
Faculdade de Tecnologia de Americana “Ministro Ralph Biasi”
Curso Superior de Tecnologia em Análise de Desenvolvimento de Sistemas

João Pedro Moreli
Rafael Rodrigues Pinheiro

Morpi:
Desenvolvimento de aplicação web de apoio à navegação acadêmica.

Americana, SP
2025

João Pedro Moreli
Rafael Rodrigues Pinheiro

Morpi:
Desenvolvimento de aplicação web de apoio a navegação acadêmica.

Trabalho de Conclusão de Curso desenvolvido
em cumprimento à exigência curricular do Curso
Superior de Tecnologia em Análise e
Desenvolvimento de Sistemas na área de
concentração em Engenharia de Software

Orientador(a): Prof. Esp. Evandro Santaclara

Este trabalho corresponde à versão final do
Trabalho de Conclusão de Curso
apresentado por João Pedro Moreli e Rafael
Rodrigues Pinheiro e orientado pelo Prof.
Esp. Evandro Santaclara.

Americana, SP
2025

FICHA CATALOGRÁFICA — Biblioteca Fatec Americana
Ministro Ralph Biasi- CEETEPS Dados Internacionais de
Catalogação-na-fonte

PINHEIRO, Rafael Rodrigues

Morpi: Desenvolvimento de aplicação web de apoio à
navegação acadêmica.. / Rafael Rodrigues Pinheiro, João Pedro
Moreli — Americana, 2025.

56t

Monografia (Curso Superior de Tecnologia em Análise e
Desenvolvimento de Sistemas) - - Faculdade de Tecnologia de
Americana Ministro Ralph Biasi — Centro Estadual de Educação
Tecnológica Paula Souza

Orientador: Prof. Esp. Evandro Santaclara

1. Banco de dados 2. Desenvolvimento de software 3. PHP —
linguagem de programação. I. PINHEIRO, Rafael Rodrigues, II.
MORELI, João Pedro III. SANTACLARA, Evandro IV. Centro Estadual
de Educação Tecnológica Paula Souza — Faculdade de Tecnologia de
Americana Ministro Ralph Biasi

CDU: 681.3.07

681.3.05

681.3.061PHP

Elaborada pelo autor por meio de sistema automático gerador de
ficha catalográfica da Fatec de Americana Ministro Ralph Biasi.

João Pedro Moreli
Rafael Rodrigues Pinheiro

Morpi: Desenvolvimento de aplicação web de apoio a navegação acadêmica

Trabalho de graduação apresentado como exigência parcial para obtenção do título de Tecnólogo em Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas pelo Centro Paula Souza – FATEC Faculdade de Tecnologia de Americana Ministro Ralph Biasi.

Área de concentração: Análise e Desenvolvimento de Sistemas.

Americana, 3 de dezembro de 2025.

Banca Examinadora:



Evandro Santaclara
Especialista
Fatec Americana "Ministro Ralph Biasi"



Wagner Siqueira Cavalcante
Mestre
Fatec Americana "Ministro Ralph Biasi"



Ana Lúcia Spigolon
Especialista
Fatec Americana "Ministro Ralph Biasi"

DEDICATÓRIA

“A verdadeira coragem não é saber quando tirar uma vida, mas quando poupar uma.”

- Gandalf (em “O Hobbit”)

AGRADECIMENTOS

Em primeiro lugar agradecer aos nossos familiares pelo amor, incentivo e apoio incondicional.

A todos os professores que acompanharam a nossa jornada, enquanto estudantes e foram essenciais a nossa formação profissional, intelectual, e principalmente pessoal.

Ao nosso orientador, Professor Esp. Evandro Santaclara, pela paciência, pela orientação precisa e por acreditar no potencial da pesquisa, mesmo nos momentos mais difíceis. Sua sabedoria foi fundamental.

RESUMO

O presente trabalho conceitua o desenvolvimento de uma aplicação web destinada a apoiar usuários de uma instituição de ensino, na organização de suas agendas escolares. O objetivo é mitigar problemas decorrentes da volatilidade dos horários, que frequentemente geram confusões e comprometem a participação dos usuários em aulas ou em comunicação com os docentes. Para a resolução desse problema, adota-se um estudo de caso que compara diferentes plataformas estudantis, a fim de subsidiar o desenvolvimento de um protótipo. Os resultados evidenciam não apenas o desconhecimento dos usuários em relação ao acesso a essas informações, mas também as limitações dos sistemas vigentes, que obscurecem a usabilidade devido à forma inadequada de distribuição dos dados, em descaso com princípios de UI/UX. Esse cenário permite problematizar a viabilidade de acesso a informações enquanto produto a ser aprimorado. Ademais, identifica-se a necessidade de facilitar tal acesso, propondo-se a incorporação de recursos de inteligência artificial como estratégia complementar, dado o potencial de aumentar a comodidade e efetividade na interação dos usuários com a plataforma.

Palavras-Chave: agenda acadêmica; filtros personalizados; gestão de tempo.

ABSTRACT

This present work conceptualizes the development of a web application designed to support users of an educational institution in organizing their academic schedules. The objective is to mitigate problems arising from schedule volatility, which often causes confusion and compromises user participation in classes or communication with faculty. To address this problem, a case study is adopted, comparing different student platforms to inform the development of a prototype. The results highlight not only the users' lack of awareness regarding access to this information but also the limitations of existing systems, which hinder usability due to inadequate data distribution, disregarding UI/UX principles. This scenario allows for the problematization of information accessibility as a product to be improved. Furthermore, a need to facilitate such access is identified, proposing the incorporation of artificial intelligence features as a complementary strategy, given its potential to increase convenience and effectiveness in user interaction with the platform.

Keywords: *academic schedule; custom filters; time management.*

LISTA DE FIGURAS

Figura 1: Grade de horários no portal SIGA.....	17
Figura 2: Visualização de disciplinas no portal SIGA	18
Figura 3: Tela de início do portal GDE	18
Figura 4: Visualização da planejador de grade no portal GDE.....	19
Figura 5: Visualização do mapa no portal GDE	19
Figura 6: Tela de login Jupiterweb	20
Figura 7: Diagrama de Caso de Uso.....	28
Figura 8: Diagrama de Classe	33
Figura 9: Modelo Entidade-Relacionamento	34
Figura 10: Tela inicial do projeto Morpi	39
Figura 11: Tela da grade de horários do Morpi	40
Figura 12: Tela inicial do Morpi.....	42
Figura 13: Tela de demonstração da consulta por IA e resultados	43
Figura 14: Interface de visualização de horários e filtros	44
Figura 15: Tela de grade após seleção de “Filtro por semestre”	45
Figura 16: Tela de grade após seleção de “Filtro por professor”	45
Figura 17: Tela de grade após seleção de “Filtro por disciplina”	46
Figura 18: Tela de grade após seleção de “Filtro por disciplina e professor”	47
Figura 19: Transição para tela de personalização	47

Figura 20: Tela de prevenção de erro	48
Figura 21: Tela de montagem da grade personalizada	49
Figura 22: Tela de visualização da sala no mapa	49
Figura 23: Grade formatada em PDF para impressão	50

LISTA DE QUADROS

Quadro 1: Aplicações	16
Quadro 2: Comparativo de funcionalidades	21
Quadro 3: Requisitos funcionais do projeto	23
Quadro 4: Detalhamento dos Requisitos Funcionais	23
Quadro 5: Requisitos não funcionais do projeto	24
Quadro 6: Caso de Uso: Inserir dados das aulas.....	28
Quadro 7: Caso de uso: Exibir horários	29
Quadro 8: Caso de uso: Aplicar filtros	29
Quadro 9: Caso de uso: Personalizar grade	30
Quadro 10: Caso de uso: Gerar PDF da grade.....	30
Quadro 11: Caso de uso: Visualizar localização da sala.....	30
Quadro 12: Caso de uso: Consultar via chat com IA	31
Quadro 13: Caso de uso: Manter banco de dados.....	31
Quadro 14: Caso de uso: Consultar banco de dados	32
Quadro 15: Dicionário de Dados: PROFESSORES.....	35
Quadro 16: Dicionário de Dados: CURSOS	36
Quadro 17: Dicionário de Dados: DISCIPLINAS.....	36
Quadro 18: Dicionário de Dados: SEMESTRES.....	36
Quadro 19: Dicionário de Dados: SALAS	37
Quadro 20: Dicionário de Dados: HORARIOS.....	37
Quadro 21: Dicionário de Dados: GRADE_HORARIA.....	38

LISTA DE ABREVIATURAS E SIGLAS

API	<i>Application Programming Interface</i> (Interface de programação de aplicativos)
CPS	Centro Paula e Souza
CSS	<i>Cascading Styles Sheet</i> (Folhas de estilo em cascata)
DSR	<i>Design Science Research</i> (Pesquisa de ciência do design)
FATEC	Faculdade de Tecnologia
FK	<i>Foreign Key</i> (Chave Estrangeira)
GDE	Gerenciador de Disciplinas e Ementas
IA	Inteligência Artificial
MER	Modelo de Entidade-Relacionamento
MySQL	<i>My Structured Query Language</i>
PDF	<i>Portable Document Format</i> (Formato Portátil de Documento)
PHP	<i>Hypertext Preprocessor</i> (Pré-processador de Hipertexto)
PK	<i>Primary Key</i> (Chave primária)
RE	<i>Requirements Engineering</i> (Engenharia de Requisitos)
SIGA	Sistema Integrado de Gestão Acadêmica
UML	<i>Unified Modeling Language</i> (Linguagem de Modelagem Unificada)
UNICAMP	Universidade Estadual de Campinas
USP	Universidade de São Paulo

SUMÁRIO

1 INTRODUÇÃO	13
2 FUNDAMENTAÇÃO TEÓRICA	15
2.1 AVALICAÇÃO HEURÍSTICA.....	15
2.2 ESTUDO COMPARATIVO DE FERRAMENTAS SIMILARES.....	16
2.2.1 SIGA.....	17
2.2.2 GDE.....	18
2.2.3 JUPITER	20
3 PROJETO DE SISTEMA MORPI	21
3.1 SOFTWARES SIMILARES	21
3.2 LEVANTAMENTO DE REQUISITOS	22
3.2.1 REQUISITOS FUNCIONAIS	23
3.2.1.1 DETALHAMENTO DOS REQUISITOS FUNCIONAIS	23
3.2.2 REQUISITOS NÃO FUNCIONAIS.....	24
3.3 RECURSOS E FERRAMENTAS	25
3.4 MODELAGEM.....	26
3.5 DIAGRAMA DE CASO DE USO	26
3.6 DIAGRAMA DE CLASSE.....	32
3.7 MODELO ENTIDADE-RELACIONAMENTO	33
3.8 DICIONÁRIO DE DADOS.....	34
3.9 PROTOTIPAÇÃO DE TELAS	38
4 FASE FINAL DO SISTEMA.....	41
5 CONSIDERAÇÕES FINAIS.....	51
REFERÊNCIAS BIBLIOGRÁFICAS.....	52

1 INTRODUÇÃO

O presente trabalho tem como objetivo de propor o desenvolvimento de uma aplicação web voltada à exibição e organização da agenda de horários de uma instituição de ensino, incorporando funcionalidades e filtros que possibilitem aos usuários personalizar e otimizar sua rotina acadêmica.

A motivação para a proposta decorre das limitações observadas em sistemas de instituições de ensino, que apresentam restrições funcionais e inconsistências de organização, dificultando o planejamento dos discentes e evidenciando a necessidade de uma solução mais eficiente em termos de usabilidade.

Este trabalho tem como objetivo suprir essas demandas, optando por focar em criar uma ferramenta web flexível que permita ao usuário visualizar uma agenda de horários da sua respectiva instituição com funcionalidades aplicadas. Logo o público-alvo são alunos de instituições de ensino.

Os objetivos específicos são:

- Proporcionar maior dinâmica na personalização de horários;
- Implementar filtros que permitam aos usuários encontrar aulas e professores específicos;
- Disponibilizar ao usuário a opção de baixar uma grade de horários personalizada;
- Informar as localizações das salas através de um mapa.

O trabalho foi desenvolvido utilizando o método de *Design Science Research* (DSR), alinhado com a disciplina de Engenharia de Software. Essa abordagem se justifica pela necessidade de conduzir, de forma sistematizada, as atividades para a elaboração de uma interface funcional e intuitiva, uma vez que o objetivo é criar uma ferramenta web fácil e agradável, com eficácia e eficiência para o público-alvo. A usabilidade foi tratada como um processo sistemático, e suas atividades foram

integradas a todas as etapas do ciclo de desenvolvimento, incluindo as fases preliminares.

O trabalho foi estruturado em cinco capítulos. O primeiro capítulo apresenta esta introdução, detalhando o problema, a motivação e os objetivos. O segundo descreve a Fundamentação teórica, revisando a literatura sobre Engenharia de Software, Usabilidade e DSR. O terceiro capítulo detalha a Metodologia, justificando o uso do DSR e incluindo a análise comparativa de sistemas acadêmicos similares. O quarto capítulo é dedicado ao Resultado, apresentando o artefato web final.

Por fim, o quinto e último capítulo apresenta as considerações finais. Neste capítulo, os objetivos traçados na introdução são retomados e avaliados com base nos resultados obtidos no capítulo quatro. Adicionalmente, são discutidas as limitações encontradas durante o desenvolvimento e apresentadas sugestões de trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

O desenvolvimento de soluções de software eficazes requer um embasamento teórico consistente que oriente as decisões de projeto e implementação. Este capítulo apresenta os conceitos e metodologias essenciais para a sustentação do trabalho, abordando pilares da Engenharia de Software e da Interação Humano-Computador (IHC). A revisão bibliográfica concentra-se especialmente na Avaliação Heurística de Jakob Nielsen como método para assegurar a qualidade de uso e a experiência do usuário. Adicionalmente, discute-se o cenário tecnológico atual por meio de um estudo comparativo de ferramentas similares, estabelecendo os parâmetros de referência e requisitos para a proposta desenvolvida.

2.1 Avaliação Heurística

Uma técnica crucial e de baixo custo fundamental para Jakob Nielsen, em Engenharia de Usabilidade, enfrenta a complexidade de se aplicar compilações que contêm centenas ou milhares de regras de usabilidade. Nielsen reconhece que o “ótimo é inimigo do bom” e defende uma redução drástica das diretrizes, advogando o uso de apenas 10 regras para a avaliação de interfaces. Essa simplificação visa garantir que algum trabalho de engenharia de usabilidade seja realizado, permitindo que, mesmo os indivíduos sem treinamento especializado encontrem muitos problemas nas interfaces. Apresentam-se as definições dessas 10 heurísticas, que servem como princípio orientadores para o projeto e avaliação de sistemas centrados no usuário:

1. Visibilidade do status do sistema, ou conceito de *feedback*, é fundamental, pois garante que o usuário saiba o que está acontecendo no sistema.
2. Correspondência entre sistema e mundo real exige que o sistema use termos, conceitos e linguagem familiares aos usuários e não jargões de desenvolvimento.
3. Controle e liberdade do usuário permitem que os usuários se movam livremente no sistema e que possuam saídas marcadas claramente para facilitar ao máximo a experiência.
4. Consistência e padrões ajudam o usuário, pois se os elementos funcionam da mesma forma em diferentes partes do sistema, a curva de aprendizado é reduzida.
5. Prevenção de erros eliminam condições propensas a falhas, apresentando opções de confirmação aos usuários antes que eles se comprometam com a ação.

6. Reconhecimento em vez de memorização. O *design*, deve focar em fazer com que as informações e ações estejam visíveis, facilitando o reconhecimento, em vez de forçar o usuário a lembrar de comandos ou passos. A memorização está listada como um dos cinco critérios básicos da usabilidade, exigindo que telas apresentem facilidade de memorização para usuários ocasionais.

7. Flexibilidade e eficiência de uso devem atender, tanto aos usuários novatos, quanto aos experientes. Os atalhos aumentam a eficiência para usuários experientes. O critério básico da usabilidade da eficiência exige que o sistema apresente um alto nível de produtividade especialmente para usuários experientes.

8. Estética e *design* minimalista é reforçada pelo *slogan* “menos é mais”. Muitas funções ou opções impõem uma carga adicional ao usuário, e poucas opções podem proporcionar uma boa usabilidade ao permitir que os usuários se concentrem na compreensão dessas opções.

9. Ajudar usuários a reconhecer, diagnosticar e recuperar erros, estabelece que a quantidade de erros deve ser reduzida e os erros devem apresentar soluções simples e rápidas. O fornecimento de boas mensagens de erro ajuda nesse processo de diagnóstico e recuperação.

10. Ajuda e documentação: É sempre preferível um sistema que o usuário possa operar sem ajuda. Um sistema hostil não se torna amigável, mesmo com a implementações de um bom sistema de ajuda. A ajuda e a documentação servem para que o usuário possa aprofundar seus conhecimentos e auxiliar na transição do usuário novato para o experiente.

2.2 Estudo comparativo de ferramentas similares

Considerando a natureza do problema que foi identificado, foram reunidas por meio de *websites*, três ferramentas similares ao projeto. Esta pesquisa foi realizada entre junho e outubro de 2025 cujos dados foram resumidos no Quadro 1.

Quadro 1 - Aplicações

Aplicação	Plataforma	Fonte
SIGA - CPS	Web	https://siga.cps.sp.gov.br/sigaaluno/applogin.aspx
GDE - Unicamp	Web	https://grade.daconline.unicamp.br/login/
Jupiter - USP	Web	https://uspdigital.usp.br/jupiterweb/webLogin.jsp

Fonte: Elaborado pelo autor (2025)

Será discutido principalmente sobre os portais ao quais houve, acesso sendo eles:

- SIGA – Sistema Integrado de Gestão Acadêmica do Centro Paula Souza: desenvolvido para oferecer recursos administrativos e de gestão para suas Faculdades de Tecnologia (FATEC) e Escolas Técnicas Estaduais (ETEC) (CENTRO PAULA SOUZA, 2025).
- Gerenciador de Disciplinas e Ementas - GDE: É uma iniciativa de alunos da Universidade Estadual de Campinas (Unicamp), que disponibiliza informações e ferramentas de controle acadêmico. Criadores Felipe Guaycuru, Rodrigo Yasuda e Carlos Avelar (UNIVERSIDADE ESTADUAL DE CAMPINAS, 2025).
- Jupiterweb: É um sistema criado para atender os alunos e os professores da Universidade de São Paulo (USP), com o fim de administrar a relação entre alunos matriculados, as disciplinas e as unidades (Universidade de São Paulo, 2025). Foi desenvolvido e é mantido pela Superintendência de Tecnologia da Informação da universidade (UNIVERSIDADE DE SÃO PAULO, 2025).

Segue a relação de algumas funcionalidades e características presentes em cada aplicação selecionada:

2.2.1 – SIGA

- Aba de visualização de horários: permite que o aluno consulte as datas e horários das aulas que está matriculado no semestre vigente conforme a Figura 1.

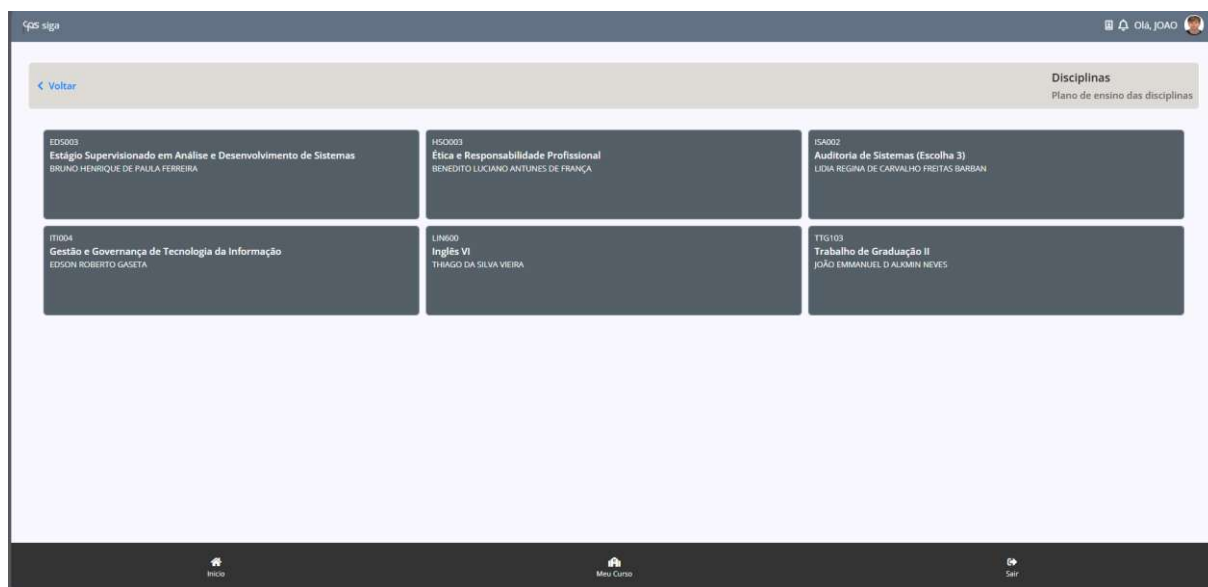
Figura 1 - Grade de horários no portal SIGA

Terça-Feira			
HSO003 Ética e Responsabilidade Profissional BENEDITO LUCIANO ANTUNES DE FRANÇA 19:00-19:50	HSO003 Ética e Responsabilidade Profissional BENEDITO LUCIANO ANTUNES DE FRANÇA 19:50-20:40	ISA002 Auditoria de Sistemas (Escolha 3) LIDIA REGINA DE CARVALHO FREITAS BARBAN 20:50-21:40	ISA002 Auditoria de Sistemas (Escolha 3) LIDIA REGINA DE CARVALHO FREITAS BARBAN 21:40-22:30
Quarta-Feira			
ITI004 Gestão e Governança de Tecnologia da Informação EDSON ROBERTO GASETA 19:00-19:50	ITI004 Gestão e Governança de Tecnologia da Informação EDSON ROBERTO GASETA 19:50-20:40		
Quinta-Feira			
ITI004 Gestão e Governança de Tecnologia da Informação EDSON ROBERTO GASETA 20:50-21:40	ITI004 Gestão e Governança de Tecnologia da Informação EDSON ROBERTO GASETA 21:40-22:30		
Sexta-Feira			
LIN600 Inglês VI THIAGO DA SILVA VIEIRA 19:00-19:50	LIN600 Inglês VI THIAGO DA SILVA VIEIRA 19:50-20:40	ISA002 Auditoria de Sistemas (Escolha 3) LIDIA REGINA DE CARVALHO FREITAS BARBAN 20:50-21:40	ISA002 Auditoria de Sistemas (Escolha 3) LIDIA REGINA DE CARVALHO FREITAS BARBAN 21:40-22:30

Fonte: Captura de tela do sistema SIGA (CPS, 2025)

- Aba de visualização das disciplinas em que o aluno está matriculado no semestre vigente, conforme Figura 2.

Figura 2 - Visualização de disciplinas no portal SIGA

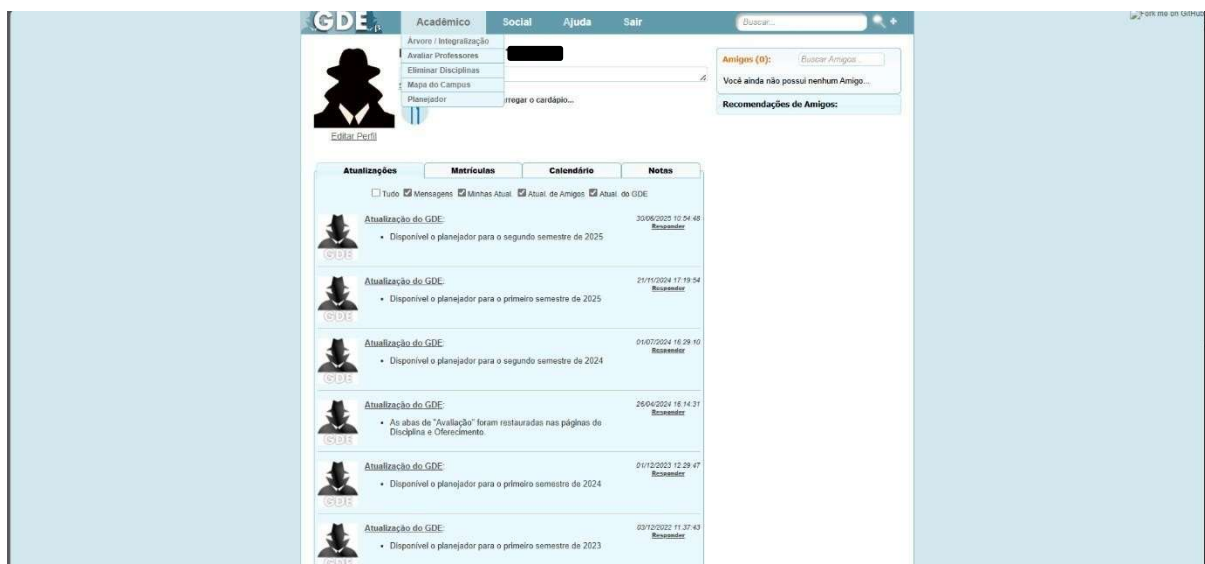


Fonte: Captura de tela do sistema SIGA (CPS, 2025)

2.2.2 – GDE

- Aba de login do portal exibindo informações pertinentes ao usuário destacando-se o link que abre a visualização de grades, conforme a Figura 3.

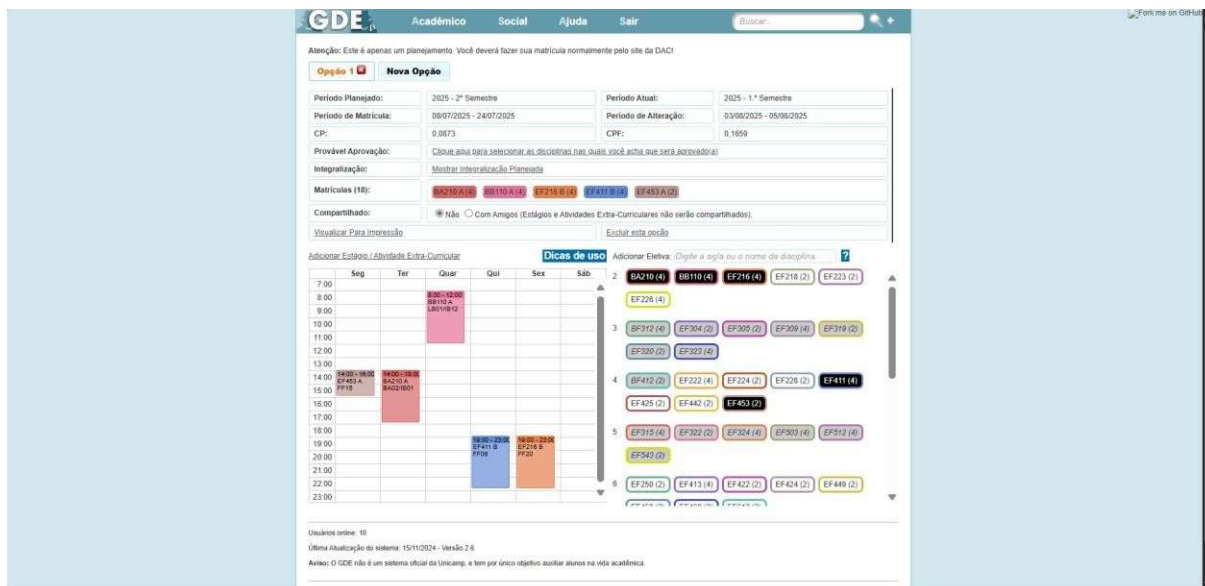
Figura 3 – Tela de início do portal GDE



Fonte: Captura de tela do sistema GDE (Unicamp, 2025)

- Aba de visualização de grades, que permite ao usuário visualizar e montar grade de aulas matriculadas, conforme Figura 4.

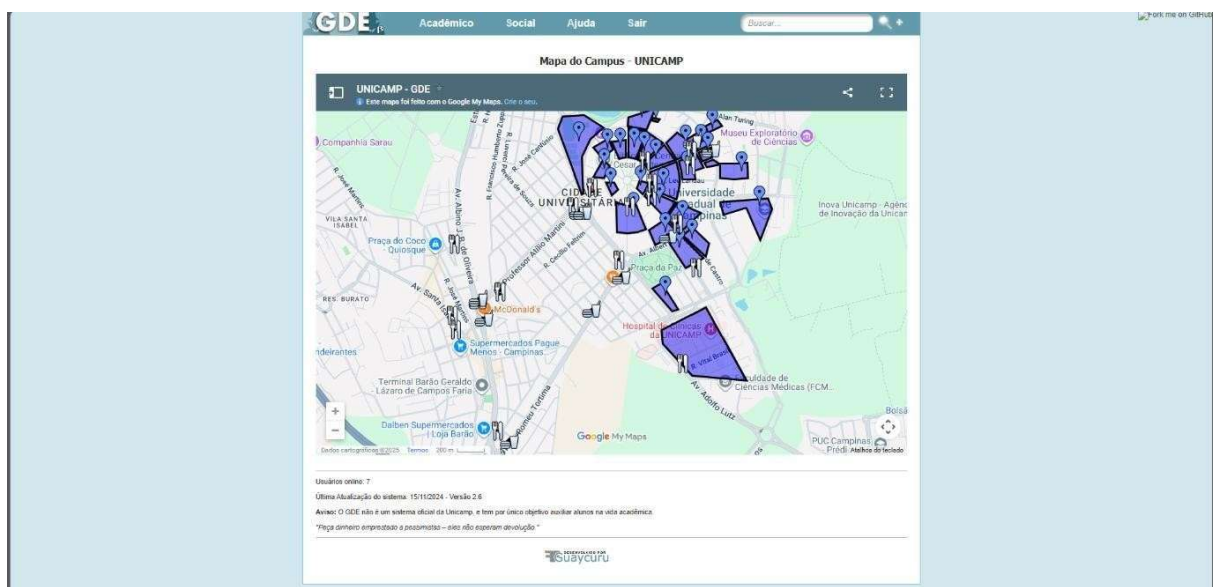
Figura 4 - Visualização da grade no portal GDE



Fonte: Captura de tela do sistema GDE (Unicamp, 2025)

- Aba de visualização do mapa da instituição que permite ao usuário se localizar pelo campus, conforme Figura 5.

Figura 5 - Visualização do mapa no portal GDE



Fonte: Captura de tela do sistema GDE (Unicamp, 2025)

2.2.3 — Jupiterweb

- Aba de visualização de grades que permite ao usuário visualizar e montar grade de aulas matriculadas, conforme Figura 6.

Figura 6 - Tela de login Jupiterweb

USP Universidade de São Paulo Brasil

11795252 - Fernanda Alves de Araujo | Aluno | [Alterar Senha](#) | [Sair](#)

Público

- Calendário Escolar
- Cursos de Ingresso
- Disciplina
- Editais
- FAQ

Acesso Restrito

- Atividades Extensionistas Curriculares (AEX)
- Matricula
- Grade horária
- Histórico escolar
- Acompanhamentos
- Cartão USP
- Emissão de documentos
- Opção de curso
- Estágio pago pela USP
- Estágio externo
- Programa de bolsas
- Auxílios permanência/PRIP
- Programa USP iFriends
- Requerimento

Grade Horária

Preparar para impressão

Unidade: [REDACTED]

Aluno: [REDACTED]

Programa: [REDACTED] Situação: Ativo

Curso: 44011 - Geologia - Integral

Complemento	Início	Fim	Motivo do Encerramento
100 - Geologia	16/03/2022		

Grade Horária

Horário		Seg	Ter	Qua	Qui	Sex	Sab	Dom
Início	Fim							
08:00	10:00		GSA0405-2025		GSA0404-2025			
10:00	12:00		GSA0405-2025		GSA0404-2025			
14:00	15:00			GSA0406-2025				
14:00	15:30					GMG0401-2025		
14:00	16:00				GSA0312-2025			
15:00	16:00			GSA0406-2025				
16:00	18:00				GSA0312-2025			

Créditos	Inscritos	Pendentes	Matriculados
Aula	0	0	32
Trabalho	0	0	1

Legenda: Inscrito Pendente Matriculado

Grade Horária gerada em 23 de Outubro de 2025 - 16:32

Observação:
Clique sobre o código da Disciplina/Turma para consultar as informações complementares.

[Créditos](#) | [Fale conosco](#)

© 1999 - 2025 - Superintendência de Tecnologia da Informação/USP

Fonte: Captura de tela do sistema Jupiterweb (USP, 2025)

3 PROJETO DO SISTEMA (MORPI)

Este capítulo detalha o processo do desenvolvimento da planilha de horários que funciona como motor central da aplicação web.

3.1 Softwares Similares

O projeto tem como base sites institucionais da FATEC (SIGA) e o da Unicamp, que busca propor melhorias e adaptações conforme a necessidade do usuário.

Levando estes aspectos em consideração, foi elaborada o Quadro 2, que mostra as principais diferenças entre a ferramenta a ser desenvolvida, o Morpi (S1), e os sistemas de referência utilizados para comparação: SIGA (S2), GDE-Unicamp(S3) e Jupiter- USP (S4).

Quadro 2 – Comparativo de funcionalidades

Funcionalidade	S1	S2	S3	S4
Horários interativos	X		X	
Sinalização de sala no mapa	X		X	
Horários organizados conforme cronograma do usuário	X	X	X	X
Filtros para pesquisas diversas	X			
Possibilidade de impressão	X	X	X	X
Criar grade específica de horários	X	X	X	X
Informar horário, matéria e professor	X	X	X	X

Fonte: Elaborado pelo autor (2025).

A análise comparativa identificou que os portais em estudo (GDE, SIGA, Jupiter) não fornecem as informações de forma estritamente voltada à funcionalidade central de pesquisa, desviando-se da simplicidade que foi priorizada como meta no desenvolvimento do Morpi, e que, segundo Nielsen (2000), também é a meta do *web design*.

O GDE, por exemplo, oferece uma funcionalidade de auxílio denominada “Dicas de uso”. Embora a intenção seja instruir o usuário, essa premissa de explicação explícita sobre como o site não possui um *design* intuitivo. Tal abordagem impõe uma carga adicional ao usuário, que deveria ser capaz de interagir com o sistema sem a

necessidade de ler material desnecessário (NIELSEN, 2000, p.111). A usabilidade é prejudicada quando o sistema falha em ser imediatamente compreensível e exige que o usuário desfoque do objetivo principal do conteúdo para aprender a operar a interface. (NIELSEN, 2000, p.119).

Em relação ao planejamento da grade curricular, o portal GDE emprega o uso de duas visualizações distintas do cronograma, recurso também contemplado pelo SIGA e Jupiter. Embora a exibição de diferentes representações possa ser encarada como uma boa ideia, não implica, necessariamente, em uma melhor eficiência da interface sob a ótica das heurísticas de usabilidade.

O SIGA demonstra uma melhor adesão à heurística de correspondência entre o sistema e o mundo real, ao utilizar símbolos na interface para executar tarefas. Esta abordagem é contrastante em relação ao GDE e Jupiter, que se restringem ao uso de caixas de texto para guiar o usuário. A dependência exclusiva de caixas de texto na navegação é desvantajosa quando o sistema não confia no princípio da consistência (NIELSEN, 2000, p. 27-28). A falta de consistência exige que o usuário invista tempo e atenção para discernir a funcionalidade, o que compromete a experiência e aumenta a probabilidade de erros.

A heurística de correspondência entre o sistema e o mundo real é fragilizada nos portais GDE e Jupiter, pois o uso de códigos de disciplina para montar as grades curriculares impõe um modelo mental abstrato ao usuário, distanciando-o da linguagem simples e familiar que deveria caracterizar o sistema.

A proposta de facilitar o acesso dos usuários a visualizar e modular uma grade horária segue os princípios de que, se a navegação for muito complexa, exige explicações externas, ou, se o usuário precisa memorizar códigos em vez de nomes familiares, a experiência falha. O *design* de usabilidade, tal como a arquitetura da informação, deve atuar como um servo silencioso, facilitando a jornada e garantindo que o tempo do usuário seja gasto consumindo o conhecimento, e não decifrando o sistema.

3.2 Levantamento de requisitos

A engenharia de requisitos (RE – *Requirements Engineering*) é o processo de descoberta, análise, documentação e conferência dos serviços e restrições de um sistema. Um requisito pode ser definido como uma descrição dos serviços fornecidos pelo sistema e as suas restrições operacionais (SOMMERVILLE, 2018).

Tradicionalmente, os requisitos são divididos em dois tipos: requisitos funcionais e requisitos não-funcionais.

3.2.1 Requisitos Funcionais

Os requisitos funcionais descrevem o que o sistema deve fazer, isto é, definem a funcionalidade desejada do software (SOMMERVILLE, 2018), e resultam na especificação das características operacionais do software (PRESSMAN, 2016).

O Quadro 3 apresenta os principais requisitos funcionais deste projeto:

Quadro 3 – Requisitos funcionais do projeto.

Identificação	Requisito Funcional	Prioridade
RF001	Inserção de todos dados das aulas no banco de dados	Essencial
RF002	Exibir Horários	Essencial
RF003	Filtros diversos (semestre, matéria professor)	Importante
RF004	Personalizar grade de horários *	Desejável
RF005	Mostrar o bloco da sala de aula no mapa	Desejável
RF006	Gerar <i>Portable Document Format</i> (PDF) da grade horária personalizada	Importante
RF007	API de Inteligência Artificial para busca	Desejável

Fonte: Elaborado pelo autor (2025).

3.2.1.1 Detalhamento dos Requisitos Funcionais

Os requisitos funcionais citados anteriormente serão explicados individualmente, detalhando suas funcionalidades:

Quadro 4 – Detalhamento dos Requisitos funcionais

Identificação	Descrição
RF001	Dados coletados do cronograma e grade horária da instituição devem ser armazenados no sistema de gerenciamento de banco de dados MySQL pelos administradores.
RF002	A ferramenta deve apresentar a grade das matérias, com seus horários atualizados, informando também seus respectivos professores, cursos e salas de forma clara e responsiva para o usuário.

RF003	Filtros diversos serão aplicados à pesquisa no banco de dados, como filtros de professores, disciplinas, dias da semana, horários, salas. O sistema deve implementar filtros dinâmicos na interface, permitindo que o usuário refine a visualização dos dados em tempo real, facilitando a localização de informações específicas sem a necessidade de recarregar a página.
RF004	O usuário deve ter a possibilidade de marcar e desmarcar disciplinas na grade horária padrão, permitindo a criação de uma grade personalizada e editável para seu planejamento acadêmico.
RF005	O sistema deve apresentar funcionalidade de exibição da localização da sala no mapa da instituição e, ao selecionar uma sala, deve exibir a localização correspondente.
RF006	Permitir ao usuário acionar um botão para exportar o conteúdo visualizado como um arquivo PDF.
RF007	Disponibilizar uma ferramenta IA, onde o usuário pode submeter consultas em linguagem natural. A ferramenta IA deve processar a pergunta, gerar uma <i>query</i> SQL correspondente, executá-la no banco de dados e retornar a informação solicitada em formato de texto.

Fonte: Elaborada pelo autor (2025)

3.2.2 Requisitos Não Funcionais

Os requisitos não funcionais (RNF) são aqueles não diretamente relacionadas às funções específicas fornecidas pelo sistema (SOMMERVILLE, 2018). Pressman (2016) os define como atributos de qualidade, de desempenho, de segurança ou restrições gerais em um sistema. O Quadro 5 apresenta alguns dos requisitos não funcionais deste projeto.

Quadro 5 – Requisitos não funcionais do projeto.

Identificação	Requisito não funcional	Categoria	Prioridade
RNF001	O sistema deve ser intuitivo e fácil de aprender para usuários inexperientes.	Usabilidade	Essencial
RNF002	O sistema deve permitir que tarefas essenciais sejam feitas de maneira prática, com número mínimo de passos.	Usabilidade	Essencial
RNF003	A interface do sistema deve ser responsiva de forma consistente em diferentes resoluções de tela.	Portabilidade	Desejável

RNF004	O banco de dados deve ser estruturado de forma que os desenvolvedores possam realizar alterações e manutenção com facilidade.	Manutenibilidade	Desejável
RNF005	Para utilização da aplicação, o usuário deve ter conexão a internet.	Desempenho	Essencial
RNF006	O sistema deve ser desenvolvido utilizando a linguagem de programação <i>Hypertext Preprocessor</i> (PHP)	Organizacional	Desejável
RNF007	O tempo de carregamento da visualização principal da agenda deve ser minimizado	Desempenho	Desejável

Fonte: Elaborado pelo autor (2025).

3.3 Recursos e Ferramentas

Esta seção contempla as ferramentas de programação e os conceitos necessários para o desenvolvimento do sistema:

- **HTML (*HyperText Markup Language*)**: Linguagem de marcação utilizada para descrever a estrutura de uma página web (DUCKETT, 2016) e exibir seu conteúdo. É a base para a criação de páginas web, permitindo a definição de elementos como textos, imagens, links e formulários. O HTML é essencial para construção da interface do usuário, garantindo que o conteúdo seja organizado e acessível em diferentes dispositivos e navegadores.
- **CSS (*Cascading Style Sheets*)**: Linguagem de estilo, utilizada para controlar a apresentação visual de páginas web. Duckett (2016) explica que o CSS emprega regras que permitem ao desenvolvedor controlar, tanto a estilização, quanto o *layout* das páginas. Permite definir cores, fontes, *layouts* e efeitos visuais, separando o conteúdo (HTML) da aparência. Com ele, é possível criar *designs* responsivos e adaptáveis, garantindo uma experiencia consistente em diferentes tamanhos de tela e dispositivos.
- **MySQL**: É um sistema de gerenciamento de banco de dados relacional de código aberto, amplamente utilizado para armazenar e gerenciar dados de aplicações web. O MySQL é conhecido por sua confiabilidade, desempenho e facilidade de uso, sendo ideal para aplicações que exigem armazenamento e recuperação eficiente de dados, como cadastros de usuários, horários. Ele

permite que o desenvolvedor armazene, pesquise, classifique e recupere dados de forma eficiente. (WELLING; THOMSON, 2003)

- PHP (*Hypertext Preprocessor*): Linguagem de *script* do lado do servidor, utilizada para desenvolver aplicações web dinâmicas e interativas, e permitir a integração com bancos de dados. O PHP é apreciado por seu alto desempenho, interfaces nativas para vários sistemas de banco de dados e facilidade de aprendizado (WELLING; THOMSON, 2003).
- JavaScript: Linguagem de programação web dinâmica e versátil, essencial para desenvolvimento web, permitindo criação de interatividade. Segundo Flanagan (2013), o JavaScript é a terceira peça da tríade tecnológica web (juntamente com HTML para conteúdo e CSS para apresentação), sendo essencial em especificar o comportamento de documentos. Lida com interatividade e define rotinas de eventos.

Além das tecnologias citadas, para auxílio na revisão textual e estruturação de códigos, foi utilizada a ferramenta de inteligência artificial Gemini (GOOGLE, 2025) nas refatorações e otimização da lógica de programação e também na revisão gramatical.

3.4 – Modelagem

Na fase da modelagem, é feita a documentação do *Web App*, e trata-se de diagramas que facilitam na compreensão do projeto de forma padronizada. A documentação deste trabalho utilizará a linguagem de modelagem *Unified Modeling Language* (UML), uma linguagem padrão para modelagem e documentação de sistemas orientados a objetos, através de diagramas de casos de uso, diagramas de classe e modelo-entidade relacionamento.

3.5 Diagrama de caso de uso

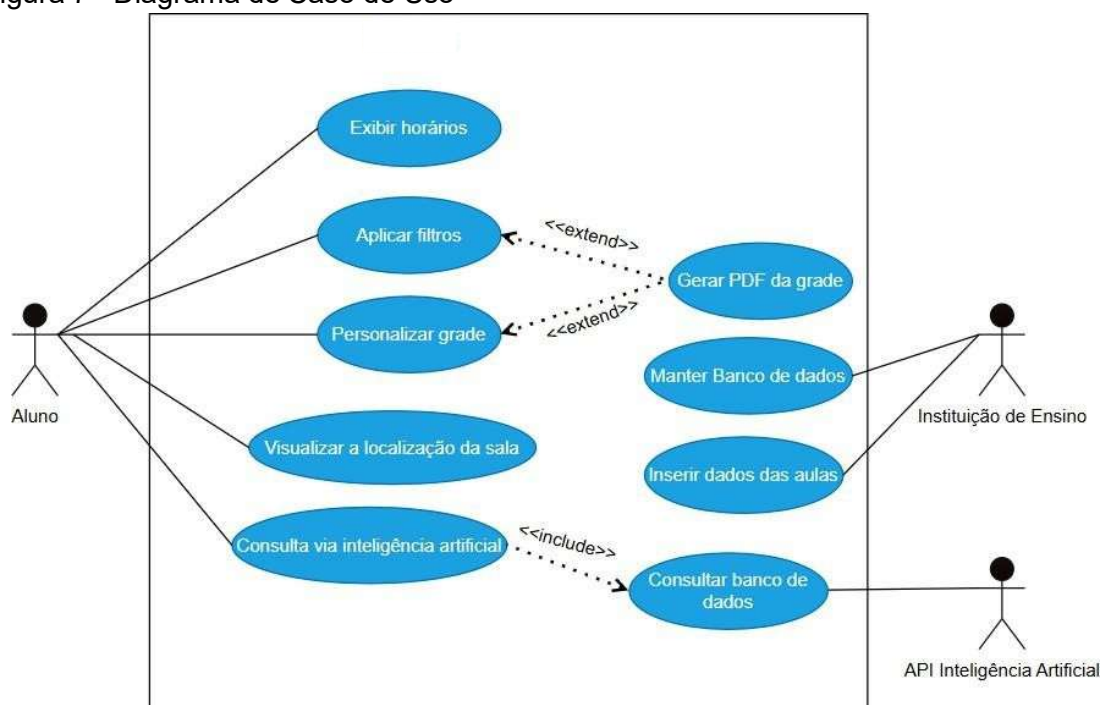
A técnica de Caso de Uso é composta por duas partes principais: O Diagrama de Caso de Uso, que é a representação gráfica, e a especificação textual, que descreve o fluxo completo de funcionalidade. Em conjunto, essas ferramentas de modelagem descrevem um cenário de funcionalidades sob a perspectiva do usuário (PRESSMAN, 2016). Tais diagramas ilustram as responsabilidades e a interação de cada ator, demonstrando o que cada um pode executar no sistema, visando os requisitos funcionais dele.

A descrição detalhada que sustenta o diagrama, traz uma narrativa de como um usuário interage com o sistema em circunstâncias específicas (PRESSMAN, 2016). Conforme Sommerville (2018), os Casos de Uso descrevem as interações entre usuários e um sistema, utilizando um modelo gráfico e um texto estruturado, sendo uma característica fundamental da Linguagem de Modelagem Unificada (UML). O diagrama captura um “contrato de comportamento”, descrevendo o que o sistema deve fazer em resposta a uma solicitação de um de seus *stakeholders* (PRESSMAN, 2016). Essa representação é estruturada através dos atores, simbolizados por um boneco tipo *stickman* (PRESSMAN, 2016; SOMMERVILLE, 2018), que representam as entidades externas (pessoas e outros sistemas) e as funcionalidades (ou casos de uso) que são representadas pelos balões (elipses), com a ação descrita em seu interior (PRESSMAN, 2016; SOMMERVILLE, 2018), que indicam o que o sistema faz, todos colocados em um retângulo para ressaltar visualmente as fronteiras do sistema (PRESSMAN, 2016), além das relações e ligações, representadas por linhas, que mostram a comunicação entre eles.

Embora o diagrama de Caso de Uso forneça uma visão simples e geral da funcionalidade do sistema (PRESSMAN, 2016; SOMMERVILLE, 2018), Pressman (2016) aponta que a maior contribuição do Caso de Uso para o processo de desenvolvimento é a descrição textual detalhada de cada caso, e não o diagrama em si.

Observam-se estes elementos dispostos no diagrama, conforme a Figura 7:

Figura 7 - Diagrama de Caso de Uso



Fonte: Elaborado pelo autor (2025).

A principal função deste diagrama é mostrar qual será o fluxo que o programa deve seguir, além de fluxos optativos adicionando informações representadas a partir do Quadro 6 ao 14.

Inicialmente, é necessário compreender como as informações são inseridas no sistema para viabilizar as consultas futuras. O Quadro 6 detalha o processo de cadastro e registro das aulas realizado pela instituição de ensino.

Quadro 6 – Caso de uso: Inserir dados das aulas

Nome do caso de uso	Inserir dados das aulas
Atores envolvidos	Instituição de Ensino
Objetivo	Registrar no banco de dados todas as informações referentes às aulas, professores, disciplinas, salas e horários.
Prioridade de desenvolvimento	Essencial
Ações do ator	<ol style="list-style-type: none"> 1. A instituição acessa o sistema com credenciais válidas. 2. Seleciona a opção 'Inserir dados das aulas'. 3. Informa as informações da disciplina, professor, sala e horário. 4. Confirma o envio para armazenamento no banco de dados.
Ações alternativas do ator	<ol style="list-style-type: none"> 1. Se o sistema detectar dados incompletos, exibirá mensagem de erro e solicitará correção.

Fonte: Elaborado pelo autor (2025)

Uma vez que os dados estão armazenados, a funcionalidade primária para o discente é o acesso a essas informações. O Quadro 7 apresenta o fluxo básico para a visualização da grade de horários.

Quadro 7 – Caso de uso: Exibir horários

Nome do caso de uso	Exibir horários
Atores envolvidos	Aluno
Objetivo	Permitir que o aluno visualize a grade de horários de seu curso e semestre.
Prioridade de desenvolvimento	Essencial
Ações do ator	1. O aluno acessa o sistema. 2. Seleciona o curso e semestre desejado. 3. O sistema exibe os horários correspondentes.
Ações alternativas do ator	1. Se não houver dados disponíveis, o sistema exibirá mensagem informando ausência de horários.

Fonte: Elaborado pelo autor (2025)

Para refinar a visualização e evitar a sobrecarga de informações, o sistema oferece recursos de triagem. O Quadro 8 descreve o funcionamento da aplicação de filtros por semestre, matéria ou professor.

Quadro 8 – Caso de uso: Aplicar filtros

Nome do caso de uso	Aplicar filtros
Atores envolvidos	Aluno
Objetivo	Permitir que o aluno filtre os horários por semestre, matéria ou professor.
Prioridade de desenvolvimento	Importante
Ações do ator	1. O aluno acessa a tela de horários. 2. Escolhe um ou mais filtros (semestre, matéria, professor). 3. O sistema atualiza a exibição conforme os filtros aplicados.
Ações alternativas do ator	1. Se nenhum filtro for selecionado, o sistema exibirá todos os horários.

Fonte: Elaborada pelo autor (2025)

Além da visualização padrão, o sistema permite que o usuário adapte a organização conforme sua necessidade específica. O Quadro 9 especifica o caso de uso voltado para a montagem e personalização da grade horária.

Quadro 9 – Caso de uso: Personalizar grade

Nome do caso de uso	Personalizar grade
Atores envolvidos	Aluno
Objetivo	Permitir que o aluno monte sua própria grade de horários selecionando as disciplinas desejadas.
Prioridade de desenvolvimento	Desejável
Ações do ator	1. O aluno acessa a opção 'Personalizar grade'. 2. Seleciona as disciplinas desejadas. 3. O sistema gera uma grade personalizada.
Ações alternativas do ator	1. Se o aluno não selecionar nenhuma disciplina, o sistema não permitirá salvar a grade.

Fonte: Elaborada pelo autor (2025).

Garantindo a portabilidade da informação organizada pelo aluno, o sistema oferece mecanismos de exportação. O Quadro 10 demonstra o fluxo para a geração do arquivo PDF da grade personalizada.

Quadro 10 – Caso de uso: Gerar PDF da grade

Nome do caso de uso	Gerar PDF da grade
Atores envolvidos	Aluno
Objetivo	Gerar e disponibilizar um arquivo PDF da grade personalizada do aluno.
Prioridade de desenvolvimento	Importante
Ações do ator	1. O aluno acessa sua grade personalizada. 2. Clica na opção 'Gerar PDF'. 3. O sistema exporta a grade em formato PDF para download.
Ações alternativas do ator	1. Caso ocorra erro na geração do arquivo, o sistema exibirá mensagem e sugerirá nova tentativa.

Fonte: Elaborada pelo autor (2025).

Visando auxiliar na orientação espacial do usuário dentro do campus, o sistema integra informações de localidade. O Quadro 11 detalha o processo de visualização da sala de aula no mapa da instituição.

Quadro 11 – Caso de uso: Visualizar localização da sala

Nome do caso de uso	Visualizar localização da sala
Atores envolvidos	Aluno
Objetivo	Permitir que o aluno visualize no mapa o bloco e localização da sala de aula.

Prioridade de desenvolvimento	Desejável
Ações do ator	1. O aluno clica no nome da sala exibida no horário. 2. O sistema mostra no mapa a localização correspondente.
Ações alternativas do ator	1. Caso o mapa não esteja disponível, o sistema exibirá mensagem de indisponibilidade temporária.

Fonte: Elaborado pelo autor (2025)

Como recurso inovador de acessibilidade e facilidade de uso, foi implementada uma interface de inteligência artificial. O Quadro 12 descreve a interação do usuário com o chat de IA para sanar dúvidas sobre a agenda.

Quadro 12 – Caso de uso: Consultar via chat com IA

Nome do caso de uso	Consultar via chat com IA
Atores envolvidos	Aluno
Objetivo	Permitir que o aluno tire dúvidas com uma IA sobre horários, professores e salas.
Prioridade de desenvolvimento	Desejável
Ações do ator	1. O aluno acessa o chat de IA. 2. Digita sua pergunta sobre aulas, horários ou professores. 3. O sistema busca informações no banco e exibe a resposta.
Ações alternativas do ator	1. Se a IA não compreender a pergunta, ela solicitará reformulação.

Fonte: Autoria própria (2025)

Para garantir a consistência e a atualização das informações ao longo do tempo, são necessárias rotinas administrativas. O Quadro 13 apresenta as atividades de manutenção e atualização dos registros no banco de dados pela instituição.

Quadro 13 – Caso de uso: Manter banco de dados

Nome do caso de uso	Manter banco de dados
Atores envolvidos	Instituição de Ensino
Objetivo	Permitir que a Instituição de Ensino atualize, corrija ou remova dados existentes no banco de dados, garantindo a consistência das informações utilizadas pelo sistema (disciplinas, salas, professores e horários).

Prioridade de desenvolvimento	Importante
Ações do ator	1. O administrador atualiza, edita ou remove informações (disciplinas, salas, professores, turmas, horários, cursos). 2. Confirma a atualização. 3. O sistema processa e registra as modificações no banco de dados.
Ações alternativas do ator	1. Caso os dados enviados estejam incompletos ou conflitantes, o sistema exibe uma mensagem solicitando correções antes de salvar.

Fonte: Autoria própria (2025)

Por fim, para suportar as requisições automatizadas, especialmente as provenientes da IA, é necessário um fluxo técnico de recuperação de dados. O Quadro 14 elucida como a API realiza as consultas internas ao banco de dados.

Quadro 14 – Caso de uso: Consultar banco de dados

Nome do caso de uso	Consultar banco de dados
Atores envolvidos	API Inteligência Artificial
Objetivo	Permitir que a API de Inteligência Artificial e outros módulos do sistema realizem consultas ao banco de dados para obter informações sobre horários, professores, salas, cursos e matérias, viabilizando respostas automatizadas e funcionalidade interna do sistema.
Prioridade de desenvolvimento	Essencial
Ações do ator	1. A API ou módulo interno envia uma requisição para consulta de dados ao sistema (ex.: horários, professor, sala, disponibilidade). 2. O sistema interpreta a requisição e identifica a consulta necessária. 3. O sistema acessa o banco de dados e executa a consulta SQL correspondente. 4. O banco retorna as informações solicitadas. 5. O sistema organiza os dados retornados e mostra ao usuário
Ações alternativas do ator	1. Caso haja erro de sintaxe na consulta ou problema no banco, o sistema retorna uma mensagem de falha e não envia informações incorretas.

Fonte: Autoria própria (2025)

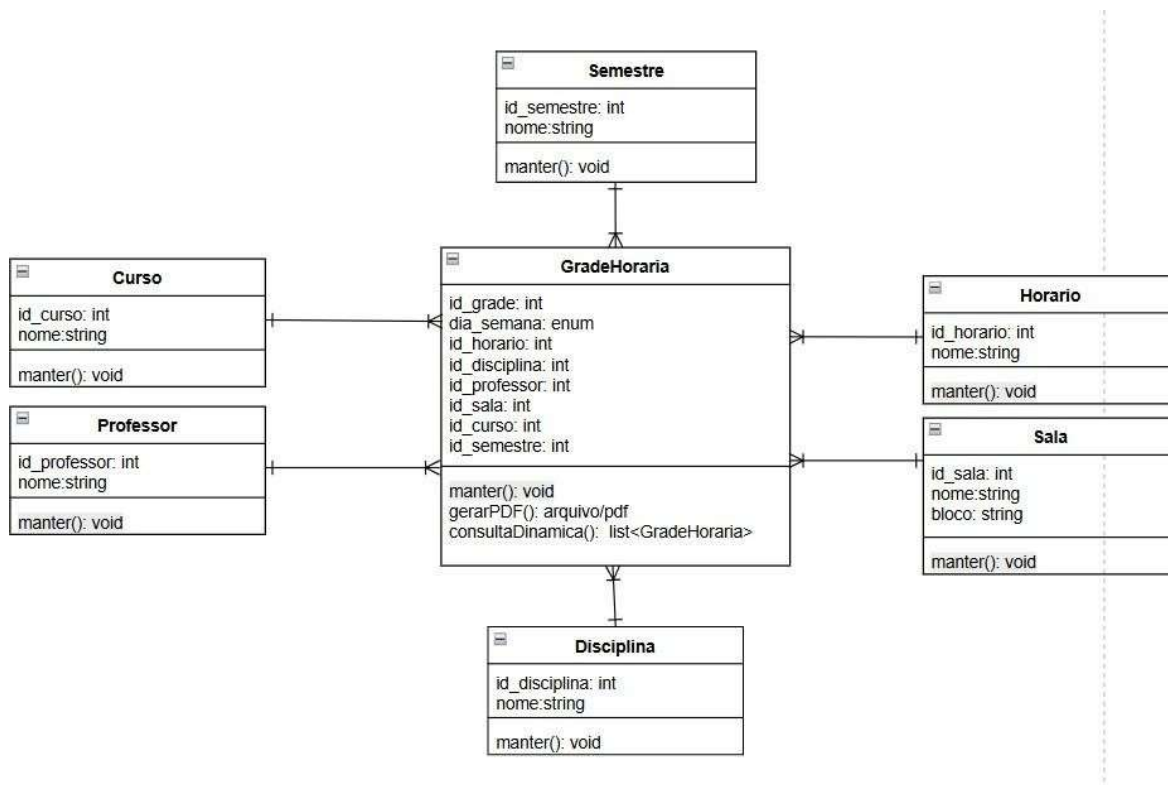
3.6 – Diagrama de Classe

O diagrama de classes é um dos tipos de diagramas de UML que fornece uma visão estática ou estrutural de um sistema. Segundo Sommerville (2018), uma classe pode ser encarada como uma definição geral de um tipo de objeto de sistema, enquanto o diagrama ilustra as classes em um sistema e as associações entre elas. Para Pressman (2016), a modelagem baseada em classes representa os objetos que o sistema irá manipular, as operações que serão aplicadas a esses objetos e as

relações entre eles. Essa representação é essencial para estabelecer uma base a partir da qual projeto pode ser conduzido.

Observam-se os elementos dispostos no diagrama de classe, na Figura 8:

Figura 8 -Diagrama de Classe do Morpi.

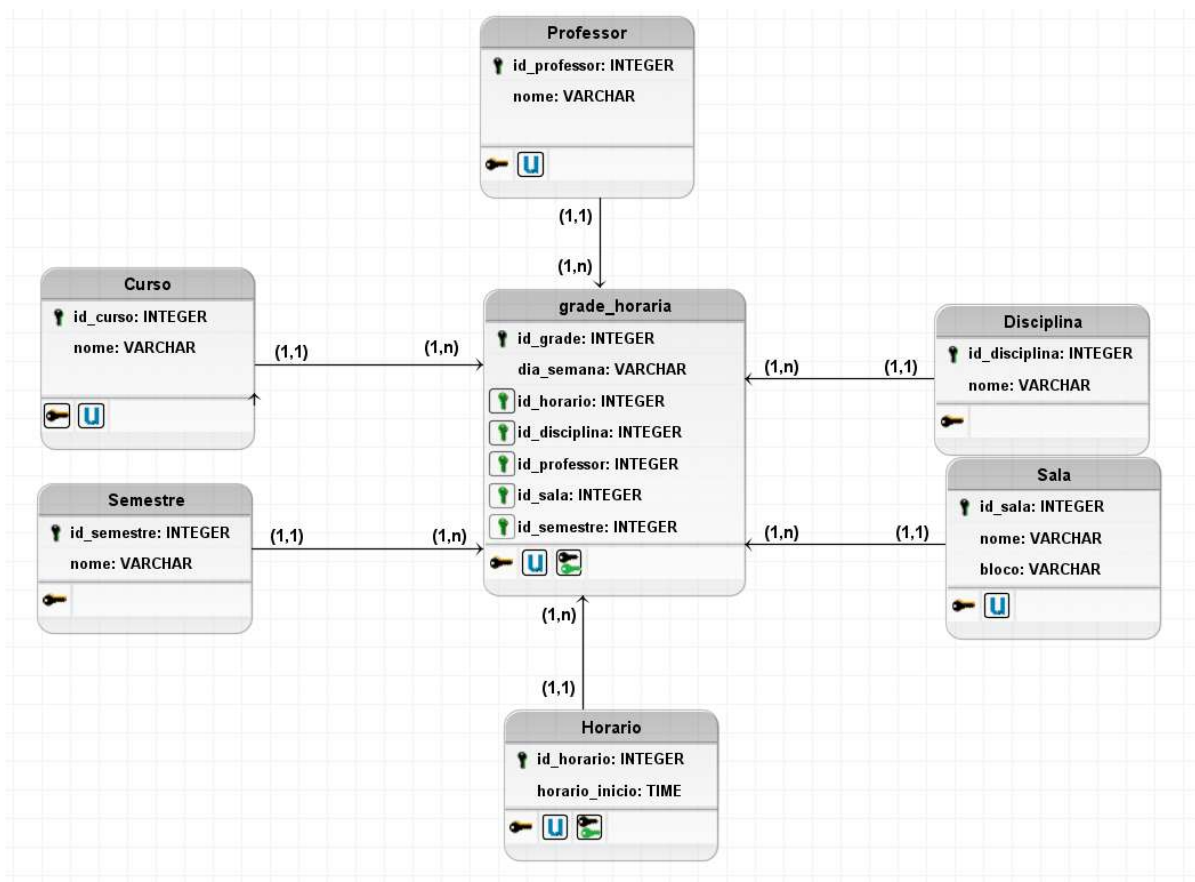


Fonte: Elaborado pelo autor.

3.7 – Modelo Entidade-Relacionamento

O Modelo de Entidade-Relacionamento (MER) transforma os requisitos estruturais do sistema em um *blueprint* lógico para o banco de dados. Sua criação é fundamental para garantir integridade, a consistência dos dados e a persistência correta das informações na aplicação web (PRESSMAN, 2016). A Figura 15 apresenta o diagrama desenvolvido para o sistema Morpi.

Figura 9 – Modelo Entidade-Relacionamento



Fonte: Elaborada pelo autor (2025).

A estrutura visa mapear as entidades acadêmicas e suas interconexões para suportar funcionalidades de consulta de horário. A arquitetura de dados é centrada na entidade `grade_horaria`, que funciona como um ponto de interconexão entre as entidades de domínio (`Professor`, `Disciplina`, `Horário`, `Sala`, `Curso`, `Semestre`). Essa entidade central é composta por chaves estrangeiras (`id_professor`, `id_disciplina`, etc.) que estabelecem as regras de integridade referencial, garantindo que um horário só possa existir se todas as suas partes constituintes (quem ensina, o que é ensinado, quando e onde) estiverem definidas (SOMMERVILLE, 2018). O modelo garante, portanto, que a base de dados suporte o requisito principal da aplicação: a busca e organização de horários complexos de forma precisa e eficiente.

3.8 – Dicionário de Dados

O Dicionário de Dados é um artefato de documentação crucial na engenharia de software, atuando como um catálogo central de metadados e uma ponte entre a definição de requisitos e a implementação do banco de dados (PRESSMAN, 2016). Ele é essencial para garantir que a arquitetura de dados seja compreendida de forma

não ambígua pela equipe de desenvolvimento, descrevendo de maneira estruturada os elementos que compõem o banco de dados (SOMMERVILLE, 2018).

Para este projeto, o Dicionário de Dados foi estruturado em formato de quadro, detalhando os elementos que compõem o banco de dados. Os campos utilizados para esta documentação possuem os seguintes significados:

- Entidade: Representa o elemento ou tabela de banco de dados sobre o qual há interesse em armazenar informações (ex: Pessoa, Objeto, Local).
- Atributo: Indica o nome de campo, conforme ele aparece em cada tabela do banco de dados.
- Domínio: Especifica o tipo de dado (ex: varchar, int, text) e seu tamanho ou restrições.
- PK – *Primary key*: informa se o atributo atua como chave primaria (identificador único) da entidade.
- FK – *Foreign key*: Indica se o atributo atua como chave estrangeira, estabelecendo relacionamento com outra entidade.
- Campo obrigatório: Define se o preenchimento do campo é mandatório (NOT NULL) ou opcional (NULL).
- Descrição: Detalha a função do atributo ou registra observações relevantes para o desenvolvimento.

Os dados da tabela “PROFESSORES” armazenam informações sobre os docentes cadastrados no sistema, conforme Quadro 15.

Quadro 15 – Dicionário de Dados: PROFESSORES

Atributo	Domínio	PK	FK	Campo obrigatório	Descrição
ID_PROFESSOR	INTEGER	PK		NOT NULL	Código de identificação do professor
NOME	VARCHAR (100)			NOT NULL	Nome completo do professor

Fonte: Autoria própria (2025)

Os dados da tabela “CURSOS” armazenam informações sobre os docentes cadastrados no sistema, conforme Quadro 16.

Quadro 16 – Dicionário de Dados: CURSOS

Atributo	Domínio	PK	FK	Campo obrigatório	Descrição
ID_CURSO	INTEGER	PK		NOT NULL	Código de identificação do curso
NOME	VARCHAR (100)			NOT NULL	Nome do curso oferecido pela instituição

Fonte: Autoria própria (2025)

Os dados da tabela “DISCIPLINAS” armazenam as informações sobre as disciplinas ministradas nos cursos, conforme Quadro 17.

Quadro 17 – Dicionário de Dados: DISCIPLINAS

Atributo	Domínio	PK	FK	Campo obrigatório	Descrição
ID_DISCIPLINA	INTEGER	PK		NOT NULL	Código de identificação da disciplina
NOME	VARCHAR (100)			NOT NULL	Nome da disciplina ministrada

Fonte: Autoria própria (2025)

Os dados da tabela “SEMESTRES” definem os períodos acadêmicos correspondentes às disciplinas e horários, conforme Quadro 18.

Quadro 18 – Dicionário de Dados: SEMESTRES

Atributo	Domínio	PK	FK	Campo obrigatório	Descrição
ID_SEMESTRE	INTEGER	PK		NOT NULL	Código de identificação do semestre
NOME	VARCHAR (50)			NOT NULL	Nome ou identificação do semestre (ex.: '1º Semestre')

Fonte: Autoria própria (2025)

Os dados da tabela “SALAS” armazenam informações referentes aos ambientes físicos onde as aulas ocorrem, conforme Quadro 19.

Quadro 19 – Dicionário de Dados: SALAS

Atributo	Domínio	PK	FK	Campo obrigatório	Descrição
ID_SALA	INTEGER	PK		NOT NULL	Código de identificação da sala
NOME	VARCHAR (50)			NOT NULL	Nome ou número identificador da sala
BLOCO	VARCHAR (20)			NOT NULL	Bloco em que a sala está localizada

Fonte: Autoria própria (2025)

Os dados da tabela “HORÁRIOS” armazenam os períodos de início das aulas, conforme Quadro 20.

Quadro 20 – Dicionário de Dados: HORARIOS

Atributo	Domínio	PK	FK	Campo obrigatório	Descrição
ID_HORARIO	INTEGE R	PK		NOT NULL	Código de identificação do horário
HORARIO_INICIO	TIME			NOT NULL	Hora de início da aula

Fonte: Autoria própria (2025)

Os dados da tabela “GRADE_HORÁRIA”, que possuem relacionamento com as tabelas “DISCIPLINAS”, “PROFESSORES”, “SALAS”, “SEMESTRES”, “HORÁRIOS” e “CURSOS” estão descritos conforme Quadro 21.

Quadro 21 – Dicionário de Dados: GRADE_HORARIA

Atributo	Domínio	PK	FK	Campo obrigatório	Descrição
ID_GRADE	INTEGER	PK		NOT NULL	Código de identificação do registro da grade
DIA_SEMANA	ENUM ('Segunda','Terça','Quarta','Quinta','Sexta','Sábado')			NOT NULL	Dia da semana referente à aula
ID_HORARIO	INTEGER		FK	NOT NULL	Código de identificação do horário da aula
ID_DISCIPLINA	INTEGER		FK	NOT NULL	Código de identificação da disciplina
ID_PROFESSOR	INTEGER		FK	NOT NULL	Código de identificação do professor
ID_SALA	INTEGER		FK	NOT NULL	Código de identificação da sala
ID_SEMESTRE	INTEGER		FK	NOT NULL	Código de identificação do semestre
ID_CURSO	INTEGER		FK	NOT NULL	Código de identificação do semestre

Fonte: Autoria própria (2025)

3.9 – Prototipação de telas

Uma versão inicial de um sistema é definida como um protótipo, cuja função principal é a demonstração de conceitos e a exploração de alternativas de design. Essa ferramenta é crucial para o desenvolvimento, pois contribui para o aprofundamento da compreensão do problema a ser resolvido e das possíveis soluções. Para garantir eficácia, o desenvolvimento desse protótipo deve ser rápido e iterativo, permitindo o controle de custos e a experimentação por parte de *stakeholders*. (SOMMERVILLE, 2018).

Com base nesses conceitos, a etapa de prototipagem deste projeto foi focada na criação de *mockups* (representações visuais) utilizando a plataforma *online* Canva. A escolha da ferramenta se justifica pela agilidade que ela proporciona na criação de telas com apelo visual próximo ao do produto, alinhando-se a necessidade de prototipação rápida.

O objetivo principal desses protótipos foi validar a proposta de *design* e a arquitetura da informação antes do início da implementação. O foco foi garantir que a interface proposta, especialmente nas seções de calendário e agenda, atingisse os requisitos de intuitividade e legibilidade definidos para o projeto.

Como ponto de partida da experiência do usuário, a interface principal foi projetada priorizando a clareza e o acesso imediato às funções essenciais, evitando a sobrecarga cognitiva. A Figura 16 apresenta a tela inicial da aplicação web Morpi, destacando a barra de busca centralizada e os atalhos rápidos de navegação.

Figura 10 – Tela inicial do projeto.



Fonte: Elaborada pelo autor (2025).

A funcionalidade central do sistema reside na organização e visualização dos cronogramas acadêmicos, que solucionam o problema da dispersão de informações. A Figura 17 exibe a interface da grade de horários, demonstrando a disposição estruturada dos dados (tabela) e a área superior dedicada aos filtros de pesquisa para refinamento da busca.

Figura 11 – Tela da grade de horários do Morpi.



Horário de Aula

Filtrar por semestre: Todos ▼

Filtrar por professor: Todos ▼

Filtrar por Disciplina: Todos ▼

Filtrar por curso: Todos ▼

Semestre	Curso	Horário	Dia da Semana	Disciplina	Professor	Sinal	Bloco

Fonte: Elaborada pelo autor (2025).

4 FASE FINAL DO SISTEMA

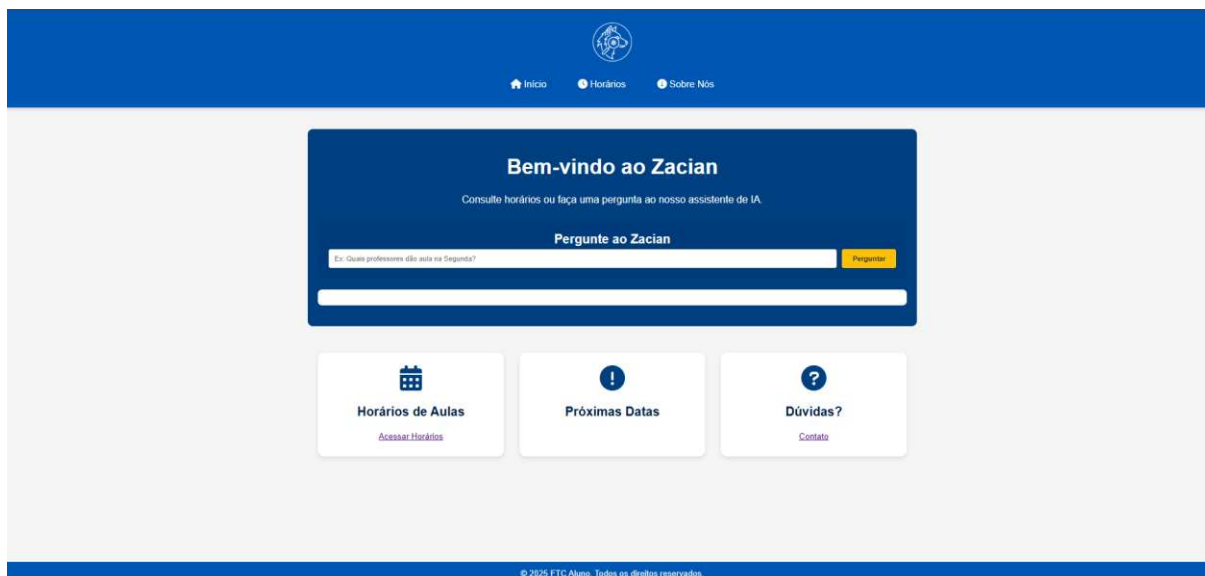
A fase final do desenvolvimento do sistema Morpi concentrou-se na demonstração e validação do artefato web, assegurando que o produto estivesse adequado à finalidade conforme os requisitos funcionais e não funcionais estabelecidos. A engenharia de software define essa etapa como o momento em que se obtém a garantia final de que o sistema satisfaz as necessidades do usuário, respondendo se está sendo construído o produto certo. (PRESSMAN, 2016). Conforme o ciclo de desenvolvimento incremental adotado, a conclusão desta etapa representa a disponibilização do incremento de software funcional (SOMMERVILLE, 2018), sendo a conclusão bem-sucedida de um artefato proposto.

A seguir serão apresentadas as telas finais e funcionalidades implementadas pelo sistema, em pleno funcionamento, documentados por meio de capturas de tela.

As funcionalidades essenciais foram desenvolvidas para serem acessíveis sem a necessidade de autenticação. Essa decisão de projeto se fundamenta no princípio de simplicidade e rápida utilização, visando eliminar a carga cognitiva imposta ao usuário e permitindo acesso imediato e direto às informações, otimizando a experiência do discente.

A Figura 18 apresenta a tela inicial da aplicação web Morpi. Esta tela é o ponto de acesso central e foi desenhada para comunicar, de imediato, o foco do sistema, direcionando o usuário para as funcionalidades.

Figura 12 – Tela inicial do Morpi.



Fonte: Elaborada pelo autor (2025).

O *design* da tela inicial demonstra uma forte adesão aos princípios de usabilidade e arquitetura de informação, conforme o referencial teórico estabelecido no capítulo 2. Essa interface prioriza a simplicidade, utilizando uma paleta de cores limpa e eliminando informações desnecessárias. Em decorrência dessa escolha, o foco principal do usuário é imediatamente dirigido ao campo de busca centralizado, que representa a funcionalidade de maior valor e cumpre a heurística de minimalismo e visibilidade do sistema.

A Figura 19 apresenta a tela de resultados após uma consulta submetida ao assistente de IA do sistema. Esta funcionalidade permite que o usuário interaja com o banco de dados usando linguagem natural. Para viabilizar essa interação com alto desempenho, o sistema utiliza a API da Groq, processando Llama 3. Essa escolha foi definida pela viabilidade econômica, uma vez que a plataforma oferece acesso gratuito robusto, ideal para a prototipagem e desenvolvimento no contexto acadêmico.

Figura 13 – Tela de demonstração da consulta por IA e resultados.



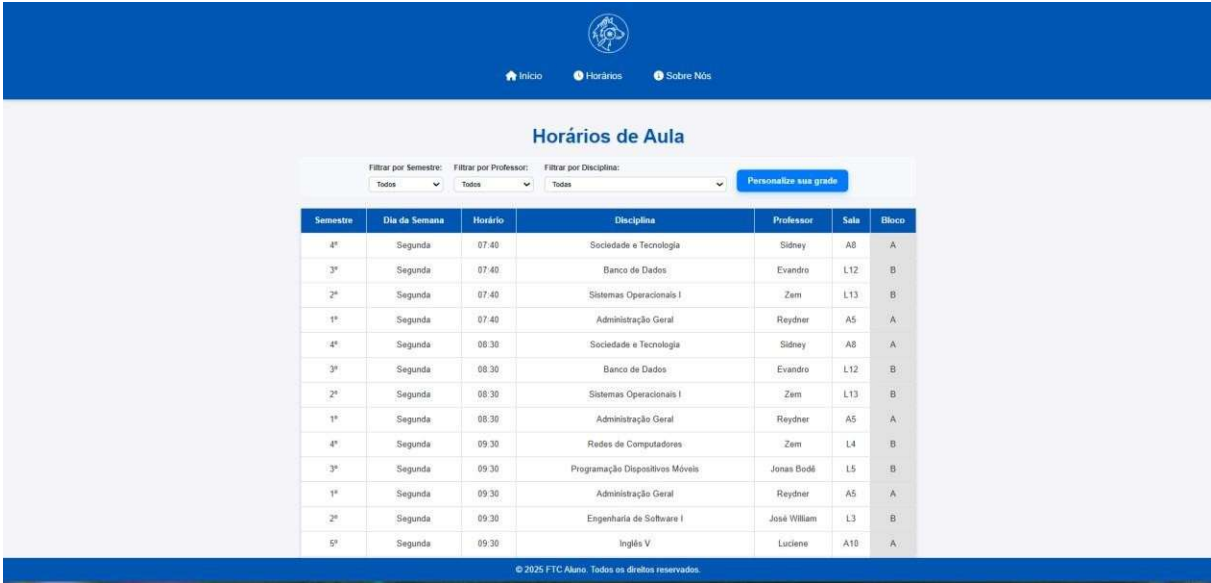
Fonte: Elaborada pelo autor (2025).

A interface de resultados demonstra alta visibilidade do sistema e controle do usuário. O sistema fornece *feedback* imediato ao exibir o resultado da pergunta, garantindo que o usuário saiba que a ação foi concluída com sucesso. Além disso o campo de busca permanece ativo, permitindo que o usuário refine a consulta ou inicie uma nova sem quebrar fluxo.

Esta tela revela também uma violação crítica da heurística de Correspondência entre o Sistema e o Mundo Real: a exibição da *query* gerada pela IA. Ela foi mantida no beta em prol de questões de acompanhamento de como a API estava gerando a *query*, e será removida na versão final do produto.

A Figura 20 apresenta a interface principal para visualização e organização dos horários de aula. Essa tela tem a função de estruturar e permitir a manipulação dos dados da grade curricular de forma eficiente, resolvendo o problema de organização identificado na fase de análise.

Figura 14 – Interface de visualização horários e filtros.



The screenshot shows a web interface titled "Horários de Aula". At the top, there is a navigation bar with links for "Início", "Horários", and "Sobre Nós". Below the navigation bar, there are three filter dropdowns: "Filtrar por Semestre:", "Filtrar por Professor:", and "Filtrar por Disciplina:". Each dropdown has a "Todos" option. To the right of these filters is a button labeled "Personalize sua grade". Below the filters is a table with the following columns: Semestre, Dia da Semana, Horário, Disciplina, Professor, Sala, and Bloco. The table contains 18 rows of data, showing a repeating pattern of classes for each day of the week (Segunda).

Semestre	Dia da Semana	Horário	Disciplina	Professor	Sala	Bloco
4º	Segunda	07:40	Sociedade e Tecnologia	Sidney	A8	A
3º	Segunda	07:40	Banco de Dados	Evandro	L12	B
2º	Segunda	07:40	Sistemas Operacionais I	Zem	L13	B
1º	Segunda	07:40	Administração Geral	Reydner	A5	A
4º	Segunda	08:30	Sociedade e Tecnologia	Sidney	A8	A
3º	Segunda	08:30	Banco de Dados	Evandro	L12	B
2º	Segunda	08:30	Sistemas Operacionais I	Zem	L13	B
1º	Segunda	08:30	Administração Geral	Reydner	A5	A
4º	Segunda	09:30	Redes de Computadores	Zem	L4	B
3º	Segunda	09:30	Programação Dispositivos Móveis	Jonas Bodê	L5	B
1º	Segunda	09:30	Administração Geral	Reydner	A5	A
2º	Segunda	09:30	Engenharia de Software I	José William	L3	B
5º	Segunda	09:30	Inglês V	Luciene	A10	A

At the bottom of the interface, there is a copyright notice: "© 2025 FTC Aluno. Todos os direitos reservados."

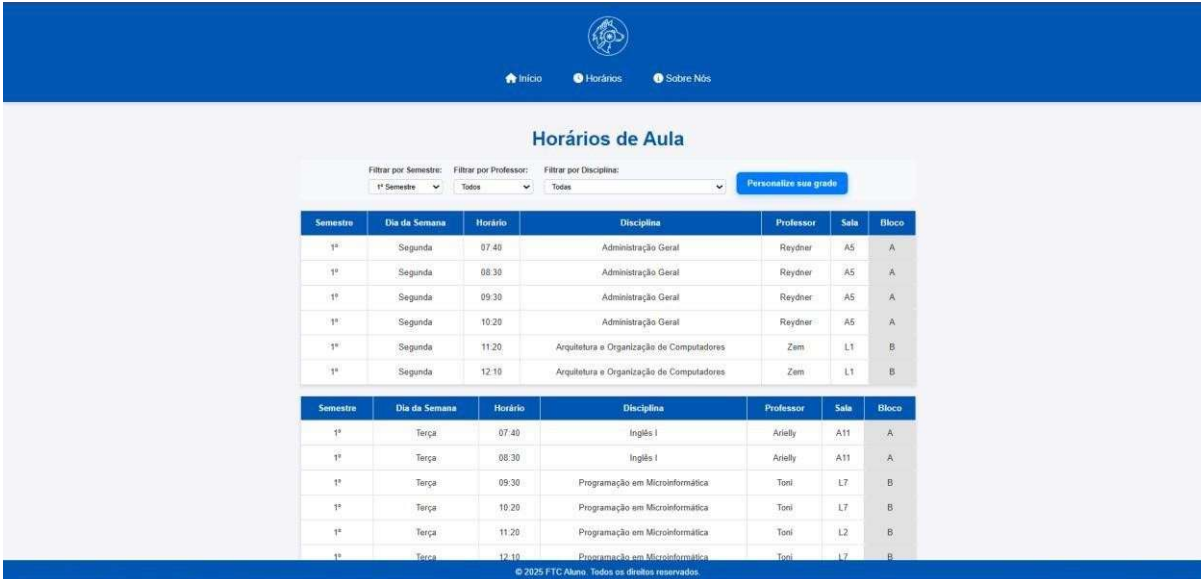
Fonte: Elaborada pelo autor (2025).

O módulo de horários de aula constitui a funcionalidade de maior valor do sistema. O *design* desta interface foi construído para superar as limitações identificadas em sistemas similares (GDE e Jupiter) ao integrar três processos essenciais: filtragem, personalização e exportação.

Inicialmente, a *interface* reforça a heurística de reconhecimento em vez de memória e flexibilidade ao oferecer múltiplos filtros (“Semestre”, “Professor”, “Disciplina”) baseados em linguagem natural. Essa abordagem permite ao usuário manipular a densidade de dados da grade curricular de forma eficiente, sem a necessidade de memorizar códigos de disciplina ou de sala. As Figura 21, 22 e 23 demonstram a aplicação desses filtros em diferentes níveis (semestre, professor e disciplina, respectivamente). O filtro atua como um recurso de recuperação de erros indireto, ao reduzir a carga de informações e evitar a exposição a dados irrelevantes de outros períodos, garantindo que o usuário se concentre apenas no seu domínio de interesse.

Para demonstrar a capacidade de segmentação por período letivo, a Figura 21 exibe o comportamento da grade após a aplicação do filtro por semestre. Esse recurso permite que o aluno visualize apenas o conjunto de disciplinas pertinentes ao seu estágio atual no curso, reduzindo a carga de informação na tela.

Figura 15 – Tela da grade após seleção de filtros “Filtro por semestre”.



Horários de Aula

Filtrar por Semestre: 1º Semestre | Filtrar por Professor: Todos | Filtrar por Disciplina: Todos | Personalize sua grade

Semestre	Dia da Semana	Horário	Disciplina	Professor	Sala	Bloco
1ª	Segunda	07:40	Administração Geral	Reydnar	A5	A
1ª	Segunda	08:30	Administração Geral	Reydnar	A5	A
1ª	Segunda	09:30	Administração Geral	Reydnar	A5	A
1ª	Segunda	10:20	Administração Geral	Reydnar	A5	A
1ª	Segunda	11:20	Arquitetura e Organização de Computadores	Zem	L1	B
1ª	Segunda	12:10	Arquitetura e Organização de Computadores	Zem	L1	B

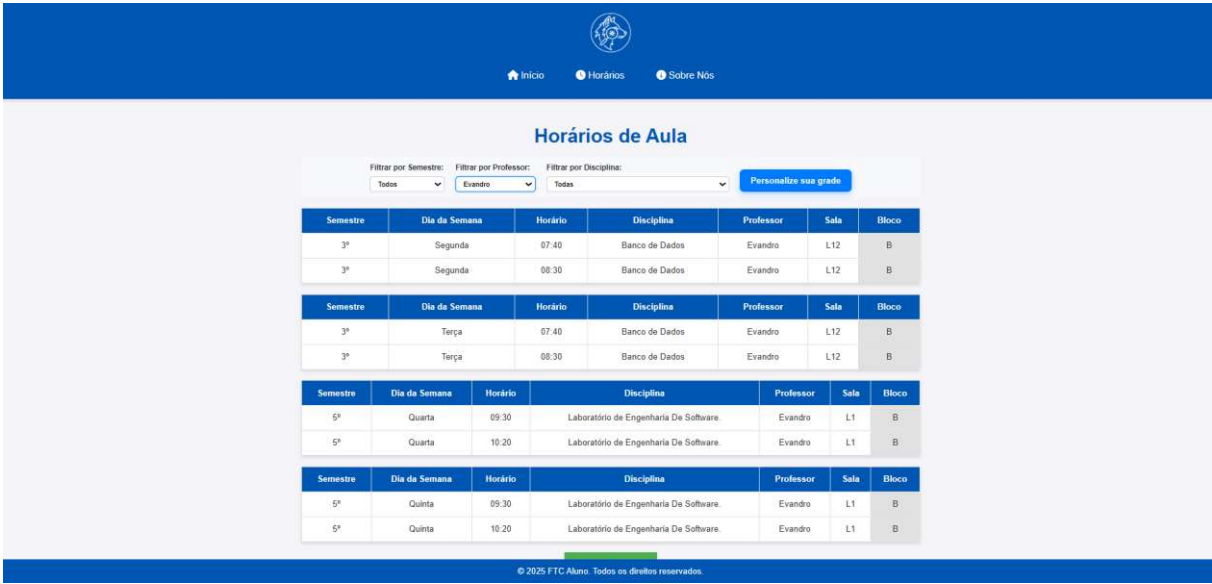
Semestre	Dia da Semana	Horário	Disciplina	Professor	Sala	Bloco
1ª	Terça	07:40	Inglês I	Arielly	A11	A
1ª	Terça	08:30	Inglês I	Arielly	A11	A
1ª	Terça	09:30	Programação em Microinformática	Toni	L7	B
1ª	Terça	10:20	Programação em Microinformática	Toni	L7	B
1ª	Terça	11:20	Programação em Microinformática	Toni	L2	B
1ª	Terça	12:10	Programação em Microinformática	Toni	L7	B

© 2025 FTC Aluno. Todos os direitos reservados.

Fonte: Elaborada pelo autor (2025).

Outra possibilidade de refinamento é a busca direcionada pelo corpo docente, facilitando a localização de aulas específicas de um educador. A Figura 22 ilustra o resultado da interface quando o filtro por professor é acionado, isolando os horários e locais vinculados ao docente selecionado.

Figura 16 – Tela da grade após seleção de “Filtro por professor”.



Horários de Aula

Filtrar por Semestre: Todos | Filtrar por Professor: Evandro | Filtrar por Disciplina: Todos | Personalize sua grade

Semestre	Dia da Semana	Horário	Disciplina	Professor	Sala	Bloco
3ª	Segunda	07:40	Banco de Dados	Evandro	L12	B
3ª	Segunda	08:30	Banco de Dados	Evandro	L12	B

Semestre	Dia da Semana	Horário	Disciplina	Professor	Sala	Bloco
3ª	Terça	07:40	Banco de Dados	Evandro	L12	B
3ª	Terça	08:30	Banco de Dados	Evandro	L12	B

Semestre	Dia da Semana	Horário	Disciplina	Professor	Sala	Bloco
5ª	Quarta	09:30	Laboratório de Engenharia De Software	Evandro	L1	B
5ª	Quarta	10:20	Laboratório de Engenharia De Software	Evandro	L1	B

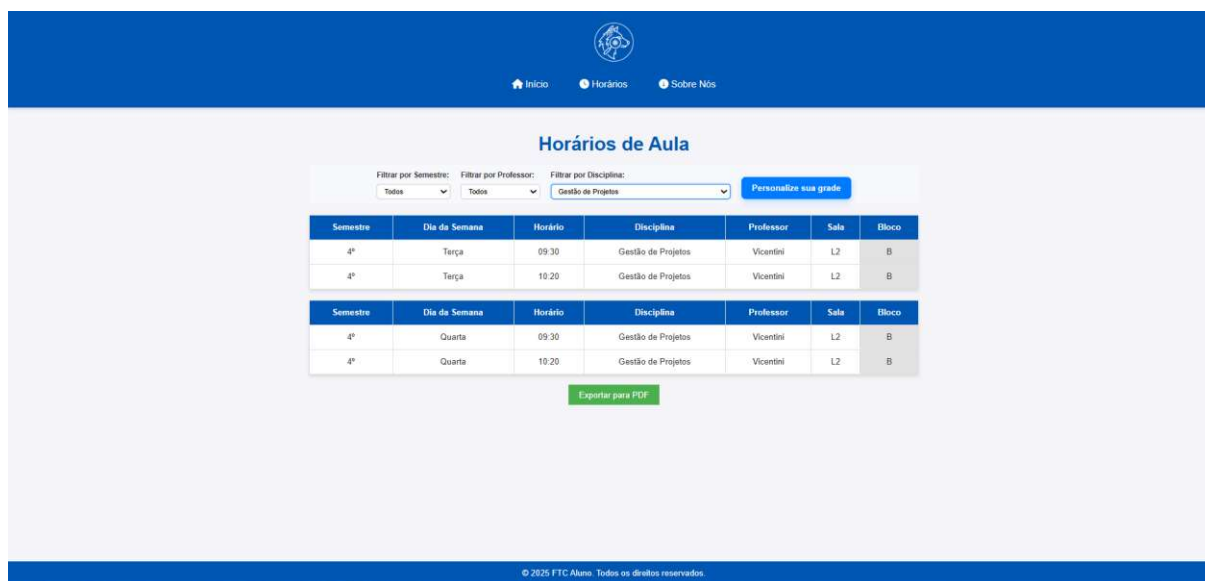
Semestre	Dia da Semana	Horário	Disciplina	Professor	Sala	Bloco
5ª	Quinta	09:30	Laboratório de Engenharia De Software	Evandro	L1	B
5ª	Quinta	10:20	Laboratório de Engenharia De Software	Evandro	L1	B

© 2025 FTC Aluno. Todos os direitos reservados.

Fonte: Elaborada pelo autor (2025).

Por fim, para uma consulta mais granular focada em um componente curricular específico, o sistema permite a triagem direta por matéria. A Figura 23 apresenta a grade de horários restrita à disciplina escolhida, eliminando dados irrelevantes e agilizando a identificação da sala e horário desejados.

Figura 17 – Tela de grade após seleção de “Filtro por disciplina”.



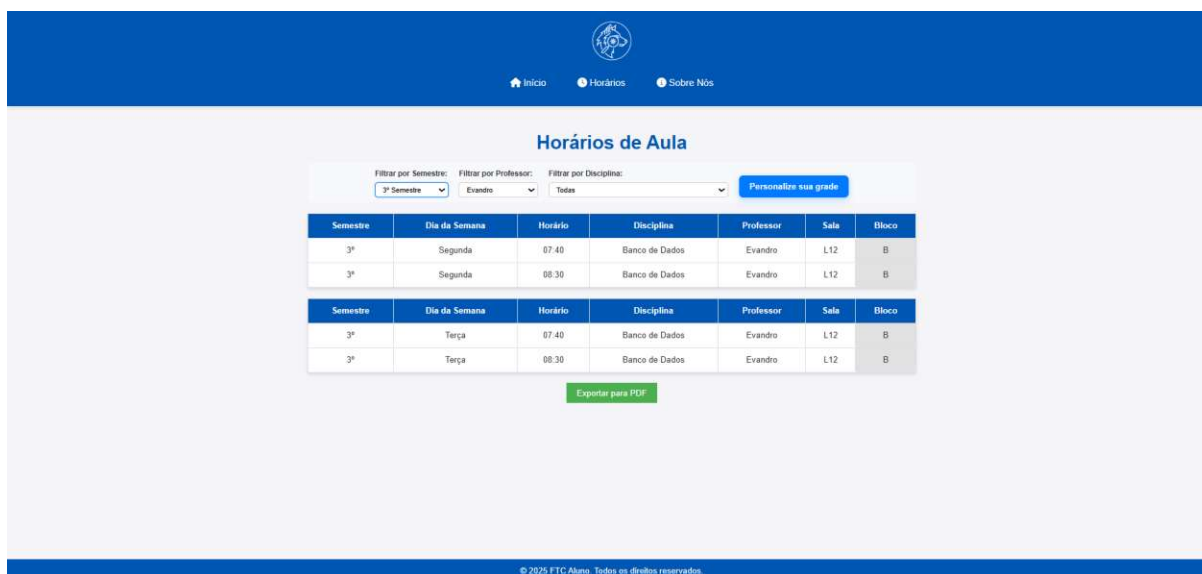
Semestre	Dia da Semana	Horário	Disciplina	Professor	Sala	Bloco
4º	Terça	09:30	Gestão de Projetos	Vicentini	L2	B
4º	Terça	10:20	Gestão de Projetos	Vicentini	L2	B

Semestre	Dia da Semana	Horário	Disciplina	Professor	Sala	Bloco
4º	Quarta	09:30	Gestão de Projetos	Vicentini	L2	B
4º	Quarta	10:20	Gestão de Projetos	Vicentini	L2	B

Fonte: Elaborada pelo autor (2025).

A Figura 24 demonstra a aplicação de filtros em múltiplos níveis, validando a heurística de flexibilidade e eficiência de uso, uma vez que oferece caminhos rápidos para o usuário especializado encontrar dados específicos, sem ter que navegar por toda a grade. A exibição estruturada e persistência dos nomes de professores e disciplinas minimizam a necessidade de memorizar códigos, que foi uma das falhas identificadas.

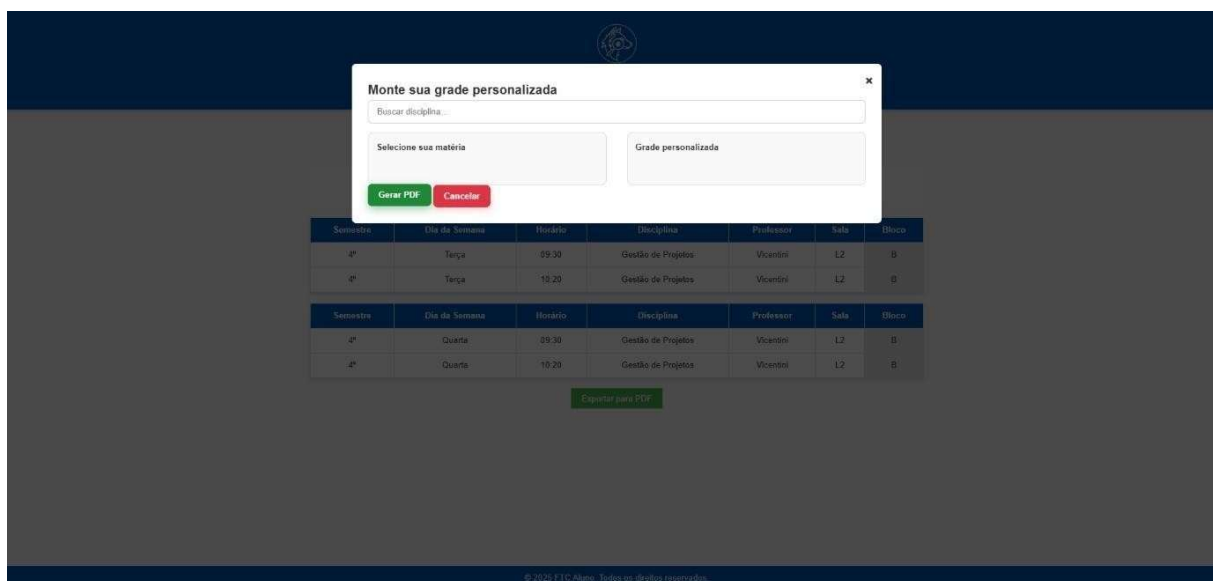
Figura 18 – Tela de grade após seleção de “Filtro por disciplina e professor”.



Fonte: Elaborada pelo autor (2025).

A Figura 25 exibe a tela pronta para a ação mais importante: a criação de uma grade personalizada. O *design* usa consistência de filtros para estruturar a busca e, em seguida, direciona o olhar do usuário ao botão “Personalize sua grade”. Este botão atua como um acelerador de funcionalidade, indicando ao usuário que ele tem a liberdade de sair da visualização estática e iniciar a manipulação e construção do artefato.

Figura 19 – Transição para tela de personalização.

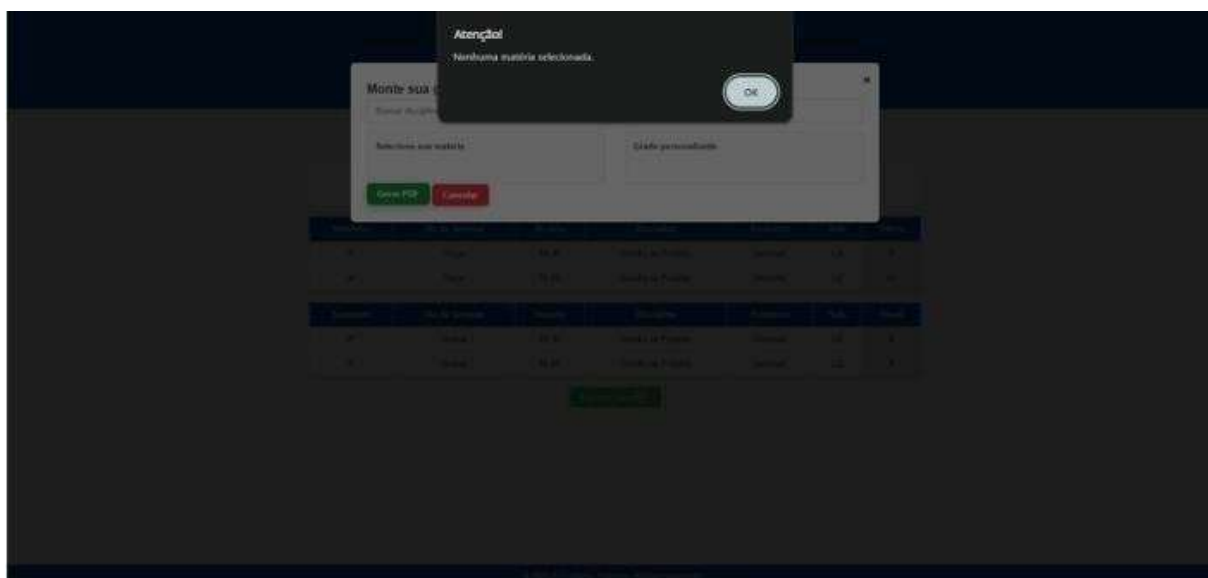


Fonte: Elaborada pelo autor (2025).

Para que não ocorra o erro de um arquivo ser gerado em branco, a Figura 26

exibe o mecanismo que impede que o usuário execute a ação sem antes satisfazer a condição necessário (seleção de uma ou mais matérias). Esse alerta oferece a aplicação de visibilidade do sistema, pois a mensagem “nenhuma matéria selecionada” é clara e direta, utiliza de linguagem familiar e permite que o usuário diagnostique e se recupere do erro imediatamente.

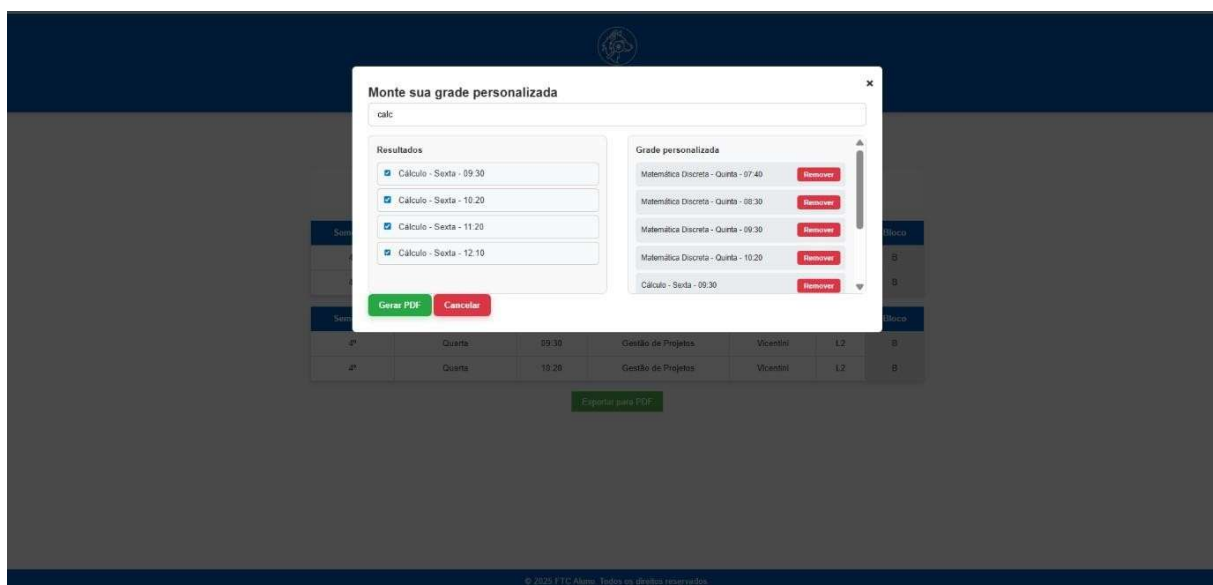
Figura 20 – Tela de prevenção de erro.



Fonte: Elaborada pelo autor (2025).

A Figura 27 apresenta o modal da personalização de grade. O usuário pode buscar disciplinas, ver os resultados disponíveis e transferir itens para a grade personalizada com *feedback* imediato (confirmação do item adicionado). Esta interatividade cumpre o objetivo de proporcionar maior dinâmica na montagem dos horários.

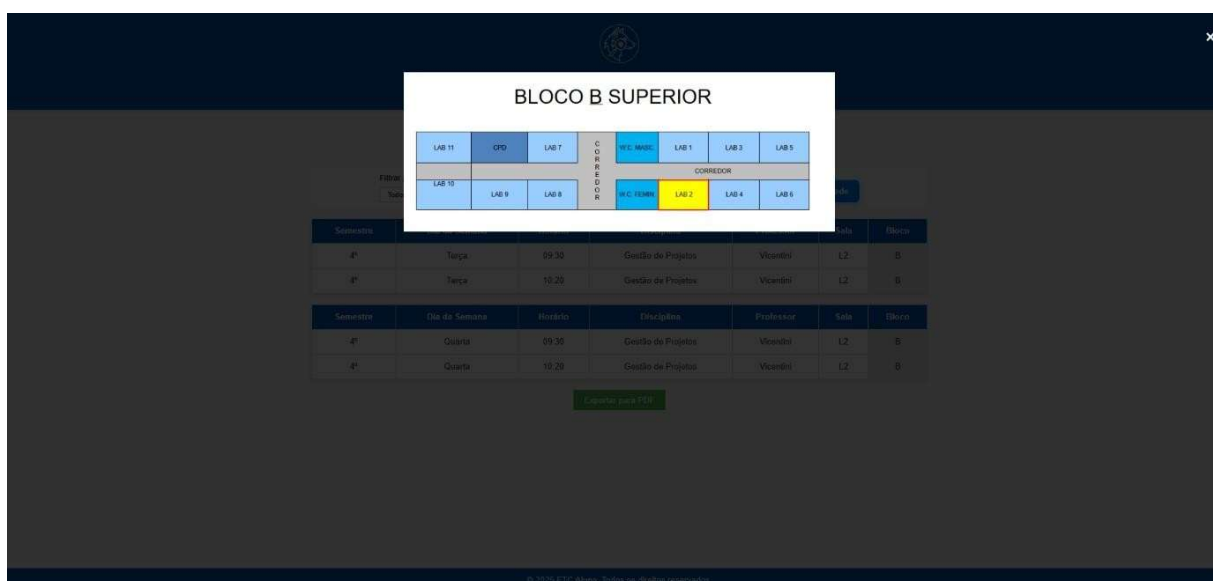
Figura 21 – Tela de montagem da grade personalizada.



Fonte Elaborada pelo autor (2025).

A Figura 28 ilustra a visualização do mapa que surge ao selecionar uma localização na grade. Esta funcionalidade é uma aplicação direta e poderosa da heurística de correspondência entre o sistema e o mundo real. Ao invés de forçar os usuários a memorizarem códigos abstratos de salas, o sistema fornece o contexto geográfico imediato, associando o dado da grade curricular a localização física baseado na planta do campus.

Figura 22 – Tela de visualização da sala no mapa.



Fonte: Elaborada pelo autor (2025).

A Figura 29 demonstra o resultado do processo de personalização: a grade formatada para impressão (PDF). Esta tela valida a utilidade e eficiência do artefato.

Após todo o processo de filtragem e escolha de disciplinas, o sistema fornece um output em um formato limpo, estruturado por dia da semana e horário, confirmando o cumprimento do objetivo de disponibilizar ao usuário a opção de baixar uma grade de horários personalizada.

Figura 23 – Grade formatada em PDF para impressão.

Grade Personalizada

Quinta

Disciplina	Horário
Matemática Discreta	07:40
Matemática Discreta	08:30
Matemática Discreta	09:30
Matemática Discreta	10:20

Sexta

Disciplina	Horário
Cálculo	09:30
Cálculo	11:20
Cálculo	10:20
Cálculo	12:10

Fonte: Elaborada pelo autor (2025).

A apresentação detalhada das interfaces e funcionalidades do sistema Morpi demonstradas nas figuras 19 a 29 validam a aplicação de formas mais eficientes, simples e respeitosas ao controle do usuário, superando limitações encontradas nos sistemas similares.

5 CONSIDERAÇÕES FINAIS.

O presente trabalho teve como objetivo principal desenvolver uma aplicação web voltada à exibição e organização da grade de horários de uma instituição de ensino, mitigando problemas decorrentes da volatilidade de cronogramas e da forma inadequada de distribuição de dados em sistemas vigentes. A partir da apresentação e análise dos dados obtidos no Capítulo 4, observa-se o cumprimento de todos os objetivos específicos, como a implementação de filtros para personalização, a disponibilização da agenda para *download* (PDF), e a inclusão da ferramenta de inteligência artificial. A solução demonstrou ser eficaz, culminando na conclusão bem-sucedida de um aplicativo funcional que demonstrou aderência aos requisitos estabelecidos.

Arelada à validação das funcionalidades, a análise heurística confirmou que a sistema Morpi demonstrou formas mais eficientes, simples e respeitosas ao controle do usuário, superando limitações encontradas nos sistemas similares. Outra questão importante diz respeito ao sucesso na construção de uma ferramenta intuitiva e com acesso imediato, validando a prioridade de usabilidade estabelecida. Contudo, o processo evidenciou a necessidade de refino contínuo. Como limitação, destaca-se que a exibição da *query* SQL na tela de busca por IA, embora mantida para acompanhamento técnico no beta, revela uma falha na heurística de Correspondência entre o Sistema e o mundo real, sendo um elemento a ser removido para a versão final. Adicionalmente, são discutidas as limitações encontradas durante o desenvolvimento, como o tempo demandado para análise de reestruturação do Banco de Dados.

Em suma, o trabalho contribuiu para o campo da Engenharia de Software ao entregar um produto avaliado que serve como modelo para soluções de gestão acadêmica centradas na usabilidade. Para trabalhos futuros, sugere-se, primordialmente, a implementação de um sistema de autenticação e login de usuário para garantir a persistência das informações de grade personalizada. Além disso propõe-se a expansão das funcionalidades API de inteligência artificial e a conversão da aplicação web para plataformas moveis garantindo maior acessibilidade para o usuário final.

REFERÊNCIAS BIBLIOGRÁFICAS

CENTRO PAULA SOUZA. **SIGA: Sistema Integrado de Gestão Acadêmica**. São Paulo, [s.d.]. Disponível em: <https://sigas.cps.sp.gov.br/sigaaaluno/applogin.aspx>. Acesso em: 12 nov. 2025.

DUCKETT, Jon. **HTML & CSS: projete e construa websites**. Rio de Janeiro: Alta Books, 2016.

FLANAGAN, David. **JavaScript: o guia definitivo**. 6. ed. Porto Alegre: Bookman, 2013.

GOOGLE. **Gemini**. Versão 1.5. [Modelo de Linguagem Grande]. Mountain View: Google, 2025. Disponível em: <https://gemini.google.com/>. Acesso em: 19 nov. 2025.

GROQ. **GroqCloud API**. [Plataforma de Inferência de IA]. Mountain View: Groq Inc., 2025. Disponível em: <https://groq.com/>. Acesso em: 19 nov. 2025.

META AI. **Llama 3**. [Modelo de Linguagem Grande]. Menlo Park: Meta, 2024. Disponível em: <https://llama.meta.com/>. Acesso em: 19 nov. 2025.

NIELSEN, Jakob. **Usability Engineering**. San Francisco: Morgan Kaufmann, Inc., 1993.

NIELSEN, Jakob. **Projetando websites**. Oxford, England: Gulf Professional Publishing, 2000.

PRESSMAN, Roger S.; MAXIM, Bruce R. **Engenharia de Software: Uma Abordagem Profissional**. Tradução de João Eduardo Nóbrega Tortello. Revisão técnica de Reginaldo Arakaki, Julio Arakaki e Renato Manzan de Andrade. 8. ed. Porto Alegre: AMGH, 2016.

SOMMERVILLE, Ian. **Engenharia de software**. Tradução: Luiz Claudio Queiroz. Revisão técnica: Fábio Levy Siqueira. 10. Ed. São Paulo: Pearson Education do Brasil, 2018.

UNIVERSIDADE DE SÃO PAULO. **Júpiter Web**. São Paulo: IME-USP, [s.d.]. Disponível em: <https://www.ime.usp.br/jupiter-web/>. Acesso em: 12 nov. 2025.

UNIVERSIDADE ESTADUAL DE CAMPINAS. **GRADE: Gerenciamento Acadêmico Digital de Ensino**. Campinas, [s.d.]. Disponível em: <https://grade.daconline.unicamp.br/login/>. Acesso em: 12 nov. 2025.

WELLING, Luke; THOMSON, Laura. **PHP and MySQL Web Development**. 2. ed. Indianapolis: Sams Publishing