REFERÊNCIAS.....</b>	<b>53</b>

## 1 INTRODUÇÃO

No contexto acadêmico, a eficácia da instituição de ensino possui alta importância, pois ela é responsável pelo aprendizado dos discentes, para isso é necessário ter servidores bem informados, experientes e organizados. Para o bom funcionamento e decisões acadêmicas da instituição é necessário que haja uma boa comunicação e troca de informações. De acordo com a agência Approach (2023), a comunicação institucional desempenha um papel fundamental para o sucesso das organizações. Alguns dos principais benefícios e importâncias incluem, a construção de uma reputação, uma comunicação mais eficaz para transmitir confiança, transparência e comprometimento, enquanto uma comunicação falha ou inconsistente pode gerar desconfiança e afetar negativamente a percepção.

Na atualidade, o agendamento de reuniões é crucial. Em um ambiente onde o tempo é valioso e as demandas são constantes, planejar e coordenar reuniões eficientemente pode afetar diretamente o desempenho e a satisfação dos servidores. Como afirma Perpétua (2023):

As reuniões são muito importantes para a comunicação e o relacionamento interpessoal dentro das empresas. É nelas que se tem a oportunidade de estudar melhor os assuntos de interesse comum. São atividades nas quais cada um deve trazer seu ponto de vista e sua maneira de buscar a solução. As reuniões devem fazer parte da dinâmica de uma empresa, visando o bom desempenho de toda equipe em prol de seus ideais e sua satisfação (PERPÉTUA, 2023).

Na situação atual do Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais *Campus* São João Evangelista (IFMG-SJE), o agendamento de reuniões é gerido pelo organizador, baseando-se em trocas de *e-mails*, mensagens e ligações, além de utilizar o Google Agenda para a marcação da reunião, assim os servidores enfrentam diversos desafios significativos ao tentar coordenar suas agendas lotadas de modo que seja possível inserir reuniões em horários disponíveis aos participantes, além de encontrar tais espaços no horário. Leva-se em conta também compromissos externos aos horários oficiais na instituição que causam ainda mais dificuldades, além de uma grande demora e tempo perdido com desorganização, cancelamentos, atrasos e mudanças repentinas.

Como tais desafios produzem uma grande complexidade referentes às atividades e compromissos desenvolvidos, uma ótima solução para tais desafios é desenvolver um sistema web para gestão de reuniões para servidores do IFMG-SJE atendendo às demandas específicas do ambiente acadêmico. Esse sistema não apenas simplificará o processo de agendamento, mas também irá melhorar a comunicação entre os servidores, reduzindo conflitos de agenda e

permitindo o melhor aproveitamento de tempo, contribuindo assim para o avanço das atividades de ensino, pesquisa e administração nas instituições de ensino, além do fato de poder ser consultado a qualquer momento e lugar, bastando possuir um dispositivo móvel ou *desktop*.

Os sistemas web têm o potencial de contribuir de várias maneiras para uma organização, inclusive agilizando os procedimentos burocráticos do cotidiano. São uma excelente opção quando o objetivo é divulgar conteúdo ou estabelecer uma presença *online*. Além disso, não requerem instalação, permitindo acesso de qualquer lugar. Por serem desenvolvidos para operar em servidores, armazenam todos os dados e informações pertinentes. Um dos principais benefícios é a acessibilidade, pois é possível acessar o sistema web de qualquer dispositivo móvel sem a necessidade de instalação individual em cada um (KANEHIRA, 2020).

Conforme destacado por Kozinets (2010), a evolução tecnológica tem impulsionado a digitalização crescente do mundo contemporâneo. Nesse contexto, os sistemas digitais surgem como ferramentas essenciais para simplificar e otimizar os desafios do dia a dia. No entanto, é importante observar que, atualmente, o IFMG-SJE ainda enfrenta limitações em seu sistema de agendamento e gestão de reuniões entre servidores, carecendo de uma solução mais integrada e eficiente.

Diante dessa realidade, surgiu a necessidade de modernizar e aprimorar o gerenciamento e controle das reuniões no instituto. Este trabalho tem como foco o desenvolvimento de um sistema eficaz que permita agendar e gerenciar as reuniões de forma mais eficiente. Ao adotar soluções tecnológicas adequadas, é possível promover uma maior organização, colaboração e produtividade no ambiente acadêmico, contribuindo para a melhoria contínua dos processos institucionais.

O objetivo geral deste trabalho foi desenvolver um sistema web, com foco no suporte ao agendamento de reuniões por parte dos servidores do IFMG-SJE. Com isso, buscou-se criar um canal de organização que facilitou o agendamento de reuniões, contribuindo para uma gestão mais eficiente das mesmas.

Os objetivos específicos do projeto foram a identificação e o levantamento detalhado dos requisitos do sistema, a elaboração de protótipos para validação das funcionalidades e o desenvolvimento completo do sistema. A hospedagem do sistema em um ambiente de produção está definida como um projeto futuro.

Atualmente, o IFMG-SJE utiliza o Google Agenda para o agendamento de reuniões, uma ferramenta amplamente conhecida e de fácil acesso. No entanto, apesar de suas funcionalidades, como criação de eventos, envio de convites e notificações automáticas, essa

solução não atende totalmente às necessidades institucionais. O Google Agenda não permite a personalização de regras específicas para o agendamento, como a verificação automática da disponibilidade dos servidores com base nos compromissos institucionais ou a priorização de horários estratégicos definidos pela gestão. Além disso, a dependência de contas individuais pode dificultar o controle centralizado das reuniões.

Em contrapartida, o sistema desenvolvido busca suprir essas limitações ao oferecer uma plataforma adaptada à realidade do IFMG-SJE, permitindo um gerenciamento mais eficiente dos agendamentos. A solução proposta centraliza as informações, reorário e melhora a organização das reuniões, garantindo maior integração entre os servidores. Além disso, o sistema permite a visualização dos horários dos usuários, possibilita anexar relatórios em reuniões cadastradas e incluir documentos para justificar a recusa de um convite de reunião, tornando o processo mais transparente e organizado. Contudo, a implementação desse novo sistema exige um período de adaptação dos usuários e manutenção contínua para assegurar seu funcionamento adequado e a adesão por parte da comunidade acadêmica.

Diante dos desafios apresentados, será vantajoso para o IFMG-SJE a modernização dos processos relacionados à gestão de reuniões, já que a implementação de um sistema web surgiu como uma solução promissora. Essa solução permitirá não apenas otimizar o agendamento e o gerenciamento dessas reuniões, mas também fortalecerá a comunicação e a organização entre os servidores. Essa modernização contribuirá significativamente para a eficiência institucional, promovendo um ambiente mais produtivo e alinhado às necessidades contemporâneas da era digital.

## 2 FUNDAMENTAÇÃO TEÓRICA

Nesta seção será apresentado a fundamentação teórica para o desenvolvimento deste trabalho, abordando desde conceitos até as tecnologias que serão utilizadas.

### 2.1 Funcionamento das Reuniões

As reuniões, tais como as de colegiado, projetos, orientações, eventos, entre outras, possuem como objetivo a comunicação e compartilhamento de informações entre professores, servidores e alunos, a respeito de inúmeras áreas e situações, a fim de aprimorar sua forma de ensino para melhorar o aprendizado dos alunos. Tais reuniões são marcadas pelo organizador, sendo necessário a análise entre todos os horários de aulas dos participantes, para encontrar um tempo acessível para que haja uma participação total dos relacionados.

### 2.2 Fluxo Atual das Reuniões

Os processos de agendamento de reuniões entre os professores do IFMG-SJE envolvem diversas etapas que vão desde a identificação da necessidade de reunião até a realização e documentação dos encontros. A seguir, algumas etapas do fluxo de trabalho no agendamento de reuniões entre os professores:

- a) **planejamento:** Nesta etapa, é feita a identificação da necessidade da reunião e a escolha criteriosa dos participantes que precisam estar presentes. Isso envolve a consideração dos temas a serem discutidos e a seleção dos professores cujas funções ou conhecimentos são relevantes para o assunto em pauta;
- b) **confirmação:** Após a escolha dos participantes, verifica-se a disponibilidade de cada um para a reunião. Em seguida, escolhe-se a data, hora e local adequados, enviam-se os convites pelo Google Agenda e aguardam-se as confirmações de presença. Essa etapa garante que todos os envolvidos estejam cientes e disponíveis para o encontro;
- c) **realização:** Esta é a etapa em que a reunião efetivamente acontece. Durante a reunião, segue-se a pauta previamente estabelecida, discute-se os assuntos em questão e tomam-se as decisões necessárias. Também durante a realização da reunião, é registrada as atas e decisões para futura referência e ações subsequentes.

No entanto a ordem dos acontecimentos podem ser alterados tanto por complicações e imprevistos que possam vir a surgir como por propriamente opinião pessoal,

levando em conta que esta ordem é somente uma ideia, não sendo estritamente necessária a ser seguida ordenadamente.

### 2.3 O Que é um Sistema Web

Um sistema web é uma ferramenta que veio para mudar significativamente a vida das pessoas de diversas maneiras, principalmente na forma como elas interagem com a internet. Os sistemas web vieram para proporcionar mais facilidade e agilidade na realização de diversas tarefas, tais como, comunicação instantânea, gerenciamento de negócios, estoques e demais. Com o surgimento dos sistemas webs, agora os sites não são mais apenas páginas estáticas, mas sim plataformas dinâmicas que oferecem uma variedade de serviços e funcionalidades (SCHAFFER, 2021).

Essa transformação digital tem impulsionado a criação de soluções inovadoras para atender as necessidades das pessoas no mundo moderno. As expectativas de interatividade e acessibilidade cresceram, trazendo consigo a necessidade de desenvolvimento de tecnologias que permitissem uma comunicação mais eficiente e a realização de tarefas complexas de forma mais simplificada e que tivessem mais disponibilidade, podendo ser acessado de qualquer lugar e em qualquer momento (SCHAFFER, 2021).

De acordo com Schaffer (2021):

Um sistema Web é uma aplicação que pode ser acessada de qualquer navegador, celular, tablet ou computador. É um tipo de site dinâmico, onde toda a experiência do usuário é praticamente feita sobre demanda e onde é possível fazer todo um gerenciamento de dados (SCHAFFER, 2021).

A *World Wide Web* (WWW), criada por Tim Berners-Lee em 1989 no *Conseil Européen pour la Recherche Nucléaire* (CERN), na Suíça, surgiu como uma solução para facilitar o compartilhamento de informações entre cientistas geograficamente dispersos, permitindo a conexão não linear de documentos por meio de hipertexto. Em 1994, Berners-Lee fundou o *World Wide Web Consortium* (W3C), com o objetivo de desenvolver padrões que garantissem a interoperabilidade e a acessibilidade da Web, mantendo-a como uma plataforma aberta e neutra. Esses avanços permitiram à Web evoluir continuamente, transformando a comunicação, o trabalho e o acesso à informação de forma global (Gale Encyclopedia of E-commerce, 2014).

Em 1990, Berners-Lee desenvolveu os três componentes essenciais da Web:

*HyperText Markup Language* (HTML), que é a linguagem de marcação usada para criar páginas web; *Uniform Resource Identifier* (URI), que é um sistema de endereçamento para identificar recursos na Web; e *Hypertext Transfer Protocol* (HTTP), que é o protocolo para a transferência de dados na Web. Esses elementos permitiram a criação da primeira página web e estabeleceram a estrutura básica para a comunicação e a distribuição de informações na Internet (Gale Encyclopedia of E-commerce, 2014).

A Web foi lançada publicamente em 1991 e, em 1993, o CERN disponibilizou o *software* da Web gratuitamente, permitindo que qualquer pessoa pudesse utilizá-lo e desenvolver novos aplicativos. Isso foi crucial para a rápida disseminação e adoção da Web em todo o mundo. A liberação do *software* com uma licença aberta incentivou a colaboração e a inovação, estabelecendo as bases para o crescimento exponencial da Web nos anos seguintes (Gale Encyclopedia of E-commerce, 2014).

## **2.4 *Integrated Development Environment (IDE)***

O *Integrated Development Environment (IDE)* é um *software* projetado para facilitar o desenvolvimento de aplicações por meio de uma *Graphical User Interface (GUI)*, reunindo as ferramentas necessárias para os desenvolvedores. Ele geralmente consiste em um editor de código-fonte, que é um editor de texto especializado que auxilia na criação de código de *software* por meio de funcionalidades como o destaque de sintaxe com indicadores visuais, preenchimento automático específico da linguagem e verificação de erros durante o desenvolvimento (Red Hat, 2023).

De acordo com Red Hat (2023), as IDE devem incluir ferramentas para a automação de compilação local, que são utilitários destinados a automatizar tarefas simples e repetitivas durante a criação de uma compilação local do *software*, facilitando o trabalho dos desenvolvedores. Outro componente essencial é o depurador (*debugger*), que é um programa utilizado para testar outros programas, permitindo aos desenvolvedores identificar e mostrar graficamente a localização de *bugs* no código original. Assim, o IDE se apresenta como uma ferramenta integrada e indispensável para o desenvolvimento eficiente e eficaz de *software*.

### **2.4.1 *Visual Studio Code***

O *Visual Studio Code* é um editor de código-fonte, que funciona localmente no computador do usuário e é compatível com os sistemas operacionais *Windows*, *MacOS* e *Linux*.

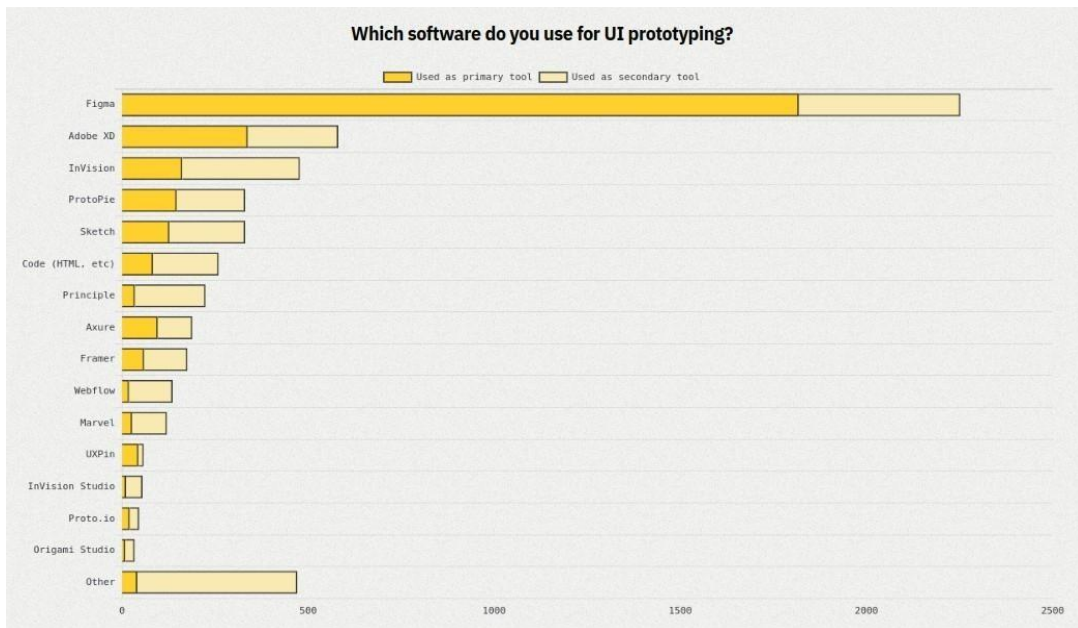
Este editor oferece suporte nativo para linguagens de programação amplamente utilizadas, como JavaScript, TypeScript e Node.js, e possui um rico ecossistema de extensões que permite ampliar suas funcionalidades para outras linguagens e ambientes de execução, incluindo C, C#, Java, *Python*, PHP, Go e .NET (VISUAL STUDIO CODE, 2024).

## 2.5 Figma

O Figma é uma plataforma de colaboração para o *design* de interfaces e prototipagem, desenvolvida pela empresa Figma, Inc., e lançada por Field e Wallace (2016). A ferramenta foi criada com o propósito de oferecer uma solução gratuita que possibilitasse a colaboração entre indivíduos e equipes, facilitando o desenvolvimento de produtos para diversas plataformas. A proposta central do Figma é proporcionar um ambiente de trabalho integrado e acessível, em que o *design* pode ser compartilhado e aprimorado coletivamente em tempo real (VILLAIN; SILVEIRA, 2024).

De acordo com uma pesquisa realizada pela Uxtools (2021), apresentada pela Figura 1, o Figma se destaca como a ferramenta mais utilizada por profissionais que atuam nas áreas de prototipação, *design* de *User Interface* (UI) e *User Experience* (UX), superando concorrentes renomados, como *Sketch* e Adobe XD. Atualmente o figma é uma das ferramentas mais completas quando se trata de criação de interfaces digitais e prototipação. Ele combina a acessibilidade da web com as funcionalidades de um aplicativo nativo. Isso quer dizer que o Figma é disponibilizado via web, permitindo o acesso através de qualquer navegador, sem a necessidade de *download* ou instalação.

Figura 1 – Pesquisa de *software* usados pelos usuários para prototipagem de UI



Fonte: Uxtools (2021)

De acordo com Santana (2023), a principal vantagem do Figma é seu caráter multiplataforma, que permite o acesso e uso da ferramenta em qualquer sistema operacional ou computador por meio de um navegador, possibilitando a colaboração entre equipes de forma eficiente. Projetado desde o início como uma plataforma colaborativa, o Figma oferece recursos integrados de multiusuários, permitindo que *designers* trabalhem simultaneamente em um projeto, com mudanças refletidas em tempo real, além de contar com sistema de comentários, tarefas e histórico de alterações. No entanto, é importante destacar que usuários sem experiência com *design* baseado em vetores podem enfrentar dificuldades iniciais, devido à diferença significativa entre a lógica desse tipo de gráfico e a dos programas baseados em *bitmap*, como o Photoshop.

## 2.6 Sistema de Controle de Versão, Git e GitHub

Um *Version Control System* (VCS) é uma ferramenta essencial no desenvolvimento de *software*, permitindo que indivíduos e equipes monitorem o histórico de alterações em seus projetos. Essas ferramentas oferecem segurança e flexibilidade ao possibilitar a recuperação de versões anteriores a qualquer momento. Além disso, os VCSs proporcionam uma visão unificada e consistente do projeto, destacando o progresso em andamento. A transparência no histórico de alterações, indicando quem fez o quê e como, ajuda os membros da equipe a permanecerem

alinhados, mesmo quando trabalham de forma independente (GitHub, 2021).

Segundo GitHub (2021), os *Distributed Version Control Systems* (DVCS), como o Git, oferecem ainda mais vantagens, permitindo que cada desenvolvedor mantenha uma cópia completa do projeto e de seu histórico localmente. Diferente dos sistemas centralizados, os DVCSs não exigem conexão constante com um repositório central, o que facilita o trabalho em qualquer lugar e a qualquer momento. O Git, o DVCS mais popular, é amplamente utilizado tanto em projetos de código aberto quanto comerciais. Ele possibilita que os desenvolvedores acompanhem toda a linha do tempo de alterações em um único local, fornecendo o contexto necessário para contribuir eficazmente. Além disso, o uso de *branches* permite que mudanças sejam propostas e discutidas sem afetar o código principal, promovendo uma colaboração eficiente.

O GitHub, uma plataforma que hospeda repositórios do Git, expande as possibilidades de colaboração e transparência nos projetos. Através de repositórios públicos, o GitHub permite que as equipes trabalhem juntas para desenvolver o melhor produto possível. A plataforma oferece diversas ferramentas para aprimorar o desenvolvimento, como a linha de comando, onde os usuários podem realizar todas as ações, como criar novas versões, deletar, salvar e outras. Há também os *issues*, que são ferramentas usadas para rastrear tarefas, bugs, melhorias ou qualquer outro tipo de trabalho a ser feito em um projeto. E os *pull requests*, que são solicitações para integrar as alterações feitas em um *branch* (ramo) de um repositório para outro, geralmente do *branch* de desenvolvimento para o principal. Além disso, as revisões de código são o processo de examinar o código escrito por um desenvolvedor antes de integrá-lo ao código principal do projeto. Com uma vasta comunidade de desenvolvedores e um ecossistema rico em integrações, o GitHub tem transformado a forma como o *software* é desenvolvido. A colaboração é centralizada no processo de desenvolvimento, onde os repositórios organizam o trabalho e facilitam a comunicação entre os membros da equipe, do planejamento à implementação das mudanças (GitHub, 2021).

## **2.7 Desenvolvimento para *Desktop***

Um sistema *desktop* é um tipo de *software* desenvolvido para ser instalado e executado diretamente em um computador, oferecendo uma experiência personalizada e de alto desempenho. Ao contrário dos aplicativos web, que são acessados por meio de navegadores, os sistemas *desktop* operam localmente no dispositivo do usuário, o que permite um controle mais detalhado sobre os recursos e funcionalidades específicas, proporcionando

uma integração mais profunda com o sistema operacional do computador (AWARI, 2023).

### 2.7.1 Vantagens e desvantagens do desenvolvimento *desktop*

O desenvolvimento de *software desktop* apresenta tanto vantagens quanto desvantagens que são cruciais para empresas e pessoas que desejam utilizar essa abordagem. Compreender esses aspectos é fundamental para que se possa tomar decisões informadas sobre a adoção de sistemas *desktop*, garantindo que as escolhas feitas atendam às suas necessidades específicas e considerem os possíveis desafios que podem surgir durante o processo de desenvolvimento e implementação.

No que se refere às vantagens do desenvolvimento *desktop* Freenfe (2023) destaca que:

Os sistemas *desktop* são populares entre as empresas pois oferecem uma forma segura de economizar tempo e recursos. No entanto, também existem algumas desvantagens que podem ser desencorajadoras para a implementação de um sistema *desktop*. Por exemplo, os sistemas *desktop* podem ser difíceis de manter e atualizar. Se o computador do usuário falhar, o sistema pode não funcionar corretamente. Além disso, os sistemas *desktop* normalmente não podem ser acessados remotamente e, por isso, se tornam mais difíceis de compartilhar entre vários usuários. (FREENFE, 2023).

Um sistema *desktop* oferece também a vantagem de funcionar independentemente da conexão com a internet, permitindo que você o utilize da mesma maneira, esteja *online* ou *offline*. No entanto, para a transmissão de documentações fiscais, como Nota Fiscal Eletrônica (NF-e) e Nota Fiscal de Consumidor Eletrônica (NFC-e), a internet é necessária devido às regulamentações estaduais. Em termos de segurança de dados, o sistema *desktop* armazena todas as informações localmente no seu computador, mas é altamente recomendável realizar *backups* regulares em mídias externas, como pendrives ou HDs externos, e mantê-los fora do alcance de possíveis ataques de vírus, como o *Ransomware* (COMPUCENTER, 2021).

Os sistemas *desktop* apresentam algumas desvantagens relevantes que podem impactar sua adoção em determinados contextos. Primeiramente, as atualizações de *software* precisam ser baixadas e instaladas manualmente pelo usuário, o que pode gerar atrasos e falhas na manutenção do sistema. Além disso, esses sistemas podem enfrentar problemas de compatibilidade com diferentes sistemas operacionais e configurações de *hardware*, dificultando a padronização em ambientes variados. Por fim, há os custos adicionais associados, uma vez que, em muitos casos, é necessário adquirir ou atualizar licenças de uso, o que pode

representar um ônus financeiro significativo (TECNOLOGIA, 2023).

### 2.7.2 Linguagens de programação para desenvolvimento desktop

Conforme abordado na seção anterior, o desenvolvimento para *desktop* envolve a escolha da linguagem de programação adequada para cada tipo de aplicação e plataforma. A seleção da linguagem pode variar conforme a plataforma alvo e as necessidades específicas do projeto. Nos próximos parágrafos deste trabalho, serão discutidas as linguagens de programação mais utilizadas para o desenvolvimento de aplicações *desktop*, assim como suas vantagens e desvantagens.

Entre as linguagens mais comuns, C# se destaca pela sua forte integração com o sistema operacional *Windows*. Desenvolvida pela Microsoft, C# é amplamente utilizada para criar aplicações *desktop* para *Windows*, com o *Visual Studio* sendo a principal IDE que oferece recursos avançados para o desenvolvimento. A principal vantagem do C# é seu suporte extensivo da Microsoft e a robusta comunidade de desenvolvedores que contribui para o seu ecossistema (SKILLSTECNOLOGICAS, 2024).

Java é amplamente reconhecido por sua capacidade de criar aplicações que funcionam em diferentes sistemas operacionais, como *Windows*, *MacOS* e *Linux*, sem a necessidade de modificações significativas no código-fonte. Esta portabilidade é possível graças à *Java Virtual Machine* (JVM), que permite que o mesmo código Java seja executado em qualquer plataforma que suporte a JVM. A popularidade de Java também é atribuída ao seu extenso conjunto de bibliotecas e *frameworks*, que facilitam o desenvolvimento e a implementação de uma ampla variedade de aplicações. Além disso, a linguagem é apoiada por uma comunidade ativa que oferece suporte contínuo e contribui para o desenvolvimento de novos recursos (SKILLSTECNOLOGICAS, 2024).

O C++ é uma linguagem de programação amplamente renomada por sua alta performance e flexibilidade, tornando-a particularmente adequada para o desenvolvimento de *software* em que a eficiência e o controle detalhado sobre os recursos de *hardware* são cruciais. A linguagem é altamente valorizada em setores que demandam um desempenho excepcional, como no desenvolvimento de jogos e em aplicações gráficas intensivas, na qual a velocidade de processamento e a capacidade de gerenciamento de recursos são fundamentais. Uma das principais vantagens do C++ é sua capacidade de oferecer um controle preciso sobre a alocação e o gerenciamento da memória e outros recursos do sistema. Isso permite que os desenvolvedores otimizem o desempenho de suas aplicações de maneira muito mais detalhada

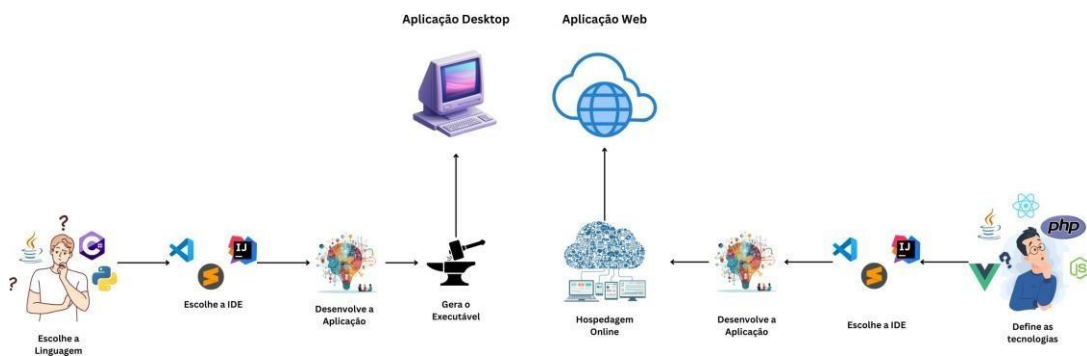
do que em muitas outras linguagens (SKILLSTECNOLOGICAS, 2024).

## 2.8 Desenvolvimento para web

O desenvolvimento web abrange a criação, codificação e programação de sites e seus componentes, incluindo a integração com sistemas, armazenamento em bancos de dados e implementação de medidas de cibersegurança, visando a estruturação detalhada de páginas na internet. Com a crescente complexidade dos sites e aplicações modernas, que demandam integrações sofisticadas com *Application Programming Interfaces* (API) e gerenciamento de grandes volumes de dados, surgiram especializações distintas como *front-end* e *back-end*. Embora ainda existam profissionais *full-stack* que dominam ambas as áreas, em projetos de grande escala, esses profissionais frequentemente assumem papéis de gestão, coordenando a integração entre as diversas áreas envolvidas no desenvolvimento web (EDUCAÇÃO, 2022).

A Figura 2 apresenta os fluxos de desenvolvimento para aplicações web e desktop, destacando as etapas envolvidas em cada abordagem. Nessa representação, é possível visualizar as especificidades de cada tipo de aplicação, incluindo as tecnologias utilizadas, os processos de implementação e as particularidades relacionadas à distribuição e execução em diferentes ambientes.

Figura 2 – Passos para desenvolver para *desktop* e web



Fonte: Elaborado pelo autor, 2024

### 2.8.1 Vantagens e desvantagens do desenvolvimento web

As aplicações web oferecem diversas vantagens, destacando-se pela facilidade de acesso a partir de qualquer dispositivo conectado à internet. Ademais, o *design* responsivo dessas

aplicações aprimora a experiência do usuário, garantindo uma visualização adequada em diferentes tipos de telas. A instalação e a manutenção dessas aplicações também apresentam benefícios significativos, pois, ao serem hospedadas em servidores, as atualizações são automaticamente disponibilizadas para todos os usuários, eliminando a necessidade de intervenções individuais em cada dispositivo (LOGIC, 2018).

O uso de sistemas web proporciona uma significativa redução de investimentos em infraestrutura tecnológica, eliminando a necessidade de adquirir computadores de alto desempenho, *hardware* adicional, ou novos *softwares*, sendo necessário apenas um computador de configuração média com acesso à internet. A segurança dos dados é assegurada por *backups* automáticos e controle de acessos restritos, garantindo a integridade das informações. A integração do sistema permite que colaboradores, clientes, fornecedores e parceiros tenham acesso a informações atualizadas sobre os processos da empresa. Além disso, a flexibilidade dos sistemas web permite que, à medida que a empresa cresce, a capacidade do servidor seja facilmente aumentada, evitando a necessidade de novos investimentos em recursos físicos. Por fim, o suporte técnico é otimizado, com a maioria dos problemas sendo resolvidos remotamente, sem a necessidade de deslocamento de funcionários (GEPER, 2011).

Apesar dos avanços tecnológicos que permitem o envio de notificações em aplicativos web, utilizando plataformas como o *Firebase*, esses aplicativos geralmente não têm acesso aos recursos nativos do dispositivo, como câmera e GPS. Além disso, o desempenho dos web *applications* tende a ser inferior ao dos aplicativos nativos, que são otimizados para cada sistema operacional específico. Outra limitação significativa dos aplicativos web é a necessidade de conexão à internet, tornando-os inacessíveis em ambientes *offline* (BRAVI, 2019).

### **2.8.2 Tecnologias base no desenvolvimento web**

Embora o HTML e o *Cascading Style Sheets* (CSS) frequentemente não sejam reconhecidos como linguagens de programação web, por serem, respectivamente, uma linguagem de marcação e uma de estilos, elas constituem a base fundamental da web. O HTML permite compreender a estrutura da página, identificar os elementos que a compõem, a hierarquia entre eles, e sua importância para o *Search Engine Optimization* (SEO), que consiste em um conjunto de práticas destinadas a melhorar o posicionamento de uma página nos mecanismos de busca. Além disso, o uso correto da semântica e dos atributos facilita a acessibilidade e a compreensão da página por desenvolvedores e outros usuários. O CSS, por

outro lado, possibilita a estilização da página, definindo tamanhos, formas, cores, *backgrounds* e adaptando os elementos para diferentes dispositivos móveis, garantindo uma interface responsiva e coerente (LOCAWEB, 2022).

JavaScript é uma linguagem de programação web orientada a objetos, amplamente utilizada para executar *scripts* no navegador do usuário, tornando a experiência de navegação mais dinâmica e fluida. Comparável à linguagem C em termos de flexibilidade, JavaScript permite a transformação e o processamento de dados enviados e recebidos pelas páginas, além de possibilitar a programação de comportamentos interativos. Por outro lado, o *Hypertext Preprocessor* (PHP) é uma linguagem de *script* que se integra ao HTML, sendo executada no servidor e enviando ao cliente apenas o HTML puro. Isso garante a segurança do código, que não é exibido ao usuário, além de proporcionar um excelente custo-benefício para projetos web, devido ao baixo custo de infraestrutura e à ampla disponibilidade de mão de obra qualificada (SCRIPTCASE, 2022).

### **2.8.3 Linguagens de programação para desenvolvimento web**

Os desenvolvedores web empregam uma variedade de linguagens de programação para criar aplicações e sites, escolhendo a mais adequada com base nas necessidades específicas de cada projeto. Dentre as principais linguagens, *Python* destaca-se por sua versatilidade, sendo amplamente utilizada no desenvolvimento web por meio de *frameworks* como Django e *Flask*, além de aplicações em automação e análise de dados. *Ruby*, por sua vez, é conhecida pelo *framework Ruby on Rails*, que permite o desenvolvimento ágil e eficiente de aplicações web. Já Java, uma linguagem popular em diversas áreas da tecnologia, é frequentemente utilizada no desenvolvimento web com o auxílio do *framework Spring* (OLIVEIRA, 2024).

Além dessas, outras linguagens também desempenham papéis cruciais no desenvolvimento web. *C Sharp* (C#), desenvolvida pela Microsoft, é amplamente utilizada na criação de aplicações web na plataforma .NET, enquanto TypeScript, uma extensão do JavaScript que incorpora tipagem estática, é preferida em projetos de grande porte e complexidade. Por fim, Golang (Go), uma linguagem desenvolvida pelo Google, é reconhecida por sua eficiência e desempenho, sendo utilizada especialmente no desenvolvimento de servidores web (OLIVEIRA, 2024).

### **2.8.4 PHP para desenvolvimento web**

O PHP é uma linguagem de programação *open-source*, amplamente utilizada para

desenvolver aplicações dinâmicas e interativas. Com ele, é possível criar desde páginas web simples até sistemas altamente sofisticados. Operando no lado do servidor, o PHP processa *scripts* e códigos, entregando o resultado final ao usuário. Além disso, o PHP possui funcionalidades avançadas para a integração com bancos de dados, o que torna mais fácil o desenvolvimento de sites que armazenam e recuperam informações. Com PHP, é possível construir um *back-end* poderoso e versátil, capaz de manipular dados, realizar consultas em bancos de dados, autenticar usuários e gerenciar a lógica de negócios de uma aplicação (PHP, 2023).

Segundo Pedroso (2022), o PHP alcançou sua enorme popularidade devido a um conjunto impressionante de características que o tornam uma escolha poderosa e versátil para desenvolvedores. Nessas características incluem sua facilidade de aprendizagem, com uma sintaxe simples e acessível, além de ser uma linguagem de código aberto, o que permite constantes melhorias feitas pela comunidade. O PHP é também altamente eficiente, capaz de processar grandes volumes de dados sem comprometer o desempenho do servidor, e é compatível com os principais bancos de dados, como MySQL e Oracle.

Além disso, Pedroso (2022), cita que o PHP é multiplataforma, podendo ser executado em diversos sistemas operacionais, como *Windows* e *Linux*. Ele é a base do WordPress, o *Content Management System* (CMS) mais popular do mundo, sistemas que permitem criar, gerenciar e editar conteúdos de sites de forma simples e eficiente, sem a necessidade de conhecimentos avançados de programação. Isso amplia sua relevância e atrai uma grande comunidade de desenvolvedores. A linguagem ainda conta com uma vasta gama de *frameworks* e bibliotecas, como *Laravel* e *Symfony*, que tornam o desenvolvimento mais ágil, simplificam a codificação e permitem a criação de soluções complexas com mais eficiência.

O PHP, quando utilizado em conjunto com a arquitetura *Model-View-Controller* (MVC), pode ser uma combinação extremamente eficaz. O MVC é um padrão arquitetural que propõe uma forma organizada de dividir as responsabilidades dentro de uma aplicação. Ele sugere que a aplicação seja separada em três camadas: a camada de apresentação ou interface do usuário (*view*), a camada de manipulação e gerenciamento dos dados (*model*), e a camada de controle das interações entre as duas anteriores (*controller*). Com essa abordagem, o código que trata da interface do usuário é separado das regras de negócio (GUEDES, 2020).

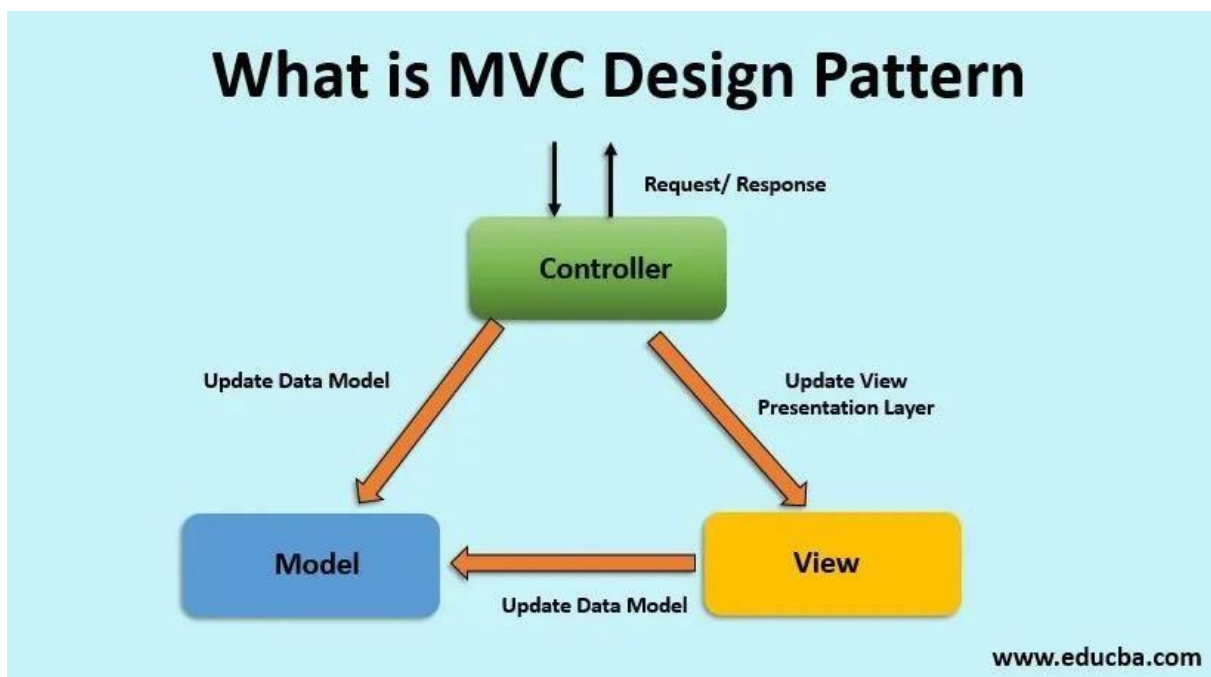
O uso do padrão MVC traz diversas vantagens para o desenvolvimento com PHP. Ele facilita a manutenção do código, permitindo uma separação clara de responsabilidades entre as camadas, o que melhora a compreensão e a reutilização. Além disso, torna o código mais

testável e granular, permitindo identificar e corrigir problemas em camadas específicas sem impactar as demais. Por exemplo, um erro de apresentação pode ser isolado na *view*, enquanto problemas de validação ou envio de dados são tratados no *controller*. Essa organização reduz o impacto de alterações e torna o processo de desenvolvimento mais eficiente e confiável (GUEDES, 2020).

Ao desenvolver com PHP, muitas vezes encontramos o conceito de MVC ao usar *frameworks*. É importante entender a origem e evolução desse padrão. O MVC foi introduzido por Trygve Reenskaug em 1979, inicialmente como *Model-View-Controller-User* (MVCU), com uma camada adicional para o usuário. Com o tempo, essa camada foi removida devido à dificuldade dos usuários em definir requisitos autônomos. Hoje, o MVC se consolidou como um padrão eficaz para separar responsabilidades (DEVMEDIA, 2017).

Além disso, a abordagem *Data-Context-Interaction* (DCI) surgiu para complementar o MVC ao reestruturar suas funções em cada camada: o *Data* atua como o *Model*, o *Context* como o *Controller*, e a *Interaction* como a *View*. Ela introduz ferramentas como *traits*, que promovem o reaproveitamento de código, e injeção de dependências, que facilita o desacoplamento entre módulos do sistema. Dessa forma, o DCI não substitui o MVC, mas aprimora sua capacidade de resolver problemas estruturais, sendo especialmente útil em versões do PHP a partir da 5.4. (DEVMEDIA, 2017).

Figura 3 – Arquitetura do MVC



Fonte: Pedamkar (2023)

A Figura 3 ilustra o padrão de design MVC, destacando as interações entre suas três camadas principais. O *Controller* atua como intermediário, processando as requisições do usuário e direcionando-as para o *Model* ou a *View*. O *Model* é responsável pela lógica de negócios e gerenciamento de dados, fornecendo informações para o *Controller* ou atualizando a *View*. A *View* apresenta os dados ao usuário de forma visual, como tabelas ou gráficos, garantindo que a interface seja clara e acessível. Essa separação de responsabilidades permite uma organização eficiente e facilita a manutenção e a escalabilidade das aplicações (PEDAMKAR, 2023).

### **2.8.5 Frameworks e bibliotecas para desenvolvimento web**

Para novos profissionais ingressando no mercado de tecnologia, alguns conceitos podem parecer abstratos ou até desconhecidos, sendo crucial compreender a função dos *frameworks* no desenvolvimento web. Um *framework* atua como um conjunto abrangente de bibliotecas, ferramentas e regras que orientam o trabalho dos desenvolvedores, facilitando a criação de aplicações ao oferecer padrões pré-definidos. Especificamente no contexto do desenvolvimento web, essa tecnologia permite que os programadores se concentrem na lógica do código, sem a necessidade de criar suportes já existentes, tornando o processo de desenvolvimento mais eficiente e menos oneroso (TRIPLETEN, 2024).

O uso de *frameworks* simplifica e acelera significativamente o trabalho no desenvolvimento web, uma vez que grande parte da programação é pré-configurada, permitindo que o código seja implementado com maior agilidade. Além disso, cada *framework* possui características e vantagens específicas, possibilitando a criação de projetos distintos e inovadores, sem a necessidade de reinventar soluções comuns. Isso garante maior eficiência e variedade na construção de aplicações web, atendendo a diferentes demandas e requisitos de mercado (TRIPLETEN, 2024).

Segundo Hanashiro (2020), as bibliotecas se diferenciam dos *frameworks* pela forma como interagem com o código do desenvolvedor. Uma biblioteca, como Moment.js ou Chart.js, é uma coleção de funções específicas que o desenvolvedor utiliza conforme necessário para resolver problemas, como criar gráficos ou manipular datas, oferecendo flexibilidade e controle sobre como e quando usá-las. Em contraste, um *framework*, como Angular ou Vue.js, fornece uma estrutura completa e um fluxo de trabalho predefinido, determinando a organização e a integração do código, e chamando o código do desenvolvedor conforme o fluxo estabelecido pelo próprio *framework*. Enquanto as bibliotecas permitem maior liberdade, os *frameworks* oferecem uma estrutura mais rígida e integrada.

Lançado em 2013, o React é uma estrutura JavaScript de código aberto amplamente utilizada para a criação de interfaces de usuário do lado do cliente. De acordo com o relatório de pesquisa anual do *Stack Overflow* de 2021, o React é o *framework* web mais utilizado, com uma taxa de adoção de 40,14%. O suporte robusto do *Facebook* contribui significativamente para sua popularidade entre os desenvolvedores, especialmente na construção de aplicativos web. No GitHub, o React conta com 187 mil estrelas e 38,3 mil *forks*, demonstrando sua ampla aceitação na comunidade de desenvolvedores (CLARK, 2022).

O React possui uma forte presença no *StackShare*, com 101,9 mil seguidores, e é utilizado por mais de 10 mil empresas, incluindo gigantes como *Facebook*, *Uber*, *Amazon*, *Instagram*, *Reddit* e *Twitter*. Suas principais características incluem uma abordagem declarativa, que facilita a construção de interfaces de usuário interativas e a depuração de *scripts*, e uma estrutura baseada em componentes, permitindo a criação de elementos reutilizáveis que gerenciam suas próprias condições. Além disso, o React oferece um fluxo de dados unidirecional, simplificando o desenvolvimento e eliminando a necessidade de requisitos adicionais. Seu suporte pela comunidade, com recursos abundantes disponíveis no GitHub e fóruns como *Stack Overflow*, *Reddit* e *Hashnode*, é outro fator que reforça sua posição como uma ferramenta indispensável no desenvolvimento web moderno (CLARK, 2022).

## 2.9 Banco de Dados

Um banco de dados é uma coleção organizada de informações estruturadas, geralmente armazenadas eletronicamente em um sistema de computador, e controladas por um *Data Base Management System* (DBMS). Em conjunto, os dados, o DBMS e os aplicativos associados formam o sistema de banco de dados, frequentemente denominado apenas banco de dados. Os dados nos bancos de dados mais comuns são modelados em linhas e colunas em tabelas, o que facilita o processamento e a consulta eficiente dessas informações. Além disso, esses dados podem ser facilmente acessados, gerenciados, modificados e atualizados, com a maioria dos bancos de dados utilizando a *Structured Query Language* (SQL) para essas operações (ORACLE, 2019).

Existem diversos tipos de bancos de dados, e a escolha do mais adequado para uma organização depende de como os dados serão utilizados. Os bancos de dados relacionais, dominantes desde a década de 1980, organizam informações em tabelas com colunas e linhas, oferecendo uma forma eficiente e flexível de acessar dados estruturados. Já os bancos de dados

orientados a objetos representam informações na forma de objetos, semelhante à programação orientada a objetos. Bancos de dados distribuídos, por sua vez, consistem em arquivos localizados em diferentes sites, permitindo a distribuição de dados entre vários computadores em diversas redes (ORACLE, 2019).

### 2.9.1 SQL e MySql

O SQL é uma linguagem computacional projetada para trabalhar com conjuntos de dados e suas inter-relações, amplamente utilizada por programas de banco de dados relacionais, como o *Microsoft Office Access*. Diferente de muitas outras linguagens de programação, o SQL é relativamente fácil de ler e entender, mesmo para iniciantes. Reconhecida como um padrão internacional por órgãos como ISO e ANSI, o SQL permite descrever conjuntos de dados para responder a diversas consultas. No entanto, é essencial utilizar a sintaxe correta ao escrever comandos em SQL, que, baseada na sintaxe da língua inglesa, compartilha diversos elementos com a sintaxe do *Visual Basic for Applications* (VBA) (MICROSOFT, 2021).

O MySQL, desenvolvido em 1994 pela empresa sueca MySQL AB, passou a ser controlado pela companhia norte-americana *Sun Microsystems* em 2008, após a aquisição da MySQL AB. Em 2010, a Oracle, outra gigante norte-americana, comprou a *Sun Microsystems*, tornando-se proprietária do MySQL desde então. O MySQL é um *Relational Data Base Management System* (RDBMS), baseado em um modelo cliente-servidor, que permite a criação e o gerenciamento de bancos de dados utilizando o modelo relacional. Esse tipo de *software*, geralmente de código aberto, organiza os dados em tabelas, facilitando o acesso e a manipulação eficiente das informações (HOSTINGER, 2024).

### 2.10 Integração de APIs

Uma *Application Programming Interface* (API), é um conjunto de regras e protocolos que permite a comunicação entre diferentes *softwares* para a troca de dados, recursos e funcionalidades. As APIs facilitam e aceleram o desenvolvimento de aplicativos ao possibilitar que os desenvolvedores integrem serviços e dados de outras fontes, em vez de criar tudo do zero. Além disso, as APIs oferecem aos proprietários de aplicativos uma maneira segura de compartilhar funcionalidades e dados, tanto internamente, dentro de uma organização, quanto com parceiros comerciais ou terceiros, mantendo outros detalhes internos do sistema ocultos para garantir a segurança (GOODWIN, 2024).

Ao permitir o compartilhamento seletivo de informações, as APIs ajudam a proteger os sistemas ao expor apenas os dados relevantes para cada solicitação específica. A documentação da API, por sua vez, funciona como um manual técnico que orienta os desenvolvedores sobre como utilizar os serviços oferecidos pela API. Uma documentação bem elaborada não só melhora a experiência dos usuários da API, mas também contribui para o sucesso da API, facilitando sua adoção e integração em diferentes projetos (GOODWIN, 2024).

## 2.11 Trabalhos Relacionados

Existem diversos trabalhos acadêmicos e desenvolvimentos de sistemas que abordam temas relacionados à gestão e organização de eventos e reuniões *online*. Esses estudos buscam responder às demandas emergentes do mercado e à evolução das necessidades tecnológicas, apresentando soluções inovadoras para a otimização de processos e a melhoria da experiência dos usuários. A seguir, são destacados alguns dos principais trabalhos nessa área, cujas contribuições são relevantes para o desenvolvimento do presente estudo.

Gonçalves (2022) desenvolveu uma aplicação com o objetivo principal de gerenciar eventos *online*, visando a necessidade do mercado da época e em demonstrar os conhecimentos adquiridos. O Trabalho em questão se trata de um protótipo de uma aplicação web voltada à gerenciar eventos acadêmicos gratuitos *online*.

Dentre inúmeros trabalhos, destaca-se também o referente à modelagem e protótipo de um sistema para gerenciamento de reuniões desenvolvido por Nascimento (2017), o mesmo descreve o trabalho como uma forma de transformar as reuniões normalmente funcionais de modo a se orientar por um processo, o mesmo possuindo fluxos a serem seguidos, sendo eles Pré-Reunião, Em Reunião e Pós-Reunião. O trabalho anunciado visa contribuir para a otimização das reuniões por meio da automatização e simplificação de processos relacionados à criação, edição e disponibilização das atas. Com isso, busca-se melhorar o gerenciamento das revisões das atas realizadas pelo coordenador, oferecendo ferramentas que facilitem a identificação e o acompanhamento de ajustes necessários. Além disso, o sistema proposto permitirá o controle mais eficiente da frequência dos participantes, garantindo registros precisos e organizados das presenças em cada reunião.

Observando por um outro lado, os trabalhos citados se relacionam por possuírem desenvolvimento web, mas se diferem em seus objetivos e tema principal, logo este trabalho também se relaciona com desenvolvimento web, no entanto se relaciona principalmente com reuniões, o que está bem presente no segundo trabalho citado. Esse trabalho se diferencia do

primeiro citado na questão do tema e diferencia-se do segundo no objetivo, buscando a organização, otimização e facilidade para marcação de reuniões, no entanto o segundo trabalho citado busca a transformação, otimização e padronização da reunião em fluxos de procedimento.

Um dos sistemas mais amplamente utilizados para o gerenciamento de eventos e reuniões online é o Google Agenda. A ferramenta permite que os usuários agendem compromissos, enviem convites, compartilhem calendários e recebam notificações automáticas, facilitando a organização e a comunicação entre os participantes. Além disso, sua integração com outras plataformas do Google, como *Gmail* e *Google Meet*, potencializa a produtividade e simplifica a realização de reuniões remotas (MADRID, 2021). No entanto, apesar de sua versatilidade, o Google Agenda pode apresentar limitações em contextos específicos, como a personalização de fluxos de trabalho para instituições acadêmicas. Dessa forma, a presente pesquisa busca desenvolver uma solução mais adaptada à realidade do IFMG-SJE, oferecendo funcionalidades que atendam às necessidades institucionais de forma eficiente e integrada.

### 3 METODOLOGIA

Esse capítulo explorou a natureza da pesquisa, a metodologia adotada e os procedimentos essenciais para atingir os objetivos definidos neste trabalho. Também destacou as ferramentas e tecnologias que foram empregadas.

#### 3.1 Natureza da Pesquisa

A pesquisa foi classificada como aplicada, pois foi voltada para a resolução de problemas práticos e específicos relacionados à gestão de reuniões dos professores no *campus*. Além disso, adotou uma abordagem qualitativa, uma vez que buscou compreender necessidades e percepções. O objetivo foi aplicar de maneira eficaz os conhecimentos teóricos já existentes para melhorar a organização e a eficiência do agendamento de reuniões.

Nesse contexto, o objetivo foi desenvolver um sistema de gestão para otimizar o agendamento das reuniões entre professores no *campus*, proporcionando uma experiência mais eficiente e prática. Para isso, a pesquisa envolveu a análise e o uso de informações relevantes presentes na literatura, além da aplicação de métodos e técnicas apropriadas para o desenvolvimento do sistema, visando alcançar resultados práticos e aplicáveis ao contexto da gestão de reuniões no *campus*.

Ao mesmo tempo, a pesquisa envolveu um processo de desenvolvimento do sistema de gestão de reuniões. Esse enfoque permitiu um desenvolvimento iterativo e incremental, no qual o sistema foi construído em etapas, com a possibilidade de ajustes e melhorias contínuas ao longo do processo.

#### 3.2 Ferramentas Empregadas

Para o desenvolvimento deste trabalho, foram empregadas diversas ferramentas, com destaque para o *Visual Studio Code*, que foi utilizado para a realização de toda a codificação necessária. Esta IDE oferece uma ampla gama de funcionalidades integradas que auxiliaram e facilitaram o processo de desenvolvimento, otimizando a execução das atividades propostas.

No que tange ao banco de dados, foram utilizados o *MySQL Workbench* e o *MySQL Server*, os quais desempenharam papéis fundamentais no desenvolvimento e monitoramento do banco de dados SQL deste trabalho. Essas ferramentas proporcionaram

suporte crucial na modelagem, administração e otimização do banco de dados, garantindo a eficácia das operações relacionadas.

Para garantir a integridade e a confiabilidade durante o desenvolvimento do projeto, foram adotadas práticas de versionamento e armazenamento de código, sendo elas, a utilização do git e github, respectivamente. Essas práticas permitiram o controle preciso das diferentes versões do *software*, possibilitando o registro detalhado de todas as alterações realizadas ao longo do processo de desenvolvimento.

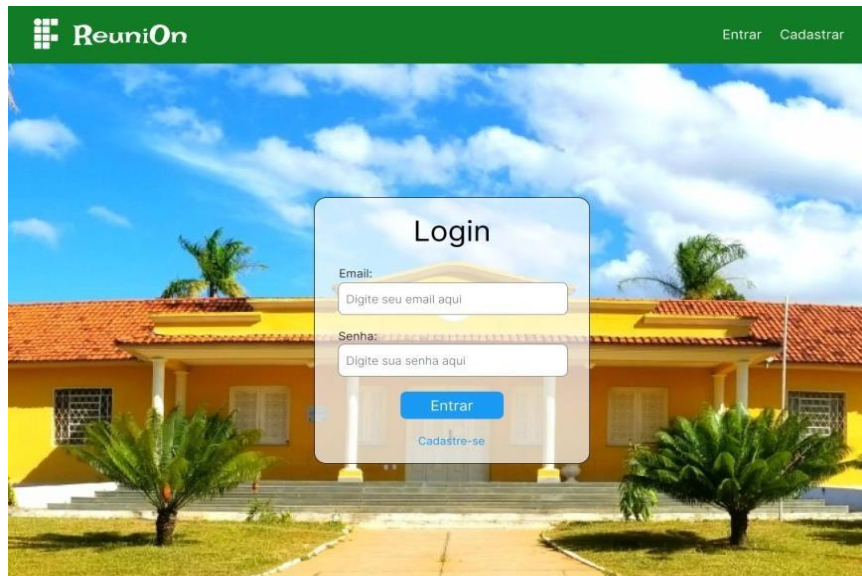
Além disso, o armazenamento organizado do código contribuiu para a estruturação eficiente do projeto, proporcionando um ambiente de trabalho mais seguro e facilitando a gestão do projeto como um todo. Dessa forma, foi possível manter a integridade do código e garantir que todas as alterações sejam rastreáveis, promovendo um desenvolvimento mais coeso e sistemático.

No desenvolvimento deste trabalho, foi utilizado o React, uma popular biblioteca JavaScript para criar interfaces de usuário. O React permitiu a criação de aplicação web dinâmica e responsiva por meio de componentes reutilizáveis, assim facilitou a manutenção do código e tornou o desenvolvimento mais ágil. Com o React, a interface do projeto foi mais intuitiva e interativa, proporcionando uma experiência de usuário aprimorada e um código mais organizado.

### 3.3 Prototipagem

A prototipagem é uma etapa crucial no desenvolvimento de sistemas, permitindo a visualização da ideia antes de sua implementação final. Esse processo é particularmente valioso para *startups*, pois reduz custos ao identificar e corrigir problemas precocemente. A prototipagem simula a experiência do usuário, demonstrando as funcionalidades planejadas e materializando a ideia abstrata em uma forma concreta. Seu principal objetivo é validar os requisitos do usuário, assegurando que o produto final atenda às expectativas e necessidades previstas (PIAZZA, 2021).

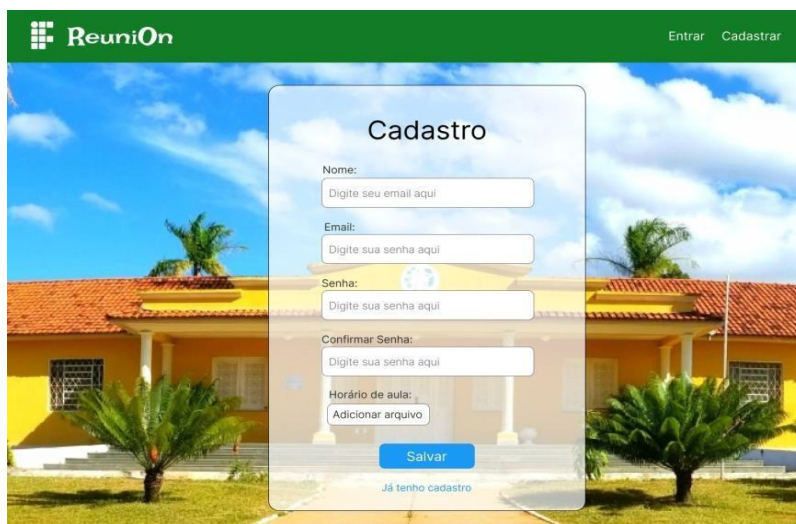
A Figura 4 ilustra a interface de *login* do sistema, onde os usuários são convidados a inserir suas credenciais de acesso, compostas pelo endereço de *e-mail* e senha. A interface segue um *design* simplificado e intuitivo, com campos bem demarcados para facilitar a interação do usuário. A funcionalidade desta tela é essencial para garantir que somente usuários autorizados possam acessar as funcionalidades do sistema, promovendo a segurança dos dados armazenados.

Figura 4 – Protótipo de Tela de *Login*

Fonte: Elaborado pelo autor, 2024

A tela de cadastro, representada pela Figura 5, desempenha um papel crucial na gestão de usuários dentro do sistema. Nesta interface, novos usuários podem ser registrados através da inserção de dados essenciais, como nome, *e-mail*, senha e o arquivo com os horários do professor. A tela foi projetada com foco na simplicidade e usabilidade, apresentando uma estrutura clara e objetiva para a inserção das informações.

Figura 5 – Protótipo de Tela de Cadastro de Usuário

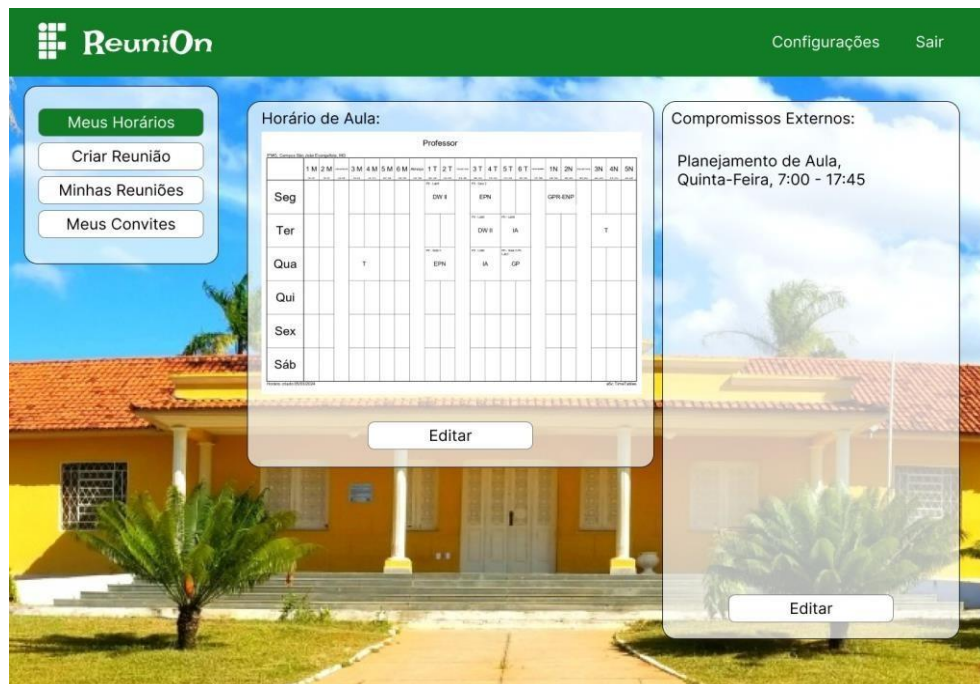


Fonte: Elaborado pelo autor, 2024

A Figura 6 apresenta a visualização dos horários de aula dos professores, onde são detalhadas as disciplinas ministradas, os dias da semana, os horários específicos e os

compromissos externos aos horários de aula. Esta visualização é fundamental para o usuário, pois permite identificar facilmente os seus períodos livres. O *layout* é claro e objetivo, proporcionando uma visão rápida e completa das informações relevantes, o que facilita a tomada de decisões no agendamento de compromissos, a tela ainda possui um menu com as funcionalidades do sistema no canto superior esquerdo e também apresenta opções de configurações e de sair da conta no canto superior direito.

Figura 6 – Protótipo de Tela de Horários



Fonte: Elaborado pelo autor, 2024

A Figura 7 mostra a tela de agendamento de reuniões, que é um dos componentes centrais do sistema. Nela, os usuários podem selecionar os integrantes da reunião, escolher um horário disponível e confirmar a marcação da reunião. Esta tela foi projetada para ser altamente funcional, minimizando erros e garantindo que o processo de agendamento seja direto e sem complicações. A interface também inclui mecanismos para evitar conflitos de horários, assegurando que as reuniões sejam marcadas de acordo com a disponibilidade real dos participantes, as funcionalidades do sistema são disponibilizadas no menu no canto superior esquerdo da tela, observa-se também as opções de configurações e sair da conta no canto superior direito.

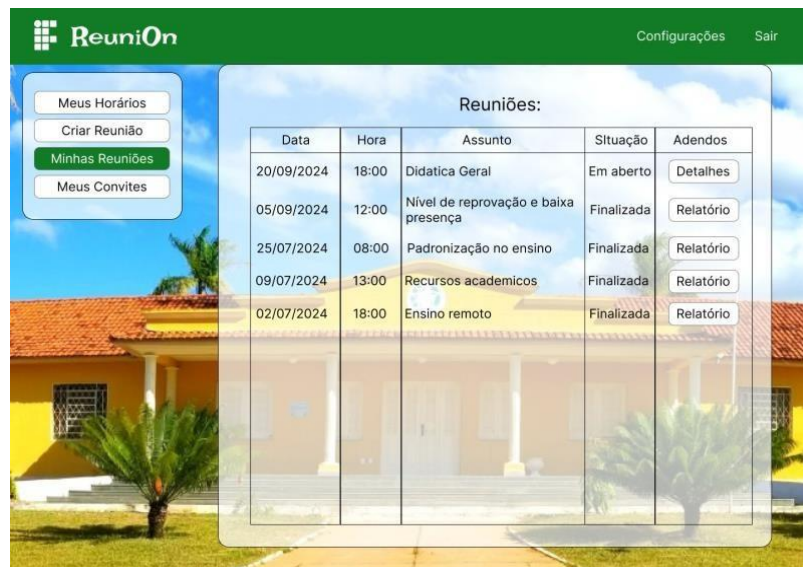
Figura 7 – Protótipo de Tela de Cadastro de Reunião

O protótipo da interface de usuário para o sistema ReuniOn apresenta uma barra superior verde com o logo 'ReuniOn' e links para 'Configurações' e 'Sair'. À esquerda, há um menu vertical com botões para 'Meus Horários', 'Criar Reunião' (destacado em verde), 'Minhas Reuniões' e 'Meus Convites'. O formulário principal, sobreposto a uma imagem de fundo de um prédio amarelo, contém os seguintes campos: 'Integrantes:' com o texto 'José, Antonio, Manuel, Alessandra.' e um botão 'Editar'; 'Data:' com um seletor de data mostrando '20/09/2024'; 'Hora:' com um seletor de hora mostrando '18:00'; 'Local:' com um campo de texto contendo 'Prédio 2 - Sala 3'; 'Assunto:' com um campo de texto contendo 'Didatica Geral'; e um botão 'Finalizar' na base.

Fonte: Elaborado pelo autor, 2024

A Figura 8 apresentada a seguir ilustra a interface de visualização das reuniões agendadas, destacando de maneira detalhada informações cruciais, como a data, o horário e o assunto de cada encontro. Além disso, a visualização permite a inclusão de detalhes adicionais, como a apresentação de relatórios ou um detalhamento mais aprofundado. Essa tela e funcionalidades são essenciais para o usuário, pois proporciona uma visão clara e organizada das reuniões, facilitando a gestão e o acompanhamento das mesmas, no canto superior da tela, também é apresentado um menu contendo as funcionalidades dos sistema, opções de sair da conta e de configurações é observados no canto superior direito da tela.

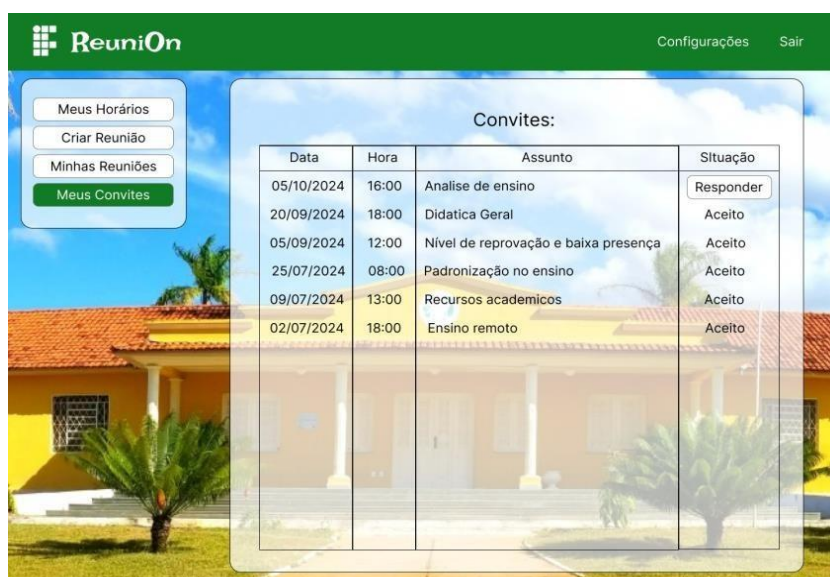
Figura 8 – Protótipo de Tela de Reuniões



Fonte: Elaborado pelo autor, 2024

A imagem apresentada pela Figura 9 retrata a interface destinada à visualização dos convites recebidos para reuniões, permitindo ao usuário avaliar e confirmar, ou não, sua participação nas mesmas. Essa tela oferece um resumo conciso e estruturado das reuniões para as quais a presença do usuário foi solicitada, incluindo informações relevantes, como data, horário, assunto e o status do convite, na tela também apresenta um menu com as funcionalidades do sistema no canto superior esquerdo, além de apresentar opções de configurações e sair da conta no canto superior direito.

Figura 9 – Protótipo de Tela de Convites



Fonte: Elaborado pelo autor, 2024

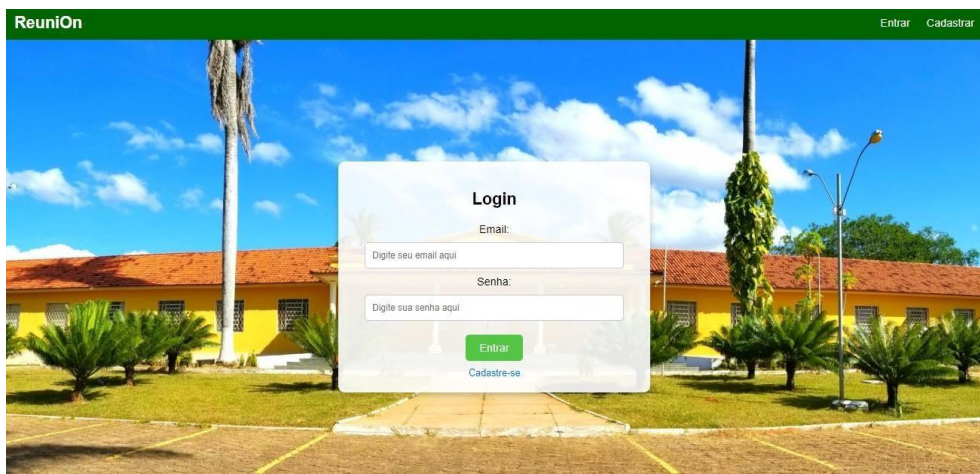
## 4 RESULTADOS E DISCUSSÕES

Esta seção apresenta os resultados obtidos ao longo do desenvolvimento deste trabalho, incluindo a descrição dos métodos empregados, as informações coletadas durante o processo e registros visuais do sistema em funcionamento. Além disso, são exibidas as funcionalidades implementadas utilizando as tecnologias HTML, CSS, JavaScript e PHP.

### 4.1 Sistema Desenvolvido

Ao iniciar o sistema "ReuniOn", o usuário é direcionado para a tela de *login*, onde deve inserir seu *e-mail* e senha para acessar a plataforma. Além disso, é disponibilizada a opção de cadastro para novos usuários, como é apresentado na Figura 10.

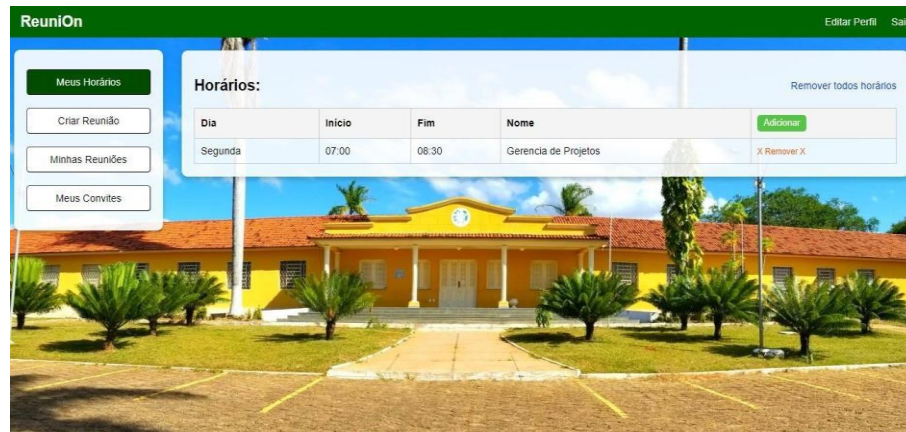
Figura 10 – Tela de *Login*



Fonte: Elaborado pelo autor, 2024

Após o *login*, a tela inicial do sistema "ReuniOn" representada pela Figura 11, apresenta um menu lateral com opções para gerenciar horários, criar reuniões, visualizar reuniões agendadas e acessar convites recebidos e enviados. No centro da interface, há uma seção dedicada à gestão de horários, sendo eles interpretados como compromissos semanais, onde são exibidos os períodos cadastrados pelo usuário, incluindo o dia da semana, horário de início e fim, além do nome da atividade ou reunião correspondente. O sistema disponibiliza funcionalidades para adicionar novos horários, remover horários individualmente ou excluir todos os registros simultaneamente. Além disso, no canto superior direito, estão disponíveis as opções de edição de perfil e *logout*.

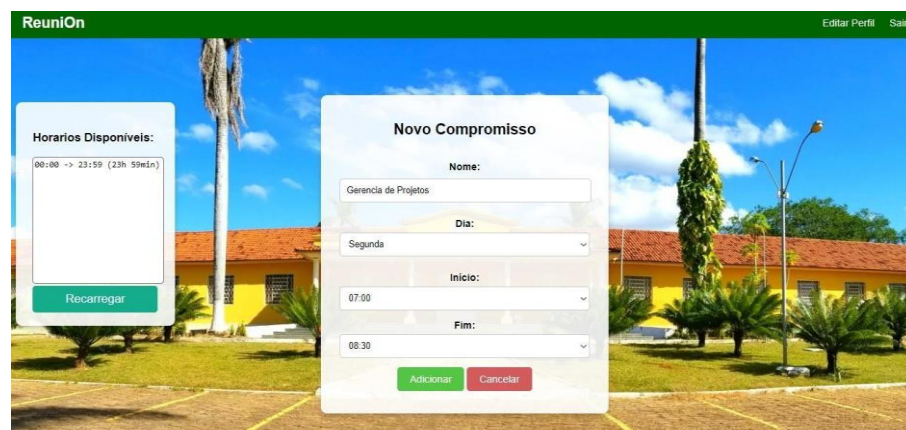
Figura 11 – Tela de Horários



Fonte: Elaborado pelo autor, 2024

A Figura 12 corresponde à tela de cadastro de um novo compromisso no sistema "ReuniOn", que inicialmente na prototipagem, era adicionado os horários individuais pela inserção de um arquivo por parte do usuário, mas devido ao prazo estipulado para conclusão, a produção desta funcionalidade foi interrompida e retirada da versão final. Ao utiliza-la, o usuário pode inserir um nome para o compromisso, selecionar o dia da semana em que ocorrerá e definir os horários de início e fim que se ajustam ao horários disponíveis do usuário de acordo com os compromissos existentes e reuniões futuras. À esquerda, há uma seção que exibe os horários disponíveis, permitindo ao usuário verificar a disponibilidade antes de cadastrar um novo compromisso. Além disso, há um botão de "Recarregar" para atualizar a disponibilidade dos horários baseando-se no dia da semana escolhido pelo usuário. Após preencher os campos necessários, o usuário pode confirmar a ação clicando no botão "Adicionar" ou cancelar o processo por meio do botão "Cancelar".

Figura 12 – Tela de Cadastro de Horário



Fonte: Elaborado pelo autor, 2024

A tela exibida na Figura 13 corresponde à funcionalidade de criação de uma nova reunião no sistema "ReuniOn". O usuário pode definir os participantes da reunião, selecionar uma data, especificar o horário de início e término que se adaptam aos horários disponíveis dos participantes, indicar o local e informar o assunto da reunião. Além disso, há um campo para adicionar um texto de convite, permitindo a inclusão de uma mensagem personalizada para os membros da reunião. Do lado inferior esquerdo, é exibida uma lista de horários disponíveis, auxiliando na escolha de um período adequado para a reunião. O sistema também oferece a opção de definir a prioridade da reunião. Após o preenchimento dos dados, o usuário pode confirmar a criação da reunião por meio do botão "Criar Reunião" ou atualizar a disponibilidade de horários utilizando o botão "Recarregar".

Figura 13 – Tela de Cadastro de Reunião

ReuniOn Editar Perfil Sair

Meus Horários

**Criar Reunião**

Minhas Reuniões

Meus Convites

**Nova Reunião:**

Integrantes: Enzo, Rafael, José, Rafaela

Obs: Clique no nome para remover.

Nome: Rafaela  Prioridade: Alta  **Adicionar**

Data: 03/02/2025

Início: 09:30

Fim: 11:30

Local: Teatro Zé Passarinho

Assunto: Reformulação do ensino

Texto do convite: Venho convocar os professores para a reunião.

Recarregar  Criar Reunião

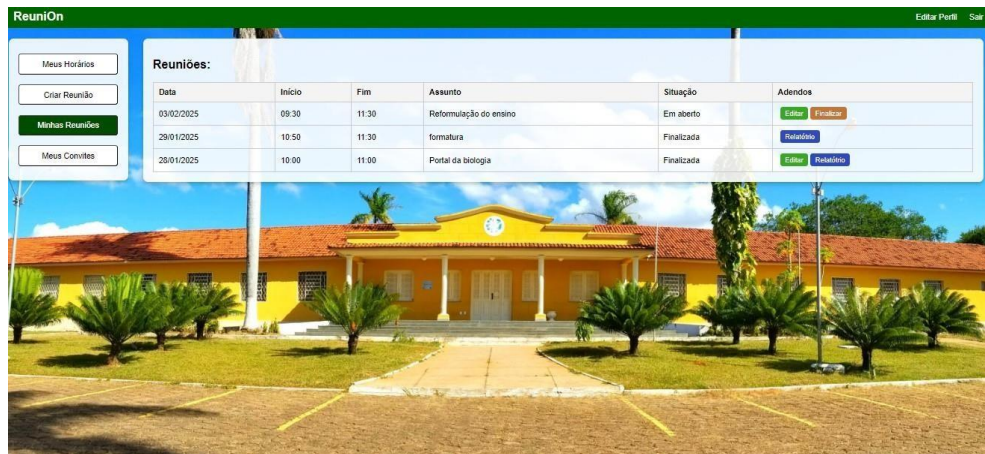
**Horários Disponíveis:**

- 08:00 -> 07:00 (7h 00min)
- 08:30 -> 13:30 (5h 00min)
- 16:30 -> 23:59 (7h 29min)

Fonte: Elaborado pelo autor, 2024

A tela apresentada pela Figura 14 exibe uma listagem de reuniões que o usuário participa, sendo integrante ou o criador. No lado esquerdo, é apresentado o menu de navegação. No painel principal, é exibida uma tabela com informações das reuniões, incluindo data, horário de início e término, assunto, situação e opções de gerenciamento. As reuniões podem estar em aberto, finalizadas ou com solicitação de reagendamento, e há botões para editar e finalizar caso o usuário seja o criador da reunião, além dos botões de detalhes e relatórios, dependendo do *status* da reunião.

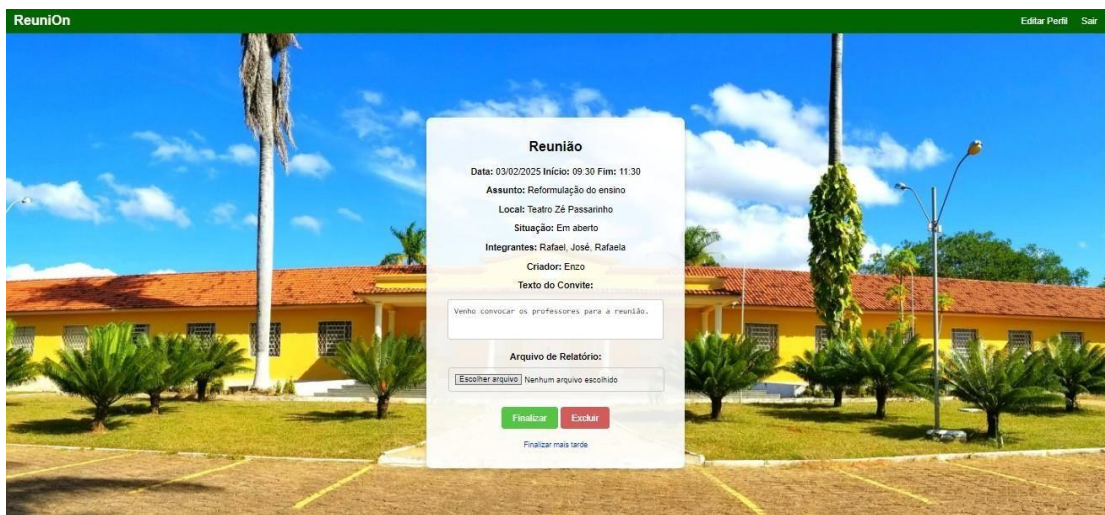
Figura 14 – Tela de Reuniões



Fonte: Elaborado pelo autor, 2024

Acessada apenas pelo criador da reunião, a tela de finalização, representada na Figura 15, apresentando informações como data, horário de início e término, assunto, local, status da reunião, integrantes e o nome do criador. Há um campo que exibe o texto do convite enviado aos participantes e uma opção para anexar um arquivo de relatório, permitindo o registro de informações relevantes sobre a reunião. Além disso, a interface disponibiliza botões para finalizar a reunião, excluí-la ou optar por finalizá-la posteriormente, proporcionando controle e flexibilidade na gestão dos compromissos.

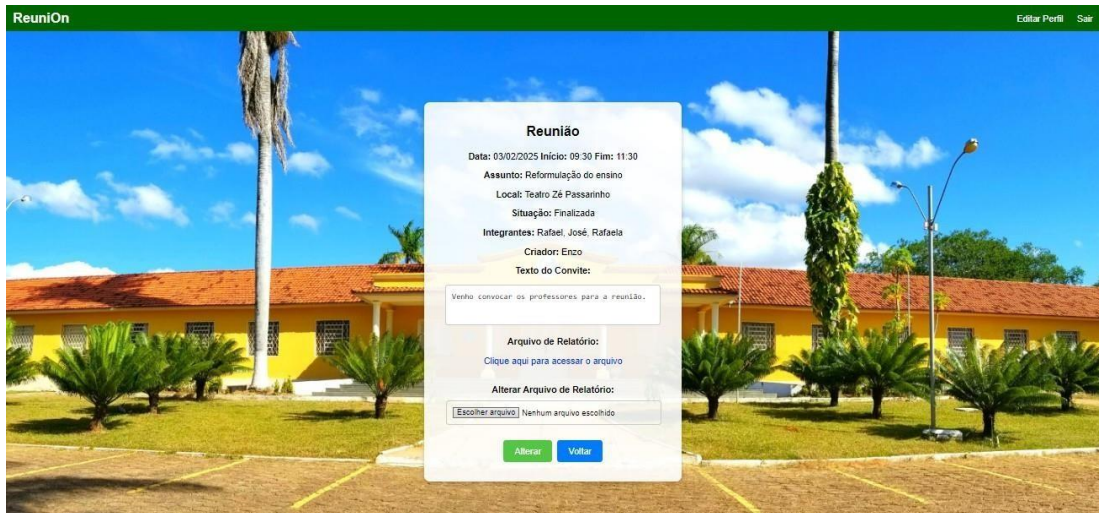
Figura 15 – Tela de Encerramento de Reunião



Fonte: Elaborado pelo autor, 2024

A Figura 16 apresenta a tela de relatório que exibe os detalhes da reunião finalizada. A interface permite o acesso ao arquivo de relatório da reunião a todos os integrantes, mas é exibido uma opção de substituí-lo por um novo arquivo apenas para o criador.

Figura 16 – Tela de Relatório de Reunião



Fonte: Elaborado pelo autor, 2024

A Figura 17 expõe a tela correspondente à interface de gerenciamento de convites para reuniões. No lado esquerdo, apresenta-se o menu de navegação. No centro da tela, uma tabela exhibe as informações dos convites recebidos e a opção de respondê-lo caso ou a opção de verificar as respostas dos convites de reuniões em que o usuário é o criador.

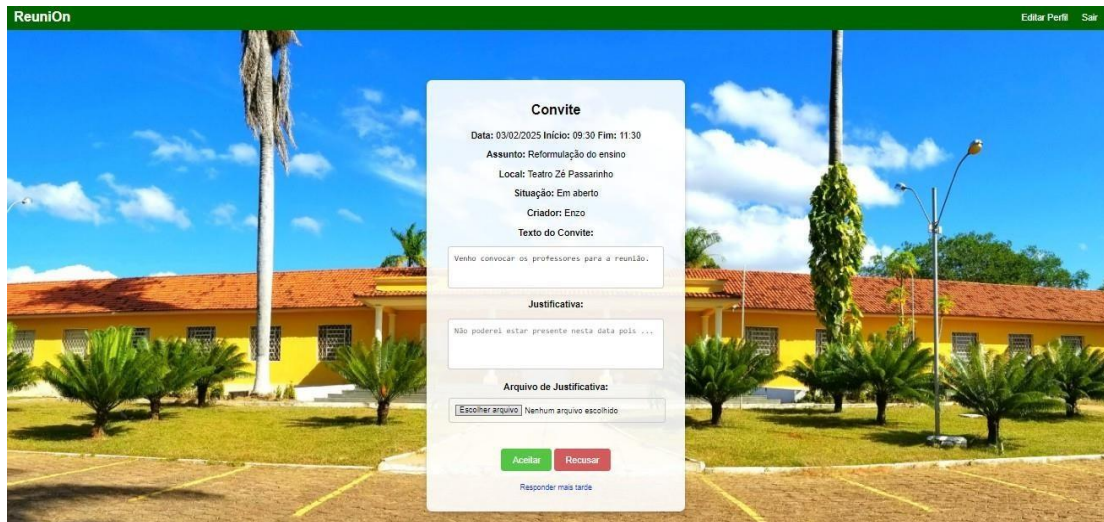
Figura 17 – Tela de Convites



Fonte: Elaborado pelo autor, 2024

A tela presente na Figura 18 exhibe os detalhes do convite para uma reunião, incluindo data, horário de início e término, assunto, local, situação da reunião e o nome do criador do evento. O usuário pode visualizar o texto do convite e, caso não possa comparecer, tem a opção de inserir uma justificativa no campo correspondente e anexar um arquivo, se necessário. Na parte inferior, há três opções de resposta: aceitar o convite, recusar (exigindo uma justificativa) ou optar por responder mais tarde.

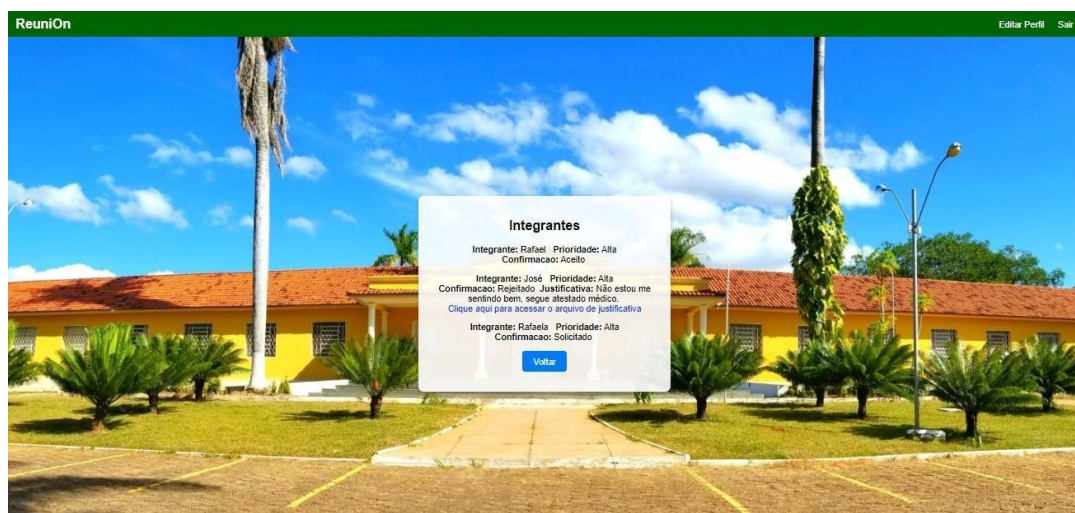
Figura 18 – Tela de Responder Convite



Fonte: Elaborado pelo autor, 2024

A Figura 19 apresenta a tela com as respostas dos integrantes da reunião, indicando a prioridade de cada participante e sua respectiva confirmação de presença. Os status apresentados incluem "Aceito", "Rejeitado" com justificativa podendo ser escrita e sendo documento anexado, no caso exibindo um link para acesso e "Solicitado" para aqueles que ainda não confirmaram.

Figura 19 – Tela de Respostas de Convites



Fonte: Elaborado pelo autor, 2024

Após a apresentação das telas e suas respectivas funcionalidades, é possível observar que o sistema desenvolvido atende às necessidades propostas, proporcionando uma interface intuitiva e de fácil navegação. As telas descritas neste capítulo foram cuidadosamente projetadas

para garantir uma experiência otimizada ao usuário, alinhada aos objetivos do projeto. O desenvolvimento dessas interfaces visa facilitar a interação e melhorar a gestão de dos agendamentos das reuniões.

## **4.2 Métodos e Procedimentos**

Os procedimentos metodológicos que foram adotados durante o desenvolvimento da aplicação seguiram uma abordagem estruturada e orientada em etapas. Dentre os principais procedimentos a serem utilizados, destacou-se a fase de definição dos requisitos, a criação de diagramas, dentre outros. A seguir, foi abordado um pouco mais detalhado cada um desses procedimentos definidos pela percepção do pesquisador.

Os requisitos foram classificados em duas categorias principais: funcionais e não funcionais. Os requisitos funcionais dizem respeito às funcionalidades e serviços que o software deve oferecer para atender às demandas e solucionar os problemas previamente identificados. Estes requisitos garantem que o sistema cumpra sua finalidade principal, atendendo às expectativas dos usuários e proporcionando as soluções necessárias (CUNHA, 2020). A seguir, está situado a Tabela 1 contendo os requisitos funcionais do trabalho.

Tabela 1 – Requisitos Funcionais

<b>Identificador</b>	<b>Requisito</b>	<b>Classificação</b>
RF01	<b>Cadastro de usuários</b> , onde os servidores podem registrar suas informações como nome, <i>e-mail</i> , disciplinas ministradas e horários de aula.	Essencial
RF02	<b>Autenticação e autorização</b> , em que os usuários utilizam permissões de acesso com a utilização de <i>login</i> com <i>e-mail</i> e senha.	Essencial
RF03	<b>Gerenciamento de horários</b> , permite o gerenciamento dos horários de aulas e disponibilidade para reuniões por parte dos usuários.	Essencial
RF04	<b>Agendamento de reuniões</b> , é gerado a partir de uma interface gráfica, se baseando nas disponibilidades dos integrantes.	Essencial
RF05	<b>Visualização de reuniões agendadas</b> , serve para manter os integrantes informados na questão de data, horário e local das reuniões.	Essencial
RF06	<b>Notificações</b> , de grande importância, elas notificam os integrantes das reuniões, solicitando a confirmação da presença.	Importante
RF07	<b>Cancelamento e reagendamento de reuniões</b> , em que de acordo com a disponibilidade dos participantes, eles podem cancelar a reunião negando sua presença no caso de tal participante ter alta relevância na reunião, assim solicitando reagendamento ao organizador.	Importante
RF08	<b>Relatórios</b> , onde estabelece dados e informações das reuniões efetuadas.	Desejável

Fonte: Elaborado pelo autor, 2024

Por outro lado, os requisitos não funcionais tratam das características operacionais do software, ou seja, como ele implementa as funcionalidades planejadas. Esses requisitos abrangem aspectos como desempenho, usabilidade, segurança, confiabilidade e escalabilidade, assegurando que o sistema seja eficiente, seguro e capaz de se adaptar às necessidades futuras (CUNHA, 2020). A seguir, está situado a Tabela 2 contendo os requisitos não funcionais do trabalho.

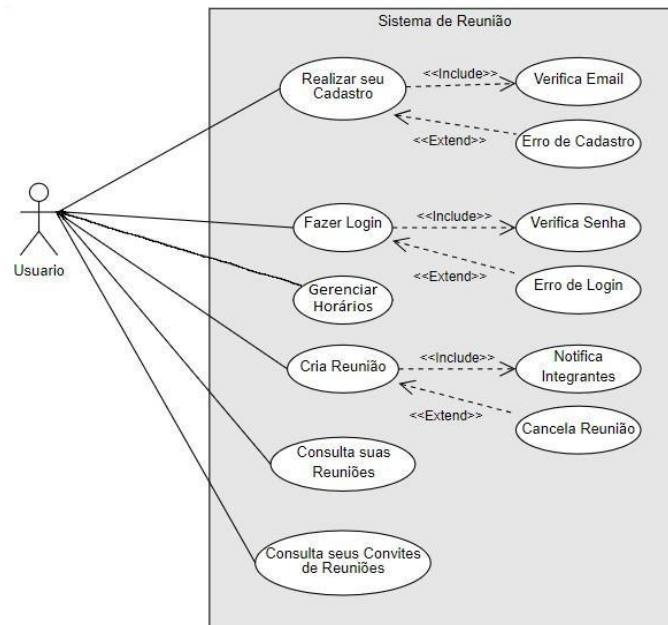
Tabela 2 – Requisitos Não Funcionais

<b>Identificador</b>	<b>Requisito</b>	<b>Classificação</b>
RNF01	<b>Usabilidade</b> , em que a aplicação é intuitiva, com uma navegação clara e acessível para usuários com diferentes níveis de conhecimento tecnológico.	Essencial
RNF02	<b>Segurança</b> , foi implementado medidas de criptografia e proteção contra acesso não autorizado.	Essencial
RNF03	<b>Escalabilidade</b> , enquanto os dados e usuários irão aumentado, o desempenho se mantêm, sem grande degradação.	Essencial
RNF04	<b>Desempenho</b> , possui um tempo de resposta curto em suas funcionalidades.	Essencial
RNF05	<b>Compatibilidade</b> , é compatível com inúmeros navegadores e dispositivos, garantido acessibilidade em diferentes plataformas.	Essencial
RNF06	<b>Manutenibilidade</b> , em que a codificação é bem estruturada e documentada, facilitando a manutenções e atualizações.	Importante
RNF07	<b>Disponibilidade</b> , estar disponível em qualquer período, minimizando tempos de inatividade, assim terá alta disponibilidade.	Desejável

Fonte: Elaborado pelo autor, 2024

Os diagramas de caso de uso são utilizados para modelar o comportamento de um sistema, auxiliando na captura de seus requisitos. Eles descrevem as funções de alto nível e o escopo do sistema, além de identificar as interações entre o sistema e seus agentes externos. Esses diagramas mostram o que o sistema faz e como os agentes o utilizam, sem detalhar seu funcionamento interno, focando apenas nas funcionalidades e na forma como elas são acessadas pelos usuários. Dessa forma, os diagramas de caso de uso facilitam a compreensão dos requisitos funcionais do sistema (IBM, 2021). A seguir é apresentado na Figura 20 o diagrama de caso de uso referente a este trabalho.

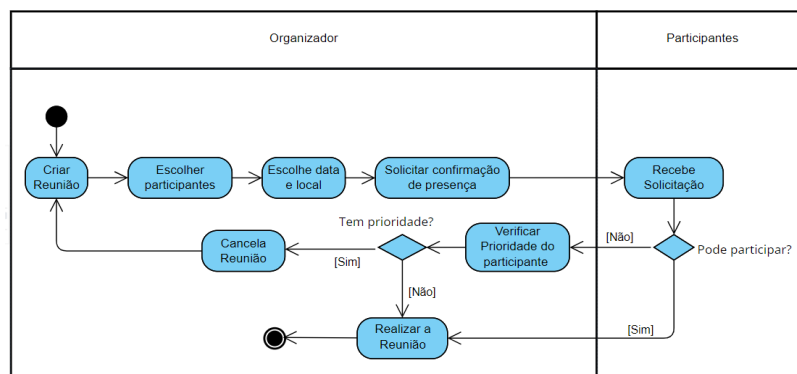
Figura 20 – Diagrama de Caso de Uso



Fonte: Elaborado pelo autor, 2024

O diagrama de atividades proporciona uma visualização detalhada do comportamento de um sistema, descrevendo a sequência de ações em um determinado processo. Embora compartilhem semelhanças com fluxogramas, ao representar o fluxo entre as ações em uma atividade, os diagramas de atividades oferecem uma visão mais abrangente. Eles permitem não apenas a representação de fluxos sequenciais, mas também de fluxos paralelos ou simultâneos, assim como fluxos alternativos, enriquecendo a análise do comportamento do sistema (IBM, 2021). A seguir, na Figura 21 é apresentado o diagrama de atividade referente a este trabalho.

Figura 21 – Diagrama de Atividade



Fonte: Elaborado pelo autor, 2024

## 5 CONSIDERAÇÕES FINAIS

O objetivo principal deste projeto foi o desenvolvimento de um sistema web focado em fornecer o suporte necessário para facilitar o agendamento de reuniões por parte dos servidores do IFMG-SJE. Tendo em vista o objetivo principal, destaca-se que por sua vez, ele se estabelece concluído, os objetivos específicos caracterizam-se como, levantamentos de requisitos necessários para o desenvolvimento do projeto, a elaboração de protótipos para validação de funcionalidades e a finalização da primeira versão pronta para utilização do sistema. Ao decorrer do projeto juntamente com várias etapas de estudo e aprendizagem, foi alcançada uma compreensão mais profunda sobre os processos de desenvolvimento de sistemas.

Após finalização do que foi planejado para ser desenvolvido nesta parte inicial do sistema, vale ressaltar que ainda existem possibilidades de escalar o sistema, implementando novas funcionalidades e integrações. Para perspectiva futura do sistema, são sugeridos alguns tópicos, como a integração do sistema com o Google Agenda, a adequação às métricas necessárias do IFMG-SJE para ser hospedado pelo instituto, o planejamento e reestruturação do sistema para a linguagem Python, utilizando seu *framework* Django, com o intuito de ser integrado ao sistema que será implementado no IFMG-SJE futuramente.

Ao final do presente projeto, destaca-se a relevância das experiências obtidas, as quais viabilizaram a utilização prática dos aprendizados adquiridos durante a trajetória acadêmica. A elaboração do sistema desenvolvido não só trouxe benefícios para a instituição, como também permitiu um entendimento mais amplo sobre as etapas essenciais na concepção de um sistema e sua integração com o banco de dados.

Em conclusão, esta trajetória contribuiu significativamente para a complementação do conhecimento teórico e prático, evidenciando a constante necessidade de atualização, aprimoramento e adaptação às práticas de desenvolvimento impulsionadas pelos contínuos avanços tecnológicos. A implementação deste projeto finaliza-se com a certeza de que o sistema desenvolvido não apenas atende aos requisitos funcionais, mas também reflete o comprometimento e a dedicação investidos pelo desenvolvedor e seus apoiadores, sempre com foco na qualidade e na melhoria da experiência do usuário.

## REFERÊNCIAS

- APPROACH. **Comunicação institucional:** o que é, qual seu objetivo + 2 exemplos. 2023. Disponível em: <https://www.approach.com.br/blog/comunicacao-institucional/>. Acesso em: 15 ago 2024.
- AWARI. **Desenvolvimento de software desktop:** Guias e melhores práticas. 2023. Disponível em: <https://awari.com.br/desenvolvimento-de-software-desktop-guias-e-melhores-praticas/>. Acesso em: 15 ago 2024.
- BRAVI. **App nativo, web app ou app híbrido?** Descubra as vantagens e desvantagens de cada tecnologia. 2019. Disponível em: <https://bravi.com.br/app-nativo-web-app-ou-app-hibrido#:~:text=Desvantagens%20do%20web%20app&text=AlÃm%20disso%2C%20eles%20tÃm%20um,nÃo%20conseguirÃa%20utilizar%20o%20aplicativo%20> . Acesso em: 21 ago 2024.
- CLARK, M. **Top 10 Frameworks Para Desenvolvimento Web.** 2022. Disponível em: <https://blog.back4app.com/pt/top-10-frameworks-para-desenvolvimento-web/> . Acesso em: 21 ago 2024.
- COMPUCENTER. **Sistema Web ou Desktop, qual melhor opção para minha empresa?** 2021. Disponível em: <https://compucenter.net.br/sistema-web-ou-desktop-qual-melhor-opcao-para-minha-empresa/> . Acesso em: 15 ago 2024.
- CUNHA, F. **Requisitos funcionais e não funcionais:** o que são? 2020. Disponível em: <https://www.mestresdawe.com.br/tecnologias/requisitos-funcionais-e-nao-funcionais-o-que-sao> . Acesso em: 04 set 2024.
- DEVMEDIA. **A arquitetura MVC no desenvolvimento em PHP.** 2017. Disponível em: <https://www.devmedia.com.br/a-arquitetura-mvc-no-desenvolvimento-em-php/23121> . Acesso em: 04 set 2024.
- EDUCAÇÃO, R. X. **Desenvolvimento web:** Tudo o que você precisa saber! 2022. Disponível em: <https://blog.xpeducacao.com.br/desenvolvimento-web/> . Acesso em: 20 ago 2024.
- FIELD, D.; WALLACE, E. **Lançamento Figma.** 2016. Disponível em: <https://www.alura.com.br/artigos/figma> . Acesso em: 09 ago 2024.
- FREENFE. **Sistema Online Versus Sistema Desktop, Como Escolher?** 2023. Disponível em: <https://blog.freenfe.com.br/categoria-dicas/sistema-online-versus-sistema-desktop-como-escolher/> . Acesso em: 15 ago 2024.
- GALE ENCYCLOPEDIA OF E-COMMERCE. **History of the Internet and World Wide Web (WWW).** 2014. Disponível em: <https://www.encyclopedia.com/economics/encyclopedias-almanacs-transcripts-and-maps/history-internet-and-world-wide-web-www> . Acesso em: 04 jul 2024.
- GEPER. **Vantagens de um sistema web.** 2011. Disponível em: <http://www.gepersistemas.com.br/content/20/vantagens-de-um-sistema-web/> . Acesso em: 20 ago 2024.

GITHUB. **Sobre o Git**. 2021. Disponível em: <https://docs.github.com/pt/get-started/using-git/about-git> . Acesso em: 11 ago 2024.

GONÇALVES, N. F. **Protótipo de uma aplicação web para gestão de eventos online**. 2022. Disponível em: <https://www.formiga.ifmg.edu.br/documents/2022/Biblioteca/TCC%20CC%20-%20Nikollas.pdf> . Acesso em: 30 ago 2024.

GOODWIN, M. **O que é uma API (interface de programação de aplicativos)?** 2024. Disponível em: <https://www.ibm.com/br-pt/topics/api> . Acesso em: 27 ago 2024.

GUEDES, M. **O que é MVC?** 2020. Disponível em: <https://www.treinaweb.com.br/blog/o-que-e-mvc> . Acesso em: 27 ago 2024.

HANASHIRO, A. **Qual a diferença entre Framework e Biblioteca?** 2020. Disponível em: <https://www.treinaweb.com.br/blog/qual-a-diferenca-entre-framework-e-biblioteca> . Acesso em: 30 ago 2024.

HOSTINGER. **O Que É MySQL? Guia Simples e Direto para Iniciantes?** 2024. Disponível em: <https://www.hostinger.com.br/tutoriais/o-que-e-mysql> . Acesso em: 21 ago 2024.

IBM. **Diagramas de Atividades**. 2021. Disponível em: <https://www.ibm.com/docs/pt-br/rational-soft-arch/9.7.0?topic=diagrams-activity> . Acesso em: 05 set 2024.

IBM. **Diagramas de Caso de Uso**. 2021. Disponível em: <https://www.ibm.com/docs/pt-br/rsm/7.5.0?topic=diagrams-use-case> . Acesso em: 04 set 2024.

KOZINETS, R. **Netnography: Doing Ethnographic Research Online**. [S.l.: s.n.], 2010. Disponível em: [https://www.researchgate.net/publication/267922181\\_Netnography\\_Doing\\_Ethnographic\\_Research\\_Online](https://www.researchgate.net/publication/267922181_Netnography_Doing_Ethnographic_Research_Online) . Acesso em: 21 ago 2024.

LOCAWEB. **Linguagens de programação web que você deve ficar de olho**. 2022. Disponível em: <https://www.locaweb.com.br/blog/temas/codigo-aberto/linguagens-de-programacao-web/> . Acesso em: 21 ago 2024.

LOGIC. **As vantagens das aplicações web**. 2018. Disponível em: <https://www.logicsp.com.br/as-vantagens-das-aplicacoes-web/> . Acesso em: 20 ago 2024.

KANEHIRA, C. **Sistemas web: o que são e qual a importância deles?**. 2020. Disponível em: <https://eescjr.com.br/blog/sistemas-web-definicao-beneficios/> . Acesso em:

MADRID, U. C. I. de. **Google Calendar**. 2021. Disponível em: <https://www.uc3m.es/uc3mdigital/guia-herramientas/google-calendar> . Acesso em: 09 ago. 2024.

MICROSOFT. **Acessar SQL: conceitos básicos, vocabulário e sintaxe**. 2021. Disponível em: <https://support.microsoft.com/pt-br/topic/acessar-sql-conceitos-basicos-vocabulario-e-sintaxe-444d0303-cde1-424e-9a74-e8dc3e460671> . Acesso em: 21 ago 2024.

NASCIMENTO, E. S. **Modelagem e Protótipo de um Sistema para Gerenciamento de Reuniões**. 2017. Disponível em: [https://www.formiga.ifmg.edu.br/documents/2017/PublicacoesTCCsBiblioteca/eduardo\\_nascimento\\_VERSAO-FINAL.pdf](https://www.formiga.ifmg.edu.br/documents/2017/PublicacoesTCCsBiblioteca/eduardo_nascimento_VERSAO-FINAL.pdf) . Acesso em: 30 ago 2024.

OLIVEIRA, D. **Conheça o dia a dia de um desenvolvedor web.** 2024. Disponível em: <https://br.hubspot.com/blog/marketing/desenvolvedor-web> . Acesso em: 23 ago 2024.

ORACLE. **O que é um Banco de Dados?** 2019. Disponível em: [https://www.oracle.com/br/database/what-is-database/#:~:text=Um%20banco%20de%%20dados%20Ã,banco%20de%20dados%20\(D BMS\)](https://www.oracle.com/br/database/what-is-database/#:~:text=Um%20banco%20de%%20dados%20Ã,banco%20de%20dados%20(D BMS)) . Acesso em: 21 ago 2024.

PEDAMKAR, P. **What is MVC Design Pattern?** 2023. Disponível em: <https://www.educba.com/what-is-mvc-design-pattern/> . Acesso em: 23 ago 2024.

PEDROSO, M. G. **PHP – Uma Introdução à Linguagem.** 2022. Disponível em: <https://www.alura.com.br/artigos/php-uma-introducao-linguagem?srsId=AfmBOoonJiA2UQFIS-xBI1iuRiOkt1pOgf6Mdo5PCQXjzyd9RIVeqkh> . Acesso em: 23 ago 2024.

PERPÉTUA, M. **10 excelentes motivos para você fazer reuniões.** 2023. Disponível em: <https://alusolda.com.br/10-excelentes-motivos-pra-voce-fazer-reunioes/> . Acesso em: 23 ago 2024.

PHP, G. D. D. D. **O que é o PHP e o que ele pode fazer?** 2023. Disponível em: [https://www.php.net/manual/pt\\_BR/introduction.php](https://www.php.net/manual/pt_BR/introduction.php) . Acesso em: 23 ago 2024.

PIAZZA, L. **Prototipagem: o que é e 10 FERRAMENTAS para fazer o seu protótipo!** 2021. Disponível em: <https://49educacao.com.br/mvp/prototipagem/> . Acesso em: 08 set 2024.

RED HAT. **O que é IDE (Ambiente de desenvolvimento integrado)?** 2023. Disponível em: <https://www.redhat.com/pt-br/topics/middleware/what-is-ide> . Acesso em: 09 ago 2024.

SANTANA, B. **Eleve Seus Designs: O Que é Figma e Como Usá-lo Corretamente!** 2023. Disponível em: <https://www.hostinger.com.br/tutoriais/figma-o-que-e> . Acesso em: 09 ago 2024.

SCHAFFER, J. **Protótipo de um sistema web para gestão e controle dos bovinos.** 2021. Disponível em: <https://arandu.iffarroupilha.edu.br/handle/itemid/262> . Acesso em: 04 jul 2024.

SCRIPTCASE. **Linguagem de Programação WEB.** 2022. Disponível em: <https://www.scriptcase.com.br/linguagem-programacao-web/> . Acesso em: 21 ago 2024.

SKILLSTECNOLOGICAS. **Desenvolvimento Desktop: Um Guia Completo para Iniciantes.** 2024. Disponível em: <https://skillstecnologicas.com/desenvolvimento-desktop/> . Acesso em: 15 ago 2024.

TECNOLOGIA, B. **Software Web x Software Desktop: Entenda as diferenças.** 2023. Disponível em: <https://bubblestecnologia.com.br/software-web-x-software-desktop-entenda-as-diferencas/> . Acesso em: 15 ago 2024.

TRIPLETEN. **Os principais frameworks para desenvolvimento web.** 2024. Disponível em: <https://tripleten.com.br/blog/os-melhores-frameworks-para-desenvolvimento-web/> . Acesso em: 23 ago 2024.

UXTOOLS. **2021 Design Tools Survey.** 2021. Disponível em: <https://uxtools.co/survey/2021/> . Acesso em: 09 ago. 2024.

VILLAIN, M.; SILVEIRA, M. I. **Figma**: o que é a ferramenta, Design e uso. 2024. Disponível em: <https://www.alura.com.br/artigos/figma> . Acesso em: 09 ago. 2024.

VISUAL STUDIO CODE. **Overview, Getting Started**. 2024. Disponível em: <https://code.visualstudio.com/docs> . Acesso em: 10 ago 2024.