



**UNIVERSIDADE FEDERAL DO CEARÁ**  
**INSTITUTO UNIVERSIDADE VIRTUAL**  
**CURSO DE GRADUAÇÃO EM SISTEMAS E MÍDIAS DIGITAIS**

**JACKSON PONTES CAVALCANTE FILHO**

**STELLA CALENDAR: UMA FERRAMENTA DE APOIO À SELEÇÃO DAS TRILHAS**  
**DO CURSO DE SISTEMAS E MÍDIAS DIGITAIS**

**FORTALEZA**

**2025**

JACKSON PONTES CAVALCANTE FILHO

STELLA CALENDAR: UMA FERRAMENTA DE APOIO À SELEÇÃO DAS TRILHAS DO  
CURSO DE SISTEMAS E MÍDIAS DIGITAIS

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Sistemas e Mídias Digitais do Instituto Universidade Virtual da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Sistemas e Mídias Digitais.

Orientador: Prof. Me. Wellington Wagner Ferreira Sarmiento

FORTALEZA

2025

Dados Internacionais de Catalogação na Publicação  
Universidade Federal do Ceará  
Sistema de Bibliotecas  
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

---

- F1s Cavalcante Filho, Jackson Pontes.  
Stella Calendar : uma ferramenta de apoio à seleção das trilhas do curso de Sistemas e Mídias Digitais /  
Jackson Pontes Cavalcante Filho. – 2025.  
66 f.
- Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Instituto UFC Virtual,  
Curso de Sistemas e Mídias Digitais, Fortaleza, 2025.  
Orientação: Prof. Me. Wellington Wagner Ferreira Sarmento.
1. Organização acadêmica. 2. Desenvolvimento de software. 3. Experiência do usuário. I. Título.  
CDD 302.23
-

JACKSON PONTES CAVALCANTE FILHO

STELLA CALENDAR: UMA FERRAMENTA DE APOIO À SELEÇÃO DAS TRILHAS DO  
CURSO DE SISTEMAS E MÍDIAS DIGITAIS

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Sistemas e Mídias Digitais do Instituto Universidade Virtual da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Sistemas e Mídias Digitais.

Aprovada em: 06 de Agosto de 2025

BANCA EXAMINADORA

---

Prof. Me. Wellington Wagner Ferreira  
Sarmiento (Orientador)  
Universidade Federal do Ceará (UFC)

---

Prof. Dr. Leonardo Oliveira Moreira  
Universidade Federal do Ceará (UFC)

---

Profa. Dra. Maria de Fátima Costa de Souza  
Universidade Federal do Ceará (UFC)

A minha mãe por fazer-me fruto de tanto esforço  
e sabedoria.

## AGRADECIMENTOS

Finalizando esta jornada, gostaria de dizer que sou grato por cada pessoa que passou pelo meu caminho durante todos esses anos de graduação.

Agradeço a mim mesmo, por não ter desistido de ser uma pessoa melhor a cada dia, que apesar dos problemas sempre tenta ser uma pessoa forte e dedicada.

Agradeço *mainha*, que sempre esteve ao meu lado nos momentos mais difíceis e também nos mais felizes, sou muito grato pela sua dedicação e cuidado.

Agradeço *pain*, que apesar de não sermos tão próximos, faz o possível e o impossível, se preciso, para me ver bem e feliz.

Agradeço minhas irmãs, que me dão forças e trazem muitos momentos de felicidade para minha vida, amo muito vocês.

Agradeço ao meu futuro esposo, Pablo José, meu maior e melhor presente da vida, por todos os dias que tivemos lado a lado, cada mensagem trocada, cada conselho e histórias que vivemos e viveremos juntos.

Agradeço aos amigos que fiz na faculdade, vocês foram essenciais para esta jornada ter sido mais leve. Agradeço de coração todos os dias, horas e minutos compartilhados juntos.

Ao meu orientador, Wellington Sarmiento, um agradecimento especial. Juntos enfrentamos desafios ao longo dessa caminhada, mas no final tudo se encaminhou como previsto.

A todos vocês, meu mais sincero obrigado. Cada pessoa, com o seu jeito, me ensinou que a vida é feita de mudanças e de diferenças, cada pessoa é única e tem seu valor. O valor de cada um de vocês me mostrou que "a vida presta" e que vale a pena ser vivida.

“Não há vantagem alguma em viver a vida correndo.”

(Shikamaru Nara)

## RESUMO

O presente trabalho descreve o refinamento da aplicação *web Stella*, voltada para auxiliar alunos do curso de Sistemas e Mídias Digitais da Universidade Federal do Ceará (UFC) na organização do horário de disciplinas com base em itinerários formativos (trilhas). A ferramenta permite aos estudantes selecionar disciplinas, verificar conflitos de horários e visualizar a distribuição semanal das aulas, além de exportar a grade em formatos *Portable Document Format (PDF)* e *Internet Calendaring and Scheduling Core Object Specification (iCalendar)* para integração com calendários digitais. A construção da Interface Gráfica com o Usuário da aplicação foi realizada utilizando a biblioteca React para o *frontend*, Node.js para o *backend* e MongoDB como banco de dados, seguindo o padrão de organização de código *feature-based folders*. Foram aplicados conceitos de usabilidade com prototipação no Figma e testes baseados na *System Usability Scale (SUS)* para avaliar a experiência do usuário. Os resultados apontaram um alto nível de usabilidade e aceitação por parte dos usuários, confirmando a eficácia da ferramenta no apoio à organização acadêmica.

**Palavras-chave:** Organização acadêmica. Desenvolvimento de software. Experiência do usuário.

## ABSTRACT

This paper describes the refinement of the **Stella** web application, designed to assist students of the Digital Media Systems program at the Federal University of Ceará (UFC) in organizing their class schedules based on formative itineraries (tracks). The tool allows students to select courses, check for schedule conflicts, and view the weekly distribution of classes. It also offers export options in *Portable Document Format (PDF)* and *Internet Calendering and Scheduling Core Object Specification (iCalendar)* formats for integration with digital calendars. The User Interface (UI) was developed using React for the *frontend*, Node.js for the *backend*, and MongoDB as the database, following a *feature-based folder* code organization pattern. Usability concepts were applied through prototyping in Figma, and the user experience was evaluated using the *System Usability Scale (SUS)*. The results indicated a high level of usability and user acceptance, confirming the tool's effectiveness in supporting academic planning.

**Palavras-chave:** Academic organization. Software development. User experience.

## LISTA DE FIGURAS

Figura 1 – Protótipo do Simulador . . . . .	18
Figura 2 – Tentativa de programação do Simulador . . . . .	18
Figura 3 – Modelo Conceitual – Simulador . . . . .	24
Figura 4 – Organização de Atividades no <i>Trello</i> . . . . .	37
Figura 5 – Estrutura da aplicação com demonstração dos componentes da ferramenta . . . . .	39
Figura 6 – Tela geral do protótipo com visualização da grade horária . . . . .	43
Figura 7 – Exemplo de conflito de horários representado com <i>toast</i> de erro . . . . .	44
Figura 8 – Botão e feedback da exportação da grade no formato ICS . . . . .	44
Figura 9 – Botão e feedback da exportação da grade em PDF . . . . .	45
Figura 10 – Exemplo de <i>modal</i> com informações detalhadas da disciplina . . . . .	45
Figura 11 – Interface com campo de busca e filtro de disciplinas . . . . .	46
Figura 12 – Tela de seleção com filtros por trilha e campos dinâmicos . . . . .	46
Figura 13 – Tela com disciplinas disponíveis e <i>checkboxes</i> de seleção . . . . .	47
Figura 14 – Estrutura de diretórios do projeto React . . . . .	48
Figura 15 – Interface da lista de disciplinas com filtros . . . . .	51
Figura 16 – Exemplo do calendário preenchido com disciplinas . . . . .	52
Figura 17 – Notificação de conflito entre disciplinas . . . . .	53
Figura 18 – Modal com informações detalhadas da disciplina . . . . .	54
Figura 19 – Botão de exportação em PDF . . . . .	55
Figura 20 – Botão de exportação em formato ICS . . . . .	56
Figura 21 – Classificação da ferramenta segundo a Escala SUS . . . . .	58
Figura 22 – Primeira página do questionário (SUS) . . . . .	65
Figura 23 – Segunda página do questionário (SUS) . . . . .	66
Figura 24 – Perguntas abertas do questionário . . . . .	67

## LISTA DE QUADROS

Quadro 1 – Tecnologias utilizadas no desenvolvimento do Stella . . . . .	25
Quadro 2 – Comparativo entre o Stella e sistemas relacionados . . . . .	28

## LISTA DE CÓDIGOS-FONTE

Código-fonte 1	– Exemplo de resposta JSON da rota GET /subject . . . . .	40
Código-fonte 2	– Trecho do componente SubjectSelector com filtros . . . . .	50
Código-fonte 3	– Configuração de eventos recorrentes com FullCalendar . . . . .	51
Código-fonte 4	– Função para verificação de conflitos de horário . . . . .	52
Código-fonte 5	– Modal . . . . .	53
Código-fonte 6	– Exportação da grade em PDF . . . . .	55
Código-fonte 7	– Evento recorrente no padrão do formato ICS . . . . .	55

## LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
CSS	Cascading Style Sheets
DOM	Document Object Model
HTTPS	Hypertext Transfer Protocol Secure
ICS	Internet Calendaring and Scheduling Core Object Specification (iCalendar)
IDE	Integrated Development Environment
JSON	JavaScript Object Notation
JWT	JSON Web Token
LGPD	Lei Geral da Proteção de Dados
MVC	Model-View-Controller
NPM	Node Package Manager
ODM	Object Data Modeling
PDF	Portable Document Format
RFA	Requisitos do Perfil de Administrador
RFE	Requisitos do Perfil de Estudante
RFG	Requisitos Gerais
SIGAA	Sistema Integrado de Gestão de Atividades Acadêmicas
SMD	Sistemas e Mídias Digitais
SUS	System Usability Scale
TCC	Trabalho de Conclusão de Curso
UFC	Universidade Federal do Ceará
UFMG	Universidade Federal de Campina Grande
UFSC	Universidade Federal de Santa Catarina
UI	User Interface
USP	Universidade de São Paulo
UTFPR	Universidade Tecnológica Federal do Paraná
UX	User Experience

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>15</b>
<b>1.1</b>	<b>Objetivos</b>	<b>17</b>
<b>1.2</b>	<b>Justificativa</b>	<b>17</b>
<b>2</b>	<b>REFERENCIAL TEÓRICO</b>	<b>20</b>
<b>2.1</b>	<b>Aplicações Web</b>	<b>20</b>
<b>2.2</b>	<b>Contexto de Uso de Aplicações Web</b>	<b>21</b>
<b>2.3</b>	<b>Desenvolvimento de Aplicações Web</b>	<b>22</b>
<b>2.4</b>	<b>Visão geral sobre o Projeto Stella</b>	<b>23</b>
<b>2.5</b>	<b>Ferramenta de Organização de Horários</b>	<b>26</b>
<b>3</b>	<b>TRABALHOS RELACIONADOS</b>	<b>27</b>
<b>3.1</b>	<b>METIS – Aplicação Web para Orientação Acadêmica</b>	<b>27</b>
<b>3.2</b>	<b>Sistemas Universitários de Matrícula</b>	<b>27</b>
<b>3.3</b>	<b>Ferramentas de Calendário e Interoperabilidade</b>	<b>28</b>
<b>3.4</b>	<b>Comparativo com o Stella</b>	<b>28</b>
<b>3.5</b>	<b>Considerações sobre Trabalhos Relacionados</b>	<b>29</b>
<b>4</b>	<b>METODOLOGIA</b>	<b>30</b>
<b>5</b>	<b>DESENVOLVIMENTO</b>	<b>32</b>
<b>5.1</b>	<b>Contexto do Desenvolvimento</b>	<b>32</b>
<b>5.1.1</b>	<i>Protótipo da Ferramenta</i>	<b>33</b>
<b>5.1.2</b>	<i>Desenvolvimento da Ferramenta</i>	<b>33</b>
<b>5.1.3</b>	<i>Integração ao Sistema Stella</i>	<b>34</b>
<b>5.2</b>	<b>Tecnologias Utilizadas</b>	<b>35</b>
<b>5.2.1</b>	<i>Frontend: React.js</i>	<b>35</b>
<b>5.2.2</b>	<i>Componente de Calendário: FullCalendar</i>	<b>35</b>
<b>5.2.3</b>	<i>Backend: Node.js com Express</i>	<b>35</b>
<b>5.2.4</b>	<i>Banco de Dados: MongoDB</i>	<b>36</b>
<b>5.2.5</b>	<i>Controle de Versão: Git</i>	<b>36</b>
<b>5.2.6</b>	<i>IDE e Ferramentas Auxiliares</i>	<b>36</b>
<b>5.3</b>	<b>Gerenciamento das Atividades com Trello</b>	<b>37</b>
<b>5.4</b>	<b>Requisitos do Sistema</b>	<b>38</b>

5.4.1	<i>Requisitos Funcionais</i> . . . . .	38
5.4.2	<i>Requisitos Não Funcionais</i> . . . . .	38
5.5	<b>Casos de Usuário</b> . . . . .	38
5.6	<b>Arquitetura da Aplicação e Estrutura do Calendário</b> . . . . .	39
5.6.1	<i>API RESTful e Comunicação com o Frontend</i> . . . . .	40
5.7	<b>Modelagem de Dados no MongoDB</b> . . . . .	41
5.7.1	<i>Coleção <code>users</code></i> . . . . .	41
5.7.2	<i>Coleção <code>subjects</code></i> . . . . .	41
5.7.3	<i>Considerações sobre a Modelagem</i> . . . . .	42
5.8	<b>Prototipação da Ferramenta</b> . . . . .	42
5.9	<b>Detalhes de Implementação</b> . . . . .	47
5.9.1	<i>Estrutura de Pastas do Projeto</i> . . . . .	47
5.9.2	<i>Componente de Seleção de Disciplinas</i> . . . . .	49
5.9.3	<i>Calendário de Visualização da Grade</i> . . . . .	51
5.9.4	<i>Verificação de Conflito de Horário</i> . . . . .	52
5.9.5	<i>Modal de Visualização Detalhada</i> . . . . .	53
5.9.6	<i>Exportação em PDF da Grade</i> . . . . .	55
5.9.7	<i>Exportação para Arquivo no Formato ICS</i> . . . . .	55
5.10	<b>Considerações Finais</b> . . . . .	56
6	<b>TESTES DE USABILIDADE E AVALIAÇÃO DA APLICAÇÃO</b> . . . . .	57
6.1	<b>Resultados da Escala SUS</b> . . . . .	57
6.2	<b>Feedback Qualitativo</b> . . . . .	58
6.3	<b>Discussão</b> . . . . .	58
7	<b>CONCLUSÃO</b> . . . . .	60
7.1	<b>Trabalhos Futuros</b> . . . . .	60
	<b>REFERÊNCIAS</b> . . . . .	62
	<b>APÊNDICES</b> . . . . .	64
	<b>APÊNDICE A – Links Complementares</b> . . . . .	64
	<b>APÊNDICE B – Perguntas do Questionário para Avaliação do Usuário</b> . . . . .	65

## 1 INTRODUÇÃO

Na Universidade Federal do Ceará (UFC), existem diversos cursos voltados para a área de tecnologia, sendo um deles o curso de Sistemas e Mídias Digitais (SMD), que tem uma estrutura curricular diversificada e multidisciplinar, com a possibilidade de cursar disciplinas de diversas áreas tecnológicas, como: programação, jogos digitais, audiovisual, etc. O curso tem a finalidade de formar bacharéis com conhecimentos especializados em duas grandes áreas principais: Sistemas Multimídia e Mídias Digitais. Desta forma, contribuindo para o desenvolvimento, no médio prazo, de novos perfis profissionais que possam sustentar o desenvolvimento de um polo regional tecnológico no estado do Ceará, fomentando e viabilizando atividades produtivas nas áreas de produção de mídias digitais e desenvolvimento de sistemas multimídia, como sistemas *web* e para dispositivos móveis, jogos digitais e animações gráficas (SMD UFC, 2024).

O curso de SMD possui uma formação baseada em itinerários formativos, as chamadas Trilhas. A partir do quinto semestre o aluno pode escolher um itinerário formativo, que consistem em um conjunto de disciplinas optativas que compõem a organização da sua formação em uma área específica, como jogos digitais, animação e audiovisual, *design* digital interativo ou sistemas multimídia. É função da coordenação do curso organizar os horários das turmas de um semestre de modo a permitir a variedade na escolha de disciplinas por parte dos alunos.

Embora o funcionamento das trilhas seja abordado na disciplina de Introdução a SMD, os estudantes só se deparam efetivamente com essa realidade, geralmente, dois anos depois do ingresso no curso, uma vez que as disciplinas que compõem o percurso entre primeiro e terceiro semestre são de natureza obrigatória e no quarto semestre há a aplicação de disciplinas eletivas, escolhidas dentre poucas opções. A partir do quinto semestre, quando essa metodologia começa de fato a ser aplicada, existe uma certa confusão entre os discentes, ainda que a coordenação faça uma divulgação, realizada através do perfil oficial do curso no Instagram, acerca do funcionamento das trilhas e sugestão de escolha para cada uma delas. Sob este viés, levando-se em consideração o aspecto inovador e tecnológico do curso, se fez necessário desenvolver formas mais interativas de esclarecer, facilitar e auxiliar os estudantes a participarem e entenderem melhor essa estrutura curricular.

Nesse contexto, o Projeto Stella foi concebido como uma resposta direta à necessidade de tornar mais acessível e interativa a compreensão das trilhas acadêmicas oferecidas pelo curso. Este, que foi um projeto desenvolvido para a disciplina de Projeto Integrado I do

curso de SMD da UFC, teve sua construção durante um semestre apenas, onde foram discutidas diversas questões sobre o *design* de experiência do usuário (UX) e o *design* de interface do usuário (UI), funcionalidades do *website* e também foram realizados diversos testes com os possíveis usuários. O objetivo principal do projeto foi trazer ao aluno do curso de SMD um sistema *web* em que o discente conseguisse ter acesso a informações sobre as trilhas presentes no curso, pudesse conhecer as disciplinas de cada trilha e seus respectivos docentes, além de poder montar uma simulação de como ficaria seu horário com as disciplinas escolhidas baseadas na disponibilidade de disciplinas ofertadas a cada semestre. O projeto foi desenvolvido para o contexto da disciplina de Projeto Integrado I, na qual era necessário conseguir um cliente e realizar um projeto. Apesar disso, acredita-se que o Stella tem potencial para se tornar um projeto que seja utilizado pela comunidade do curso de SMD. Sendo assim, o projeto segue em aberto para melhorias e discussões sobre novas funcionalidades, novos testes, sua publicação, etc.

Dessa forma, uma das funcionalidades propostas no projeto Stella pelo autor deste trabalho e sua equipe "Jovens Titãs", responsáveis pela produção do Stella na disciplina de Projeto Integrado I, o simulador de horários, era considerada uma das mais relevantes, mas acabou não sendo totalmente desenvolvida e implementada na época devido à sua complexidade e às limitações de tempo enfrentadas durante a disciplina (no Apêndice A, há um link para o GitHub do Projeto onde é possível ver em sua descrição, todas as funcionalidades que foram possíveis concluir durante o período da disciplina). Neste Trabalho de Conclusão de Curso (TCC), essa ideia foi reformulada e desenvolvida de maneira mais robusta e completa, resultando em uma ferramenta funcional para organização de horários.

A ferramenta concebida neste trabalho permite que os estudantes visualizem de forma interativa as disciplinas disponíveis, montem suas grades horárias, identifiquem possíveis conflitos e exportem suas grades em formatos como PDF e ICS para integração com calendários digitais. Todo o desenvolvimento foi realizado de forma independente, com maior atenção à experiência do usuário, utilizando tecnologias modernas e melhores práticas de desenvolvimento. Essa abordagem representa um avanço significativo em relação ao projeto original e entrega aos discentes uma solução prática e inovadora para o planejamento acadêmico no curso de Sistemas e Mídias Digitais.

## 1.1 Objetivos

O objetivo geral do presente trabalho é aperfeiçoar uma aplicação *web* que possibilite aos discentes do curso de SMD da UFC organizarem de forma simplificada e personalizada, seus horários de aula, a fim de poderem otimizar melhor seus tempos de estudo, trabalho e lazer. A partir deste objetivo geral, desenvolveu-se os seguintes objetivos específicos:

1. Criar uma prototipação da nova ferramenta no Figma.
2. Desenvolver a ferramenta de horários baseado no protótipo criado.
3. Garantir que as principais funcionalidades da ferramenta estejam funcionando, como: visualização de disciplinas seleção de disciplinas, filtros de seleção/pesquisa, interação com calendário e exportação semanal (formato PDF) e mensal (formato ICS) do calendário.
4. Fazer teste de usabilidade com alunos do próprio curso.

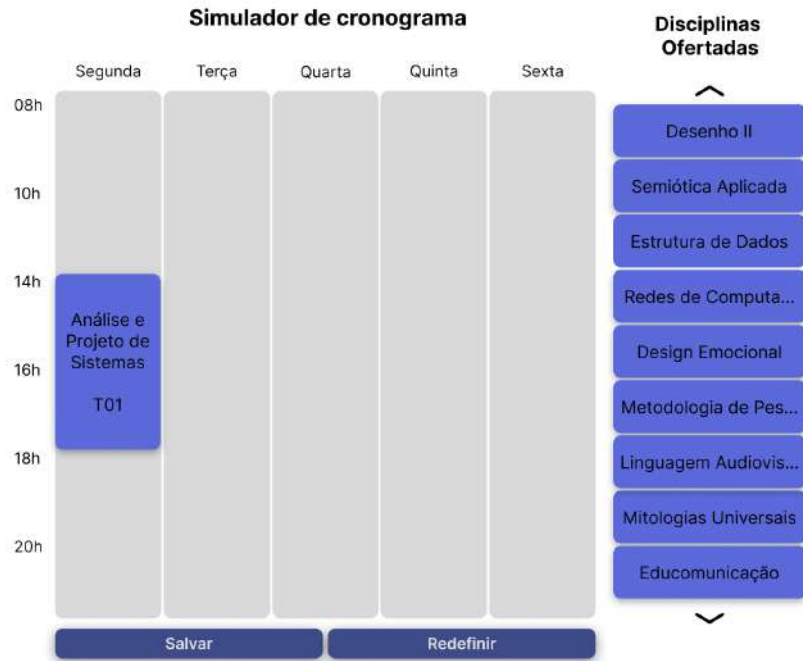
## 1.2 Justificativa

O curso de SMD da UFC adota, a partir do quinto semestre, um modelo curricular baseado em itinerários formativos, chamados de Trilhas. Apesar de inovador, esse modelo ainda apresenta dificuldades de compreensão por parte dos estudantes, especialmente no que diz respeito à organização das disciplinas optativas e à construção de uma grade horária viável e livre de conflitos.

A forma como essas informações são tradicionalmente divulgadas, por meio de planilhas ou publicações em redes sociais, não é suficiente para garantir clareza, acessibilidade e interatividade no processo de planejamento acadêmico. Essa limitação pode gerar insegurança, sobrecarga cognitiva e escolhas inadequadas por parte dos discentes, comprometendo seu rendimento e engajamento ao longo da formação.

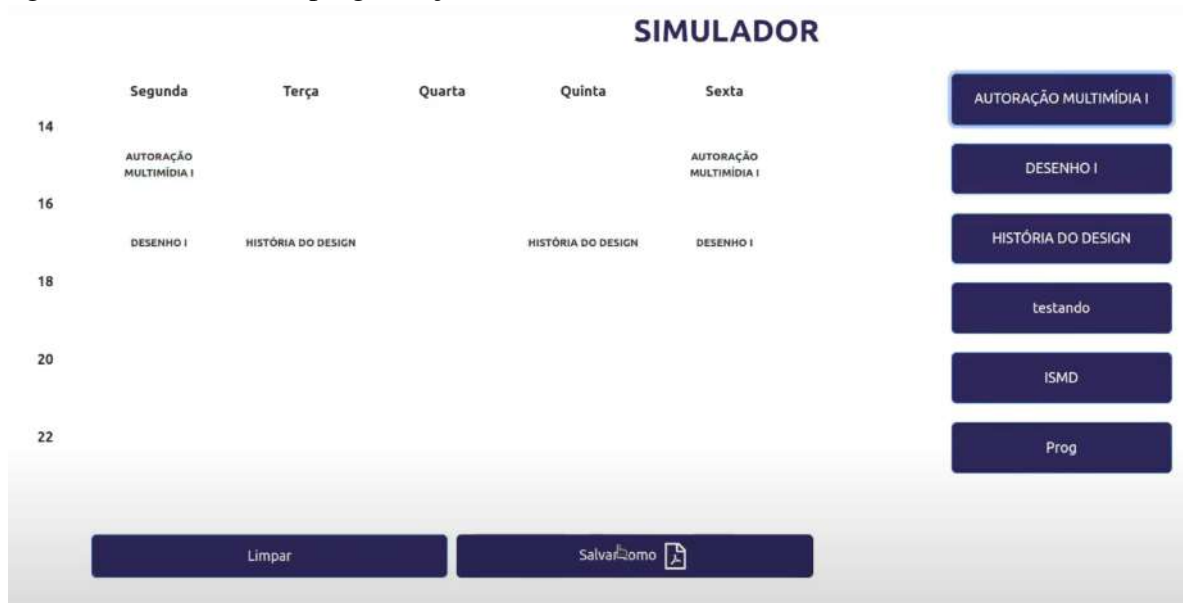
O Projeto Stella surgiu com o objetivo de preencher essa lacuna, propondo uma plataforma digital interativa para consulta das trilhas. No entanto, uma de suas funcionalidades mais desejadas, o simulador digital, não foi plenamente desenvolvida durante a disciplina de Projeto Integrado I, sendo feito apenas um protótipo no Figma, e uma tentativa de desenvolvimento seguindo o protótipo. A Figura 1 mostra o protótipo feito no Figma e a Figura 2 mostra a tentativa de desenvolvimento do simulador, ambos trabalhos feitos pela equipe "Jovens Titans" durante a disciplina de Projeto Integrado I:

Figura 1 – Protótipo do Simulador



Fonte: Elaborado pelo autor (2025).

Figura 2 – Tentativa de programação do Simulador



Fonte: Elaborado pelo autor (2025).

Portanto, a presente proposta é justificada pela necessidade de aprimorar a experiência acadêmica dos estudantes por meio de um recurso digital intuitivo e funcional. Ao permitir a personalização da grade, evitar conflitos de horário e oferecer integração com calendários digitais, a ferramenta contribuirá significativamente para a autonomia, planejamento e organização dos alunos. Além disso, reforça o papel da tecnologia educacional como aliada na gestão da trajetória universitária, alinhando-se aos princípios do próprio curso que valoriza a interdisciplinaridade entre *design*, sistemas e mídias digitais.

## 2 REFERENCIAL TEÓRICO

Neste capítulo será apresentada uma breve introdução sobre aplicações *web* e também tópicos sobre as tecnologias e metodologias utilizadas para a produção e funcionamento do projeto, além da apresentação de como será implementada a nova funcionalidade. Sendo assim, a primeira parte deste referencial teórico abordará o tema aplicações *web*, onde será discutido sobre sua definição e algumas vantagens do seu uso. A segunda parte traz seu Contexto de Uso, onde serão explorados alguns exemplos. Em seguida, a terceira parte explica um pouco sobre como se dá o desenvolvimento dessas aplicações, além de explorar e explanar algumas vantagens e desvantagens. Logo após esses tópicos relacionados as aplicações *web*, o projeto Stella e seu desenvolvimento será apresentado, desde a ideia inicial, pesquisas, divisões de trabalho, organização, até o seu desenvolvimento. E, por fim, no último tópico será abordado como será o processo de desenvolvimento da ferramenta organizadora de horários, ressaltando tecnologias que serão utilizadas, além de explicar seu propósito dentro do projeto.

### 2.1 Aplicações Web

As aplicações *web* são sistemas de software que utilizam tecnologias e padrões da *web* para permitir a interação dos usuários por meio de navegadores. Diferente dos aplicativos tradicionais, que precisam ser instalados localmente nos dispositivos, as aplicações *web* são acessadas via internet e executadas no servidor, com a interface apresentada no navegador do cliente.

De acordo com Loudon e Loudon (2014), as aplicações *web* "consistem em páginas da *web* dinâmicas capazes de fornecer informações e permitir a interação com o usuário, sendo atualizadas em tempo real conforme as necessidades e ações do usuário". Essa definição destaca a natureza interativa e dinâmica dessas aplicações, que podem variar desde simples sites informativos até complexos sistemas de gestão empresarial, *e-commerces*, redes sociais, entre outros.

A principal vantagem das aplicações *web* é a sua facilidade de acesso e manutenção. Por estarem centralizadas em servidores, as atualizações e correções podem ser realizadas de maneira rápida e eficiente, sem a necessidade de intervenção direta no dispositivo do usuário. Além disso, a compatibilidade multiplataforma permite que sejam acessadas a partir de qualquer dispositivo com um navegador, seja um computador, *tablet* ou *smartphone*.

Um exemplo clássico de aplicação *web* é o Google Docs, que permite a criação, edição e compartilhamento de documentos em tempo real, tudo isso através do navegador, sem a necessidade de instalar um software específico no dispositivo do usuário.

A seguir, serão explorados aspectos adicionais relacionados ao contexto de uso das aplicações *web* e sua necessidade no âmbito acadêmico.

## **2.2 Contexto de Uso de Aplicações Web**

Está cada vez mais comum que empresas e organizações em geral utilizem a *web* para prestar algum tipo de serviço ou fazer algum tipo de divulgação. Isso se deve ao crescimento, nos últimos anos, do número de pessoas que utilizam smartphones ou computadores para acessar informações.

Conforme o NÚCLEO DE INFORMAÇÃO E COORDENAÇÃO DO PONTO BR (NIC.br) (2023), com o aumento do acesso à internet, empresas e organizações em geral têm reconhecido a importância de estar presentes no ambiente digital para alcançar seu público-alvo de maneira mais eficaz. Segundo Sobral (2017), sites e mídias sociais já não se configuram mais como diferenciais entre as organizações, mas são ferramentas essenciais que estão relacionadas diretamente com a sua competitividade e sobrevivência no mercado. A presença *online*, por meio de *websites* e aplicações *web*, tornou-se essencial para a divulgação de informações e prestação de serviços. Para Sobral (2017), “é na internet que clientes atuais e em potencial buscam e encontram endereços, produtos e serviços, sendo o ambiente virtual também um importante meio de comunicação entre consumidor/usuário e empreendedor.”.

O interesse em utilizar aplicações *web* vai além das empresas privadas. Governos e outros tipos de organizações também têm adotado essa abordagem para facilitar o acesso a serviços públicos e informações essenciais para a população. Por exemplo, a digitalização de serviços governamentais, como a emissão de documentos e o acesso a benefícios sociais, tem proporcionado maior agilidade e comodidade para a maioria dos cidadãos.

No contexto da universidade também é notável o interesse em divulgar informações para os docentes/discentes sobre o que acontece no mundo universitário. Com isso, a maioria das universidades possui seu próprio *website* contendo informações e notícias sobre tudo que envolve a instituição. Vale observar que dentro da própria Universidade existem nichos que, dependendo da situação, se faz necessário a criação de um software, na forma de aplicação *web*, que ajude na divulgação de informações importantes entre os estudantes de um curso.

É o caso do curso de Sistemas e Mídias Digitais da Universidade Federal do Ceará, que para manter a divulgação das trilhas do curso mais viva ainda, se fez necessário desenvolver formas mais interativas de esclarecer, facilitar e auxiliar os estudantes a participarem e entenderem esse processo de formação, resultando na produção da aplicação *web* Stella.

Com isso, pode-se perceber a importância do desenvolvimento de um meio que seja útil e eficiente para a divulgação de informações, e levando isso em consideração, o desenvolvimento de aplicações *web* é uma boa alternativa de divulgação.

A seguir, será abordado um pouco sobre os aspectos do processo de desenvolvimento de uma aplicação *web* e quais suas vantagens e desvantagens em relação ao seu uso.

### 2.3 Desenvolvimento de Aplicações Web

O desenvolvimento de aplicações *web* tem se tornado cada vez mais pertinente no cenário tecnológico atual, impulsionado pela crescente demanda por soluções digitais *online* eficientes e acessíveis. As aplicações *web* oferecem diversas vantagens, mas também apresentam algumas desvantagens. Explorando ambos os lados dessa tecnologia, pode-se observar suas potencialidades e limitações. Dessa forma, artigos apontam vantagens e desvantagens das aplicações *web* como soluções tecnológicas digitais.

Algumas das vantagens incluem (SYSTEM, 2024; BRAVI, 2024):

- a) **Disponibilidade e Conveniência:** poder usufruir de software sem ter que passar pelo processo de instalação destes recursos num computador pessoal; além de poder acessar os documentos e informações contidas nestes softwares de qualquer computador. Um exemplo é o Google Drive que permite aos usuários armazenar, acessar e compartilhar arquivos a partir de qualquer dispositivo com internet.
- b) **Responsividade e Compatibilidade Multidispositivo:** a maioria das aplicações *web* modernas são projetadas para serem responsivas, ou seja, adaptam-se automaticamente a diferentes tamanhos de telas e dispositivos.
- c) **Atualizações Centralizadas e Manutenção:** aplicações *web* são atualizadas de forma centralizada em um servidor, o que significa que os usuários sempre acessam a versão mais recente sem necessidade de *downloads* ou instalações adicionais.

Sobre as desvantagens vale citar (SYSTEM, 2024; BRAVI, 2024):

- a) **Dependência de Conexão com a Internet:** uma das principais desvantagens das aplica-

ções *web* é a necessidade de ter uma conexão constante com a internet.

- b) **Desempenho:** aplicações *web* podem enfrentar limitações de desempenho em comparação com aplicações nativas, especialmente em tarefas que exigem processamento intenso.

O desenvolvimento *web* é fundamentado na arquitetura cliente-servidor de aplicações distribuídas, podendo adotar padrões suplementares que auxiliam na organização e construção dessas aplicações. Um exemplo amplamente utilizado, e que foi aplicado no *backend* do Stella ao realizar o presente trabalho, é o padrão *Model-View-Controller (MVC)*, que separa o modelo de negócio (*Model*) da interface com o usuário (*View*), por meio de uma camada intermediária chamada controlador (*Controller*), responsável por gerenciar a comunicação entre as outras duas partes. Outro padrão relevante em aplicações *web* é o de Microserviços, no qual as funcionalidades do sistema são divididas em pequenos serviços autônomos que interagem entre si de forma coesa. Há ainda outras abordagens arquiteturais aplicadas ao desenvolvimento *web*, como as discutidas por Oliveira Júnior e Fortes (JÚNIOR; FORTES, 2022) e no material da Microsoft apresentado no *Microsoft Ignite* (Microsoft Corporation, 2025). Essas abordagens convergem para a necessidade de dividir as aplicações em múltiplas camadas, o que promove a modularização do desenvolvimento e da manutenção do sistema. Essa modularização possibilita a adição de novas funcionalidades com menor risco de comprometer o código já existente e facilita o entendimento da estrutura do sistema, sobretudo em projetos com grande número de requisitos.

## 2.4 Visão geral sobre o Projeto Stella

O projeto Stella foi desenvolvido pela equipe "Jovens Titans" sob a orientação do professor Me. Wellington Wagner F. Sarmiento e da professora Dra. Cátia Luzia Oliveira da Silva, docentes do curso noturno de Sistemas e Mídias Digitais. O projeto foi idealizado pelo professor Dr. Alysson Diniz, antigo coordenador do curso noturno de Sistemas e Mídias Digitais, durante a disciplina de Projeto Integrado I. O objetivo principal do projeto era desenvolver uma aplicação *web* que facilitasse a compreensão e organização das trilhas formativas oferecidas a partir do quinto semestre do curso.

A partir da organização do projeto, foi possível perceber três modelos de públicos interessados, como *stakeholders*, ou seja, perfis que se beneficiariam diante do sistema. Entretanto, dos três modelos, apenas dois utilizariam possivelmente a aplicação: alunos do curso de Sistemas e Mídias Digitais, para precisar colher as informações sobre as trilhas de uma

forma mais interativa e organizada, e a coordenação que preencheria o sistema semestralmente com a liberação das disciplinas correspondentes. Além disso, há o perfil dos professores que participariam indiretamente com as informações sobre as disciplinas ofertadas e estarão cientes do projeto.

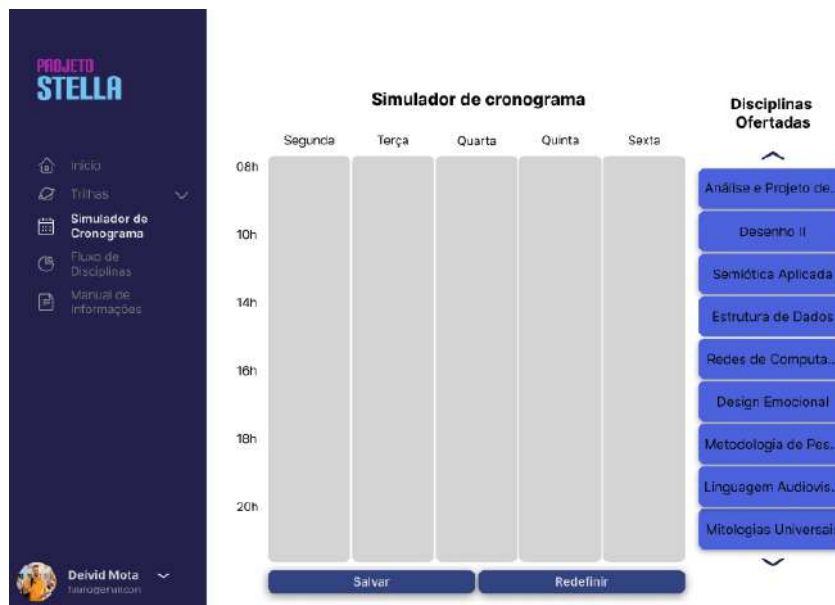
Nesse contexto, a aplicação reuniria os dados das trilhas, a partir do 4º semestre do curso de Sistemas e Mídias Digitais, onde os alunos poderiam identificar e organizar melhor quais disciplinas iriam compor sua grade curricular e seguir com seu planejamento profissional.

Assim, cada perfil se beneficiaria com a aplicação para diferentes fins, mas compondo um só sistema prático, interativo e funcional.

Para a realização do projeto, após toda a discussão sobre público-alvo, conversa com professores, secretários e líderes das trilhas, foi feita uma análise hierárquica das tarefas, na qual foi projetado um modelo de exploração para a área do estudante, em que as atividades destinadas seriam: entender as trilhas, ajustar o cronograma, conhecer os professores responsáveis por cada disciplina. Já para os administradores as principais características são de instruções para ajustar informações e dados necessários a serem distribuídos aos alunos como: alterar dados e confirmar disciplinas.

No processo de desenvolvimento da aplicação, foi criado o Modelo Conceitual (ver Figura 3). Durante o desenvolvimento dos modelos, o principal objetivo foi garantir uma experiência amigável, fugindo de aspectos formalistas e evitando etapas desnecessárias, reduzindo o cansaço e criando uma interação agradável e, ocasionalmente, lúdica.

Figura 3 – Modelo Conceitual – Simulador



Fonte: Elaborado pelo autor (2025).

Também foram levantados requisitos funcionais do sistema para ficar mais organizado qual a relação de cada funcionalidade desenvolvida, divididos em: Requisitos Gerais (RFG), Requisitos do Perfil de Administrador (RFA) e Requisitos do Perfil de Estudante (RFE).

O levantamento das tecnologias também são etapas fundamentais no desenvolvimento de software. Escolher as tecnologias certas garante que o projeto seja viável, eficiente e sustentável. Dessa forma foram selecionadas as tecnologias presentes no Quadro 1, para possibilitar o desenvolvimento do Stella.

Quadro 1 – Tecnologias utilizadas no desenvolvimento do Stella

ID	Software	Tipo	Link
1	Figma	Software de prototipação de interface gráfica	< <a href="https://www.figma.com/downloads/">https://www.figma.com/downloads/</a> >
2	Trello	Software de organização	< <a href="https://trello.com/">https://trello.com/</a> >
3	Maze	Software de teste de usabilidade	< <a href="https://t.maze.co/99892936">https://t.maze.co/99892936</a> >
4	VS Code	Ambiente de desenvolvimento	< <a href="https://code.visualstudio.com/">https://code.visualstudio.com/</a> >
5	Git	Ferramenta de versionamento	< <a href="https://git-scm.com/">https://git-scm.com/</a> >
6	React JS	Biblioteca JavaScript usada nas interfaces da aplicação <i>web</i>	< <a href="https://pt-br.reactjs.org/">https://pt-br.reactjs.org/</a> >
7	Node JS	Interpretador JS no servidor <i>web</i> , utilizado na API	< <a href="https://nodejs.org/pt-br/">https://nodejs.org/pt-br/</a> >
8	MongoDB	Software de banco de dados	< <a href="https://www.mongodb.com/pt-br">https://www.mongodb.com/pt-br</a> >
9	Mongoose	Mapeamento objeto-relacional	< <a href="https://mongoosejs.com/">https://mongoosejs.com/</a> >
10	Postman	Teste de requisições	< <a href="https://www.postman.com/">https://www.postman.com/</a> >

Fonte: Elaborado pelo autor (2025).

Por fim, o Projeto Stella foi definido como uma aplicação híbrida (*Web e Mobile*), que pode ser acessada através de navegadores *web*, desenvolvida em função do curso de Sistemas e Mídias Digitais para a compreensão mais prática dos alunos em relação às trilhas. A aplicação é interativa, procurando alcançar a simplicidade e funcionalidade almejadas por seu público-alvo, com o objetivo de contribuir para sua organização e gerenciamento de tempo. No Apêndice A, há um link para os repositórios do Projeto no Github onde é possível ver em sua descrição, todas as funcionalidades que foram possíveis concluir durante o período da disciplina de Projeto Integrado I, além do *design* proposto da aplicação.

Entendendo como funciona e como se configura o projeto Stella, pode-se discutir agora sobre a implementação que será realizada e sobre a nova ferramenta a ser desenvolvida

neste trabalho.

## **2.5 Ferramenta de Organização de Horários**

A ideia central de organização de horários, proposta neste trabalho, é permitir que os estudantes visualizem as disciplinas ofertadas, filtrem por trilhas ou professores, selecionem as disciplinas de interesse e montem sua própria grade horária, evitando conflitos de horário e permitindo exportações em formatos úteis, como PDF e o padrão de documentos para agendas *online* ICS.

Essa solução busca não apenas otimizar o processo de escolha de disciplinas, mas também proporcionar uma experiência mais autônoma e acessível para o planejamento acadêmico dos alunos. Sua implementação será apresentada detalhadamente nos capítulos seguintes, abordando as decisões técnicas, prototipação e testes realizados ao longo do processo de desenvolvimento.

### 3 TRABALHOS RELACIONADOS

Este capítulo apresenta sistemas e pesquisas que se aproximam do propósito deste projeto, focando em ferramentas de simulação de grade horária, apoio à matrícula e orientação acadêmica interativa. O objetivo é identificar boas práticas e lacunas que justificam e fundamentam as decisões de projeto adotadas no sistema Stella.

#### 3.1 METIS – Aplicação Web para Orientação Acadêmica

O software METIS, desenvolvido por Andrade (2020) na Universidade Federal da Bahia (UFBA), é uma aplicação web que visa auxiliar alunos na simulação de matrícula. A plataforma permite que o estudante visualize sua grade curricular, selecione disciplinas e identifique automaticamente conflitos de horários. Além disso, o sistema oferece diferentes perfis de usuários (aluno, coordenador, professor) e autenticação institucional.

Entre os recursos notáveis do METIS estão o uso de notificações Toastr para alertas não intrusivos e a personalização de *feedbacks* durante o processo de matrícula. Embora compartilhe o propósito de organização acadêmica com o Stella, o METIS não possui integração com calendários digitais nem funcionalidades de exportação em formatos como PDF ou ICS.

#### 3.2 Sistemas Universitários de Matrícula

Diversas universidades brasileiras desenvolvem ferramentas próprias para simulação de matrícula e organização acadêmica:

- **MatrUSP (USP) e Grade na Hora (UTFPR):** permitem simulações com base em grade curricular e horários disponíveis, com visualização semanal;
- **MatrUFSC (UFSC):** evoluiu com funcionalidades como busca por combinações de disciplinas e feedback visual aprimorado;
- **Cursos UFCG e UFCG-PRO:** permitem planejamento de matrícula e integração com o sistema oficial da instituição, com destaque para marcação de disciplinas já cursadas e controle de pré-requisitos.

Tais sistemas cumprem bem sua função acadêmica, mas apresentam limitações em aspectos como personalização visual (interfaces pouco flexíveis, sem opções para o aluno ajustar cores, disposição ou forma de visualização da grade), usabilidade (fluxos de interação pouco intuitivos, exigindo múltiplos cliques ou navegação por várias telas para realizar tarefas simples)

e interoperabilidade de dados (ausência de exportação em formatos padrão, como PDF ou ICS, ou integração com outros sistemas acadêmicos e aplicativos de calendário).

### 3.3 Ferramentas de Calendário e Interoperabilidade

Sistemas como o SIGAA (usado pela UFC) fornecem informações sobre disciplinas e turmas, mas não permitem a visualização interativa da grade semanal nem oferecem funcionalidades de exportação.

Por outro lado, ferramentas como o Google Calendar popularizaram o uso do formato ICS para integração entre calendários. Essa inspiração motivou a implementação da exportação neste formato, para a ferramenta de Organização de Horários, integrada ao Stella, possibilitando que os alunos e alunas adicionem suas matrizes curriculares do semestre diretamente a aplicativos de calendário digital, ampliando a utilidade prática da ferramenta.

### 3.4 Comparativo com o Stella

O Quadro 2 apresenta um comparativo entre o Stella e outros sistemas apresentados, destacando os principais recursos presentes ou ausentes em cada solução.

Quadro 2 – Comparativo entre o Stella e sistemas relacionados

<b>Recurso</b>	<b>Stella</b>	<b>Trabalhos Relacionados</b>
Simulação de matrícula	Seleção dinâmica de disciplinas com checkbox e visualização em tempo real	METIS e MatrUSP apresentam grade fixa e estática
Visualização semanal	Integração com FullCalendar usando eventos recorrentes (rrule)	METIS usa tabela estática com horários fixos
Verificação de conflito	Toast de erro com destaque da disciplina em choque	METIS apresenta alertas simples
Exportação em PDF	Exportação da grade horária via html2canvas e jsPDF	Ausente ou pouco explorado nos demais trabalhos
Exportação em ICS	Implementação completa com suporte a calendários digitais	Não encontrada nos sistemas analisados

Fonte: Elaborado pelo autor (2025).

### 3.5 Considerações sobre Trabalhos Relacionados

A partir da análise apresentada no Quadro 2, observa-se que o Stella se diferencia de outros sistemas em praticamente todos os recursos avaliados.

No quesito **simulação de matrícula**, enquanto soluções como METIS e MatrUSP utilizam grades fixas e estáticas, o Stella adota um mecanismo dinâmico de seleção com *checkbox*, permitindo ao aluno montar e modificar sua grade em tempo real, com retorno visual imediato. Essa abordagem favorece a experimentação e reduz o tempo necessário para encontrar a combinação ideal de disciplinas.

Quanto à **visualização semanal**, o Stella se beneficia da integração com a biblioteca FullCalendar e do uso de eventos recorrentes (*rrule*), garantindo flexibilidade e clareza na apresentação dos horários. Já sistemas como o METIS se limitam a tabelas estáticas, o que restringe a interação e a personalização.

Na **verificação de conflitos**, a solução proposta exibe notificações visuais (*toasts*) e destaca as disciplinas conflitantes diretamente na interface, permitindo ao usuário identificar rapidamente o problema. Em contraste, o METIS fornece apenas alertas simples, sem integração visual no calendário.

Outro diferencial é a **exportação da grade em PDF**, viabilizada pelas bibliotecas `html2canvas` e `jsPDF`, recurso ausente ou pouco explorado nos trabalhos relacionados. A **exportação em formato ICS** amplia ainda mais a interoperabilidade, pois permite a importação direta em calendários digitais como Google Calendar e Outlook — funcionalidade não encontrada nos sistemas analisados.

Esses resultados evidenciam que, embora existam iniciativas com propostas semelhantes, a implementação do Stella prioriza tanto a usabilidade quanto a integração com ferramentas externas, atendendo a necessidades que os outros sistemas ainda não contemplam de forma completa.

Por fim, optou-se pela utilização do MongoDB como banco de dados devido à familiaridade adquirida pelo autor em experiências anteriores, além de já ter sido adotado na primeira versão do projeto durante a disciplina de Projeto Integrado I.

## 4 METODOLOGIA

O método científico adotado para validação deste trabalho é o experimental, onde a solução proposta foi desenvolvida e sua adequação às necessidades dos usuários, ou público-alvo, é testada através de uma avaliação de usabilidade. Neste caso, as demais variáveis dos envolvidos no sistema analisado, tais como desempenho, disponibilidade, escalabilidade ou complexidade algorítmica, são ignoradas e apenas a variável a ser testada é destacada, no caso, a facilidade de uso do sistema.

Para o desenvolvimento da ferramenta, foram utilizadas diversas tecnologias que foram previamente apresentadas, garantindo que cada etapa do processo seja organizada e conduzida de maneira eficiente.

O Trello foi utilizado como uma ferramenta de gerenciamento de tarefas e planejamento do projeto. Essa plataforma permite a criação de cartões de tarefas, organização e acompanhamento das atividades através das colunas, que representam o *status* de cada atividade, assim, garantindo que todos os requisitos do projeto sejam cumpridos dentro dos prazos estabelecidos.

Na etapa de *design*, foi empregado o Figma para a criação e prototipação do *layout* da ferramenta. O Figma permite que o *design* seja visualizado e ajustado conforme necessário, assegurando que a interface do usuário seja intuitiva e esteja alinhada com as expectativas da aplicação final. Para esta etapa, foi utilizado o método *Design Thinking*, que propõe uma abordagem centrada no ser humano, voltada à solução de problemas de forma criativa e colaborativa (BROWN, 2010).

A aplicação do *Design Thinking* seguiu suas cinco etapas principais, adaptadas ao contexto deste projeto:

1. **Empatia:** foram analisadas as necessidades dos alunos do curso de Sistemas e Mídias Digitais por meio de conversas informais e revisão da primeira versão do Stella, identificando pontos de dificuldade na organização das disciplinas.
2. **Definição:** a partir das informações coletadas, definiu-se o problema central como a necessidade de uma interface clara, intuitiva e integrada, que permitisse organizar a grade horária de forma personalizada e detectar conflitos.
3. **Ideação:** foram levantadas possíveis soluções para o problema, incluindo novos recursos, melhorias na navegação e integração com formatos de exportação compatíveis com outros sistemas.

4. **Prototipação:** criação, no Figma, de telas representando as principais funcionalidades da ferramenta (seleção de disciplinas, visualização semanal do calendário, verificação de conflitos e exportação), servindo como base para o desenvolvimento em React.
5. **Teste:** após a finalização da programação a partir do protótipo, a ferramenta foi apresentada à alguns alunos do curso para validação e teste SUS.

Essa aplicação prática do *Design Thinking* permitiu que o desenvolvimento fosse conduzido de forma iterativa, incorporando o feedback dos usuários e garantindo que a solução final atendesse de maneira eficaz às demandas identificadas.

Para o desenvolvimento da funcionalidade, foi mantido o *framework* React JS, devido à sua flexibilidade e capacidade de construir interfaces de usuário interativas e dinâmicas. Dentro do React, foi integrado o componente Full Calendar, uma biblioteca popular e robusta desenvolvida em JavaScript, que facilitou a criação e gestão de eventos em formato de calendário. Isso é essencial para a funcionalidade da ferramenta, que exigiu a manipulação de horários e datas de forma eficiente.

A comunicação entre o *frontend* e o *backend* foi realizada por meio de uma *Application Programming Interface* (API) construída com Node.js seguindo o padrão arquitetural RESTful<sup>1</sup>. O Node.js foi escolhido por sua capacidade de lidar com múltiplas requisições simultâneas de forma eficiente, característica essencial para o bom funcionamento da ferramenta.

Para garantir o versionamento e o controle das diferentes versões do código ao longo do desenvolvimento, foi utilizado o Git. Essa ferramenta possibilita a colaboração entre os membros da equipe, bem como a reversão de alterações indesejadas, garantindo a integridade do código.

Por fim, todo o desenvolvimento foi realizado utilizando o *Integrated Development Environment* (IDE) *Visual Studio Code* (VS Code), que oferece uma série de extensões e ferramentas que facilitam a escrita e o *debugging* do código, tornando o processo de desenvolvimento mais produtivo e organizado.

Cada uma dessas ferramentas e tecnologias foi escolhida cuidadosamente para garantir que a aplicação proposta fosse desenvolvida de forma eficiente, funcional e dentro dos padrões de qualidade estabelecidos para o Projeto Stella, durante seu desenvolvimento.

---

<sup>1</sup> Refere-se a um conjunto de princípios arquiteturais utilizados no desenvolvimento de APIs para aplicações web. Para mais detalhes, ver o verbete “REST” da Wikipédia, disponível em: <<https://pt.wikipedia.org/wiki/REST>>.

## 5 DESENVOLVIMENTO

Este capítulo apresenta o desenvolvimento da ferramenta organizadora de grade horária integrada ao Projeto Stella. São descritas as tecnologias utilizadas, justificando suas escolhas frente a alternativas, os requisitos funcionais e não funcionais, o *design* da arquitetura de sistema (*frontend e backend*), o processo de prototipação, estratégias de teste, e por fim a implementação da ferramenta em si.

### 5.1 Contexto do Desenvolvimento

Para se iniciar a produção de uma ferramenta, é importante saber como está o código legado do projeto, e, claro, seu funcionamento. Para isso, foi inicialmente feita uma revisão do sistema por completo, verificando o código e suas devidas funcionalidades. Cada funcionalidade foi testada no ambiente de desenvolvimento para certificar-se de que o sistema estivesse funcionando devidamente para então planejar o desenvolvimento da nova funcionalidade.

Após os testes e verificação do código, foi realizada uma definição dos requisitos da nova funcionalidade, que será apresentado em um dos capítulos posteriores. Além disso, como o desenvolvimento da ferramenta foi feito utilizando o método "Kanban", as etapas de produção foram separadas por atividades de acordo com cada requisito a ser desenvolvido.

O conceito de "Kanban" está ligado a uma metodologia ágil de gerenciamento de projetos de software que busca conduzir cada tarefa por um fluxo predefinido de trabalho (EQUIPE TOTVS, 2023). Por sua vez, estas são organizadas utilizando-se de colunas e cartões, na qual as colunas servem para organizar o status da tarefa e os cartões indicam qual a tarefa, além de sua descrição, prazo, etc.

Para cada requisito funcional a ser desenvolvido foi criado um "cartão" no aplicativo Trello, uma ferramenta de gestão de projetos baseada em quadros virtuais, que ajuda na organização de tarefas, de forma visual e colaborativa (CHÉROLET, BRENDA, 2024), na qual foi posto o título de cada atividade, a descrição, dificuldade, desenvolvedor designado para realizar tal tarefa e prazo.

Inicialmente, todos os cartões criados ficavam concentrados na coluna chamada "Backlog", seguindo um fluxo de acordo com cada *status* da atividade. No total, eram utilizadas seis colunas, com *status* diferentes. A primeira, como já mencionado, concentrava todas as atividades criadas. A segunda, chamada "A fazer", indicava todas as atividades aptas a serem

iniciadas, ou seja, os cartões dessa coluna já estavam completos e contavam com informações bem definidas para o início do trabalho. A terceira, “Fazendo”, representava as atividades em execução. A quarta, “Revisão de Código”, sinalizava que o código estava sendo revisado antes da fase de testes. Em seguida, vinha a coluna “Teste”, que indicava que a funcionalidade estava sendo testada. Por fim, havia a coluna “Feito”, para onde o cartão era movido após o requisito ser implementado, revisado, testado e publicado.

Com isso, foram definidas as etapas de projeto, descritas nos tópicos a seguir.

### **5.1.1 Protótipo da Ferramenta**

A produção do protótipo da ferramenta representou um passo importante no projeto, pois serviu como modelo a ser seguido durante o desenvolvimento. O protótipo foi elaborado no Figma, uma plataforma colaborativa para construção de interfaces e protótipos, fundada por Dylan Field e Evan Wallace em 2016 (WIKIPÉDIA, 2025). Após a criação do protótipo, seguindo o padrão de *design* estabelecido no projeto, foram criadas novas atividades no Trello, seguindo o fluxo definido no Kanban, com o propósito de iniciar o desenvolvimento da ferramenta em código para então construir, testar e, posteriormente, integrar ao Projeto Stella. O protótipo da interface, criado no Figma, pode ser consultado no Apêndice A, assim como o repositório completo do código-fonte hospedado no GitHub.

O protótipo contemplou as principais telas e fluxos de interação esperados para a ferramenta. Isso incluiu: a seleção de disciplinas disponíveis no semestre, a visualização da grade horária sendo montada em tempo real, opções para remover ou ajustar disciplinas escolhidas, além do botão de exportação em PDF e o botão de exportação do calendário em formato digital (arquivo no formato ICS).

Após a finalização do protótipo, foi realizada uma validação interna por parte do autor e do orientador, com o objetivo de garantir a clareza da interface e a aderência às funcionalidades previamente definidas. A partir desse protótipo, iniciou-se a transição para a implementação do código, utilizando a *stack* tecnológica previamente descrita.

### **5.1.2 Desenvolvimento da Ferramenta**

A etapa de desenvolvimento da ferramenta foi realizada com base nos requisitos definidos e validados durante as fases anteriores, especialmente a prototipação. O código foi desenvolvido utilizando a biblioteca React JS para o *frontend*, que permite a criação de interfaces

dinâmicas e responsivas. Para o gerenciamento visual dos horários, foi utilizado o componente *FullCalendar*, que oferece suporte à visualização de eventos em formato semanal, facilitando a organização da grade horária pelos alunos.

No *backend*, foi empregada a plataforma Node.js, com a estruturação de uma API RESTful responsável por intermediar as requisições entre o *frontend* e o banco de dados. A persistência das informações foi feita por meio do banco de dados MongoDB, que possibilita flexibilidade no armazenamento e consulta de documentos, adaptando-se bem à natureza dinâmica das informações manipuladas.

Durante o desenvolvimento, o sistema de controle de versões Git foi utilizado para garantir o versionamento e a rastreabilidade das alterações no código. O repositório foi organizado de forma clara, com *branches* específicas para novas funcionalidades e correções, seguindo boas práticas de desenvolvimento colaborativo.

As funcionalidades da ferramenta foram implementadas de forma incremental, priorizando-se inicialmente as funções principais: seleção de trilhas e disciplinas, visualização e edição da grade horária e exportação do resultado em PDF e formato ICS. Após a implementação das funcionalidades básicas, foram realizados testes funcionais para garantir que cada parte da aplicação estivesse funcionando conforme o esperado.

Cada atividade teve como objetivo entregar uma parte funcional do sistema, permitindo a validação contínua do que estava sendo desenvolvido. A comunicação entre *frontend* e *backend* foi constantemente testada, visando assegurar a integridade dos dados trafegados e a consistência da experiência do usuário.

Ao final do desenvolvimento, foi realizada uma rodada completa de testes e ajustes, preparando a ferramenta para sua integração definitiva ao sistema principal.

### **5.1.3 Integração ao Sistema Stella**

Por fim, após a produção do código e diversos testes feitos pelo próprio autor deste trabalho, como: teste do filtro de seleção de disciplina, teste de busca por texto, teste de exportação do calendário semanal em (PDF), etc., a ferramenta foi integrada totalmente ao Stella, sendo criado e testado uma rota na aplicação com acesso à ferramenta, permitindo ser utilizada pelos usuários do aplicativo.

## 5.2 Tecnologias Utilizadas

As tecnologias utilizadas foram escolhidas com base em critérios como popularidade, suporte da comunidade, curva de aprendizado, integração com outras ferramentas e aderência aos requisitos do projeto.

### 5.2.1 Frontend: React.js

O *React.js* foi a biblioteca escolhida para o desenvolvimento da interface do usuário. Trata-se de uma biblioteca JavaScript criada pelo Facebook que permite a criação de interfaces reativas por meio de componentes reutilizáveis.

Além das vantagens como componentização, *Virtual DOM* e comunidade ativa, foram utilizadas bibliotecas auxiliares como *React Bootstrap* para a estilização, *html2canvas* e *jsPDF* para exportação da grade em PDF, e integração com *FullCalendar* para a visualização da grade semanal.

### 5.2.2 Componente de Calendário: FullCalendar

Para a visualização e manipulação de eventos em uma grade semanal, foi adotado o *FullCalendar*, uma biblioteca JavaScript amplamente utilizada em aplicações de calendário.

Segundo o site oficial do *FullCalendar*<sup>1</sup>, a biblioteca oferece suporte para eventos recorrentes, personalização visual, integração com React e exportação por meio do padrão ICS.

Foram utilizadas configurações específicas como *rrule* e *duration* para indicar eventos recorrentes com durações reais (como 2h ou 4h), respeitando os horários reais de cada disciplina.

### 5.2.3 Backend: Node.js com Express

O *backend* foi implementado utilizando *Node.js* e o *framework Express*, adotando o padrão arquitetural MVC, que até então não era utilizado no projeto. Os controladores (*controllers*) centralizam a lógica de negócio, as rotas (*routes*) organizam os *endpoints*, e os modelos (*models*) foram definidos com *Mongoose* para mapear os dados no banco.

Foram criados/reformulados endpoints REST para:

- Autenticação e registro de usuários (reformulado);
- Listagem e cadastro de disciplinas por semestre (criado);

---

<sup>1</sup> <<https://fullcalendar.io/>>. Acesso em: 07 de julho de 2025.

- Cadastro em lote de disciplinas usando `insertMany()` (criado);

#### 5.2.4 Banco de Dados: MongoDB

O banco de dados adotado foi o MongoDB, acessado por meio da plataforma *MongoDB Atlas*, que oferece hospedagem em nuvem com alta disponibilidade e escalabilidade. A escolha por um banco NoSQL se deu devido à sua flexibilidade na modelagem de dados, especialmente relevante para este projeto, que exige o armazenamento de disciplinas com múltiplos horários e variações de configuração por trilha.

A modelagem utilizou *arrays* embutidos para os horários de cada disciplina, reduzindo a necessidade de múltiplas coleções e facilitando as operações de leitura e escrita. Para facilitar a integração com o *backend*, foi utilizado o *Object Data Modeling* (ODM) *Mongoose*, que possibilitou a definição de *schemas* com validações automáticas, consultas mais otimizadas e maior consistência dos dados. Essa abordagem reduziu a probabilidade de erros na camada de persistência e simplificou a manutenção do código.

#### 5.2.5 Controle de Versão: Git

O controle de versão do código foi realizado com o Git, uma ferramenta amplamente consolidada no desenvolvimento de software colaborativo. Os repositórios do Stella foram hospedados no GitHub (tanto o *frontend* quanto o *backend*), o que proporcionou integração com *pull requests*, controle de histórico e visualização de alterações.

As funcionalidades foram desenvolvidas em *branches* específicas, seguindo boas práticas de *feature branching*. Esse método facilitou a organização e a rastreabilidade das mudanças, permitindo que novas implementações ou correções fossem testadas de forma isolada antes de serem integradas à *branch* principal. Além disso, essa estratégia garantiu maior segurança no fluxo de desenvolvimento, prevenindo a introdução de falhas em versões estáveis.

#### 5.2.6 IDE e Ferramentas Auxiliares

A IDE utilizada foi o *Visual Studio Code* (VS Code), escolhida por sua leveza, ampla biblioteca de extensões e integração nativa com o Git. Foram empregadas extensões como ESLint e Prettier para padronização e formatação automática do código, GitLens para melhor visualização do histórico e contribuições no repositório, além de suporte a *snippets* para React,

acelerando a escrita de componentes.

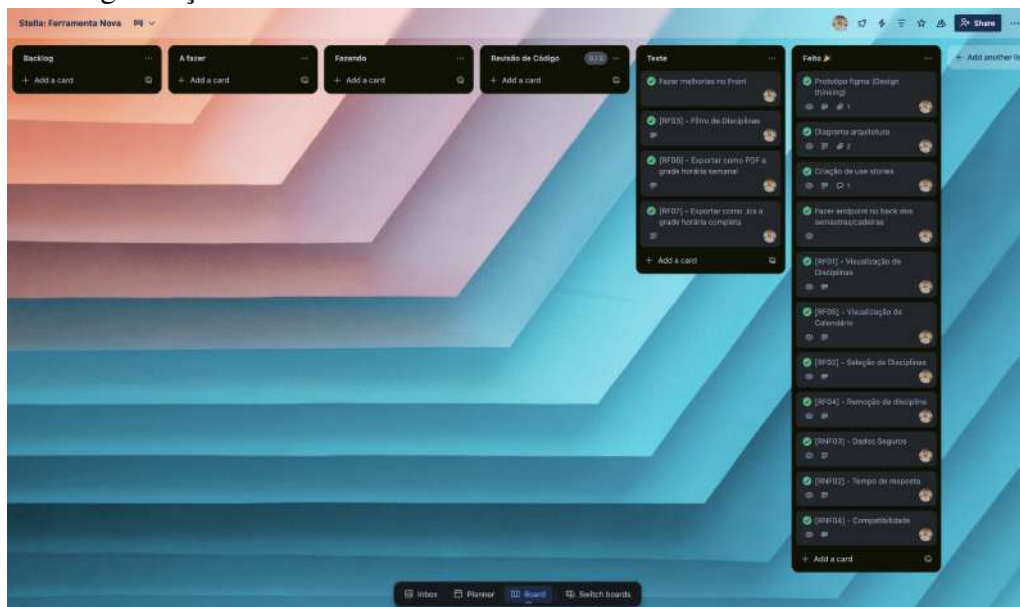
### 5.3 Gerenciamento das Atividades com Trello

Durante o desenvolvimento da ferramenta, foi utilizado o Trello para organização das tarefas com base no método Kanban. As atividades foram organizadas em colunas como: "Backlog", "A fazer", "Fazendo", "Revisão de Código", "Teste" e "Feito".

Essa abordagem visual proporcionou uma visão clara do progresso do projeto, facilitando a priorização das funcionalidades e o acompanhamento das etapas de desenvolvimento de forma estruturada. Com isso, foi possível manter uma organização eficaz do fluxo de trabalho, mesmo sendo um projeto individual.

Além disso, o Trello permitiu registrar comentários, prazos e anotações relevantes para cada tarefa, contribuindo para o planejamento e a execução autônoma de todas as fases do projeto.

Figura 4 – Organização de Atividades no *Trello*



Fonte: Elaborado pelo autor (2025).

Cada cartão representava uma tarefa específica, como "Criação de User Stories" ou "RF01 - Visualização de disciplinas". Essa organização visual permitiu acompanhar o progresso e ajustar prioridades.

## 5.4 Requisitos do Sistema

As sub-seções a seguir apresentam os requisitos funcionais e não funcionais necessários para atingir o objetivo principal deste trabalho, que é a produção e integração da ferramenta organizadora de grade horária no sistema Stella.

### 5.4.1 *Requisitos Funcionais*

- RF01 – Permitir ao aluno selecionar as disciplinas disponíveis no semestre.
- RF02 – Exibir uma grade horária semanal com as disciplinas selecionadas.
- RF03 – Remover disciplinas da grade.
- RF04 – Permitir filtrar disciplinas por nome, código de disciplina, nome da disciplina, trilha ou professor.
- RF05 – Exportar a grade final como arquivo PDF.
- RF06 – Exportar a grade final como arquivo ICS.
- RF07 – Impedir seleção de disciplinas com choque de horários.

### 5.4.2 *Requisitos Não Funcionais*

- RNF01 – Armazenar dados com segurança, seguindo a Lei Geral da Proteção de Dados (LGPD).
- RNF02 – Ser compatível com os principais navegadores modernos.

## 5.5 Casos de Usuário

Os casos de usuário a seguir descrevem, em linguagem simples, as interações previstas entre os usuários e o sistema Stella, baseados nos requisitos funcionais levantados.

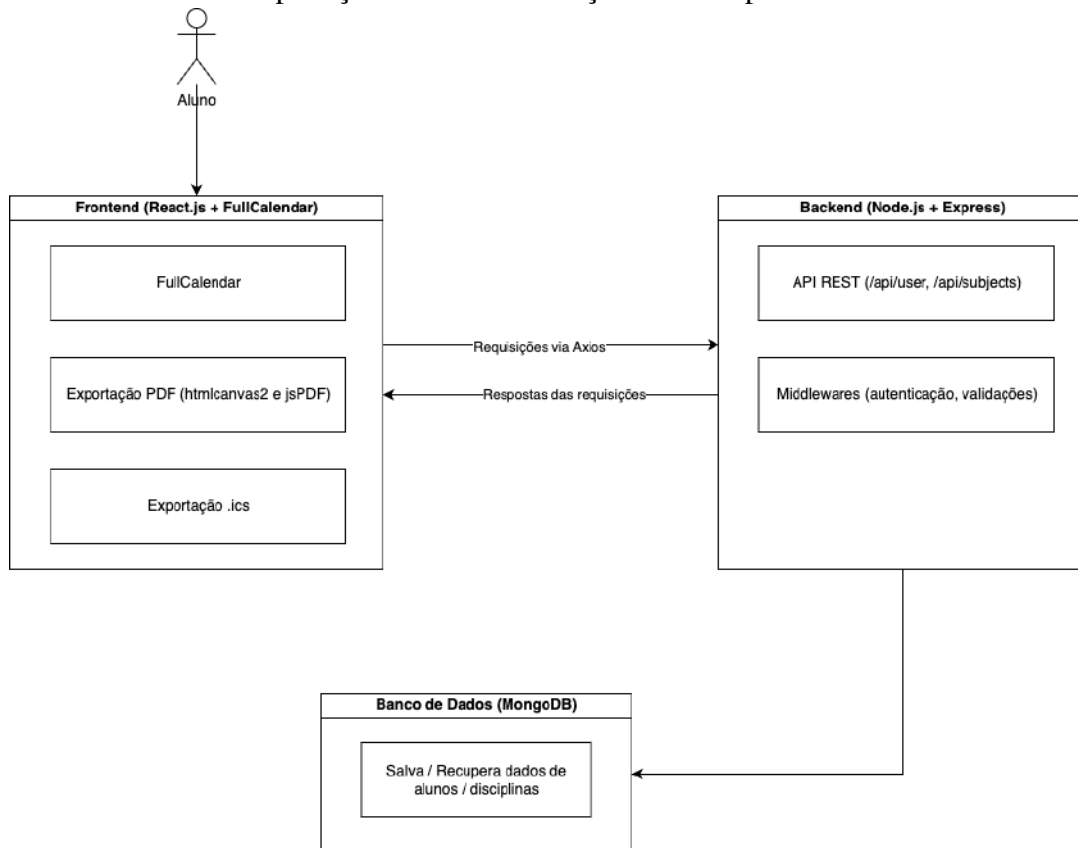
- **CU01 – Seleção de Disciplinas:** Como estudante, desejo visualizar e selecionar disciplinas ofertadas no semestre para montar minha grade horária.
- **CU02 – Filtro por Trilha ou Nome:** Como estudante, desejo filtrar as disciplinas por código de disciplina, nome da disciplina, trilha, nome ou professor para facilitar a busca pelas opções do meu interesse.
- **CU03 – Verificação de Conflito:** Como estudante, desejo ser alertado em caso de conflito de horários entre disciplinas ao montar minha grade.

- **CU04 – Exportação da Grade:** Como estudante, desejo exportar minha grade em formato PDF ou ICS para salvar ou importar no meu calendário pessoal.
- **CU05 – Visualização Detalhada:** Como estudante, desejo acessar informações detalhadas sobre cada disciplina (carga horária, descrição, professor) antes de adicioná-la à grade.
- **CU06 – Exclusão de Disciplinas:** Como estudante, desejo poder remover disciplinas adicionadas à grade a qualquer momento, para refinar minha simulação.
- **CU07 – Proteção de Dados:** Como usuário, desejo que minhas informações estejam protegidas durante a navegação e uso do sistema.

## 5.6 Arquitetura da Aplicação e Estrutura do Calendário

A arquitetura da aplicação foi planejada e desenvolvida de forma individual, com base em uma estrutura dividida em três camadas principais: *frontend*, *backend* e banco de dados. Essa separação favorece a manutenção, escalabilidade e organização do projeto.

Figura 5 – Estrutura da aplicação com demonstração dos componentes da ferramenta



Fonte: Elaborado pelo autor (2025).

A comunicação entre o *frontend* e o *backend* é realizada por meio de requisições

HTTPS assíncronas utilizando a biblioteca Axios. Já a comunicação entre o *backend* e o banco de dados ocorre por meio do *driver* oficial do MongoDB para Node.js, instalado via NPM.

Essa arquitetura modular permitiu o desenvolvimento organizado e coerente da aplicação, possibilitando a integração fluida entre as diferentes camadas da solução proposta.

### 5.6.1 API RESTful e Comunicação com o Frontend

O *backend* da ferramenta foi desenvolvido seguindo o paradigma RESTful, com rotas organizadas para facilitar a manutenção e evolução do sistema. A API foi responsável por fornecer ao *frontend* todas as informações necessárias, como a lista de disciplinas, trilhas formativas e dados de autenticação dos usuários.

As principais rotas implementadas foram:

- **POST /auth/register**: cadastro de novos usuários com criptografia de senha via bcrypt;
- **POST /auth/login**: autenticação de usuários com geração de JWT;
- **GET /subject**: listagem de todas as disciplinas cadastradas;
- **POST /subject**: criação de novas disciplinas;
- **PUT /subject/:id**: edição de disciplinas existentes;
- **DELETE /subject/:id**: remoção de disciplinas.

As respostas da API seguem o formato JSON, o que facilita a integração com o *frontend* React. O exemplo abaixo ilustra o retorno da rota de listagem de disciplinas:

Código-fonte 1 – Exemplo de resposta JSON da rota GET /subject

```
1  [
2    {
3      "_id": "665a71d6f1f457acbe5b25fa",
4      "name": "Design de Interface",
5      "track": "Design Digital Interativo",
6      "subjectCode": "SMD101",
7      "teacher": "Prof. João Silva",
8      "credits": 4,
9      "workload": "64h",
10     "description": "Disciplina sobre princípios e técnicas de design de interfaces digitais.",
```

```

11     "semester": 3,
12     "startDate": "2025-08-05T00:00:00.000Z",
13     "endDate": "2025-12-15T00:00:00.000Z",
14     "schedule": [
15       { "day": "Segunda-feira", "startTime": "08:00", "
           endTime": "10:00" },
16       { "day": "Quarta-feira", "startTime": "10:00", "
           endTime": "12:00" }
17     ]
18   }
19 ]

```

Essa organização permite que o *frontend* consuma os dados e renderize dinamicamente as disciplinas no componente `SubjectSelector.jsx`, que será apresentado posteriormente.

## 5.7 Modelagem de Dados no MongoDB

O banco de dados utilizado no projeto foi o MongoDB, estruturado de forma não relacional e orientado a documentos. Para mapear os dados entre a aplicação Node.js e o banco, foi utilizado o pacote `mongoose`, que permite a definição de esquemas e validações para os documentos.

### 5.7.1 Coleção *users*

A coleção `users` armazena os dados dos usuários registrados na aplicação.

- `name (String)`: Nome completo do usuário.
- `email (String)`: Endereço de e-mail (único).
- `password (String)`: *Hash* da senha.

### 5.7.2 Coleção *subjects*

A coleção `subjects` armazena as disciplinas ofertadas por semestre e trilha.

- `name (String)`: Nome da disciplina.

- `subjectCode` (*String*): Código da disciplina no sistema.
- `track` (*String*): Nome da trilha formativa à qual a disciplina pertence.
- `teacher` (*String*): Nome do professor responsável.
- `semester` (*Number*): Número do semestre em que a disciplina é ofertada.
- `startDate` (*String*): Data de início do semestre (formato ISO AAAA-MM-DD).
- `endDate` (*String*): Data de término do semestre (formato ISO AAAA-MM-DD).
- `schedule` (*Array*): Lista de horários da disciplina, contendo:
  - `day` (*String*): Dia da semana em que a disciplina ocorre (ex: Segunda-feira).
  - `timeCode` (*String*): Código do horário no sistema (ex: M34, T12).
  - `startTime` (*String*): Horário de início no formato HH:MM.
  - `endTime` (*String*): Horário de término no formato HH:MM.
- `credits` (*Number*): Quantidade de créditos atribuídos à disciplina.
- `workload` (*String*): Carga horária total da disciplina (ex: 64h, 96h).
- `description` (*String*): Breve descrição do conteúdo programático da disciplina.

### 5.7.3 Considerações sobre a Modelagem

A modelagem foi pensada para ser extensível e flexível, permitindo a adição de novos campos (como turma, sala ou tipo de atividade) sem necessidade de alterações estruturais complexas, o que é uma das vantagens do uso de um banco NoSQL.

Além disso, foram aplicadas boas práticas como uso de campos normalizados (nome, trilha, horários), validações no *schema* com o Mongoose, e inserções em lote com o método `insertMany()` para popular o banco com múltiplas disciplinas.

## 5.8 Prototipação da Ferramenta

A prototipação da ferramenta foi realizada no *Figma*, uma das ferramentas mais populares para criação de protótipos entre os *UX Designers* (RICHARD, 2025). O uso do Figma permitiu o planejamento visual das funcionalidades da ferramenta antes da sua implementação, facilitando o alinhamento com os objetivos do projeto e proporcionando uma base sólida para as decisões de desenvolvimento *frontend*.

Durante essa etapa, foram desenhadas todas as principais telas da aplicação, com foco na usabilidade, clareza de informações e hierarquia visual. Cada tela foi pensada para guiar

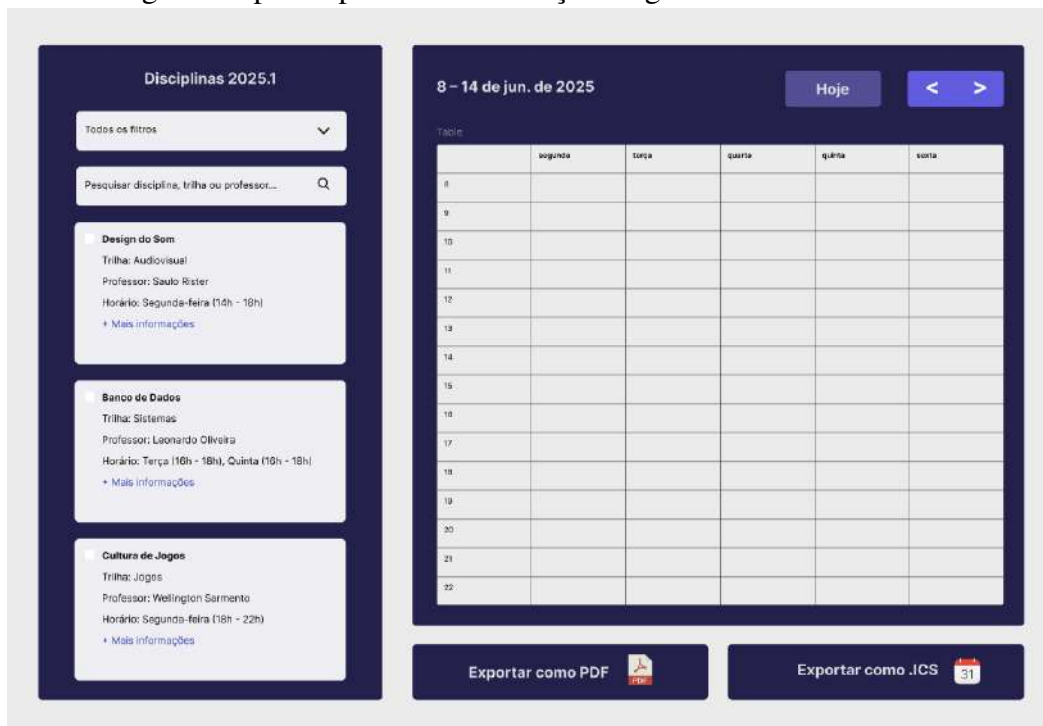
o estudante durante o processo de seleção de disciplinas e montagem da grade horária.

As principais decisões tomadas no protótipo foram:

- Adição de filtros por código de disciplina, nome da disciplina, trilha e professor para facilitar a busca;
- Uso de caixas de seleção (*checkboxes*) para escolha das disciplinas;
- Visualização da grade semanal em formato de calendário com cores no padrão do sistema;
- Inclusão de *toasts* e modais para *feedback* de ações, como erro de conflito ou exibição de detalhes da disciplina;
- Posicionamento estratégico dos botões de exportação em PDF e ICS;

A Figura 6 apresenta a visualização geral da grade horária no formato de calendário semanal. Esse *layout* foi planejado para facilitar a compreensão dos horários ocupados e permitir que o aluno identifique possíveis conflitos de forma visual e intuitiva. A tela serviu como base para o desenvolvimento da programação e também inspirou outras funcionalidades que foram sendo incorporadas ao longo das etapas de prototipação e testes.

Figura 6 – Tela geral do protótipo com visualização da grade horária



Fonte: Elaborado pelo autor (2025).

A Figura 7 representa a exibição de uma notificação de erro (*toast*) indicando conflito entre disciplinas selecionadas. Essa funcionalidade proporciona *feedback* instantâneo ao usuário e permite que o usuário tenha ciência de quais disciplinas estão com conflitos.

Figura 7 – Exemplo de conflito de horários representado com *toast* de erro

The screenshot displays a course schedule interface. On the left, a sidebar titled "Disciplinas 2025.1" lists three courses: "SMD001 - Design do Som", "SMD002 - Banco de Dados", and "SMD003 - Cultura de Jogos". The main area shows a calendar for "8 - 14 de jun. de 2025". A conflict is highlighted with a purple "Conflito de horários" toast message: "Conflito detectado entre 'Banco de Dados' e 'Design do Som'". The calendar grid shows two overlapping class blocks for "SMD002 - Banco de Dados" on Wednesday and Friday. At the bottom, there are buttons for "Exportar como PDF" and "Exportar como .ICS".

Fonte: Elaborado pelo autor (2025).

A Figura 8 mostra a interface com a funcionalidade de exportação para arquivos no formato ICS, permitindo que os alunos importem sua grade para aplicativos como Google Calendar ou Outlook.

Figura 8 – Botão e feedback da exportação da grade no formato ICS

This screenshot shows the same course schedule interface as Figure 7, but with a successful export. A purple "Download iniciado!" toast message is displayed: "Seu calendário está sendo baixado em formato .ICS e em breve estará disponível para ser importado no seu calendário digital favorito!". The calendar grid and sidebar are identical to the previous figure. The "Exportar como .ICS" button is now highlighted with a calendar icon showing the date 31.

Fonte: Elaborado pelo autor (2025).

A Figura 9 mostra a opção de exportar a grade horária em formato PDF.

Figura 9 – Botão e feedback da exportação da grade em PDF

The screenshot displays a user interface for course management. On the left, a sidebar titled 'Disciplinas 2025.1' lists three courses: 'SMD001 - Design do Som', 'SMD002 - Banco de Dados', and 'SMD003 - Cultura de Jogos'. Each course entry includes its category, professor, and schedule. On the right, a weekly grid for '8 - 14 de jun. de 2025' shows the schedule for each day. A notification at the top right states: 'Download (Incluído) Sua grade horária semanal está sendo baseada em formato PDF e estará disponível em breve!'. At the bottom, there are two buttons: 'Exportar como PDF' and 'Exportar como ICS'.

Fonte: Elaborado pelo autor (2025).

A Figura 10 apresenta um *modal* ao clicar no botão de título “Mais informações”.

Figura 10 – Exemplo de *modal* com informações detalhadas da disciplina

This screenshot shows the same interface as Figure 9, but with a modal window open for the 'SMD001 - Design do Som' course. The modal contains the following details:
 

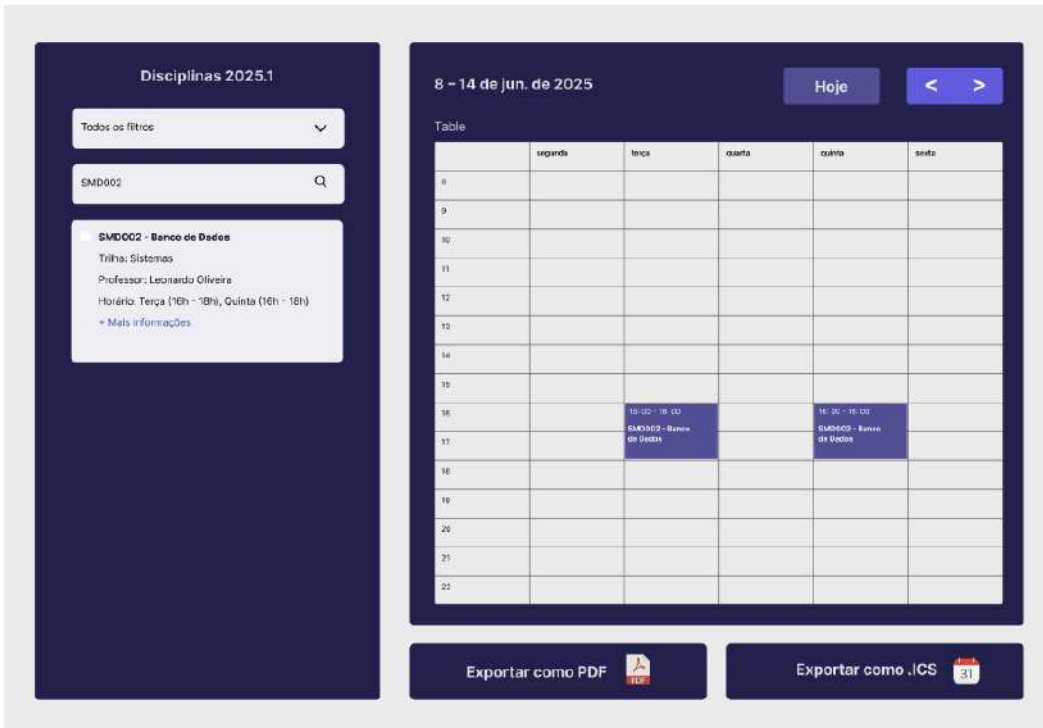
- Título:** SMD001 - Design do Som
- Triha:** Audiovisual
- Professor:** Saulo Rister
- Carga Horária:** 64h
- Créditos:** 4
- Descrição:** Explore o conteúdo técnico sobre o som e ferramentas digitais sonoras para edições (DAW).
- Horários:** Terça-feira - 18:00 às 19:00, Quinta-feira - 16:00 às 19:00

 The background interface is dimmed, and the modal has a close button (X) in the top right corner.

Fonte: Elaborado pelo autor (2025).

A Figura 11 mostra a funcionalidade de busca textual.

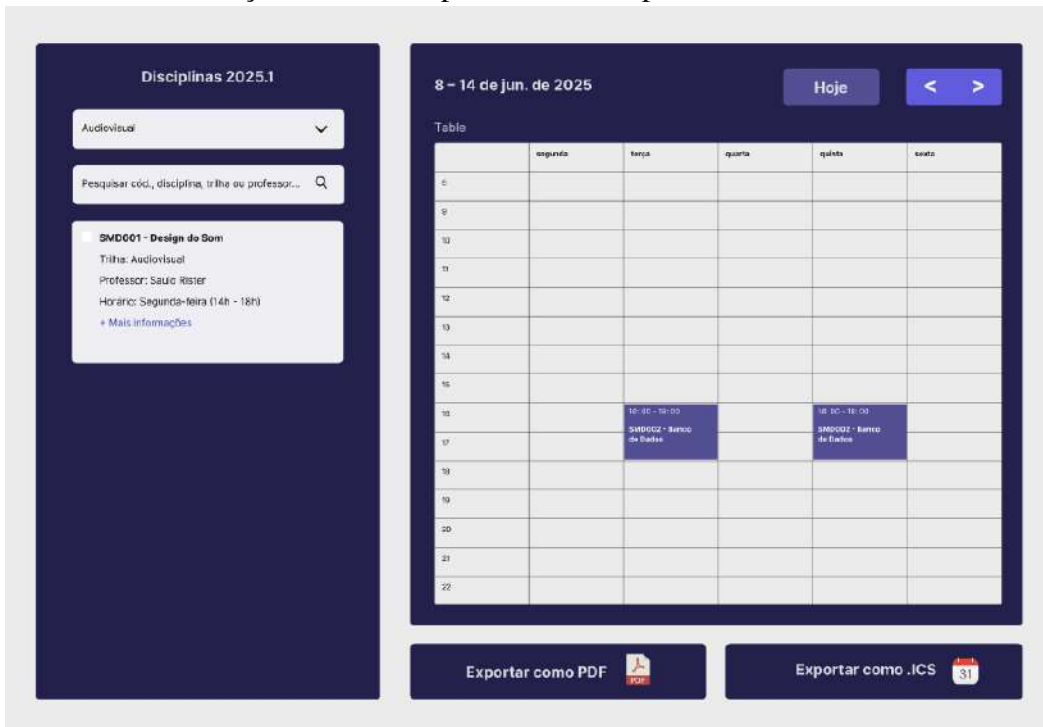
Figura 11 – Interface com campo de busca e filtro de disciplinas



Fonte: Elaborado pelo autor (2025).

A Figura 12 demonstra a aplicação de um dos filtros de seleção por trilha formativa.

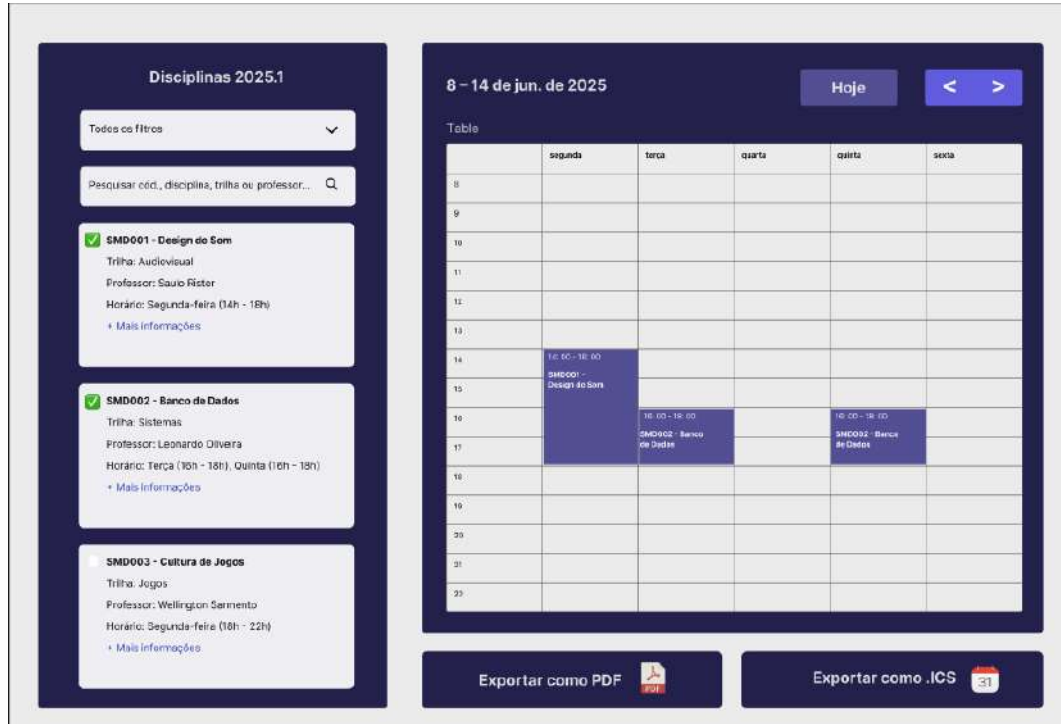
Figura 12 – Tela de seleção com filtros por trilha e campos dinâmicos



Fonte: Elaborado pelo autor (2025).

A Figura 13 mostra como as disciplinas são exibidas com *checkboxes*. O aluno pode selecionar e adicionar à sua grade, visualizando as alterações em tempo real.

Figura 13 – Tela com disciplinas disponíveis e *checkboxes* de seleção



Fonte: Elaborado pelo autor (2025).

Por fim, o protótipo feito no Figma serviu como guia visual durante toda o desenvolvimento e implementação da ferramenta, promovendo consistência entre a proposta inicial e a versão final.

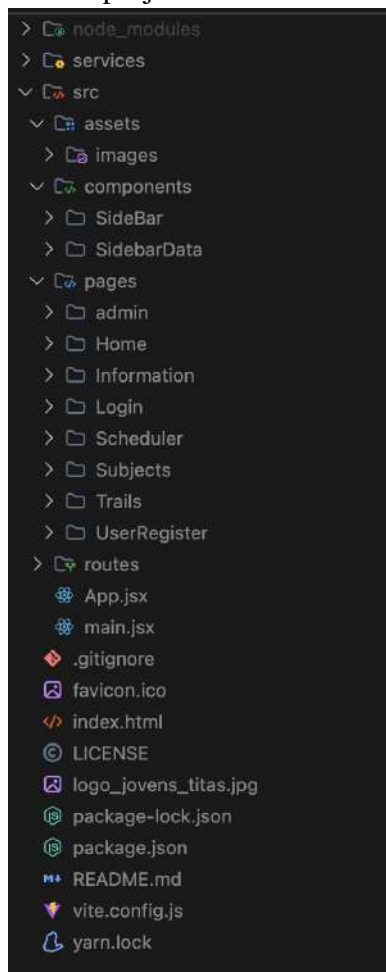
## 5.9 Detalhes de Implementação

Esta seção apresenta exemplos práticos do código-fonte e da estrutura do projeto Stella, bem como o que foi implementado com a integração da ferramenta, com o objetivo de demonstrar de forma clara como a ferramenta foi implementada tecnicamente.

### 5.9.1 Estrutura de Pastas do Projeto

A Figura 14 ilustra a estrutura de diretórios do projeto no *frontend*, cuidadosamente organizada para garantir uma separação clara de responsabilidades entre os diferentes módulos da aplicação. Essa organização foi essencial para manter o código limpo, facilitar o trabalho e possibilitar futuras expansões com menor risco de introdução de erros.

Figura 14 – Estrutura de diretórios do projeto React



Fonte: Elaborado pelo autor (2025).

A organização do *frontend* foi estruturada com base no padrão *feature-based folders*, que propõe a divisão do código em pastas agrupadas por funcionalidades ou domínios da aplicação, ao invés de agrupar por tipo técnico (como componentes, serviços, etc). Esse padrão é amplamente utilizado em projetos React modernos por favorecer a escalabilidade, o isolamento de responsabilidades e a facilidade de manutenção do código (RASOULI, 2025).

Entre os principais benefícios da arquitetura baseada em *features* estão:

- **Isolamento funcional:** cada funcionalidade possui seus próprios componentes, serviços e lógica, reduzindo acoplamentos e tornando o código mais modular;
- **Escalabilidade:** novas funcionalidades podem ser adicionadas sem afetar diretamente outras partes do sistema;
- **Facilidade de navegação:** ao manter todos os arquivos relacionados em um único diretório, desenvolvedores conseguem localizar e modificar o código com mais rapidez;
- **Reuso e coesão:** elementos específicos de uma feature não ficam espalhados em pastas

genéricas, o que contribui para a organização (Feature-Sliced Design, 2025).

A estrutura de diretórios do projeto está representada na Figura 14. Nela, destacam-se as seguintes pastas:

- **/components:** Contém os componentes reutilizáveis da interface, como a barra lateral (`SideBar.jsx`), modais e botões personalizados. Esses elementos são utilizados em diversas páginas e possuem estilo próprio encapsulado para evitar conflitos.
- **/pages:** Cada subpasta representa uma página ou rota da aplicação. Por exemplo, a pasta `/Scheduler` contém todos os arquivos relacionados à ferramenta de grade horária, incluindo o componente principal (`index.jsx`), o seletor de disciplinas (`SubjectSelector.jsx`) e o calendário (`CalendarView.jsx`).
- **/services:** Reúne as funções responsáveis pela comunicação com a API *backend*. É neste diretório que estão centralizadas chamadas HTTPS feitas com Axios, como a função `getAllSubjects()`, que retorna a lista de disciplinas cadastradas no banco de dados.
- **/assets:** Armazena imagens, ícones e estilos CSS globais utilizados na aplicação.
- **/routes:** Define as rotas protegidas e públicas da aplicação usando o *React Router*, permitindo o controle de acesso com base na autenticação do usuário.
- **index.js:** Arquivo raiz que inicializa o React, renderiza o componente principal (`App.jsx`) e aplica configurações globais como o `BrowserRouter` para gerenciamento das rotas.

Essa organização está alinhada às boas práticas de desenvolvimento frontend e reflete um padrão robusto para aplicações modernas em React (ADAMAKIS, 2023).

### 5.9.2 Componente de Seleção de Disciplinas

A listagem de disciplinas disponíveis é feita no componente `SubjectSelector.jsx`, responsável por exibir as disciplinas com *checkbox*, filtro por trilha ou pesquisa por nome, código da disciplina, trilha ou professor e o botão de detalhes, presente nos *cards* de cada disciplina.

Este componente realiza uma requisição ao *backend* utilizando a biblioteca Axios para buscar todas as disciplinas cadastradas no sistema. Enquanto a resposta da API está sendo aguardada, um indicador visual de carregamento (*loader*) é exibido, proporcionando feedback ao usuário e melhorando a experiência durante o tempo de espera.

Ao receber os dados, o componente processa a lista de disciplinas e monta dinamicamente os elementos visuais correspondentes, atualizando o estado interno com as informações recebidas. A interface permite que o usuário realize múltiplas seleções por meio dos *checkboxes*,

além de aplicar filtros para facilitar a busca, seja pelo nome da disciplina, pelo código, pela trilha a que pertence ou pelo professor responsável.

O botão de detalhes, presente em cada card, possibilita a visualização ampliada das informações específicas daquela disciplina, permitindo uma análise mais profunda sem sair da tela principal. Essa interação é implementada utilizando estados locais para controlar a exibição condicional dos detalhes, garantindo uma navegação fluida e intuitiva.

#### Código-fonte 2 – Trecho do componente SubjectSelector com filtros

```
1  useEffect(() => {
2    setLoading(true);
3    getAllSubjects()
4      .then(setSubjects)
5      .finally(() => setLoading(false));
6  }, []);
7
8  <Form.Check
9    type="checkbox"
10   label={<strong>{subject.name}</strong>}
11   checked={selectedSubjects.some(s => s._id === subject._id
12     )}
13   onChange={e => onSelect(subject, e.target.checked)}
14 />
15 <Button
16   variant="link"
17   size="sm"
18   onClick={() => onOpenDetails(subject)}
19 >
20   + Mais informacoes
21 </Button>
```

A Figura 15 mostra a interface renderizada do componente.

Figura 15 – Interface da lista de disciplinas com filtros



Fonte: Elaborado pelo autor (2025).

### 5.9.3 Calendário de Visualização da Grade

A grade semanal de disciplinas é exibida utilizando o componente *FullCalendar*, configurado com suporte a eventos recorrentes. A biblioteca *rrule* foi utilizada para representar aulas repetidas semanalmente durante o semestre.

#### Código-fonte 3 – Configuração de eventos recorrentes com FullCalendar

```

1 events.push({
2   title: `${entry.startTime} - ${entry.endTime}\n${subject.name}`,
3   rrule: {
4     freq: 'weekly',
5     byweekday: [dayNum], // Segunda-feira = MO, etc.
6     dtstart: `${subject.startDate}T${entry.startTime}:00`,
7     until: `${subject.endDate}T23:59:59`
8   },

```

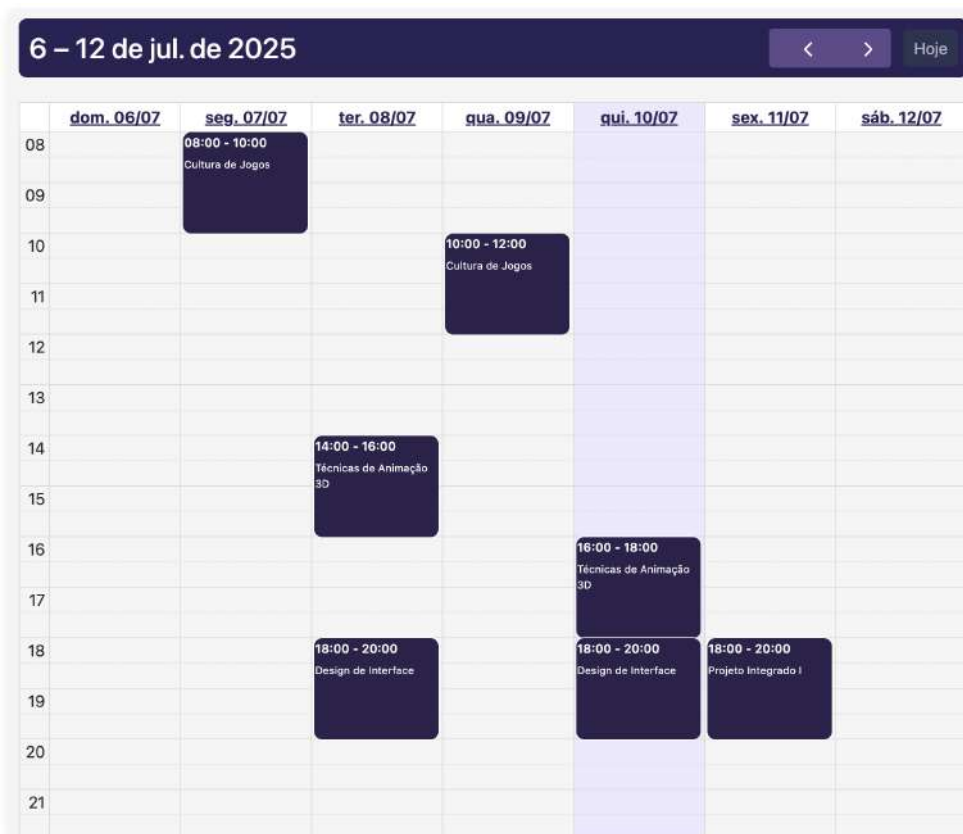
```

9   duration: getDuration(`${entry.startTime}:00`, `${entry.
    endTime}:00`),
10  });

```

Cada disciplina adicionada à grade gera um ou mais eventos no calendário. Essa configuração permite que o *FullCalendar* exiba visualmente todas as aulas de forma organizada.

Figura 16 – Exemplo do calendário preenchido com disciplinas



Fonte: Elaborado pelo autor (2025).

#### 5.9.4 Verificação de Conflito de Horário

Para evitar sobreposição de horários, foi implementada a função `overlaps()`, responsável por comparar os intervalos de tempo das disciplinas já selecionadas e identificar conflitos.

Código-fonte 4 – Função para verificação de conflitos de horário

```

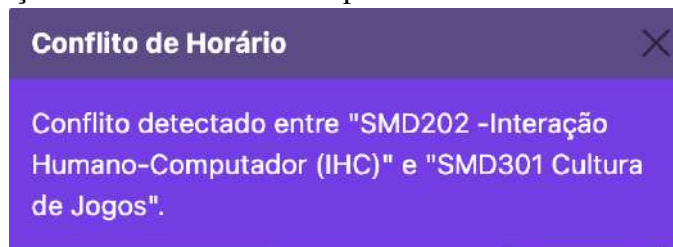
1 function overlaps(start1, end1, start2, end2) {
2   return (start1 < end2) && (start2 < end1);

```

3 }

Caso seja detectado um conflito, uma mensagem (*toast*) é exibida informando o nome das disciplinas em choque. Dessa forma, o usuário fica ciente de que não é possível selecionar disciplinas com choque de horário, além disso, quando esse choque acontece, a disciplina que foi selecionada primeira fica marcada com *checkbox* e com sua visualização no calendário.

Figura 17 – Notificação de conflito entre disciplinas



Fonte: Elaborado pelo autor (2025).

### 5.9.5 Modal de Visualização Detalhada

O botão *Mais informações* exibe um *modal* com dados completos da disciplina, como carga horária, créditos, docente responsável e horários.

Código-fonte 5 – Modal

```

1      <Modal show={showModal} onHide={closeModal}>
2          <Modal.Header closeButton>
3              <Modal.Title>{modalSubject?.subjectCode} - {
4                  modalSubject?.name}</Modal.Title>
5          </Modal.Header>
6          <Modal.Body>
7              <p><strong>Trilha:</strong> {modalSubject?.track
8                  }</p>
9              <p><strong>Professor:</strong> {modalSubject?.
10                 teacher}</p>
11             <p><strong>Carga Horária:</strong> {modalSubject
12                 ?.workload}</p>

```

```

9      <p><strong>Cr ditos:</strong> {modalSubject?.
      credits}</p>
10     <p><strong>Descri o:</strong> {modalSubject?.
      description}</p>
11     <p><strong>Hor rios:</strong><br />
12     {modalSubject?.schedule.map((s, i) => (
13     <div key={i}>
14     {s.day} {s.startTime} s {s.endTime}
15     </div>
16     ))}
17     </p>
18     </Modal.Body>
19     </Modal>

```

É possível sair do modal clicando fora dele (em qualquer outra parte da tela) ou clicando no botão "X", disponível na lateral superior direita do *modal*.

Figura 18 – Modal com informações detalhadas da disciplina



Fonte: Elaborado pelo autor (2025).

### 5.9.6 Exportação em PDF da Grade

A ferramenta permite exportar a grade visualizada no calendário para um arquivo PDF. A funcionalidade foi construída utilizando *html2canvas* para capturar o DOM do calendário como imagem e *jsPDF* para gerar o arquivo PDF.

Código-fonte 6 – Exportação da grade em PDF

```

1  const calendar = document.querySelector('.fc');
2  html2canvas(calendar, { scale: 2, useCORS: true }).then((
   canvas) => {
3    const imgData = canvas.toDataURL('image/png');
4    const pdf = new jsPDF('landscape');
5    pdf.addImage(imgData, 'PNG', 10, 10, 280, 160);
6    pdf.save('grade_horaria.pdf');
7  });

```

Figura 19 – Botão de exportação em PDF



Fonte: Elaborado pelo autor (2025).

### 5.9.7 Exportação para Arquivo no Formato ICS

A ferramenta também exporta a grade para arquivos no formato ICS, compatíveis com Google Calendar, Apple Calendar, Outlook e etc. Cada disciplina é representada como um evento com regra de repetição (*RRULE*).

Código-fonte 7 – Evento recorrente no padrão do formato ICS

```

1  BEGIN:VEVENT
2  UID: 1dsad212as
3  SUMMARY:Banco de Dados
4  DTSTART;TZID=America/Fortaleza:20250812T140000
5  DTEND;TZID=America/Fortaleza:20250812T160000

```

```
6 RRULE:FREQ=WEEKLY;BYDAY=TU,TH;UNTIL=20251201T235900Z
7 DESCRIPTION:Disciplina da trilha de Sistemas 64h
8 CATEGORIES:Sistemas
9 END:VEVENT
```

Figura 20 – Botão de exportação em formato ICS



Fonte: Elaborado pelo autor (2025).

## 5.10 Considerações Finais

Neste capítulo, procurou-se mostrar o processo de criação e integração das novas funcionalidades acrescentadas ao Stella, de forma que esse possa entender as decisões de projeto tomadas e a organização adotada no processo de desenvolvimento. Destaca-se a integração entre *frontend* e *backend*, bem como as funcionalidades avançadas como exportação e verificação de conflitos, que elevam a experiência do usuário.

## 6 TESTES DE USABILIDADE E AVALIAÇÃO DA APLICAÇÃO

Com o objetivo de avaliar a experiência do usuário com a ferramenta desenvolvida, foi realizado um teste de usabilidade utilizando o método SUS, um instrumento amplamente utilizado para mensurar a usabilidade de sistemas interativos. O SUS foi desenvolvido por John Brooke em 1986, e consiste em um questionário de 10 itens com afirmações alternadas entre positivas e negativas, sendo respondidas em uma escala de *Likert* de 5 pontos (BROOKE, 1996).

O teste foi aplicado após a finalização da versão funcional da ferramenta, por meio de um formulário *online* feito pelo Google Forms. Participaram 16 alunos do curso de Sistemas e Mídias Digitais da Universidade Federal do Ceará (UFC), público-alvo do Projeto Stella. O objetivo foi avaliar a facilidade de uso, clareza da interface e eficiência da funcionalidade principal (organizador de grade horária).

Cada participante recebeu instruções sobre a ferramenta, realizou tarefas como montar uma grade, visualizar conflitos e exportar em PDF ou iCalendar, e em seguida preencheu o questionário SUS. A coleta foi anônima, e os dados foram analisados conforme o método proposto por Sauro e Lewis (SAURO; LEWIS, 2016), em que os escores individuais são convertidos para uma escala de 0 a 100. A estrutura completa do formulário utilizado pode ser consultada no Apêndice B.

### 6.1 Resultados da Escala SUS

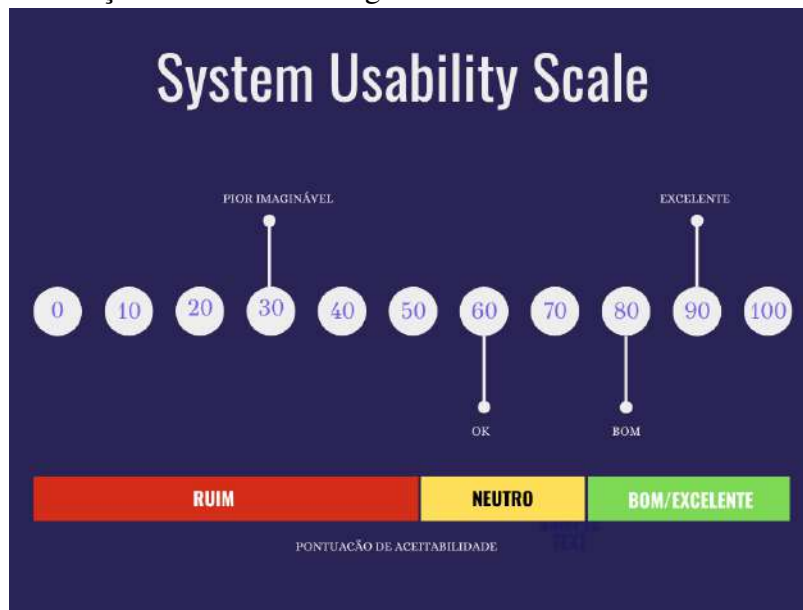
Dos 20 participantes esperados, 16 participaram do teste e responderam ao questionário. A pontuação média obtida na Escala SUS foi de **81,25 pontos**, com **mediana de 82,5**, indicando um nível de usabilidade considerado bom/excelente, segundo a classificação de Bangor et al. (BANGOR *et al.*, 2009). Os resultados variaram de 45 a 100 pontos. Dentre os respondentes, 87,5% atribuíram notas acima de 70, sendo que 62,5% deram notas superiores a 80.

"A visualização da grade à medida que as cadeiras eram selecionadas facilitou muito o entendimento."

*(Comentário de um participante)*

A Figura 21 ilustra visualmente o desempenho da ferramenta com base na Escala SUS.

Figura 21 – Classificação da ferramenta segundo a Escala SUS



Fonte: Elaborado pelo autor (2025).

## 6.2 Feedback Qualitativo

Além do questionário objetivo, os 16 participantes da pesquisa responderam quatro perguntas abertas. As principais observações foram:

- **Facilidades identificadas:** navegação intuitiva, visualização clara da grade, *layout* limpo e organizado.
- **Dificuldades relatadas:** localização pouco evidente de alertas (*toasts*) e pequenas dúvidas na escolha dos horários.
- **Pontos positivos:** design agradável, paleta de cores bem definida, organização das disciplinas e funcionalidades de exportação.
- **Sugestões de melhoria:** destacar visualmente os conflitos, permitir personalização das cores da grade e melhorar a experiência em telas menores.

"Melhorar a navegação, ser algo mais direto, pois às vezes me perdi entre as etapas."

(*Sugestão de um participante*)

## 6.3 Discussão

A avaliação SUS demonstrou que a ferramenta possui um nível de usabilidade satisfatório e bem aceito entre os alunos. A combinação de métricas quantitativas e qualitativas indica que o sistema cumpre seu propósito com clareza e eficiência. As sugestões obtidas servirão

como base para futuras melhorias, reforçando o compromisso com a experiência do usuário e a evolução contínua da ferramenta.

## 7 CONCLUSÃO

Este trabalho teve como objetivo a continuidade e o aprimoramento do Projeto Stella, com foco na implementação de uma ferramenta digital para organização personalizada da grade horária pelos estudantes do curso de Sistemas e Mídias Digitais da Universidade Federal do Ceará. Por meio do desenvolvimento dessa funcionalidade, buscou-se oferecer uma solução prática e interativa que auxiliasse os alunos no planejamento de suas disciplinas optativas a partir do quinto semestre.

Durante o processo de desenvolvimento, foram utilizadas tecnologias modernas como React.js no *frontend* e Node.js com Express no *backend*, além da biblioteca *FullCalendar* para visualização da grade semanal. A integração entre as camadas da aplicação, o uso de rotas protegidas e a manipulação de dados com MongoDB garantiram a flexibilidade e a eficiência da ferramenta.

A utilização dessas ferramentas e metodologias possibilitaram o alcance dos objetivos propostos para este trabalho, como a prototipação de uma nova ferramenta no Figma, bem como o seu desenvolvimento e a garantia das suas principais funcionalidades: a visualização e seleção das disciplinas, a possibilidade de filtragem e pesquisa e a interatividade com calendários semanais e mensais. Além disso, a execução do teste de usabilidade demonstrou boa aceitação dos discentes que são vistos como o público-alvo da ferramenta neste momento.

A aplicação permite a seleção de disciplinas por semestre e trilha, verifica automaticamente conflitos de horários e possibilita a exportação da grade em formatos PDF e *iCalendar*, compatíveis com calendários digitais. Também foi projetada com atenção à compatibilidade com navegadores modernos e à usabilidade geral.

Os resultados demonstraram que os objetivos propostos foram alcançados. Os testes de usabilidade aplicados indicaram que a ferramenta é intuitiva, funcional e cumpre seu papel de facilitar a organização acadêmica dos discentes.

### 7.1 Trabalhos Futuros

Como possibilidades de aprimoramento e continuidade do projeto, destacam-se:

- Implementação de autenticação completa com níveis de acesso (aluno, professor, coordenação);
- Criação da funcionalidade de salvamento da grade no banco de dados para recuperação

posterior;

- Permitir o cadastro e a edição de trilhas e disciplinas por parte da coordenação;
- Disponibilização de estatísticas visuais sobre a escolha de trilhas pelos alunos;
- Aperfeiçoamento da navegação e da responsividade do sistema.

Com essas melhorias, espera-se que o Stella continue evoluindo e contribuindo de forma significativa para o fortalecimento da comunicação e da organização acadêmica no âmbito do curso.

## REFERÊNCIAS

- ADAMAKIS, F. **A front-end application folder structure that makes sense**. 2023. Disponível em: <<https://fadamakis.com/a-front-end-application-folder-structure-that-makes-sense-ecc0b690968b>>. Acesso em: 14 jul. 2025.
- ANDRADE, R. B. L. **METIS: uma aplicação web para orientação acadêmica e pré-matrícula**. 2020. Salvador - Bahia, 28 dez. 2020. Disponível em: <<https://rodrigors.github.io/files/orientacao/tcc-roberto-bruno-leal-andrade.pdf>>. Acesso em: 14 jul. 2025.
- BANGOR, A.; KORTUM, P. T.; MILLER, J. T. Determining what individual sus scores mean: Adding an adjective rating scale. **Journal of Usability Studies**, Usability Professionals' Association, v. 4, n. 3, p. 114–123, 2009.
- BRAVI. **App nativo, web app ou app híbrido? Descubra as vantagens e desvantagens de cada tecnologia**. 2024. <<https://bravi.com.br/app-nativo-web-app-ou-app-hibrido>>. Acesso em: 10 ago. 2024.
- BROOKE, J. Sus: A quick and dirty usability scale. **Usability evaluation in industry**, Taylor & Francis, v. 189, n. 194, p. 4–7, 1996.
- BROWN, T. **Design Thinking: uma metodologia poderosa para decretar o fim das velhas ideias**. Rio de Janeiro: Alta Books, 2010.
- CHÉROLET, BRENDA. **O que é e como usar o Trello?** 2024. Disponível em: <<https://www.educamaisbrasil.com.br/educacao/dicas/o-que-e-e-como-usar-o-trello>>.
- EQUIPE TOTVS. **Kanban: conceito, como funciona, vantagens e implementação**. 2023. Disponível em: <<https://www.totvs.com/blog/negocios/kanban/#:~:text=Ou%20seja%2C%20o%20Kanban%20%C3%A9,mais%20no%20dia%20a%20dia.>> Acesso em: 06 mai. 2025.
- Feature-Sliced Design. **Feature-Sliced Design: Architectural methodology for frontend projects**. 2025. Disponível em: <<https://feature-sliced.design/>>.
- JÚNIOR, H. O.; FORTES, R. **Notas Didáticas: Padrões Arquiteturais em Aplicações Web**. [S.l.], 2022. Acesso em: 27 jul. 2025.
- LOUDON, K. C.; LOUDON, J. P. **Management Information Systems: Managing the Digital Firm**. 13. ed. [S.l.]: Pearson, 2014.
- Microsoft Corporation. **Common Web Application Architectures**. 2025. Acesso em: 27 jul. 2025. Disponível em: <<https://learn.microsoft.com/pt-pt/dotnet/architecture/modern-web-apps-azure/common-web-application-architectures>>.
- NÚCLEO DE INFORMAÇÃO E COORDENAÇÃO DO PONTO BR (NIC.br). **Resumo executivo: Pesquisa TIC Domicílios 2022**. [S.l.], 2023. Acesso em: 27 jul. 2024.
- RASOULI, N. Scalable react projects with feature-based architecture. 2025. Disponível em: <<https://dev.to/naserrasouli/scalable-react-projects-with-feature-based-architecture-117c>>.

**RICHARD. 5 melhores frameworks para protótipos de design e como eles estão transformando a criação de interfaces inovadoras.** 2025. Acesso em: 05 abr. 2025. Disponível em: <<https://bias.academy/blog/ux-design/melhores-frameworks-para-prototipos/>>.

**SAURO, J.; LEWIS, J. R. Quantifying the user experience: Practical statistics for user research.** [S.l.]: Morgan Kaufmann, 2016.

**SMD UFC. Sobre o curso - Sistemas e Mídias Digitais.** 2024. <<https://smd.ufc.br/pt/sobre-o-curso/>>. Acesso em: 21 ago. 2024.

**SOBRAL, L. H. C. Competitividade na era da informação digital. Inclusão Social,** 2017. Acesso em: 10 ago. 2024. Disponível em: <<https://revista.ibict.br/inclusao/article/view/3872>>.

**SYSTEM, C. Aplicações Web: o que são, os tipos e as vantagens.** 2024. <<https://www.coopersystem.com.br/aplicacoes-web-o-que-sao-os-tipos-e-as-vantagens/>>. Acesso em: 10 ago. 2024.

**WIKIPÉDIA. Figma.** 2025. Disponível em: <<https://en.wikipedia.org/wiki/Figma>>. Acesso em: 05 mai. 2025.

## APÊNDICE A – LINKS COMPLEMENTARES

A seguir, apresentam-se os links para os repositórios de código-fonte do sistema Stella e para o protótipo de interface da ferramenta de grade horária desenvolvido no Figma:

- **Repositório no GitHub (APP):** <<https://github.com/iJacKP/app-stella>>
- **Repositório no GitHub (API):** <<https://github.com/iJacKP/api-stella>>
- **Protótipo da ferramenta no Figma:** <<https://encurtador.com.br/qpRFf>>

**Observação:** Os links foram disponibilizados com o objetivo de permitir análise complementar do desenvolvimento, código e interface do sistema.

## APÊNDICE B – PERGUNTAS DO QUESTIONÁRIO PARA AVALIAÇÃO DO USUÁRIO

A seguir, são apresentadas as telas do questionário utilizado para a coleta de dados dos participantes da pesquisa. O instrumento foi implementado por meio do Google Forms e continha perguntas objetivas (Escala SUS) e abertas, conforme ilustrado nas figuras abaixo.

Figura 22 – Primeira página do questionário (SUS)

1. Eu usaria frequentemente esta ferramenta para montar minha grade. \*

1 2 3 4 5

Discordo Totalmente      Concordo Totalmente

2. Achei a ferramenta desnecessariamente complexa. \*

1 2 3 4 5

Discordo Totalmente      Concordo Totalmente

3. Achei a ferramenta fácil de usar. \*

1 2 3 4 5

Discordo Totalmente      Concordo Totalmente

4. Eu precisaria de ajuda de alguém para usar a ferramenta.

1 2 3 4 5

Discordo Totalmente      Concordo Totalmente

5. As funcionalidades (filtro, seleção, calendário, exportação) funcionam de forma integrada. \*

1 2 3 4 5

Discordo Totalmente      Concordo Totalmente

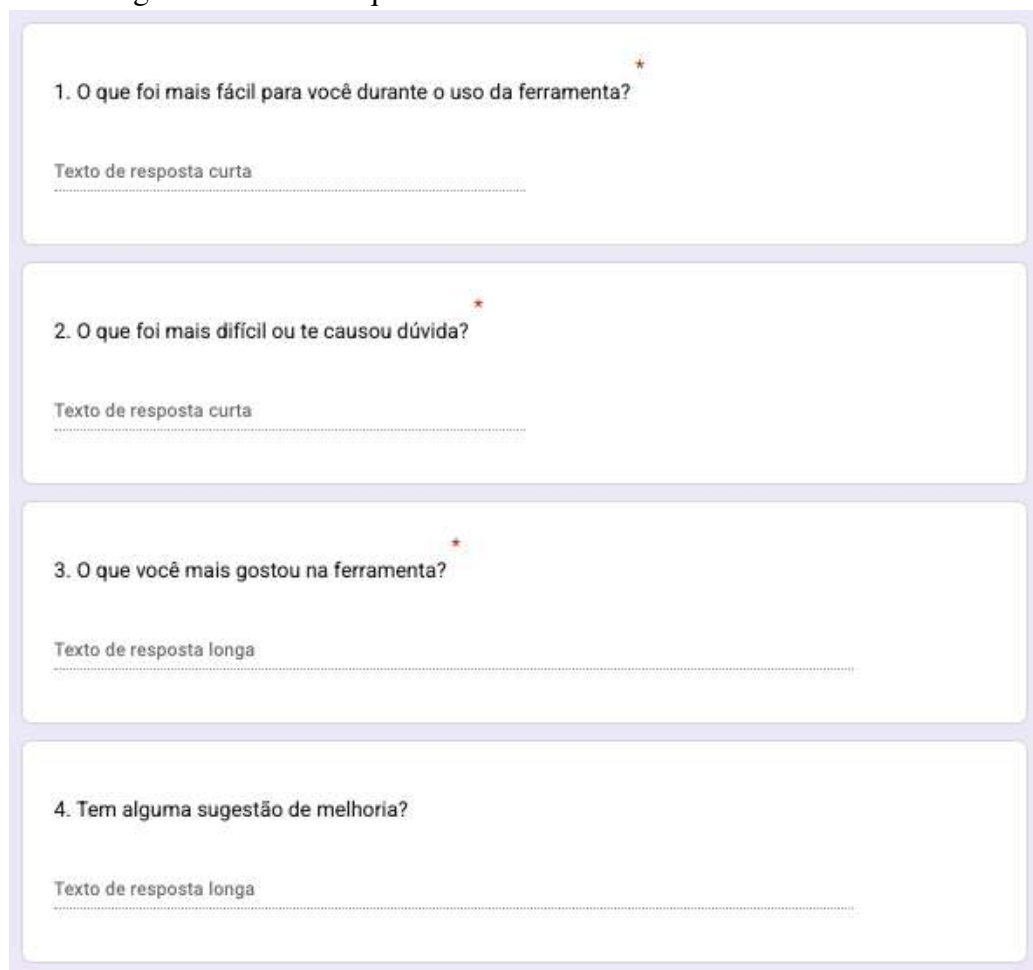
Fonte: Captura de tela do Google Forms (2025).

Figura 23 – Segunda página do questionário (SUS)

6. Achei que havia inconsistências ao usar a ferramenta. *						
	1	2	3	4	5	
Discordo Totalmente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Concordo Totalmente
7. Pessoas que usarem a ferramenta vão aprender a usá-la rapidamente. *						
	1	2	3	4	5	
Discordo Totalmente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Concordo Totalmente
8. A ferramenta me pareceu confusa em alguns momentos. *						
	1	2	3	4	5	
Discordo Totalmente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Concordo Totalmente
9. Eu me senti confiante ao usar a ferramenta. *						
	1	2	3	4	5	
Discordo Totalmente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Concordo Totalmente
10. Precisei aprender muitas coisas novas antes de usar a ferramenta. *						
	1	2	3	4	5	
Discordo Totalmente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Concordo Totalmente

Fonte: Captura de tela do Google Forms (2025).

Figura 24 – Perguntas abertas do questionário



1. O que foi mais fácil para você durante o uso da ferramenta? \*

Texto de resposta curta

2. O que foi mais difícil ou te causou dúvida? \*

Texto de resposta curta

3. O que você mais gostou na ferramenta? \*

Texto de resposta longa

4. Tem alguma sugestão de melhoria?

Texto de resposta longa

Fonte: Captura de tela do Google Forms (2025).