



Universidade de Brasília - UnB

Faculdade de Ciências e Tecnologias em Engenharia - FCTE

Engenharia de Software

Sistema de Planejamento Acadêmico Assistido

Autora: Hellen Faria

Professor Orientador: Dr. Henrique Gomes de Moura

Brasília, DF

2025



Hellen Faria

Sistema de Planejamento Acadêmico Assistido

Projeto Experimental submetido ao curso de graduação em Engenharia de Software da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software

Universidade de Brasília - UnB

Faculdade de Ciências e Tecnologias em Engenharia - FCTE

Orientador: Dr. Henrique Gomes de Moura

Brasília, DF

2025

Hellen Faria

Sistema de Planejamento Acadêmico Assistido/ Hellen Faria. – Brasília, DF, 2025-

107 p. : il. (algumas color.) ; 30 cm.

Orientador: Dr. Henrique Gomes de Moura

TCC – Universidade de Brasília - UnB

Faculdade de Ciências e Tecnologias em Engenharia - FCTE , 2025.

1. planejamento. 2. assistido. I. Dr. Henrique Gomes de Moura. II. Universidade de Brasília. III. Faculdade de Ciências e Tecnologias em Engenharia. IV. Sistema de Planejamento Acadêmico Assistido

CDU 02:141:005.6

Hellen Faria

Sistema de Planejamento Acadêmico Assistido

Projeto Experimental submetido ao curso de graduação em Engenharia de Software da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software

Aprovado em Brasília, DF, 24 de julho de 2025:

Dr. Henrique Gomes de Moura
Orientador

Dr. Wander Cleber Maria Pereira da Silva
Convidado 1

Dr. Mauricio Serrano
Convidado 2

Brasília, DF
2025

Dedico este projeto aos meus pais, Valter Solano e Meirilene Vigorito, pelo apoio incondicional ao longo de toda a minha jornada. Estiveram sempre ao meu lado, me incentivando em minhas decisões e me ajudando a seguir o caminho certo sempre que precisei.

(Hellen Faria)

Agradecimentos

Gostaria de expressar minha profunda gratidão à Faculdade de Ciências e Tecnologias em Engenharia (FCTE) da Universidade de Brasília (UnB) por ter me proporcionado um ambiente acadêmico enriquecedor e todos os recursos necessários para o desenvolvimento deste trabalho. Sou imensamente grata a todos os professores do curso de Engenharia de Software, cujos ensinamentos foram a base sólida da minha formação.

Um agradecimento especial vai ao meu orientador, Dr. Henrique Gomes de Moura pelo apoio inestimável, paciência e contribuições essenciais em cada etapa deste projeto. Além disso, gostaria de destacar a fundamental ajuda do Lucas Heller, que me auxiliou incansavelmente na escrita e no desenvolvimento de todo o trabalho, sendo um suporte crucial para a sua concretização.

*"Por vezes sentimos que aquilo que fazemos
não é senão uma gota de água no mar.
Mas o mar seria menor se lhe faltasse uma gota."
(Madre Teresa de Calcuta)*

Resumo

O planejamento e a organização do percurso acadêmico são fatores decisivos para o sucesso dos estudantes nas universidades públicas, podendo favorecer a conclusão do curso ou contribuir para a evasão. Esse planejamento envolve a organização de horários, a escolha de disciplinas e a gestão da carga horária, aspectos que afetam diretamente a trajetória acadêmica. Uma pesquisa conduzida com estudantes da Universidade de Brasília revelou dificuldades recorrentes no planejamento acadêmico, tais como conflitos de horários (87%) e a clara demanda por uma organização estratégica mais eficiente (74%). Com base nesses dados, este estudo propõe e **apresenta o desenvolvimento de um sistema automatizado** para o planejamento acadêmico, projetado para considerar critérios como pré-requisitos, distribuição equilibrada de disciplinas e preferências individuais. A **implementação e a validação** da solução confirmaram funcionalidades como a geração automática de grades e a detecção de conflitos, atestando a viabilidade prática do sistema. Os resultados obtidos indicaram que 91,7% dos participantes apoiaram a ferramenta, reconhecendo sua utilidade na redução da sobrecarga e na melhoria da gestão do tempo. Conclui-se, portanto, que o sistema desenvolvido pode contribuir significativamente para a redução da evasão universitária, ao oferecer uma solução ajustável às necessidades dos estudantes.

Palavras-chave: planejamento acadêmico, percurso acadêmico, grade horária, sistema automatizado.

Abstract

The planning and organization of the academic path are decisive factors for student success in public universities, potentially favoring course completion or contributing to dropout rates. This planning involves scheduling, course selection, and workload management, aspects that directly affect the academic trajectory. A survey conducted with students from the University of Brasília revealed recurring difficulties in academic planning, such as timetable conflicts (87%) and a clear demand for more efficient strategic organization (74%). Based on these data, this study proposes and **presents the development of an automated system** for academic planning, designed to consider criteria such as prerequisites, balanced distribution of courses, and individual preferences. The **implementation and validation** of the solution confirmed functionalities such as automatic timetable generation and conflict detection, attesting to the system's practical feasibility. The results obtained indicated that 91.7% of participants supported the tool, recognizing its usefulness in reducing overload and improving time management. It is concluded, therefore, that the developed system can contribute significantly to reducing university dropout rates by offering a solution adjustable to the students' needs.

Key-words: academic planning, academic journey, class schedule, automated system.

Lista de ilustrações

Figura 1 – Fluxo detalhado do gerenciamento de riscos no Scrum, desde a concepção do projeto até a iteração contínua.	20
Figura 2 – Estrutura modular do SIGAA e suas principais funcionalidades acadêmicas e administrativas.	22
Figura 3 – Estrutura e funcionamento do padrão arquitetural Model-View-Controller (MVC).	24
Figura 4 – Exemplo de Diagrama Entidade-Relacionamento (DE-R) representando conexões entre entidades em um banco de dados.	28
Figura 5 – Ilustração do funcionamento do algoritmo K-Nearest Neighbors (KNN)	36
Figura 6 – Interface do HoráriosUFCG exibindo a grade horária com disciplinas selecionadas	38
Figura 7 – Tela do GradeUFOP apresentando grades curriculares geradas automaticamente	40
Figura 8 – Distribuição dos respondentes da pesquisa por campus da Universidade de Brasília, destacando a predominância de participantes do Campus Gama.	44
Figura 9 – Métodos e ferramentas utilizadas pelos estudantes para planejamento da grade horária, evidenciando a preferência por soluções digitais. . . .	45
Figura 10 – Principais dificuldades relatadas pelos estudantes na montagem da grade curricular, incluindo conflitos de horário e indisponibilidade de vagas. .	46
Figura 11 – Percepções dos estudantes sobre a importância do planejamento acadêmico estratégico. Onde maioria dos respondentes acreditam na importância de um planejamento a longo prazo.	47
Figura 12 – Representação da Arquitetura do Sistema dividido em Camadas Principais: Interface do Usuário, Lógica de Negócios (<i>BLL</i>) e Acesso a Dados (<i>DAL</i>).	55
Figura 13 – Diagrama de Entidade-Relacionamento (DER) - Representação das Entidades e Relacionamentos do Sistema.	56
Figura 14 – Diagrama Lógico de Dados (DLD) - Estrutura de Tabelas, Relacionamentos e Tipos de Dados do Sistema.	59
Figura 15 – Interface Inicial do SPAA - Visão Geral das Disciplinas e Funcionalidades do Sistema.	60
Figura 16 – Interface de Configuração da Grade Horária - Seleção de Preferências Acadêmicas e Geração Automática de Grade no SPAA.	61
Figura 17 – Interface de Visualização do Fluxograma Acadêmico - Planejamento da Sequência de Disciplinas no SPAA.	62

Figura 18 – Estrutura de pastas dos projetos de testes unitários e de integração. . .	65
Figura 19 – Relatório consolidado da ferramenta de cobertura de testes, exibindo a porcentagem de linhas e ramificações de código cobertas.	66
Figura 20 – Estrutura de diretórios dos testes E2E.	67
Figura 21 – Relatório de execução gerado pelo Robot Framework.	70
Figura 22 – Representação Gráfica dos Perfis Profissionais e Suas Afinidades. . . .	73
Figura 23 – Tela inicial do formulário de coleta de dados	83
Figura 24 – Seção do formulário: Perguntas de perfil dos respondentes (1 e 2) . . .	84
Figura 25 – Seção do formulário: Métodos atuais de planejamento (Perguntas 3 e 4)	85
Figura 26 – Seção do formulário: Dificuldades no planejamento acadêmico (Pergun- tas 5 e 6)	86
Figura 27 – Seção do formulário: Opiniões sobre planejamento estratégico (Perguntas 7 e 8)	87
Figura 28 – Seção do formulário: Ferramentas existentes (Pergunta 9)	87
Figura 29 – Seção do formulário: Necessidade de sistema automatizado (Perguntas 10 e 11)	88
Figura 30 – Seção do formulário: Expectativas e funcionalidades (Perguntas 12 e 13)	89
Figura 31 – Seção do formulário: Interesse em participação e feedback (Perguntas 14, 15 e 16)	90
Figura 32 – Questão 1 do Formulário de Mapeamento de Perfil.	97
Figura 33 – Questão 2 do Formulário de Mapeamento de Perfil.	98
Figura 34 – Questão 3 do Formulário de Mapeamento de Perfil.	99
Figura 35 – Questão 4 do Formulário de Mapeamento de Perfil.	100
Figura 36 – Questão 5 do Formulário de Mapeamento de Perfil.	101
Figura 37 – Questão 6 do Formulário de Mapeamento de Perfil.	102
Figura 38 – Questão 7 do Formulário de Mapeamento de Perfil.	103
Figura 39 – Questão 8 do Formulário de Mapeamento de Perfil.	104
Figura 40 – Questão 9 do Formulário de Mapeamento de Perfil.	105
Figura 41 – Questão 10 do Formulário de Mapeamento de Perfil.	106

Lista de tabelas

Tabela 1 – Comparação das principais funcionalidades dos sistemas de planejamento acadêmico analisados, incluindo o SPAA	41
Tabela 2 – Requisitos Funcionais do Projeto com Prioridade Must Have	50
Tabela 3 – Requisitos Funcionais do Projeto com Prioridade Should Have	51
Tabela 4 – Requisitos Funcionais do Projeto com Prioridade Could Have	51
Tabela 5 – Requisitos Não Funcionais Essenciais para o Funcionamento do Sistema - Must Have	52
Tabela 6 – Requisitos Não Funcionais Recomendados para Melhor Desempenho do Sistema - Should Have	52
Tabela 7 – Requisitos Não Funcionais do Projeto- Wont Have	52
Tabela 8 – Requisitos Funcionais do MVP	54
Tabela 9 – Planejamento das Sprints de Desenvolvimento do MVP	94

Lista de abreviaturas e siglas

UnB	Universidade de Brasília
FCTE	Faculdade de Ciências e Tecnologias em Engenharia
SIGAA	Sistema Integrado de Gestão de Atividades Acadêmicas
SEI	Sistema Eletrônico de Informações
PNAES	Política Nacional de Assistência Estudantil
IES	Instituições de Educação Superior
TCC	Trabalho de Conclusão de Curso
HST	High School Timetabling
MVC	Model-View-Controller
SGBD	Sistema Gerenciador de Banco de Dados
API	Application Programming Interfaces
AAA	Arrange-Act-Assert
WCAG	Web Content Accessibility Guidelines
BD	Banco de Dados
ME-R	Modelo Entidade-Relacionamento
DE-R	Diagrama Entidade-Relacionamento
DDL	Linguagem de Definição de Dados
BLL	Business Logic Layer
DAL	Data Access Layer
UFCG	Universidade Federal de Campina Grande
UFRN	Universidade Federal do Rio Grande do Norte
SPAA	Sistema de Planejamento Acadêmico Assistido
E2E	Testes de Ponta a Ponta
RPA	Automação de Processos Robóticos
PPC	Projeto Pedagógico de Curso

Sumário

Resumo	7
Abstract	8
Lista de ilustrações	9
Lista de tabelas	11
1 INTRODUÇÃO	16
1.1 Contexto	16
1.2 Justificativa	17
1.3 Problema	17
1.4 Objetivos	17
1.4.1 Objetivos Gerais (OG)	17
1.4.2 Objetivos específicos (OE)	18
2 REFERENCIAL TEÓRICO	19
2.1 Métodos Ágeis	19
2.2 Sistema Automatizado	21
2.3 Meta-Heurística	22
2.4 Arquitetura Model-View-Controller (MVC)	23
2.5 Banco de Dados	24
2.6 Modelagem de Dados	25
2.6.1 Modelo Entidade-Relacionamento (ME-R)	25
2.6.1.1 Conceitos Fundamentais	26
2.6.1.2 Cardinalidade de Relacionamentos	26
2.6.1.3 Grau do Relacionamento	27
2.6.1.4 Agregação	27
2.6.1.5 Generalização e Especialização	27
2.6.2 Diagrama Entidade-Relacionamento (DE-R)	27
2.6.3 Linguagem de Definição de Dados (DDL)	28
2.7 MoSCoW	29
2.8 Tecnologias	30
2.8.1 ASP.NET Core	30
2.8.2 xUnit	31
2.8.3 Python	31
2.8.4 Robot Framework	32
2.8.5 Ngrok	33

2.9	Coleta de Dados	33
2.10	Protótipação	34
2.11	Algoritmo K-Nearest Neighbors (KNN)	35
2.12	Trabalhos Relacionados	37
2.12.1	HoráriosUFCG: Ferramenta para o auxílio no planejamento de horários e escolha de disciplinas para a matrícula	37
2.12.2	Controle Universitário: Um Aplicativo de controle dos estudos e do andamento do graduando ao longo do curso superior	38
2.12.3	GradeUFOP: Uma aplicação web para ajuste de grades curriculares	39
2.12.4	Comparação das Soluções Existentes com o SPAA	40
3	METODOLOGIA	42
3.1	Coleta de Dados	42
3.2	Análise de Dados	43
3.2.1	Perfil dos Respondentes	43
3.2.2	Métodos Atuais de Planejamento	44
3.2.3	Problemas e Dificuldades Enfrentadas	45
3.2.4	Opiniões sobre Planejamento Estratégico	46
3.2.5	Ferramentas Existentes	48
3.2.6	Necessidade e Motivação para um Sistema Automatizado	48
3.2.7	Participação no Desenvolvimento	48
3.3	Requisitos	49
3.3.1	Requisitos Funcionais	49
3.3.2	Requisitos Não Funcionais	52
3.4	MVP	53
3.5	Arquitetura	55
3.6	Modelagem do Banco de Dados	56
3.7	Prototipação de telas	59
4	TESTES CAIXA-BRANCA E CAIXA-PRETA	63
4.1	Testes Unitários com Abordagem TDD Adaptada	63
4.2	Testes na Camada de Apresentação (SPAA.App)	64
4.2.1	Testes Unitários	64
4.2.2	Testes de Integração	64
4.3	Organização e Cobertura Geral dos Testes	65
4.4	Testes de Ponta a Ponta (E2E)	66
4.4.1	Estrutura dos Testes E2E	66
4.4.2	Cenários de Teste Implementados	67
4.4.2.1	Cenário 1: Registro de Novo Aluno	68
4.4.2.2	Cenário 2: Autenticação de Aluno	68

4.4.2.3	Cenário 3: Anexar Histórico Escolar (Primeiro Acesso)	69
4.4.2.4	Cenário 4: Gestão de Disponibilidade de Horários	69
4.4.3	Resultados e Análise	70
5	SISTEMA DE PERFIS DA APLICAÇÃO SPAA	71
5.1	Perfis Derivados do PPC do Curso	71
5.2	Aplicação do Algoritmo de Mapeamento de Perfil	72
6	CONCLUSÃO	75
6.1	Reafirmação da Viabilidade e Contribuições do SPAA	75
6.2	Qualidade e Confiabilidade da Solução	76
6.3	Caminhos para o Futuro: Expansão e Sustentabilidade do SPAA	76
6.3.1	Implementação do MVP2 e Expansão de Funcionalidades	76
6.3.2	Expansão da Cobertura e Abrangência dos Testes	78
6.3.3	Considerações Finais sobre o Potencial do SPAA	79
	REFERÊNCIAS	80
A	FORMULÁRIO DE COLETA DE DADOS	83
B	REPOSITÓRIO DOS TESTES DE PONTA A PONTA (E2E)	91
C	REPOSITÓRIO DO CÓDIGO-FONTE DA APLICAÇÃO	92
D	CRONOGRAMA DE DESENVOLVIMENTO DO MVP (SPRINTS)	93
E	FORMULÁRIO DE MAPEAMENTO DE PERFIL DO ESTUDANTE	96
F	REPOSITÓRIO DE CÓDIGO-FONTE DOS SCRIPTS DE EXTRA- ÇÃO DE DADOS DO SIGAA	107

1 Introdução

Este capítulo tem como objetivo apresentar o contexto do problema abordado neste trabalho, destacando a relevância do planejamento acadêmico no ambiente universitário, especialmente nas universidades públicas brasileiras. Inicialmente, é abordado o cenário atual e as dificuldades enfrentadas pelos estudantes na organização de suas grades curriculares, bem como o impacto desses desafios na evasão acadêmica. Em seguida, são discutidas a justificativa e a relevância do tema, o problema que motivou este estudo e os objetivos propostos.

1.1 Contexto

O planejamento acadêmico é um dos pilares fundamentais para o sucesso dos estudantes nas universidades públicas, exercendo influência direta sobre a organização e o progresso de suas trajetórias acadêmicas. Uma gestão eficiente da grade curricular, que considere os pré-requisitos das disciplinas, a carga horária equilibrada e a seleção estratégica de componentes curriculares, pode prevenir atrasos na formação e promover um desempenho acadêmico mais satisfatório. Contudo, a ausência de ferramentas apropriadas para esse planejamento contribui significativamente para a evasão universitária, um problema recorrente nas instituições públicas brasileiras.

De acordo com o Censo da Educação Superior de 2018, apenas 37,9% dos estudantes que ingressaram em 2010 concluíram seus cursos no período regular, enquanto 56,8% desistiram. Esses números reforçam a necessidade de estratégias que melhorem os índices de retenção e promovam a conclusão dos cursos ([Ministério da Educação \(MEC\), 2019](#)). O modelo teórico de Vincent Tinto (1975) correlaciona a evasão à falta de integração acadêmica e social dos estudantes, evidenciando como a dificuldade de adaptação ao ambiente universitário, a ausência de orientação personalizada e a falta de vínculos acadêmicos influenciam negativamente a motivação e o desempenho ([TINTO, 1975](#)).

Neste trabalho, o foco será nos estudantes da Universidade de Brasília, do campus Gama (FCTE), visando compreender as dificuldades específicas enfrentadas por esse grupo e propor uma solução que auxilie no planejamento de suas grades curriculares.

1.2 Justificativa

A evasão acadêmica é uma problemática que não apenas impacta o progresso individual dos estudantes, mas também compromete a eficiência dos cursos oferecidos. Fatores como dificuldades financeiras, sobrecarga acadêmica e falta de suporte personalizado no planejamento da grade curricular são barreiras significativas para a permanência no ensino superior ([MOROSINI et al., 2012](#)). Apesar de políticas como a Política Nacional de Assistência Estudantil (Pnaes), que busca apoiar financeiramente estudantes em situação de vulnerabilidade, muitas vezes as necessidades específicas relacionadas à organização da trajetória acadêmica não são atendidas ([Ministério da Educação \(MEC\), 2024](#)).

A integração acadêmica, destacada pelo modelo de Tinto, está diretamente ligada ao sucesso do estudante. Um planejamento eficiente da grade curricular, que alinhe metas acadêmicas claras e equilibradas com as demandas do curso, pode reduzir o estresse, aumentar a sensação de progresso e promover o engajamento do aluno nas atividades acadêmicas e sociais. Essas ações contribuem diretamente para a permanência no curso e a conclusão da formação ([CABRERA et al., 1992](#)).

1.3 Problema

Estudantes enfrentam dificuldades no planejamento de suas grades curriculares devido à ausência de ferramentas automatizadas que considerem fatores essenciais como pré-requisitos, restrições curriculares e carga horária equilibrada. Esse déficit resulta em escolhas equivocadas, sobrecarga acadêmica e até mesmo no abandono do curso. Assim, surge a necessidade de uma solução tecnológica que otimize esse processo, tornando-o mais eficiente e personalizado.

1.4 Objetivos

1.4.1 Objetivos Gerais (OG)

O objetivo deste trabalho é desenvolver um sistema automatizado de planejamento acadêmico que auxilie os estudantes da FCTE na organização de suas grades curriculares de maneira personalizada e eficiente.

1.4.2 Objetivos específicos (OE)

Para alcançar o objetivo geral, este estudo busca atingir os seguintes objetivos específicos:

- Automatizar a montagem da grade curricular com base no histórico acadêmico;
- Recomendar disciplinas optativas alinhadas ao perfil e às preferências do estudante;
- Auxiliar o estudante no planejamento acadêmico a longo prazo, promovendo uma organização eficiente que evite sobrecarga e contribua para a conclusão do curso no tempo previsto.

Dessa forma, os objetivos específicos delineados acima visam garantir que o sistema desenvolvido proporcione uma experiência mais intuitiva e eficiente para os estudantes. Ao considerar o histórico acadêmico, as preferências individuais e a carga horária recomendada, espera-se que a ferramenta contribua para uma tomada de decisão mais informada, reduzindo os desafios enfrentados no planejamento da grade curricular.

2 Referencial Teórico

Este capítulo tem como objetivo fornecer uma visão geral dos conceitos e tecnologias fundamentais que sustentam a proposta deste trabalho, explicando as abordagens adotadas e os fundamentos teóricos que orientam a construção da solução.

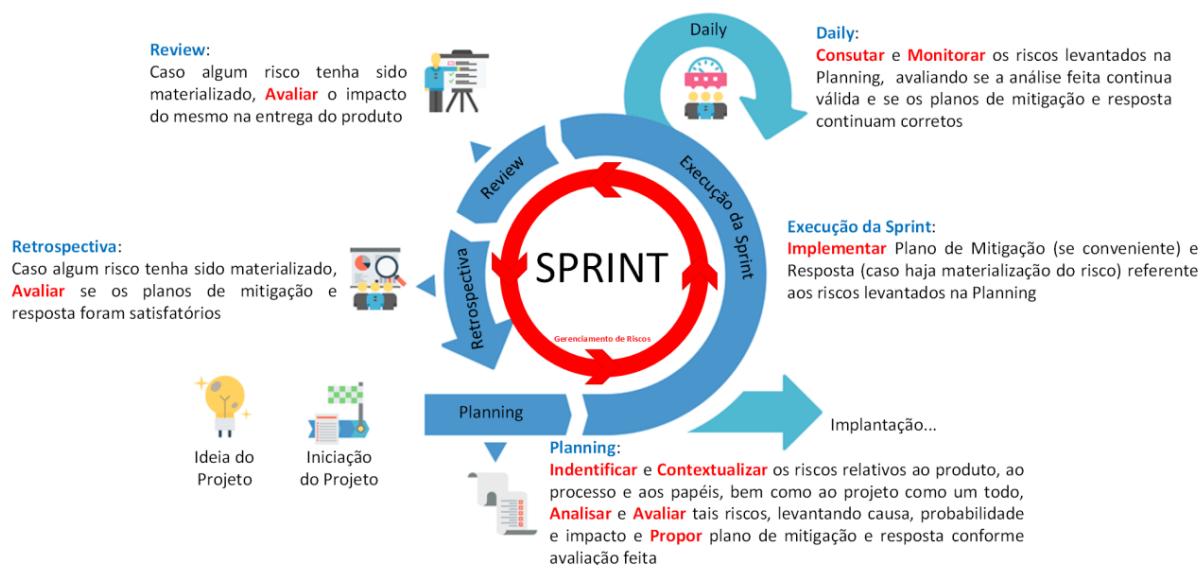
2.1 Métodos Ágeis

As metodologias ágeis surgiram para superar as limitações das abordagens tradicionais no desenvolvimento de *software*, muitas vezes caracterizadas por processos inflexíveis e ineficientes diante de mudanças frequentes nos requisitos. Diferentemente dessas abordagens, os métodos ágeis priorizam a colaboração contínua entre equipes, a entrega incremental de valor e a adaptabilidade, colocando as pessoas e suas interações acima de processos rígidos e burocráticos. O Manifesto Ágil, publicado em 2001, consolidou os princípios fundamentais dessas metodologias, incluindo a ênfase na comunicação direta, na aceitação de mudanças durante o desenvolvimento e na entrega frequente de *software* funcional (SOARES, 2004).

Dentre as metodologias ágeis, o scrum se destaca como um *framework* amplamente adotado para gerenciar a complexidade no desenvolvimento de produtos. Por meio de ciclos iterativos denominados *sprints*, que geralmente têm duração de 1 a 4 semanas, ele organiza o trabalho de equipes multidisciplinares, permitindo a entrega de funcionalidades utilizáveis ao final de cada ciclo. Essa abordagem iterativa não apenas viabiliza ajustes frequentes com base no progresso do projeto, mas também promove maior transparência e alinhamento entre todos os *stakeholders* (SCHWABER; SUTHERLAND, 2016).

Na Figura 1, é apresentado um fluxo detalhado do Scrum, incorporando práticas de gerenciamento de riscos em todas as suas fases. O processo tem início com a concepção da ideia do projeto e a fase de iniciação, onde os requisitos são levantados e os objetivos são definidos. Em seguida, entra-se no ciclo iterativo do Scrum, que abrange o planejamento da *sprint* (*Sprint Planning*), a execução acompanhada por reuniões diárias (*Daily Meetings*), a revisão dos incrementos desenvolvidos (*Sprint Review*) e a reflexão sobre melhorias contínuas (*Sprint Retrospective*). Em cada uma dessas etapas, são considerados aspectos como a identificação, análise, monitoramento e mitigação de riscos, garantindo um desenvolvimento ágil mais seguro e adaptável a mudanças.

Figura 1 – Fluxo detalhado do gerenciamento de riscos no Scrum, desde a concepção do projeto até a iteração contínua.



Fonte: Visual Paradigm(Adaptado).

Os fundamentos do scrum repousam em três pilares essenciais: transparência, que assegura que objetivos, progresso e desafios sejam visíveis para todos os envolvidos; inspeção, que permite avaliações regulares dos resultados obtidos; e adaptação, que facilita respostas rápidas a mudanças inesperadas. Para sustentar esses pilares, o *framework* se estrutura em papéis claramente definidos, eventos regulares como *sprint reviews* e *daily scrums*, e artefatos como o *product backlog*, que organizam as prioridades do trabalho (SCHWABER; SUTHERLAND, 2016).

Para sustentar esses pilares, o *framework* define papéis específicos, como o *Product Owner*, responsável por maximizar o valor do produto; o *Scrum Master*, que promove o cumprimento das práticas do Scrum; e o *Development Team*, encarregado da entrega dos incrementos. Além disso, eventos como *sprint reviews* e *daily scrums* oferecem momentos de interação e revisão, enquanto artefatos como o *product backlog* e o *sprint backlog* ajudam a organizar e priorizar o trabalho (SCHWABER; SUTHERLAND, 2016).

No contexto deste trabalho, a escolha do scrum como *framework* de desenvolvimento justifica-se pela necessidade de criar um sistema acadêmico assistido que seja dinâmico e capaz de se adaptar rapidamente às demandas dos usuários. O ambiente acadêmico, com suas constantes mudanças de requisitos — sejam decorrentes de regulamentações institucionais ou das necessidades específicas de estudantes e professores —, exige uma abordagem flexível e colaborativa. A entrega incremental de valor, característica do Scrum, é particularmente vantajosa para garantir que o sistema evolua de maneira contínua, incorporando *feedback* dos *stakeholders* em cada etapa. Dessa forma, é possível não apenas atender às expectativas imediatas dos usuários, mas também ajustar o desenvolvimento

de acordo com suas necessidades emergentes, garantindo um produto final alinhado às suas demandas reais.

2.2 Sistema Automatizado

Os sistemas automatizados são ferramentas tecnológicas projetadas para otimizar processos, consolidar informações e fornecer soluções práticas em diferentes contextos. No ambiente acadêmico, esses sistemas desempenham um papel estratégico, oferecendo suporte às instituições de ensino e aos estudantes na gestão de informações e atividades. Eles possibilitam maior eficiência e acessibilidade nos processos educacionais e administrativos, alinhando-se às demandas tecnológicas do setor e contribuindo para a modernização das práticas institucionais (CARVALHO et al., 2018).

A integração de sistemas automatizados no ensino superior trouxe avanços significativos na gestão acadêmica e administrativa. Esses sistemas permitem um acompanhamento mais eficiente do desempenho discente, simplificam processos como matrículas, gestão de frequência e monitoramento de atividades complementares, e centralizam dados relevantes para a gestão educacional. Além disso, promovem a personalização do aprendizado, oferecendo soluções adaptadas às necessidades específicas dos estudantes, como recomendações de disciplinas ou análise de desempenho individual (VASCONCELOS et al., 2012).

Na Universidade de Brasília (UnB), o Sistema Integrado de Gestão de Atividades Acadêmicas (SIGAA) é amplamente utilizado para facilitar a interação entre estudantes, professores e a administração universitária. Esse sistema é composto por módulos, como os da figura 2 abaixo, que abrangem atividades acadêmicas e administrativas, como gestão de matrículas, acompanhamento de frequência, submissão de trabalhos, registro de atividades complementares e consulta ao calendário acadêmico. Sua centralização de informações permite uma experiência integrada para discentes e docentes, otimizando fluxos de trabalho e promovendo maior organização (SIGAA, 2009).

Figura 2 – Estrutura modular do SIGAA e suas principais funcionalidades acadêmicas e administrativas.



Fonte: SIGAA - Sistema Integrado de Gestão de Atividades Acadêmicas

Além do SIGAA, outros sistemas automatizados desempenham funções complementares na gestão acadêmica da UnB. O Sistema Eletrônico de Informações (SEI) é utilizado para a tramitação digital de processos administrativos, reduzindo o uso de papel e acelerando procedimentos internos (SEI, 2009). Já a plataforma Aprender3, baseada no Moodle, é amplamente adotada para o ensino presencial e a distância, facilitando o envio de materiais, a realização de avaliações, a interação em fóruns e o acompanhamento do progresso acadêmico (APRENDER3, 2001). Essas ferramentas não apenas ampliam o acesso à educação, mas também favorecem a inclusão digital e a flexibilidade nas modalidades de ensino.

Embora sistemas como o SIGAA, SEI e Aprender3 apresentem diversas vantagens, é importante considerar os desafios associados à sua implementação e manutenção. Problemas como resistência de usuários à adoção de novas tecnologias, custos de infraestrutura e a necessidade de capacitação contínua são questões que demandam atenção para garantir o pleno aproveitamento dessas ferramentas.

2.3 Meta-Heurística

Uma meta-heurística é uma abordagem geral para problemas de busca que não se limita a um problema específico. Esses algoritmos são projetados para evitar o confinamento

em soluções subótimas locais, permitindo uma exploração mais ampla do espaço de soluções. Além disso, eles podem incorporar conhecimentos específicos do problema por meio de heurísticas personalizadas, aumentando a eficiência da busca. Métodos como Algoritmos Genéticos, *Simulated Annealing* e Enxame de Partículas têm sido amplamente empregados em problemas de otimização combinatória. De acordo com Becceneri et al. (2008)([BECCENERI, 2008](#)), a principal força desses métodos reside na sua capacidade de equilibrar a exploração do espaço de busca com a intensificação em regiões promissoras.

No contexto educacional, especialmente em problemas como o *High School Timetabling* (HST), que lida com a alocação de horários escolares, as meta-heurísticas têm se mostrado particularmente eficazes. O desafio envolve criar horários que atendam a diversas restrições, como a disponibilidade de professores, a capacidade das salas de aula e a ausência de conflitos entre disciplinas, ao mesmo tempo em que se considera as preferências institucionais ([BURKE; NEWALL, 2004](#)).

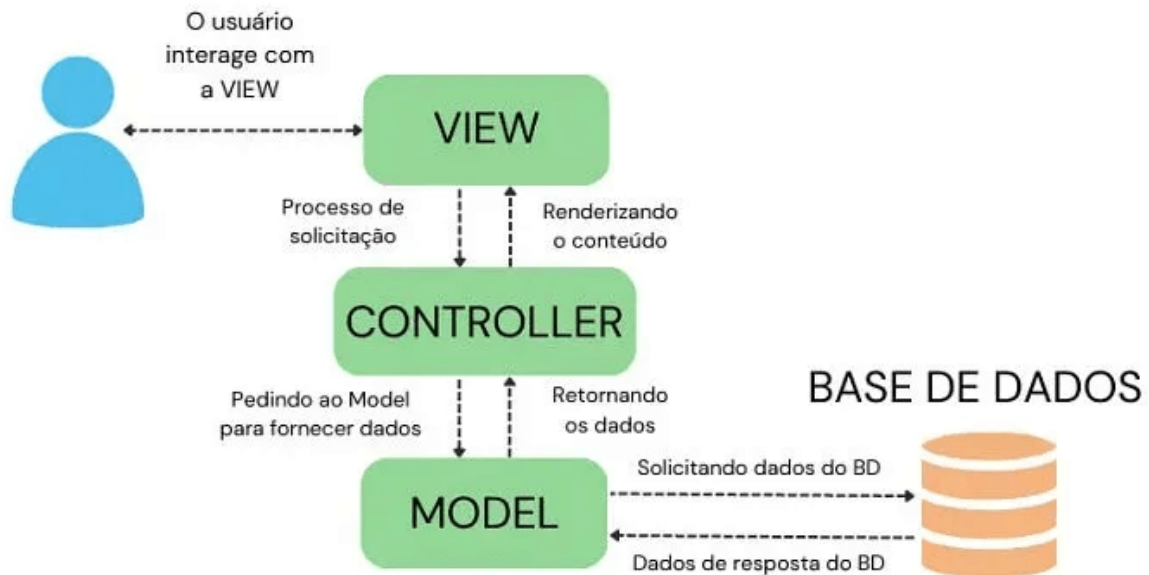
Aplicando essas teorias ao problema do planejamento acadêmico automatizado no contexto universitário, observa-se que as meta-heurísticas fornecem ferramentas poderosas para lidar com as complexidades envolvidas. Esses métodos exploram o espaço de soluções de modo a gerar alocações de horários que atendam às restrições acadêmicas, como os pré-requisitos das disciplinas, a disponibilidade de horários dos estudantes e o equilíbrio da carga horária.

2.4 Arquitetura Model-View-Controller (MVC)

O padrão arquitetural *Model-View-Controller* (MVC) é amplamente reconhecido por sua eficácia na organização de aplicações em três componentes principais: *model* (Modelo), *view* (Visão) e *controller* (Controlador). Essa estrutura facilita o desenvolvimento e a manutenção de sistemas, promovendo a separação de responsabilidades. O *model* é responsável exclusivamente por armazenar as informações necessárias para o funcionamento da aplicação. Deve conter apenas dados, sem incluir regras de negócio, funcionando como uma camada de "pura informação", sem lógica de processamento. Por outro lado, o *controller*, também chamado de "adaptador", é onde as regras de negócio são atribuídas, garantindo que a camada do *model* permaneça desacoplada de qualquer lógica que não seja estritamente relacionada aos dados. O *view*, por sua vez, é encarregado da apresentação dos dados ao usuário de forma clara e acessível, com o objetivo de exibir a interface gráfica para interação. O *controller* atua como intermediário, processando as interações do usuário e manipulando os dados no *model* para atualizar o *view*. Esse modelo é altamente modular e aplicável a diversas plataformas, como sistemas *desktop*, *web* e *mobile* ([NORMANDO, 2023](#)), como visto abaixo na figura 3, uma representação desse padrão, demonstrando a interação do usuário com a interface (*View*), o controle das solicitações pelo *Controller* e

o gerenciamento dos dados no *Model*, com integração à base de dados para processamento e resposta..

Figura 3 – Estrutura e funcionamento do padrão arquitetural Model-View-Controller (MVC).



Fonte: Leandro Raposo.

Em aplicações *web*, o padrão MVC pode ser exemplificado por uma aplicação financeira que realiza cálculos de juros simples e compostos. O usuário interage com a interface gráfica (*view*) para inserir dados e selecionar o tipo de cálculo desejado. A ação gerada pelo usuário é processada pelo *controller*, que organiza as informações e as envia ao *Model*, responsável por realizar os cálculos e, se necessário, armazenar os dados em um banco de dados. O resultado é então retornado ao *controller*, que o encaminha para o *view*, que exibe o resultado ao usuário. A separação das responsabilidades entre esses componentes facilita a manutenção e evolução do sistema, tornando-o mais modular e flexível (DevMedia, 2023).

Para o desenvolvimento de um site, o padrão MVC será utilizado para organizar a aplicação, garantindo que a lógica de negócios, o gerenciamento de dados e a interface com o usuário sejam devidamente separados. Essa abordagem não só melhora a modularidade do sistema, mas também facilita a implementação de futuras melhorias e funcionalidades.

2.5 Banco de Dados

Um banco de dados é uma coleção estruturada de dados inter-relacionados, projetados para representar aspectos específicos do mundo real, também denominado universo de

discurso. Os bancos de dados variam em tamanho e complexidade, desde pequenas listas até grandes sistemas que armazenam milhões de registros. Características fundamentais de um banco de dados incluem sua organização lógica e coerente, a reflexão das mudanças no mundo real e sua adequação a uma proposta específica, geralmente voltada para um grupo de usuários e suas respectivas aplicações (ELMASRI; NAVATHE, 2008).

Os bancos de dados podem ser administrados de forma manual ou por meio de um Sistema Gerenciador de Banco de Dados (SGBD). O SGBD facilita a definição, construção, manipulação e compartilhamento de dados, garantindo a segurança, consistência e proteção contra falhas (ELMASRI; NAVATHE, 2008).

No contexto de uma universidade, um banco de dados pode armazenar informações cruciais, como dados de alunos, cursos, disciplinas, notas e pré-requisitos. Esses dados são organizados para garantir a integridade e consistência das informações (ELMASRI; NAVATHE, 2008).

Para o projeto em questão, o banco de dados é essencial, pois serve como a base para o armazenamento e organização de dados acadêmicos, como histórico escolar e configurações da grade curricular. Ele assegura a integridade dos dados e fornece as informações necessárias para o algoritmo de meta-heurística, que utiliza essas informações para gerar soluções otimizadas que atendem às restrições acadêmicas. Além disso, o banco de dados facilita a integração com o sistema *web*, permitindo uma gestão centralizada e eficiente das informações, essencial para o planejamento acadêmico.

2.6 Modelagem de Dados

A modelagem de dados na área de *software* pode ser definida como o processo de desenvolvimento de modelos que detalham as características de um sistema. No contexto de bancos de dados, os modelos de dados são criados para descrever a estrutura de um banco de dados (BD) e as bases de dados associadas.

2.6.1 Modelo Entidade-Relacionamento (ME-R)

O Modelo de Entidade-Relacionamento (ME-R) é amplamente utilizado devido à sua simplicidade e eficiência. Esse modelo baseia-se na percepção do mundo real, que é composto por uma coleção de objetos básicos, denominados entidades, e pelos relacionamentos entre essas entidades. A modelagem de dados envolve uma série de análises conceituais e lógicas para determinar a melhor estrutura de armazenamento e manutenção das informações no banco de dados. A utilização de ferramentas específicas para modelagem de dados pode ou não ser adotada (Universidade de Brasília, s.d.).

2.6.1.1 Conceitos Fundamentais

- **Entidade:** Representação abstrata de um objeto do mundo real.
- **Atributo:** Elemento de dado que contém informações que descrevem uma entidade.
- **Domínio do Atributo:** Conjunto de valores que um atributo pode armazenar.
- **Relacionamento:** Estrutura que indica a associação entre duas ou mais entidades.

Cada um desses conceitos desempenha um papel fundamental na estruturação e organização dos dados dentro de um banco de dados relacional. As entidades representam os principais objetos do sistema, enquanto os atributos definem suas características e identificam suas particularidades. O domínio do atributo garante a integridade dos dados, restringindo os valores possíveis para cada campo armazenado. Já os relacionamentos permitem a associação entre diferentes entidades, estabelecendo conexões que refletem a lógica do mundo real.

A correta aplicação desses conceitos no desenvolvimento de um modelo entidade-relacionamento é essencial para garantir um banco de dados bem estruturado, eficiente e coerente com os requisitos do sistema. Dessa forma, a modelagem adequada facilita não apenas a implementação do banco, mas também sua manutenção e escalabilidade ao longo do tempo.

2.6.1.2 Cardinalidade de Relacionamentos

A cardinalidade de um relacionamento descreve o número de instâncias de uma entidade associadas a outra. Alguns exemplos de cardinalidade incluem:

- **Um-para-um (1 : 1):** Uma entidade A está associada no máximo a uma entidade B, e vice-versa.
- **Um-para-muitos (1 : N):** Uma entidade A está associada a várias entidades B, mas uma entidade B está associada a no máximo uma entidade A.
- **Muitos-para-um (N : 1):** Uma entidade A está associada a no máximo uma entidade B, mas uma entidade B pode estar associada a várias entidades A.
- **Muitos-para-muitos (N : N):** Uma entidade A pode estar associada a várias entidades B, e uma entidade B pode estar associada a várias entidades A.

A definição correta da cardinalidade é essencial para garantir a integridade e coerência dos dados no banco de dados. Ela influencia diretamente na estrutura das tabelas

e na forma como os dados serão armazenados e acessados.

Além disso, a cardinalidade impacta as regras de negócio do sistema, pois define as restrições e dependências entre as entidades. Um entendimento preciso dessas relações permite uma modelagem mais eficiente e evita redundâncias ou inconsistências nos dados.

2.6.1.3 Grau do Relacionamento

O grau de um relacionamento refere-se ao número de entidades que se relacionam. Alguns tipos de relacionamentos incluem:

- **Relacionamento Binário:** Quando há relacionamento entre duas entidades.
- **Relacionamento Ternário:** Quando existe um relacionamento entre três ou mais entidades.

O grau do relacionamento é um aspecto fundamental na modelagem de dados, pois determina a complexidade das associações entre as entidades. Relacionamentos binários são os mais comuns e frequentemente utilizados em bancos de dados relacionais, pois facilitam a normalização e a integridade dos dados.

Por outro lado, relacionamentos ternários ou de ordem superior podem ser necessários para representar cenários mais complexos, onde múltiplas entidades interagem simultaneamente.

2.6.1.4 Agregação

A agregação é uma limitação do modelo Entidade-Relacionamento, pois ele não permite expressar relacionamentos entre relacionamentos. A agregação é uma abstração na qual os relacionamentos são tratados como entidades de nível superior.

2.6.1.5 Generalização e Especialização

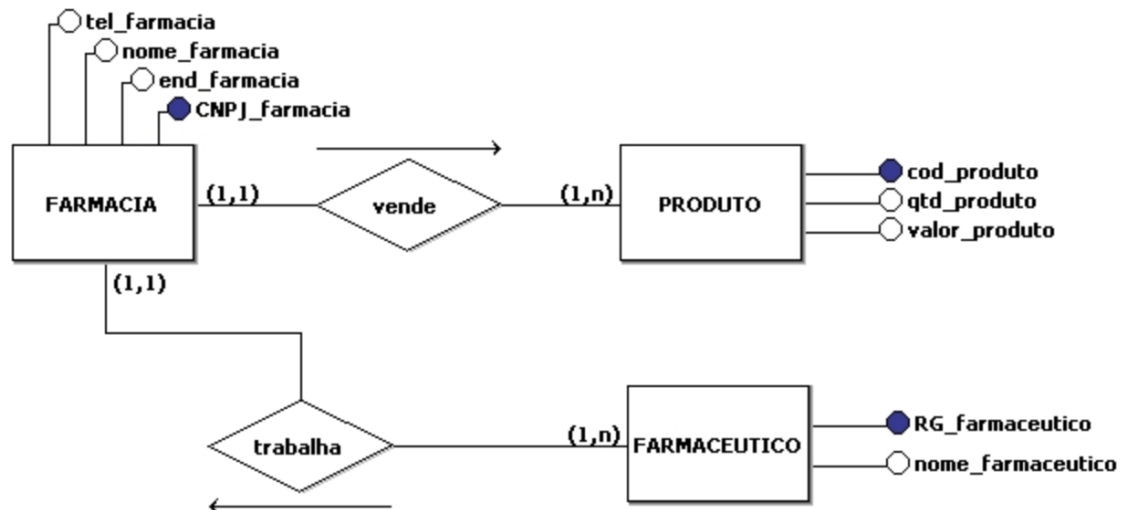
Em alguns casos, um conjunto de entidades pode ser dividido em categorias, cada uma com atributos específicos. A generalização e especialização permitem agrupar entidades de maneira hierárquica, conforme suas características.

2.6.2 Diagrama Entidade-Relacionamento (DE-R)

A estrutura lógica de um banco de dados pode ser representada graficamente por meio de um Diagrama Entidade-Relacionamento (DE-R), que ilustra como as entidades

estão conectadas. A presença de um substantivo geralmente indica uma entidade, a presença de um verbo indica um relacionamento, e um adjetivo ou advérbio pode sugerir atributos ou características dos relacionamentos (Universidade de Brasília, s.d.), assim como mostra a Figura 4 abaixo.

Figura 4 – Exemplo de Diagrama Entidade-Relacionamento (DE-R) representando conexões entre entidades em um banco de dados.



Fonte: Sae UnB.

O uso do DE-R é essencial para compreender a estrutura do banco de dados antes da implementação, facilitando a identificação de possíveis redundâncias e inconsistências. Além disso, essa representação gráfica auxilia na comunicação entre desenvolvedores, analistas e outros *stakeholders*, garantindo um entendimento claro das relações entre os dados.

2.6.3 Linguagem de Definição de Dados (DDL)

É um padrão utilizado para definir a estrutura de dados em sistemas de gerenciamento de banco de dados. Em *SQL*, isso abrange comandos para criar, alterar e excluir estruturas no banco de dados (Universidade de Brasília, s.d.).

- **CREATE:** Utilizado para criar novos objetos no banco de dados.
- **ALTER:** Usado para modificar objetos existentes no banco.
- **DROP:** Remove permanentemente objetos do banco de dados.

Os comandos DDL são fundamentais para a organização e manutenção da estrutura

do banco de dados, permitindo a criação e modificação de tabelas, índices e outros objetos essenciais. Seu uso correto é essencial para garantir a integridade e a eficiência do armazenamento dos dados, além de facilitar a evolução do sistema conforme novas necessidades surgem. Assim, compreender e aplicar a DDL de maneira adequada contribui para um banco de dados bem estruturado e de fácil manutenção.

2.7 MoSCoW

O método MoSCoW é amplamente utilizado em abordagens ágeis para priorização de requisitos, sendo conhecido por sua simplicidade e eficácia. Ele classifica os requisitos em quatro categorias: *Must Have* (essenciais), *Should Have* (importantes, mas não cruciais), *Could Have* (desejáveis se houver recursos disponíveis) e *Won't Have* (não incluídos na iteração atual). Essa estrutura hierárquica permite que equipes de desenvolvimento mantenham o foco no que é indispensável para o sucesso do projeto, ao mesmo tempo em que oferecem flexibilidade para lidar com mudanças e limitações (MIRANDA, 2022).

No planejamento inicial, os requisitos são atribuídos a essas categorias com base em sua relevância e no esforço estimado para implementá-los. A alocação típica de recursos segue a proporção de 60% para *Must Have*, 20% para *Should Have* e 20% para *Could Have*, estabelecendo margens de segurança que garantem a entrega dos itens essenciais mesmo em cenários de subestimação de esforço. Esse equilíbrio estratégico é fundamental para projetos que demandam previsibilidade e organização, especialmente em contextos com restrições de tempo e orçamento (MIRANDA, 2022).

Durante a execução, o método segue uma ordem de prioridade clara. Os itens da categoria *Must Have* são desenvolvidos primeiro, assegurando que o núcleo do sistema seja funcional antes de considerar requisitos secundários. Apenas após a conclusão dos itens essenciais, os esforços são direcionados para as categorias *Should Have* e *Could Have*, na ordem estabelecida. Caso os recursos se esgotem antes de concluir todas as funcionalidades, os itens de menor prioridade são automaticamente descartados, preservando a integridade do projeto principal (MIRANDA, 2022).

Essa abordagem também se destaca por sua capacidade de lidar com incertezas e mudanças inesperadas ao longo do desenvolvimento. Ao estabelecer *buffers* de recursos nas categorias menos prioritárias, o MoSCoW reduz o impacto de estimativas imprecisas, oferecendo um nível adicional de segurança para a entrega de requisitos essenciais (MIRANDA, 2022). Essa característica é particularmente relevante para sistemas acadêmicos como o Sistema de Planejamento Acadêmico Assistido, onde diferentes necessidades e preferências dos usuários exigem uma gestão cuidadosa de prioridades.

Outro aspecto importante do método é sua ênfase na colaboração entre equipes

técnicas e *stakeholders*. A categorização dos requisitos é feita com base em critérios objetivos, como dependências e valor agregado, o que promove uma comunicação clara sobre as prioridades do projeto (MIRANDA, 2022). Isso pode ser especialmente útil em sistemas que envolvem grande volume de dados ou múltiplos interesses, onde decisões informadas podem evitar conflitos e sobrecarga de funcionalidades.

O MoSCoW é flexível o suficiente para ser adaptado a diferentes contextos e necessidades específicas. A capacidade de ajustar prioridades ao longo do desenvolvimento torna o método uma escolha ideal para um sistema que precisa equilibrar demandas complexas e recursos limitados, garantindo entregas funcionais que atendam às expectativas dos usuários finais (MIRANDA, 2022).

2.8 Tecnologias

2.8.1 ASP.NET Core

Um *framework* é um conjunto de ferramentas e bibliotecas que fornece uma estrutura base para o desenvolvimento de *software*, facilitando a criação de aplicações ao fornecer soluções pré-definidas para problemas comuns, como gerenciamento de dados, autenticação e interface do usuário (ROBERTS; JOHNSON et al., 1996).

O *ASP.NET Core* é um *framework* de código aberto e moderno para o desenvolvimento de aplicações *web* (*World Wide Web*) e *APIs* (*Interfaces de Programação de Aplicações*). Sua principal vantagem é a capacidade de rodar em múltiplas plataformas, como Windows, macOS e Linux, utilizando o mesmo código. Além disso, o *ASP.NET Core* se destaca pela alta performance, flexibilidade e escalabilidade, o que o torna ideal para a criação de soluções robustas. Ele oferece integração nativa com bancos de dados, funcionalidades para autenticação e autorização de usuários, e permite a construção de *APIs* eficientes, que atendem aos requisitos de diversas aplicações (LOCK, 2018).

Neste trabalho, o *ASP.NET Core* será utilizado para estruturar a aplicação *web* e as *APIs* que interagem com o banco de dados. A escolha dessa tecnologia foi estratégica, considerando sua performance superior, flexibilidade e capacidade de escalar, elementos essenciais para o processamento eficiente de grandes volumes de dados. Sua arquitetura multiplataforma também facilita a implementação da solução em ambientes de produção distintos, otimizando custos e garantindo a escalabilidade da aplicação.

2.8.2 xUnit

Os testes unitários são uma parte essencial do desenvolvimento de *software*, sendo responsáveis por garantir a qualidade do código. Eles envolvem o teste de pequenas unidades de código, como funções ou métodos, de forma isolada, com o objetivo de verificar se o comportamento dessas unidades está correto. O processo de teste unitário busca detectar erros e falhas em fases iniciais do desenvolvimento, antes que eles se propaguem para outras partes do sistema. Isso é importante para manter a confiabilidade do *software* ao longo do seu ciclo de vida, facilitando a manutenção e o refatoramento sem comprometer o funcionamento da aplicação (MYERS; BADGETT; SANDLER, 2012).

O *xUnit* é uma biblioteca de código aberto amplamente utilizada para a realização de testes unitários no *framework* .NET. Ele adota o padrão de testes *Arrange-Act-Assert* (AAA), que divide o teste em três etapas principais: *Arrange* (Organizar), sendo a preparação do ambiente de teste, onde os dados necessários são configurados. *Act* (Agir), a execução do código a ser testado. *Assert* (afirmar), a verificação do resultado obtido para garantir que o comportamento do código esteja correto. Esse padrão facilita a leitura e a manutenção dos testes, permitindo que os desenvolvedores escrevam e atualizem os testes de forma mais robusta e compreensível. Além disso, o *framework* oferece flexibilidade e simplicidade, permitindo que os testes sejam facilmente integrados ao fluxo de trabalho de desenvolvimento (xUnit Community, 2023).

Neste trabalho, o *xUnit* será utilizado para garantir uma boa cobertura de testes unitários nas camadas de negócio e dados, e também para implementar testes de integração na camada de aplicação. Isso assegurará que tanto as funcionalidades individuais quanto as interações entre os componentes da aplicação estejam funcionando conforme o esperado. A implementação de testes com *xUnit* contribuirá para aumentar a confiabilidade do sistema, permitindo a detecção precoce de erros e a manutenção da integridade da aplicação ao longo de seu ciclo de vida.

2.8.3 Python

Python é uma linguagem de programação interpretada, de alto nível e de propósito geral, conhecida por sua sintaxe clara e legível, o que facilita tanto o aprendizado quanto o desenvolvimento. Com uma vasta coleção de bibliotecas e *frameworks*, Python é amplamente utilizado em diversas áreas, como desenvolvimento *web*, análise de dados, inteligência artificial, automação e *scripting* (Python Software Foundation, 2025). Sua versatilidade e a grande comunidade de desenvolvedores contribuem para um ecossistema rico em soluções e suporte.

Neste trabalho, Python será utilizado para automações específicas, incluindo o *web scraping* de dados e a subsequente população do banco de dados. Essa abordagem é fundamental para preparar o ambiente de testes e para o carregamento inicial de informações essenciais para o funcionamento da aplicação. A escolha de Python para essas tarefas se deve à sua eficiência em manipulação de dados, à disponibilidade de bibliotecas robustas para *parsing* de HTML (*BeautifulSoup*, *Scrapy*) e interação com bancos de dados, e à sua capacidade de integrar-se perfeitamente com ferramentas de automação, como o *Robot Framework*. Essa utilização estratégica de Python garante que o sistema possa ser testado com um volume realista de dados, simulando cenários complexos e validando a performance e a robustez da solução.

2.8.4 Robot Framework

O *Robot Framework* é uma ferramenta de automação de testes genérica e de código aberto, amplamente utilizada para testes de aceitação, automação de processos robóticos (*RPA*) e automação de testes de regressão. Sua principal característica é a abordagem baseada em palavras-chave (*keyword-driven*), que permite escrever testes de forma legível e de alto nível, quase como uma linguagem natural. Isso facilita a colaboração entre desenvolvedores, testadores e *stakeholders* não técnicos, pois os casos de teste são facilmente compreendidos por qualquer pessoa ([Robot Framework Foundation, 2025](#)).

Além da sua legibilidade, o *Robot Framework* é extremamente extensível. Ele pode ser integrado com outras bibliotecas e ferramentas, como Selenium WebDriver para automação de navegadores *web*, Appium para automação *mobile* e até mesmo bibliotecas Python customizadas para interações específicas. Essa flexibilidade o torna uma escolha poderosa para testes de ponta a ponta (*E2E*), onde a interação com a interface do usuário é crucial e complexa.

Neste trabalho, o *Robot Framework* será empregado para a criação e execução de testes de ponta a ponta (*E2E*). Ele simulará a interação de usuários reais com a aplicação, verificando os fluxos completos do sistema, desde o login até a realização de operações complexas. A escolha do *Robot Framework* é estratégica, pois sua sintaxe intuitiva e a capacidade de integrar-se com o Selenium permitirão o desenvolvimento de testes robustos e de fácil manutenção, que validarão o comportamento da aplicação sob a perspectiva do usuário final, garantindo que os requisitos funcionais sejam plenamente atendidos.

2.8.5 Ngrok

No cenário de desenvolvimento de sistemas, especialmente aqueles que operam em ambientes locais (como um computador pessoal ou rede interna), a necessidade de expor uma aplicação para acesso externo é frequente. Isso é particularmente relevante durante fases de teste, demonstrações a *stakeholders* ou colaborações remotas, onde firewalls e configurações de rede locais impedem o acesso direto de IPs externos. Nesse contexto, ferramentas de tunelamento reverso tornam-se indispensáveis.

O *Ngrok* é um serviço de tunelamento reverso de código aberto que permite criar URLs públicas seguras para aplicações locais ([ngrok, 2025b](#)). Ele estabelece um túnel seguro desde um servidor em execução no ambiente local (computador do desenvolvedor) até a nuvem do *Ngrok*, e então expõe essa conexão através de um domínio público temporário e acessível via internet. Essa capacidade permite que aplicações que normalmente estariam restritas à rede interna (como um servidor *web* de desenvolvimento, uma API local, ou mesmo um serviço rodando em uma porta específica) sejam acessadas por qualquer pessoa em qualquer lugar do mundo, sem a necessidade de configurar firewalls, roteadores ou serviços de VPN. O *Ngrok* suporta diversos protocolos, incluindo *HTTP*, *HTTPS* e *TCP*, garantindo flexibilidade para diferentes tipos de aplicações.

Além de sua funcionalidade principal de tunelamento, o *Ngrok* oferece recursos adicionais valiosos para desenvolvedores, como inspeção de tráfego (permitindo visualizar requisições e respostas), reprodução de requisições, e proteção por senha para os túneis, aumentando a segurança em demonstrações e testes ([ngrok, 2025a](#)). Essa combinação de facilidade de uso, segurança e recursos de depuração torna o *Ngrok* uma ferramenta popular para testes de integração, webhooks, pagamentos online, e colaboração remota.

O *Ngrok* foi essencial no desenvolvimento e testes do Sistema de Planejamento Acadêmico Assistido (SPAA). Sua simplicidade e segurança foram cruciais para acelerar o processo, já que ele facilitou a exposição temporária do aplicativo. Isso foi fundamental para validar o SPAA com usuários reais e para as apresentações de progresso do projeto.

2.9 Coleta de Dados

A coleta de dados desempenha um papel fundamental no desenvolvimento de soluções orientadas pelas necessidades dos usuários. Ela é uma etapa importante para entender as necessidades dos usuários e desenvolver soluções que atendam ao contexto estudado. Para este trabalho, foi escolhido o formulário como principal instrumento de coleta, por sua capacidade de capturar informações qualitativas e quantitativas de maneira eficiente. Essa abordagem combina perspectivas subjetivas e contextuais, obtidas por meio

de perguntas abertas, com dados objetivos e estruturados, coletados por meio de perguntas fechadas, oferecendo uma análise ampla e detalhada das respostas obtidas (STICKDORN et al., 2018).

A abordagem qualitativa permite identificar percepções, motivações e desafios enfrentados pelos usuários, aspectos que são essenciais para compreender as complexidades do problema em questão. Por outro lado, os dados quantitativos fornecem métricas claras e mensuráveis que auxiliam na identificação de padrões e tendências, permitindo uma validação mais robusta das hipóteses levantadas (STICKDORN et al., 2018).

No contexto deste trabalho, a aplicação do formulário foi estruturada para captar dados que evidenciem as dificuldades enfrentadas pelos estudantes no planejamento acadêmico, ao mesmo tempo em que permite mapear tendências e necessidades.

2.10 Protótipação

A prototipação é uma etapa no *design* de serviços que possibilita a experimentação e a validação de conceitos antes de sua implementação. Esse processo envolve a criação de representações tangíveis de ideias, permitindo explorar soluções, identificar falhas e realizar iterações para aprimorar o produto ou serviço (STICKDORN et al., 2018).

No presente estudo, o processo de prototipação inicia com a definição de requisitos detalhados, fundamentais para estruturar as funcionalidades do sistema acadêmico assistido. Esses requisitos orientam a criação de *wireframes*, que funcionam como esboços visuais simplificados das interfaces e fluxos do sistema. A escolha por *wireframes* se justifica por sua capacidade de estruturar ideias de forma rápida e eficiente, facilitando revisões e ajustes iniciais com base no *feedback* dos *stakeholders*. Essa abordagem permite alinhar as expectativas e reduzir riscos ainda nas etapas iniciais do desenvolvimento (STICKDORN et al., 2018).

A aplicação de métodos digitais de prototipação complementa esse processo ao possibilitar a criação de protótipos interativos. Esses protótipos permitem simular a experiência do usuário e realizar ajustes frequentes, promovendo uma iteração contínua ao longo do desenvolvimento (STICKDORN et al., 2018). No contexto de sistemas acadêmicos, essa técnica é particularmente útil para testar fluxos de navegação, como o planejamento de disciplinas ou a consulta de horários, garantindo que as interfaces atendam às necessidades específicas dos usuários e respeitem boas práticas de usabilidade.

Além disso, o processo de prototipação inclui a participação de *stakeholders* por meio de atividades colaborativas, conhecidas como *co-design* (STICKDORN et al., 2018). A interação direta entre equipe de desenvolvimento, estudantes e professores contribui para incorporar diferentes perspectivas e validar as soluções propostas. No caso do sistema

acadêmico assistido, essa colaboração permite que as funcionalidades desenvolvidas reflitam as demandas operacionais e as expectativas do público-alvo, reduzindo a lacuna entre o *design* e a aplicação prática.

2.11 Algoritmo K-Nearest Neighbors (KNN)

O algoritmo **K-Nearest Neighbors (KNN)** é uma técnica de aprendizado de máquina supervisionado, amplamente empregada para resolver problemas de classificação e regressão. Sua operação fundamenta-se na premissa de que instâncias de dados com características similares tendem a se agrupar no espaço dimensional. Dessa forma, a determinação da classe ou do valor de um novo ponto de dados é realizada com base nos K exemplos mais próximos presentes no conjunto de treinamento.

O processo de classificação (ou regressão) via KNN pode ser compreendido em três etapas fundamentais:

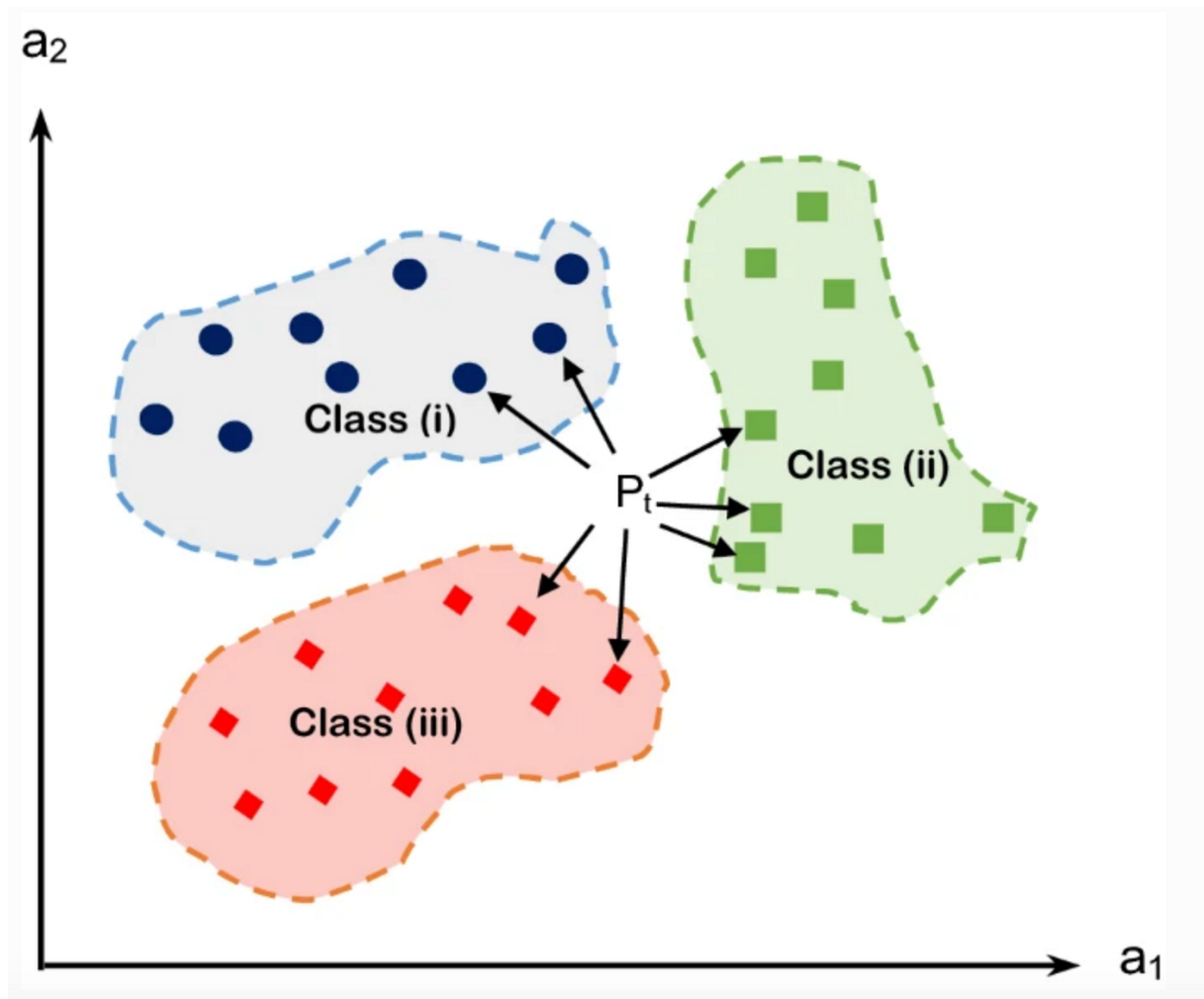
1. **Cálculo de Distâncias:** A distância entre o novo ponto de dados (a ser classificado) e cada um dos pontos existentes no conjunto de treinamento é computada. As métricas de distância mais comuns incluem a Distância Euclidiana, conforme apresentada na equação abaixo, mas outras métricas como Manhattan ou Minkowski também podem ser aplicadas, dependendo da natureza dos dados.

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (2.1)$$

2. **Seleção dos K Vizinhos Mais Próximos:** Após o cálculo das distâncias, são selecionados os K pontos de dados do conjunto de treinamento que apresentaram as menores distâncias em relação ao novo ponto.
3. **Atribuição da Classe/Valor:** Para problemas de classificação, a classe do novo ponto é definida pela maioria das classes entre os K vizinhos selecionados (voto majoritário). Em problemas de regressão, o valor é atribuído pela média dos valores dos K vizinhos.

Para ilustrar o conceito, observe a **Figura 5**, que demonstra o processo de classificação de um novo ponto, P_t , com base em seus vizinhos mais próximos. Os pontos já classificados estão agrupados em "Classe (i)", "Classe (ii)" e "Classe (iii)". A classificação de P_t dependerá dos seus vizinhos mais próximos, de acordo com o valor de K escolhido.

Figura 5 – Ilustração do funcionamento do algoritmo K-Nearest Neighbors (KNN)



Fonte: Data Hackers

A escolha do valor de K é um fator crítico para o desempenho do algoritmo, impactando diretamente a capacidade de generalização do modelo. Um K muito pequeno pode levar ao sobreajuste (*overfitting*), tornando o modelo excessivamente sensível a ruídos nos dados. Por outro lado, um K muito grande pode resultar em subajuste (*underfitting*), suavizando demais as fronteiras de decisão e ignorando padrões importantes. A determinação do K ideal geralmente envolve técnicas de validação cruzada.

Neste projeto, a adoção do KNN é justificada pela sua comprovada eficácia em problemas de baixa dimensionalidade, aliada à sua capacidade de se adaptar a novos contextos sem a exigência de um treinamento prévio. Essa característica o torna particularmente adequado para aplicações de recomendação baseadas na similaridade entre perfis.

2.12 Trabalhos Relacionados

Este capítulo apresenta uma análise de trabalhos relacionados ao planejamento acadêmico, destacando diferentes abordagens para organização de horários, gestão de disciplinas e acompanhamento do percurso universitário.

2.12.1 HoráriosUFCG: Ferramenta para o auxílio no planejamento de horários e escolha de disciplinas para a matrícula

O trabalho desenvolvido na Universidade Federal de Campina Grande (UFCG) busca melhorar o planejamento acadêmico dos estudantes, especialmente no período de matrículas. O sistema de Controle Acadêmico da UFCG, embora amplamente utilizado, apresenta limitações ao fornecer apenas uma tabela estática das disciplinas disponíveis, o que dificulta a organização por parte dos alunos. Para contornar esse problema, o grupo Guardians criou inicialmente um sistema *web* que capturava dados por *scraping* e os apresentava em uma interface visual. Contudo, essa solução foi descontinuada devido à dependência de atualizações manuais (FERRÃO, 2023).

Posteriormente, foi desenvolvido o HoráriosUFCG, uma ferramenta que permite aos alunos planejar suas matrículas de forma mais eficiente e autônoma. A aplicação organiza as disciplinas em uma matriz que relaciona dias e horários, como mostra na Figura 6, detecta conflitos e oferece filtros para facilitar a seleção. Avaliado com o *Computer System Usability Questionnaire*, o sistema obteve resultados positivos em usabilidade e funcionalidade, comprovando sua eficácia e relevância para o planejamento acadêmico dos estudantes (FERRÃO, 2023).

Figura 6 – Interface do HoráriosUFCG exibindo a grade horária com disciplinas selecionadas

	SEG	TER	QUI	SEX	SAB
08h - 10h	CÁLCULO DIFERENCIAL E INTEGRAL I - t01	LABORATÓRIO DE ESTRUTURA DE DADOS - t01	PROGRAMAÇÃO II - t01	FUND DE MATEMÁTICA P/C.D.A COMPUTAÇÃO II - t02	
10h - 12h	PROGRAMAÇÃO II - t01 PROGRAMAÇÃO II - t02	FUND DE MATEMÁTICA P/C.D.A COMPUTAÇÃO II - t02	LABORATÓRIO DE ESTRUTURA DE DADOS - t01	LABORATÓRIO DE PROGRAMAÇÃO II - t02 LABORATÓRIO DE PROGRAMAÇÃO II - t03	
14h - 16h	CÁLCULO DIFERENCIAL E INTEGRAL I - t07	ÁLGEBRA LINEAR I - t04			
16h - 18h			CÁLCULO DIFERENCIAL E INTEGRAL I - t07 ÁLGEBRA LINEAR I - t04		

Fonte: HoráriosUFCG

Enquanto o trabalho da Universidade Federal de Campina Grande foca no planejamento de horários e na resolução de conflitos em matrículas semestrais, a solução apresentada propõe uma abordagem mais personalizada. Além de organizar as disciplinas, o sistema desenvolvido considera os objetivos específicos de cada aluno, como metas acadêmicas e carga horária ideal, contribuindo para a construção de uma grade curricular estratégica e alinhada às necessidades individuais. Essa abordagem diferencia-se ao ampliar o suporte ao estudante, indo além da simples organização de horários para promover um planejamento acadêmico mais integrado e eficiente.

2.12.2 Controle Universitário: Um Aplicativo de controle dos estudos e do andamento do graduando ao longo do curso superior

O trabalho em análise apresenta o desenvolvimento de um aplicativo móvel voltado para auxiliar estudantes no gerenciamento de disciplinas e tarefas durante o ensino superior. O aplicativo permite que os alunos acompanhem suas notas, faltas e atividades a serem entregues, além de visualizar as disciplinas já cursadas, pendentes e seus pré-requisitos. Uma característica importante é a possibilidade de cadastrar a grade curricular do curso ou inseri-la manualmente, facilitando o acompanhamento do progresso acadêmico. O diferencial do projeto está na pré-configuração das matrizes curriculares de cursos específicos, permitindo maior praticidade para os usuários dessas áreas (GOMES, 2021).

Outro ponto destacado no trabalho é o foco na usabilidade do aplicativo, com a realização de testes para identificar erros e implementar melhorias. As funcionalidades

propostas buscam ajudar os estudantes a organizarem melhor suas responsabilidades acadêmicas e a se programarem semestre a semestre. Dessa forma, o aplicativo tem como objetivo principal auxiliar na gestão do tempo e na visualização do desempenho acadêmico, contribuindo para a organização e eficiência dos estudantes durante o curso (GOMES, 2021).

Em contraste com o aplicativo analisado, que se concentra no acompanhamento e organização de tarefas semestrais, nosso projeto oferece uma solução mais abrangente para o planejamento estratégico da trajetória acadêmica completa. Nossa ferramenta permite que os estudantes planejem suas grades curriculares considerando pré-requisitos e a sequência ideal de disciplinas, otimizando o tempo de conclusão do curso. O foco é proporcionar uma visão global do percurso acadêmico, do início ao fim, enquanto o trabalho analisado prioriza a gestão de curto prazo.

2.12.3 GradeUFOP: Uma aplicação web para ajuste de grades curriculares

O trabalho em análise descreve o desenvolvimento do GradeUFOP, uma aplicação *web* voltada para otimizar a montagem das grades curriculares dos alunos da Universidade Federal de Ouro Preto (UFOP). A plataforma utiliza técnicas avançadas de raspagem de dados para coletar informações sobre cursos, disciplinas e horários diretamente da universidade. Com essas informações, o sistema integra-as em uma interface interativa que permite aos alunos criar grades personalizadas, levando em consideração suas preferências, restrições acadêmicas e disponibilidade de horários. A solução emprega um algoritmo que gera múltiplas opções de grade, dando aos alunos a liberdade de escolher a melhor combinação de disciplinas. A interface é projetada para ser intuitiva, permitindo fácil visualização e configuração das "slots" de disciplinas de acordo com as necessidades individuais dos usuários (TORRES, 2024).

Os resultados do desenvolvimento do GradeUFOP indicam que a aplicação tem o potencial de transformar o processo de planejamento acadêmico na UFOP. Com base em testes realizados, observou-se que o sistema contribui significativamente para a redução de erros, otimiza o tempo gasto pelos alunos durante o processo de matrícula e facilita a visualização das opções de grade. A Figura 7 ilustra a tela do GradeUFOP apresentando grades curriculares geradas automaticamente (TORRES, 2024).

Figura 7 – Tela do GradeUFOP apresentando grades curriculares geradas automaticamente

Grades Disponíveis

Grade 2/10

Horário	Segunda	Terça	Quarta	Quinta	Sexta	Sábado
13:30 - 15:10	CSI148	-	-	CSI403	CEA064	-
15:20 - 17:00	CSI403	-	CEA064	CSI148	-	-
19:00 - 20:40	-	ELT561	-	-	ELT502	-
21:00 - 22:40	-	-	ELT502	ELT561	-	-

Disciplinas Escolhidas

Código	Turma	Disciplina	Professor
CEA064	21	FISICA IV	JUVENIL SIQUEIRA DE OLIVEIRA FILHO(T+P)
ELT502	21	ELETROMAGNETISMO	FELIPE EDUARDO MOREIRA COTA(T)
CSI148	21	ANALISE NUMERICA	DARLAN NUNES DE BRITO(T+P) / LUIZ OTAVIO MENEZES TELES(...)
ELT561	22	ELETRONICA I	WELBERT ALVES RODRIGUES(T+P)
CSI403	11	ENGENHARIA DE SOFTWARE I	EULER HORTA MARINHO(T)

ANTERIOR PRÓXIMA

Fonte: GradeUFOP

O GradeUFOP concentra-se na otimização das grades curriculares, buscando evitar conflitos de horário e considerando as preferências individuais e as disponibilidades de disciplinas. Em contraste, o trabalho aqui apresentado tem como objetivo oferecer uma visão global da trajetória acadêmica, facilitando o planejamento ao longo de toda a graduação. A ferramenta permite que os alunos visualizem as disciplinas pendentes, bem como os pré-requisitos necessários, promovendo um planejamento estratégico e integrado. Ambos os sistemas se mostram complementares, atendendo a diferentes necessidades dos estudantes, com ênfases distintas na forma de gestão das informações ao longo da jornada acadêmica.

2.12.4 Comparação das Soluções Existentes com o SPAA

A Tabela 1 apresenta uma análise comparativa entre diferentes sistemas de planejamento acadêmico, destacando suas funcionalidades, contribuições e limitações. Esse levantamento evidencia as lacunas que o SPAA busca preencher, reforçando sua proposta inovadora.

Tabela 1 – Comparação das principais funcionalidades dos sistemas de planejamento acadêmico analisados, incluindo o SPAA

Funcionalidade	HoráriosUFCG	Controle Univ.	GradeUFOP	SPAA
Planejamento de grade semestral	X		X	X
Geração automática de grades				X
Deteção de conflitos de horário	X			X
Planejamento de longo prazo				X
Planejamento de curto prazo		X		
Consideração de pré-requisitos		X	X	X
Carga horária equilibrada			X	X
Projeção de fluxo acadêmico				X
Integração com histórico acadêmico		X	X	X

Fonte: Autoria própria.

Os sistemas analisados concentram-se majoritariamente no planejamento semestral e no fornecimento de informações acadêmicas. Em contraste, o SPAA se diferencia ao adotar uma abordagem mais ampla, permitindo um planejamento estratégico de longo prazo. Ele incorpora critérios essenciais, como otimização do tempo de curso, equilíbrio da carga horária e personalização do planejamento acadêmico, oferecendo uma solução mais completa e adaptável às necessidades dos estudantes.

3 Metodologia

Neste capítulo, são detalhados os métodos e as abordagens utilizadas para o desenvolvimento da solução proposta, desde a coleta e análise dos dados até a definição da arquitetura do sistema e a prototipação das interfaces de usuário.

3.1 Coleta de Dados

A coleta de dados foi realizada por meio de um formulário *online*, com o objetivo de compreender as dificuldades enfrentadas pelos estudantes no planejamento acadêmico e suas expectativas em relação a um sistema automatizado de apoio. A escolha do questionário como método de coleta seguiu a recomendação do livro *Isto é Design de Serviço* (STICKDORN et al., 2018), que enfatiza a combinação de dados quantitativos e qualitativos para se obter uma visão mais ampla e profunda do problema. Tal abordagem permite capturar tanto a frequência de problemas comuns quanto as percepções subjetivas dos participantes, enriquecendo a análise dos dados. O formulário completo pode ser consultado no **Apêndice A**.

O formulário foi disponibilizado entre os dias 09 e 17 de dezembro de 2024, com a finalidade de alcançar o maior número possível de estudantes em um intervalo de tempo reduzido, permitindo a coleta de uma amostra representativa das dificuldades e necessidades dos alunos da Universidade de Brasília (UnB). Durante o período de coleta, o questionário obteve 77 respostas, possibilitando uma análise robusta dos dados coletados.

A estrutura do formulário foi dividida em sete blocos principais, com o intuito de explorar aspectos específicos do processo de planejamento acadêmico dos estudantes:

- **Bloco 1: Perfil dos Respondentes.** Este bloco buscou identificar se os participantes eram ou haviam sido estudantes da UnB, além do campus ao qual estavam vinculados, com o objetivo de contextualizar o perfil da amostra.
- **Bloco 2: Métodos Atuais de Planejamento.** Neste bloco, investigou-se como os estudantes atualmente organizam suas grades horárias, incluindo o uso de ferramentas e métodos organizacionais.
- **Bloco 3: Problemas e Dificuldades Enfrentados.** Este bloco explorou os principais problemas enfrentados pelos estudantes no planejamento da grade curricular, como conflitos de horários, indisponibilidade de vagas, dificuldades com pré-requisitos e impacto no progresso acadêmico.

- **Bloco 4: Opiniões sobre Planejamento Estratégico.** Buscou-se compreender a opinião dos estudantes sobre a utilidade de um planejamento acadêmico de longo prazo e os motivos pelos quais acreditam que essa abordagem poderia ser vantajosa.
- **Bloco 5: Ferramentas Existentes.** Este bloco investigou o uso ou conhecimento de plataformas existentes que auxiliam no planejamento da grade curricular.
- **Bloco 6: Necessidade e Motivação para um Sistema Automatizado.** O foco deste bloco foi avaliar a percepção dos estudantes sobre a necessidade de uma ferramenta automatizada, as funcionalidades consideradas indispensáveis e as preocupações em relação ao uso de tal sistema.
- **Bloco 7: Participação no Desenvolvimento.** Por fim, buscou-se avaliar o interesse dos estudantes em receber mais informações ou participar de testes relacionados ao desenvolvimento do sistema.

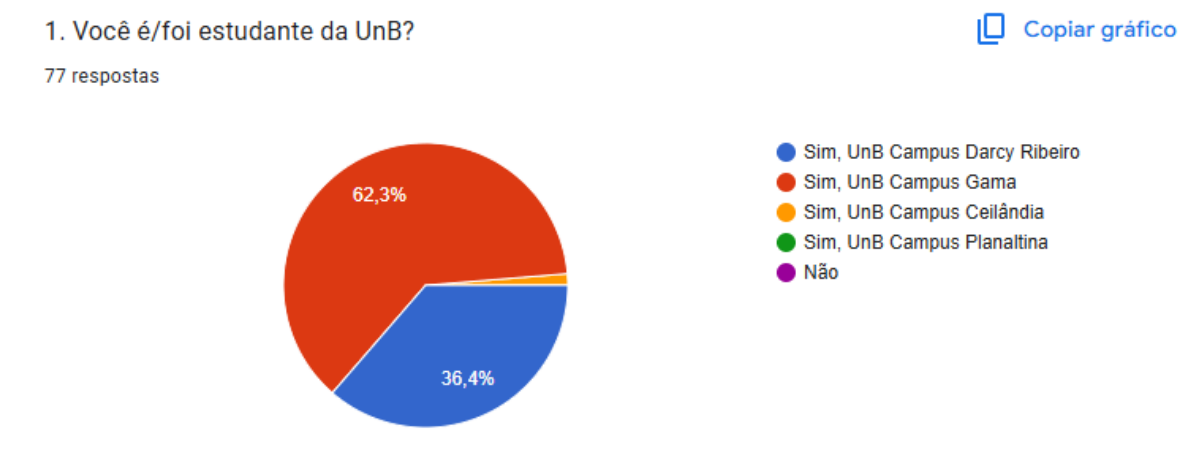
Essa estrutura foi projetada para captar tanto a realidade atual enfrentada pelos estudantes quanto suas expectativas em relação a possíveis soluções tecnológicas, oferecendo uma visão abrangente dos desafios e das oportunidades para o desenvolvimento de um sistema que atenda às necessidades identificadas.

3.2 Análise de Dados

3.2.1 Perfil dos Respondentes

Com base nos dados coletados, observa-se que o formulário foi amplamente respondido por estudantes ou ex-estudantes do Campus Gama, que representaram 62,3% dos respondentes, conforme apresentado na Figura 8. Esse número é significativamente superior ao do Campus Darcy Ribeiro, que reuniu 36,4% das respostas, e ao do Campus Ceilândia, com apenas 1,3%. Não houve registro de participação de respondentes do Campus Planaltina nem de indivíduos externos à UnB.

Figura 8 – Distribuição dos respondentes da pesquisa por campus da Universidade de Brasília, destacando a predominância de participantes do Campus Gama.



Fonte: Autoria própria.

Esses resultados indicam que o formulário obteve maior adesão entre os estudantes do Campus Gama, sugerindo um maior engajamento ou interesse desse público em relação à pesquisa.

3.2.2 Métodos Atuais de Planejamento

Com base nos dados coletados, os respondentes relataram o uso de diversos métodos para a montagem de suas grades horárias. A maioria dos estudantes organiza manualmente as disciplinas, utilizando planilhas (*Excel* ou *Google Sheets*), ferramentas digitais como *Trello* ou *Notion*, ou métodos tradicionais como papel e caneta. Observa-se também que muitos utilizam o SIGAA para consulta de horários e organização prévia. Entre os métodos mencionados, destacam-se:

- Uso de planilhas eletrônicas, como *Excel* e *Google Sheets*, mencionados por 56,1% dos respondentes.
- Aplicativos de produtividade, como *Trello* e *Notion*, utilizados por 28,1%.
- Ferramentas específicas para a UnB, como o site SuaGradeUnB (5,4%).
- Métodos tradicionais, como papel e caneta (3,6%).

Esses resultados indicam uma preferência significativa por ferramentas digitais, que facilitam o planejamento e a organização das grades horárias, mas também evidenciam a diversidade de abordagens utilizadas pelos estudantes. A Figura 9 a seguir ilustra

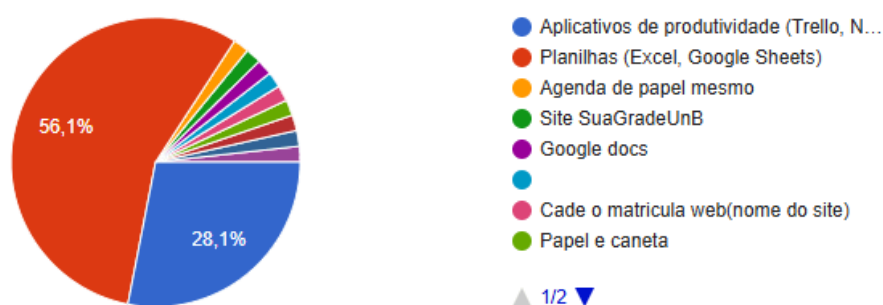
a distribuição dos métodos e ferramentas utilizados pelos participantes, mostrando a variedade de opções empregadas para o planejamento acadêmico.

Figura 9 – Métodos e ferramentas utilizadas pelos estudantes para planejamento da grade horária, evidenciando a preferência por soluções digitais.

3. Se você utiliza algum método ou ferramenta para montar sua grade manualmente, qual é?

 Copiar gráfico

57 respostas



Fonte: Autoria própria.

A diversidade de ferramentas utilizadas reflete as diferentes necessidades e preferências dos estudantes, o que é um indicativo da flexibilidade do processo de planejamento acadêmico. Essas informações são importantes para compreender como os alunos se organizam e quais recursos consideram mais eficazes para a criação de suas grades horárias.

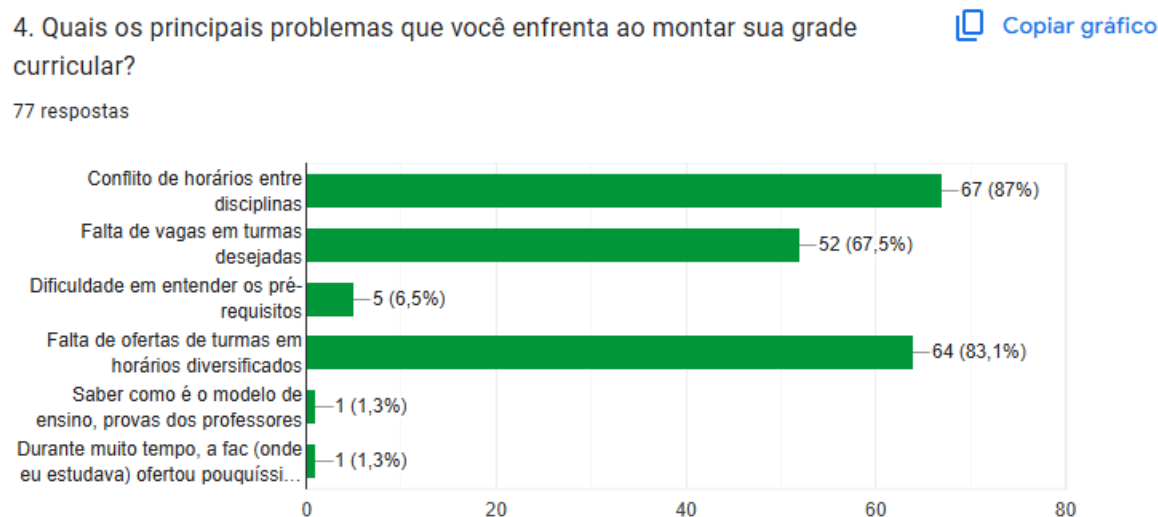
3.2.3 Problemas e Dificuldades Enfrentadas

Com base nas respostas dos participantes, os principais problemas enfrentados pelos estudantes na organização de suas grades curriculares incluem o **conflito de horários entre disciplinas**, mencionado por 87% dos respondentes, e a **falta de vagas em turmas desejadas**, citada por 67,5%. A **falta de ofertas de turmas em horários diversificados** também é uma dificuldade relevante, com 83,1% dos estudantes relatando essa preocupação. Essas respostas sugerem que a organização dos horários de disciplinas e a disponibilidade de turmas em horários flexíveis são questões cruciais para o planejamento acadêmico.

Por outro lado, questões como a **dificuldade em entender os pré-requisitos** e a falta de informações sobre os **modelos de ensino ou provas dos professores** foram mencionadas por uma minoria, com 6,5% e 1,3% das respostas, respectivamente. Isso indica que, embora sejam aspectos relevantes, esses fatores não são vistos como as principais barreiras para o planejamento da grade curricular. Esses dados estão ilustrados

na Figura 10 a seguir, que apresenta a distribuição das respostas sobre os principais problemas enfrentados pelos estudantes.

Figura 10 – Principais dificuldades relatadas pelos estudantes na montagem da grade curricular, incluindo conflitos de horário e indisponibilidade de vagas.



Fonte: Autoria própria.

A análise dessas dificuldades oferece insights valiosos sobre as principais barreiras enfrentadas pelos alunos ao planejar suas grades acadêmicas. Identificar essas dificuldades pode orientar a criação de soluções mais eficazes para melhorar a experiência de organização das disciplinas, tornando-a mais eficiente e menos sujeita a obstáculos.

3.2.4 Opiniões sobre Planejamento Estratégico

A partir dos dados coletados, observa-se que 74% dos participantes acreditam que um planejamento estratégico de longo prazo, que organize as disciplinas para os próximos semestres de forma equilibrada, seria útil para sua organização acadêmica, enquanto 26% preferem planejar semestre a semestre. Entre os principais argumentos apresentados pelos respondentes, destacam-se:

- A possibilidade de organização antecipada, permitindo conciliar estudos com trabalho, estágio e outras atividades pessoais.
- Maior previsibilidade e controle sobre o tempo de conclusão do curso, reduzindo atrasos causados por imprevistos ou problemas relacionados a pré-requisitos.
- Facilitação do planejamento estratégico para maximizar o aproveitamento acadêmico,

evitando sobrecargas em determinados semestres e equilibrando a carga horária.

- O interesse em ferramentas automatizadas para ajudar na organização, otimizando o tempo e minimizando erros no planejamento.

Esses resultados, ilustrados na Figura 11 a seguir, reforçam a importância de uma abordagem mais estruturada e tecnológica para o planejamento acadêmico. A adoção de sistemas que auxiliem nesse processo pode trazer benefícios significativos para a experiência dos estudantes, contribuindo para sua organização, produtividade e sucesso na graduação.

Figura 11 – Percepções dos estudantes sobre a importância do planejamento acadêmico estratégico. Onde maioria dos respondentes acreditam na importância de um planejamento a longo prazo.



Fonte: Autoria própria.

Esse tipo de planejamento estratégico de longo prazo pode ser um diferencial importante para os estudantes, permitindo que eles planejem com antecedência e com maior controle, o que pode resultar em uma experiência acadêmica mais fluida e sem

grandes surpresas.

3.2.5 Ferramentas Existentes

Em relação ao uso de ferramentas existentes para o planejamento acadêmico, 5,4% dos participantes mencionaram o uso do site SuaGradeUnB, uma plataforma voltada para a organização de grades de cursos na Universidade de Brasília. Este dado sugere que, embora a maioria dos estudantes utilize métodos manuais ou aplicativos de produtividade, há uma busca crescente por ferramentas mais especializadas que possam facilitar o processo de organização acadêmica. No entanto, a adoção dessas plataformas ainda é limitada, indicando que há espaço para a criação de soluções mais acessíveis ou eficazes.

3.2.6 Necessidade e Motivação para um Sistema Automatizado

Quando questionados sobre a necessidade de um sistema automatizado de planejamento acadêmico, 91,7% dos participantes se mostraram favoráveis à implementação de uma solução desse tipo. As funcionalidades mais desejadas incluem a **possibilidade de visualização automática de conflitos de horário, ajustes dinâmicos de turmas e horários**, além da **indicação de disciplinas com base em pré-requisitos e carga horária**. Os estudantes também destacaram a importância de um sistema que integrasse as informações acadêmicas da UnB e que fosse acessível via dispositivos móveis, oferecendo praticidade e flexibilidade. Esses dados revelam uma clara demanda por soluções tecnológicas que ajudem na organização das grades acadêmicas de forma inteligente e adaptável às necessidades dos estudantes.

3.2.7 Participação no Desenvolvimento

Por fim, a pesquisa questionou os participantes sobre sua disposição em participar de futuras fases do desenvolvimento do sistema. 43,9% expressaram interesse em participar de testes do sistema, o que demonstra uma preocupação com a eficácia e a usabilidade da ferramenta proposta. Esse engajamento potencial pode ser explorado para garantir que o sistema atenda de forma eficaz às necessidades dos estudantes, permitindo ajustes e melhorias contínuas com base no feedback real dos usuários.

3.3 Requisitos

A definição dos requisitos para o sistema proposto foi realizada com base nos dados coletados durante a pesquisa, analisando as necessidades e expectativas dos estudantes da Universidade de Brasília (UnB) em relação a um sistema automatizado de apoio ao planejamento acadêmico. Esse processo foi orientado pelos seguintes objetivos principais:

- Resolver os problemas mais frequentes identificados: Conflitos de horários entre disciplinas e dificuldades relacionadas a pré-requisitos foram priorizados, pois foram os desafios mais citados pelos participantes da pesquisa. Embora o sistema não possa resolver diretamente a falta de vagas, ele pode informar os pré-requisitos das disciplinas, identificar possíveis conflitos de horários e alertar os usuários sobre turmas que apresentam alta demanda.
- Atender às demandas específicas dos estudantes: Funcionalidades consideradas essenciais, como sugestões automáticas de organização de disciplinas e identificação de conflitos de horários, foram incorporadas aos requisitos.
- Promover usabilidade e acessibilidade: Facilitar a adoção do sistema pelos estudantes, integrando informações acadêmicas da universidade e garantindo praticidade no uso.

Para estabelecer a prioridade de implementação dos requisitos, utilizou-se o método MoSCoW, que classifica os requisitos em quatro categorias, como foi explicado na Seção 2.7.

3.3.1 Requisitos Funcionais

Os requisitos funcionais descrevem as funcionalidades do sistema. A Tabela 2 apresenta os requisitos funcionais classificados como *Must Have*, ou seja, essenciais para o funcionamento do sistema.

Tabela 2 – Requisitos Funcionais do Projeto com Prioridade Must Have

ID	Descrição do Requisito Funcional
RF01	O sistema deve permitir o cadastro de novos usuários, solicitando dados como nome, e-mail, senha e matrícula.
RF02	O sistema deve permitir que o usuário faça login, utilizando matrícula e senha previamente cadastrados.
RF03	O sistema deve permitir que o usuário faça logout, encerrando a sessão ativa.
RF04	O sistema deve permitir que o usuário redefina sua senha, utilizando um processo de recuperação que envolva validação por e-mail.
RF05	O sistema deve acessar e consultar o histórico acadêmico do aluno, exibindo disciplinas cursadas e pendentes.
RF06	O sistema deve permitir que o aluno configure preferências de dias e horários disponíveis para cursar disciplinas, a fim de personalizar a sugestão de horários.
RF07	O sistema deve gerar automaticamente uma grade curricular personalizada, considerando disciplinas pendentes, pré-requisitos, carga horária necessária e preferências de horários configuradas pelo aluno.
RF08	O sistema deve identificar conflitos de horário entre disciplinas e avisar o usuário.
RF09	O sistema deve permitir que o usuário exclua sua conta, após confirmação de identidade, removendo todos os dados associados.

Fonte: Autoria própria.

A Tabela 3 apresenta os requisitos funcionais classificados como *Should Have*, que são importantes, mas não essenciais para o funcionamento inicial do sistema.

Tabela 3 – Requisitos Funcionais do Projeto com Prioridade Should Have

ID	Descrição do Requisito Funcional
RF10	O sistema deve permitir que o usuário gerencie seu perfil pessoal, possibilitando a edição de informações como nome e senha.
RF11	O sistema deve coletar dados acadêmicos e pessoais do aluno, construindo um perfil que permita sugerir disciplinas optativas alinhadas aos interesses do usuário.
RF12	O sistema deve permitir que o aluno visualize a grade horária sugerida e edite manualmente as disciplinas e horários, adaptando-a às suas necessidades.

Fonte: Autoria própria.

Por fim, a Tabela 4 apresenta os requisitos funcionais classificados como *Could Have*, que são desejáveis, mas não prioritários.

Tabela 4 – Requisitos Funcionais do Projeto com Prioridade Could Have

ID	Descrição do Requisito Funcional
RF13	O sistema deve fornecer uma projeção visual da grade curricular para os semestres futuros, apresentando um fluxograma das disciplinas pendentes e estimando um tempo para a conclusão do curso.
RF14	O sistema deve exibir informações detalhadas sobre as disciplinas, incluindo pré-requisitos, professor responsável e carga horária.
RF15	O sistema deve destacar disciplinas prioritárias, que sejam essenciais para a conclusão de cadeias de pré-requisitos, sugerindo-as como prioritárias na grade horária.
RF16	O sistema deve ser capaz de enviar notificações para os alunos, como lembretes de prazos de matrícula e alertas sobre alterações em horários de disciplinas.
RF17	O sistema deve permitir que os usuários administrativos (administradores) tenham acesso exclusivo a funcionalidades avançadas, como a gestão de dados do sistema e monitoramento de atividades dos usuários.

Fonte: Autoria própria.

3.3.2 Requisitos Não Funcionais

Os requisitos não funcionais descrevem as condições gerais que o sistema deve atender, como desempenho, segurança e compatibilidade com plataformas. Estes requisitos são classificados de acordo com sua prioridade. A Tabela 5 apresenta os requisitos essenciais para o funcionamento do sistema, enquanto a Tabela 6 traz requisitos desejáveis, mas não obrigatórios. Por fim, a Tabela 7 lista os requisitos que não são considerados para esta versão do sistema.

Tabela 5 – Requisitos Não Funcionais Essenciais para o Funcionamento do Sistema - Must Have

ID	Descrição do Requisito Não Funcional
RNF01	O sistema deve ser responsivo, permitindo acesso eficiente em navegadores de diferentes tamanhos e plataformas.
RNF02	O sistema deve armazenar dados de forma segura, garantindo a confidencialidade e integridade das informações acadêmicas dos usuários.
RNF03	O sistema deve apresentar um desempenho eficiente, com tempos de resposta adequados para todas as funcionalidades.

Fonte: Autoria própria.

Tabela 6 – Requisitos Não Funcionais Recomendados para Melhor Desempenho do Sistema - Should Have

ID	Descrição do Requisito Não Funcional
RNF04	O sistema deve garantir acessibilidade para pessoas com deficiências, respeitando normas como <i>WCAG (Web Content Accessibility Guidelines)</i> .
RNF05	O sistema deve ser escalável, suportando o aumento do número de usuários e acessos simultâneos sem comprometer o desempenho.
RNF06	O sistema deve recomendar disciplinas optativas que se alinhem ao perfil do estudante.

Fonte: Autoria própria.

Tabela 7 – Requisitos Não Funcionais do Projeto- Wont Have

ID	Descrição do Requisito Não Funcional
RNF07	O sistema deve ser compatível com as principais tecnologias e ferramentas utilizadas pela UnB, integrando-se ao SIGAA e outros sistemas relevantes.

Fonte: Autoria própria.

3.4 MVP

O *Produto Mínimo Viável* (MVP) do sistema foi definido com base na priorização dos requisitos por meio do método MoSCoW. O foco do *MVP* é garantir a implementação das funcionalidades essenciais para o planejamento acadêmico, assegurando que o sistema atenda às principais necessidades dos estudantes identificadas na pesquisa.

Os requisitos classificados como *Must Have* são fundamentais para o funcionamento básico do sistema e devem ser implementados na primeira versão. Já os requisitos *Should Have* são desejáveis e complementam a experiência do usuário, podendo ser adicionados posteriormente, conforme a disponibilidade de recursos. O cronograma detalhado das *sprints* para o desenvolvimento deste MVP, incluindo as atividades e os requisitos associados a cada etapa, está apresentado no **Apêndice D**. Adicionalmente, o código-fonte completo da aplicação pode ser consultado no **Apêndice C**.

A Tabela 8 apresenta os requisitos funcionais prioritários que compõem o *MVP*.

Tabela 8 – Requisitos Funcionais do MVP

ID	Descrição do Requisito	Prioridade
RF01	O sistema deve permitir o cadastro de novos usuários, solicitando dados como nome, e-mail, senha e matrícula.	Must Have
RF02	O sistema deve permitir que o usuário faça login, utilizando matrícula e senha previamente cadastrados.	Must Have
RF03	O sistema deve permitir que o usuário faça logout, encerrando a sessão ativa.	Must Have
RF04	O sistema deve permitir que o usuário redefina sua senha, utilizando um processo de recuperação via e-mail.	Must Have
RF05	O sistema deve acessar e consultar o histórico acadêmico do aluno, exibindo disciplinas cursadas e pendentes.	Must Have
RF06	O sistema deve permitir que o aluno configure preferências de dias e horários disponíveis para cursar disciplinas.	Must Have
RF07	O sistema deve gerar automaticamente uma grade curricular personalizada, considerando disciplinas pendentes, pré-requisitos e preferências de horários.	Must Have
RF08	O sistema deve identificar conflitos de horário entre disciplinas e avisar o usuário.	Must Have
RF09	O sistema deve permitir que o usuário exclua sua conta, removendo todos os dados associados.	Must Have
RF10	O sistema deve permitir que o usuário gerencie seu perfil pessoal, editando informações como nome e senha.	Should Have
RF11	O sistema deve coletar dados acadêmicos e sugerir disciplinas optativas alinhadas ao perfil do aluno.	Should Have
RF12	O sistema deve permitir que o aluno edite manualmente a grade horária sugerida.	Should Have

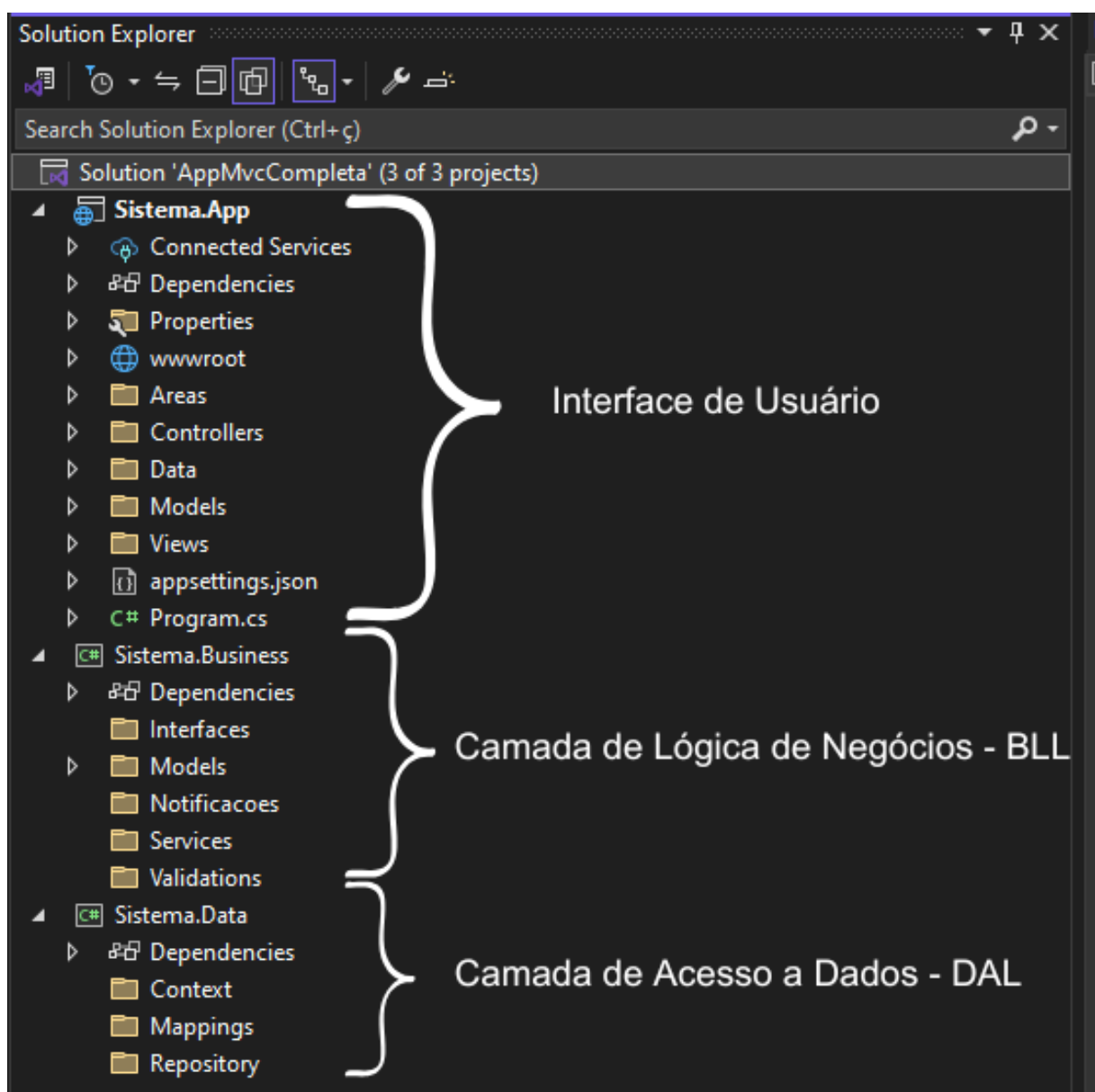
Fonte: Autoria própria.

A definição desses requisitos possibilita a construção de um sistema funcional desde sua versão inicial, permitindo a validação da solução com usuários reais. Com base no feedback dos estudantes, ajustes poderão ser realizados antes da implementação de novos recursos.

3.5 Arquitetura

A arquitetura proposta para o sistema foi estruturada com base no padrão *MVC*, descrito na Seção 2.4, utilizando uma organização lógica em camadas, conforme ilustrado na Figura 12. O sistema foi planejado para ser dividido em três camadas principais: Interface do Usuário, *BLL* (*Business Logic Layer*, ou Camada de Lógica de Negócios) e *DAL* (*Data Access Layer*, ou Camada de Acesso a Dados).

Figura 12 – Representação da Arquitetura do Sistema dividido em Camadas Principais: Interface do Usuário, Lógica de Negócios (*BLL*) e Acesso a Dados (*DAL*).



Fonte: Autoria própria.

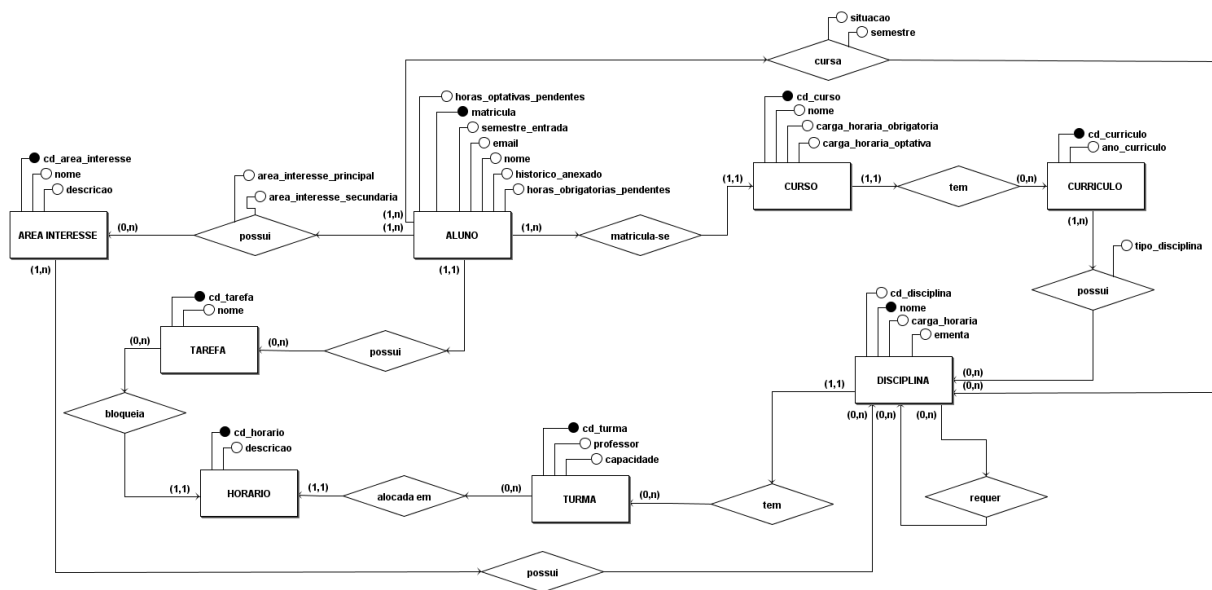
Nesse modelo arquitetural, as interações têm início na camada de Interface do

Usuário, que é responsável por receber as solicitações dos usuários. Essa camada se comunica diretamente com a BLL, que gerencia o processamento das regras de negócio do sistema. Sempre que necessário, a *BLL* realiza chamadas à *DAL*, que se encarrega do acesso e manipulação dos dados de forma segura e eficiente. Essa separação clara entre as camadas visa garantir uma maior organização, modularidade e facilidade de manutenção ao longo do ciclo de vida do sistema (Microsoft Learn, s.d.).

3.6 Modelagem do Banco de Dados

Para a modelagem do banco de dados do sistema, iniciamos com a elaboração de um Diagrama de Entidade-Relacionamento (DER), apresentado na Figura 13. O DER oferece uma visão conceitual de alto nível da estrutura de dados, sendo uma representação intuitiva das entidades principais do sistema, seus atributos e as conexões lógicas entre elas. Ele é fundamental para a compreensão da arquitetura de dados do sistema, mesmo para usuários não técnicos.

Figura 13 – Diagrama de Entidade-Relacionamento (DER) - Representação das Entidades e Relacionamentos do Sistema.



Fonte: Autoria própria.

Conforme ilustrado no DER da Figura 13, as principais entidades que compõem o sistema são:

Entidades

As entidades do sistema foram definidas com os seguintes atributos, onde as chaves primárias estão sublinhadas:

- **ALUNO** (Matrícula, Nome, Email, Semestre de Entrada, Histórico Anexado, Horas Obrigatórias Pendentes, Horas Optativas Pendentes)
- **CURSO** (Código do Curso, Nome, Carga Horária Obrigatória, Carga Horária Optativa)
- **CURRICULO** (Código do Currículo, Ano do Currículo)
- **DISCIPLINA** (Nome, Código da Disciplina, Carga Horária, Ementa)
- **ÁREA DE INTERESSE** (Código da Área de Interesse, Nome, Descrição)
- **TURMA** (Código da Turma, Professor, Capacidade)
- **HORÁRIO** (Código do Horário, Descrição)
- **TAREFA** (Código da Tarefa, Nome)

Relacionamentos

A seguir, são apresentados os relacionamentos entre as entidades do sistema, com suas respectivas cardinalidades, que estabelecem as conexões lógicas entre os dados:

- **ALUNO - matricula-se - CURSO**
Um aluno matricula-se em um único curso, e um curso pode ter vários alunos matriculados. **Cardinalidade: N : 1**
- **CURSO - tem - CURRICULO**
Um curso pode ter vários currículos ao longo do tempo, mas um currículo pertence a um único curso. **Cardinalidade: 1 : N**
- **CURRICULO - possui - DISCIPLINA**
Um currículo é composto por várias disciplinas, e uma disciplina pode fazer parte de vários currículos. A relação possui o atributo "Tipo de Disciplina"(obrigatória, optativa, etc.). **Cardinalidade: N : M**
- **ALUNO - cursa - DISCIPLINA**

Um aluno pode cursar várias disciplinas, e uma disciplina pode ser cursada por vários alunos. A relação armazena o semestre e a situação do aluno na disciplina (aprovado, reprovado, cursando). **Cardinalidade: N : M**

- **DISCIPLINA - requer - DISCIPLINA (Auto-relacionamento)**

Uma disciplina pode exigir várias outras como pré-requisito, e também pode ser pré-requisito para várias disciplinas. **Cardinalidade: N : M**

- **ALUNO - possui - ÁREA DE INTERESSE**

Um aluno pode ter várias áreas de interesse, e uma área de interesse pode ser indicada por vários alunos. **Cardinalidade: N : M**

- **ÁREA DE INTERESSE - possui - DISCIPLINA**

Uma área de interesse pode ter várias disciplinas, e uma disciplina pode pertencer a várias áreas de interesse. **Cardinalidade: N : M**

- **DISCIPLINA - tem - TURMA**

Uma disciplina pode ter várias turmas, mas uma turma pertence a uma única disciplina. **Cardinalidade: 1 : N**

- **TURMA - alocada em - HORÁRIO**

Várias turmas podem ser alocadas no mesmo horário, mas uma turma possui um único horário definido. **Cardinalidade: N : 1**

- **ALUNO - possui - TAREFA**

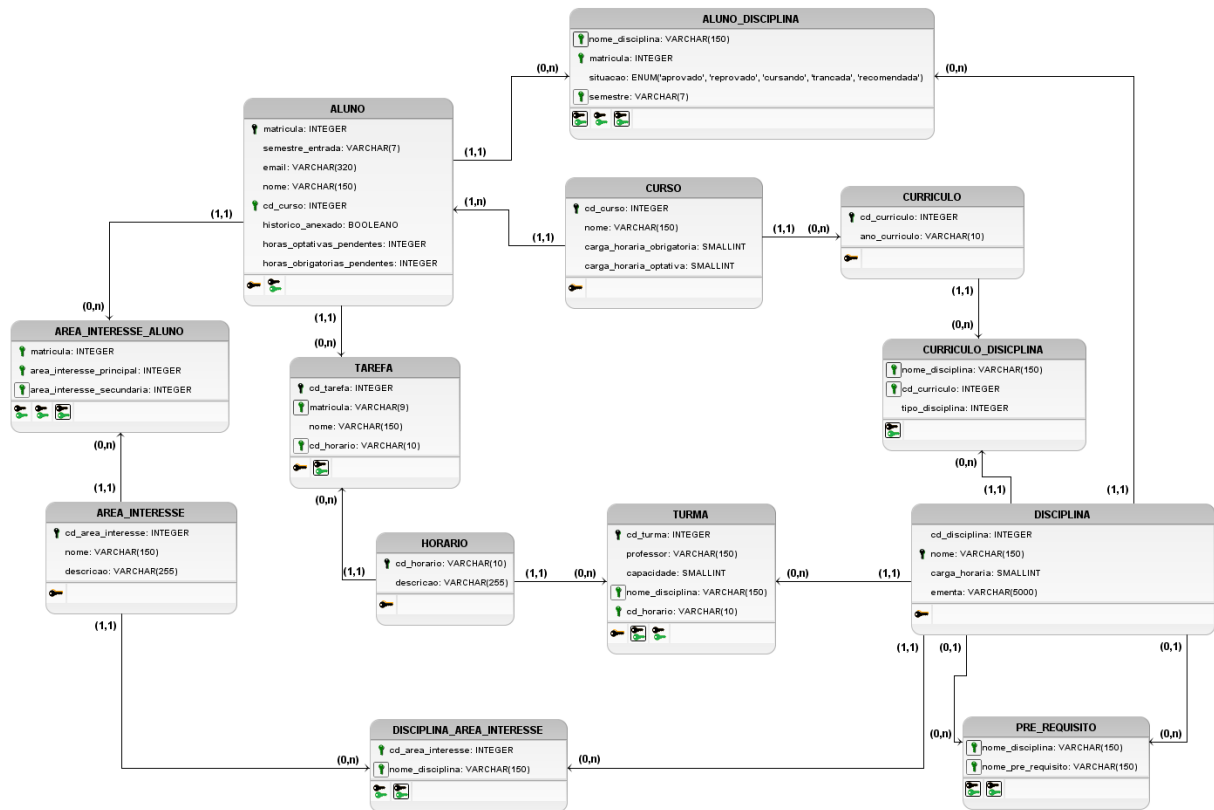
Um aluno pode ter várias tarefas (como provas ou trabalhos), mas uma tarefa pertence a um único aluno. **Cardinalidade: 1 : N**

- **TAREFA - bloqueia - HORÁRIO**

Uma tarefa ocorre em um horário, mas um horário pode ser de várias tarefas. **Cardinalidade: 1 : N**

Após a definição conceitual com o DER, prosseguiu-se para a representação mais detalhada através do Diagrama Lógico de Dados (DLD), ilustrado na Figura 14. O DLD detalha o projeto em seu nível lógico, definindo a estrutura final dos dados através de tabelas, suas chaves primárias e estrangeiras que estabelecem os relacionamentos, e os atributos com seus respectivos tipos de dados, servindo como base para a implementação física do banco de dados.

Figura 14 – Diagrama Lógico de Dados (DLĐ) - Estrutura de Tabelas, Relacionamentos e Tipos de Dados do Sistema.



Fonte: Autoria própria.

A estrutura de banco de dados aqui apresentada, abrangendo desde o modelo conceitual até o lógico, fornece o alicerce para o desenvolvimento robusto e escalável do sistema proposto, garantindo a integridade e a organização das informações acadêmicas.

3.7 Prototipação de telas

Neste capítulo, é apresentada a prototipação proposta para o Sistema de Planejamento Acadêmico Assistido (SPAA). O processo de prototipação teve como objetivo validar a viabilidade da solução, assegurando que as funcionalidades essenciais definidas na Seção 3.3 fossem desenvolvidas de acordo com os requisitos identificados.

A prototipação abrange tanto a modelagem da interface quanto a implementação inicial das principais funcionalidades do sistema. As imagens apresentadas ao longo deste capítulo ilustram diferentes aspectos do protótipo, permitindo uma visão detalhada de sua estrutura e operação.

No entanto, vale ressaltar que nem todas as telas do sistema estão incluídas neste

capítulo, apenas aquelas que representam as funcionalidades principais, essenciais para compreender o funcionamento da solução proposta.

A Figura 15 exibe a tela inicial do sistema, onde o usuário pode visualizar as disciplinas pendentes e concluídas, acompanhar seu progresso acadêmico, consultar sua grade horária atual, receber notificações do sistema e acessar as principais funcionalidades.

Figura 15 – Interface Inicial do SPAA - Visão Geral das Disciplinas e Funcionalidades do Sistema.



Fonte: Autoria própria.

A Figura 16 ilustra a tela da funcionalidade principal, permitindo ao aluno gerenciar sua grade curricular de forma intuitiva. Nela, o estudante pode: selecionar horários e dias da semana preferenciais para cursar disciplinas, enviar sua disponibilidade para o sistema processar, limpar a seleção caso precise recomeçar, escolher as turmas sugeridas automaticamente pelo sistema com base em sua disponibilidade e salvar a grade final para referência futura.

Figura 16 – Interface de Configuração da Grade Horária - Seleção de Preferências Acadêmicas e Geração Automática de Grade no SPAA.

SPAA
Universidade de Brasília

Nome de Usuário ▾

PROGRESSO ACADÊMICO
28%
Visualizar Fluxograma

Notificações

- Prazo de Matrícula Extraordinária se encerra em 3 dias ✕
- Horário de disciplinas atualizado ✕
- Nova turma de Física 1 Experimental ✕

Horas faltantes:
Obrigatória: 1860 horas
Optativa: 560 horas

Selecione os horários e dias que você tem preferência.

Horários	Seg	Ter	Qua	Qui	Sex	Sáb
08:00	---	---	---	---	---	---
10:00	---	---	---	---	---	---
12:00	---	---	---	---	---	---
14:00	---	---	---	---	---	---
16:00	---	---	---	---	---	---
18:00	---	---	---	---	---	---

Enviar Disponibilidade Limpar Seleção
+ Salvar Grade

Turmas Obrigatorias Disponíveis

Disciplina	Código	Professor	Salas	Info	Seleção
FISICA 1 EXPERIMENTAL	IFD0173	Professor	35M12	(i)	☒
GESTÃO DA PRODUÇÃO E QUALIDADE	FGA0184	Professor	35M12	(i)	☒
MÉTODOS DE DESENVOLVIMENTO DE SO...	FGA0138	Professor	35M12	(i)	☒

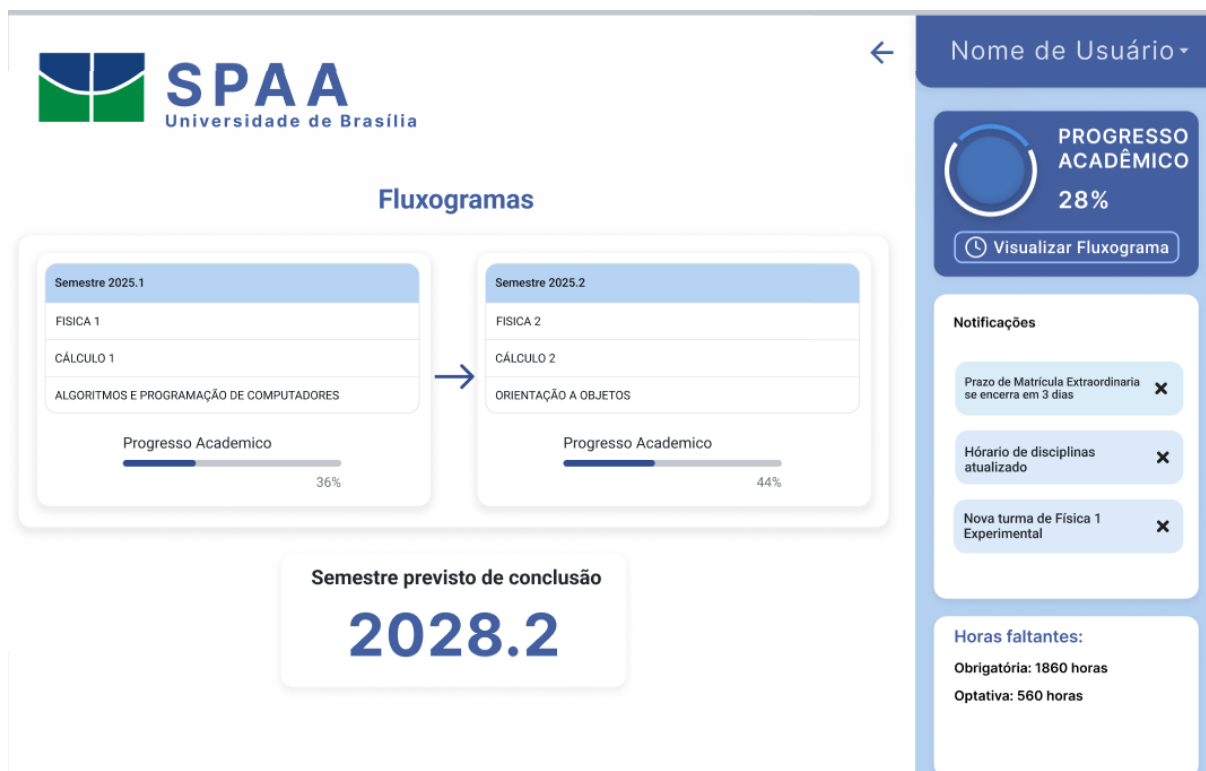
Turmas Optativas Disponíveis

Disciplina	Código	Professor	Salas	Info	Seleção
FISICA 1 EXPERIMENTAL	IFD0173	Professor	35M12	(i)	☒

Fonte: Autoria própria.

Por fim, a Figura 17 exibe a tela de fluxogramas, onde o aluno pode visualizar a sequência de semestres com as disciplinas pendentes. O fluxograma indica o percurso acadêmico ideal, demonstrando que, ao seguir essa trajetória, o aluno concluirá o curso na data prevista pelo sistema.

Figura 17 – Interface de Visualização do Fluxograma Acadêmico - Planejamento da Sequência de Disciplinas no SPAA.



Fonte: Autoria própria.

A prototipação desempenhou um papel fundamental na validação dos conceitos iniciais do sistema, possibilitando ajustes na interface e aprimoramentos no fluxo de interação.

4 Testes Caixa-Branca e Caixa-Preta

Para assegurar a qualidade, a confiabilidade e a manutenibilidade do sistema, foi implementada uma estratégia de testes abrangente, combinando diferentes metodologias. A validação do sistema foi realizada por meio de **testes de caixa-branca**, que exigem conhecimento da estrutura interna do código, e complementada por **testes de caixa-preta**, que focam no comportamento externo e nos requisitos funcionais.

Essa estratégia mista se materializou nos seguintes tipos de testes, aplicados em diferentes níveis da arquitetura:

- **Testes Unitários e de Integração (Caixa-Branca):** Utilizados para garantir, respectivamente, o correto funcionamento das menores unidades de código e a colaboração entre os diferentes componentes internos do sistema.
- **Testes de Ponta a Ponta (*E2E*) (Caixa-Preta):** Empregados para validar os fluxos completos da aplicação, simulando a interação real do usuário e garantindo que o sistema atende aos requisitos de negócio.

4.1 Testes Unitários com Abordagem TDD Adaptada

Nas camadas `SPAA.Business` (responsável pelas regras de negócio) e `SPAA.Data` (responsável pelo acesso e persistência de dados), que constituem o núcleo funcional do sistema, foi adotada uma **abordagem adaptada de Desenvolvimento Orientado a Testes (TDD - Test-Driven Development)**. O TDD convencional preconiza um ciclo rigoroso de "escrever teste, fazê-lo falhar, escrever código para passar no teste, e refatorar"([BECK, 2002](#)). Embora o TDD completo seja reconhecido por promover um design de código robusto e de alta qualidade, sua aplicação integral demanda um tempo de desenvolvimento considerável.

Considerando as restrições de prazo do projeto, optou-se por uma adaptação da metodologia, focando na escrita de testes antes do código de produção nas áreas mais críticas e complexas do sistema. Essa escolha foi motivada pela natureza essencial dessas camadas e seu baixo acoplamento com a infraestrutura externa, o que facilita a testabilidade e a validação independente. A priorização do TDD nessas seções permitiu guiar o design do código de forma eficiente, assegurar a cobertura dos requisitos de negócio e das operações de dados fundamentais, e construir um software mais robusto e modular nos seus pontos nevrálgicos. Como Robert C. Martin reforça em sua obra "Código Limpo"([MARTIN, 2009](#)):

“O código limpo é aquele que foi escrito por alguém que se importa. Ele não é apenas funcional, mas também é testável e de fácil manutenção. O TDD ajuda o desenvolvedor a manter o foco nos requisitos e a evitar acoplamentos desnecessários.”

A aplicação estratégica e adaptada do *TDD* resultou em uma cobertura de testes excepcional nessas camadas, atingindo **97%** na camada **SPAA.Data** e **85.4%** na camada **SPAA.Business**. É importante notar que, devido à não implementação completa da metodologia *TDD* em todas as funcionalidades, a **cobertura de ramificação** não atingiu os níveis máximos possíveis, refletindo a otimização entre a aplicação rigorosa do *TDD* e as limitações de tempo do projeto. No entanto, os índices de cobertura alcançados demonstram a alta confiabilidade e a validação criteriosa dos componentes essenciais do sistema.

4.2 Testes na Camada de Apresentação (SPAA.App)

Na camada **SPAA.App**, responsável por orquestrar as requisições e a apresentação, foi adotada uma estratégia de testes mista, combinando testes unitários e testes de integração, ambos implementados após o desenvolvimento do código.

4.2.1 Testes Unitários

O foco dos testes unitários foi direcionado exclusivamente aos **controllers**. A decisão de não testar unitariamente as *views* foi estratégica, pois sua validação é mais eficaz e realista quando realizada por meio de **testes de ponta a ponta (E2E)**, que simulam a interação real do usuário. Dessa forma, os testes unitários da camada **SPAA.App** validaram a lógica de orquestração dos controllers: tratamento de rotas, manipulação de dados de entrada e formatação das respostas.

4.2.2 Testes de Integração

Os testes de integração também foram concentrados na camada **SPAA.App**. A justificativa para essa decisão é que esta camada serve como a "ponte" entre o mundo externo (requisições HTTP) e a lógica de negócio interna. É nos controllers que as chamadas aos serviços da camada **SPAA.Business** são orquestradas.

Portanto, testar essa integração é crucial para garantir que o fluxo completo de uma requisição — desde sua chegada ao controller até a execução da regra de negócio

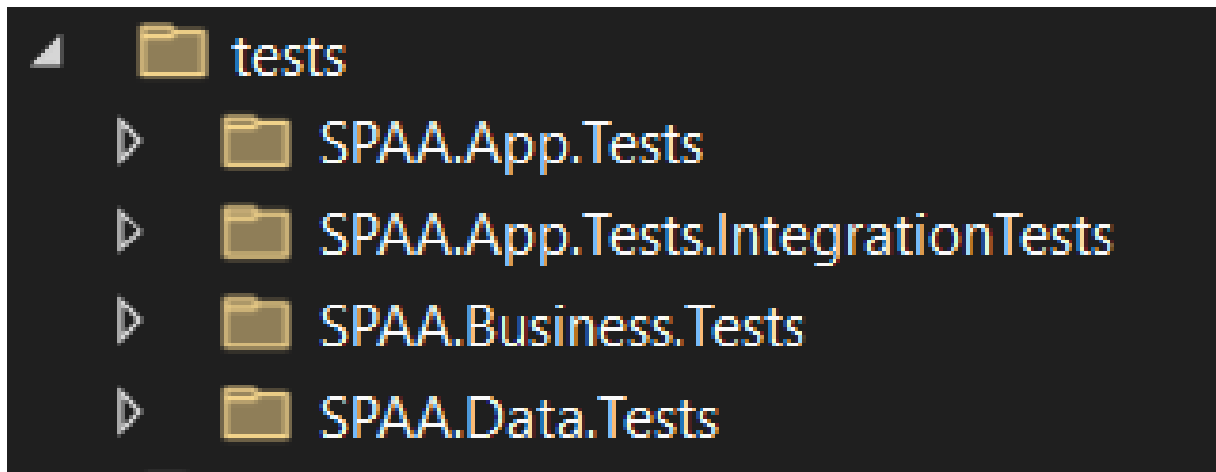
correspondente e o retorno da resposta — funcione como esperado. Esses testes validam a correta colaboração entre a infraestrutura web (framework) e os serviços de negócio, garantindo que o sistema como um todo responda adequadamente às interações do usuário.

A estratégia combinada de testes unitários e de integração na camada `SPAA.App` resultou em uma cobertura de **62.5%** para esta camada.

4.3 Organização e Cobertura Geral dos Testes

Para manter a clareza e o desacoplamento, a estrutura de testes do projeto foi organizada em quatro projetos distintos, espelhando a arquitetura do sistema e separando os tipos de teste, conforme ilustrado na Figura 18.

Figura 18 – Estrutura de pastas dos projetos de testes unitários e de integração.



Fonte: Autoria própria.

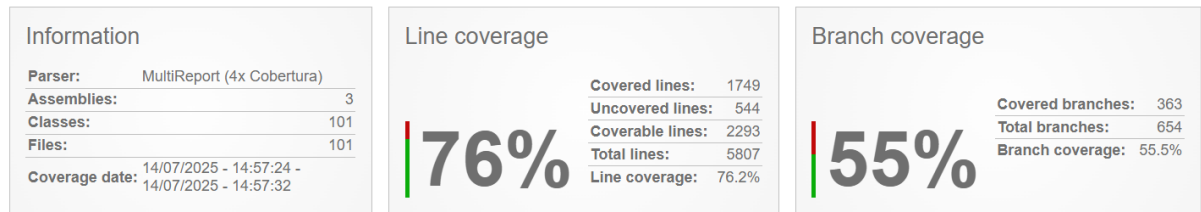
Os projetos são:

- **SPAA.App.Tests:** Focado nos testes unitários dos controllers.
- **SPAA.App.Tests.IntegrationTests:** Contém os testes de integração da camada de aplicação.
- **SPAA.Business.Tests:** Abriga os testes unitários para os serviços e regras de negócio.
- **SPAA.Data.Tests:** Destinado aos testes unitários para os repositórios e operações de acesso a dados.

A execução de toda a suíte de testes de caixa-branca validou a eficácia da estratégia

adotada. Conforme detalhado no relatório da Figura 19, o projeto alcançou uma cobertura geral de **76%** das linhas de código e **55%** de cobertura de ramificação (*branch coverage*).

Figura 19 – Relatório consolidado da ferramenta de cobertura de testes, exibindo a porcentagem de linhas e ramificações de código cobertas.



Fonte: Autoria própria.

Esses resultados evidenciam um alto grau de validação em todo o sistema, conferindo maior segurança contra regressões e garantindo a qualidade do código produzido. A combinação de alta cobertura nas camadas de núcleo com testes de integração na camada de apresentação assegura que tanto as unidades individuais quanto suas interações funcionem conforme o esperado.

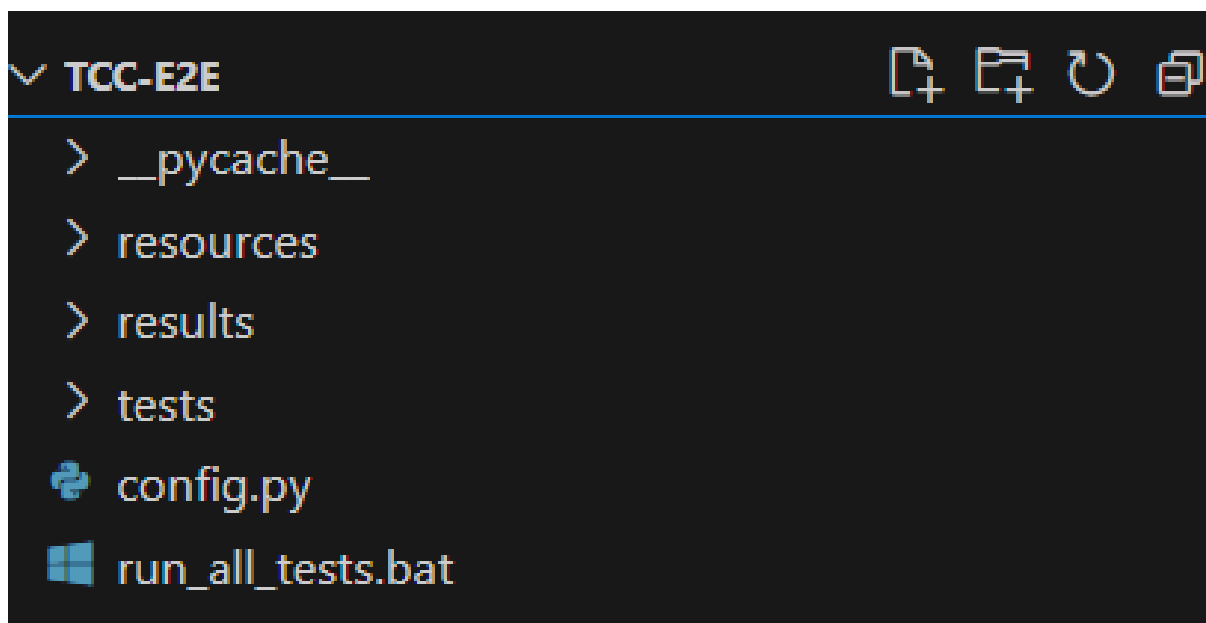
4.4 Testes de Ponta a Ponta (E2E)

Os testes de ponta a ponta (*End-to-End - E2E*) foram empregados como uma estratégia de validação de caixa-preta, focando na verificação dos fluxos de usuário completos da aplicação, simulando o comportamento de um usuário real em um ambiente integrado. Para a automação desses testes, foi utilizada a ferramenta *Robot Framework*, descrita na Seção 2.8.4, em conjunto com a linguagem *Python* e a biblioteca *Selenium*, permitindo a interação direta com a interface web do sistema.

4.4.1 Estrutura dos Testes E2E

A organização dos artefatos de teste foi estruturada para garantir manutenibilidade, reusabilidade e clareza. A estrutura de diretórios, conforme ilustrado na Figura 20, segue as melhores práticas de automação com *Robot Framework*. O repositório completo contendo os scripts de teste *E2E* pode ser consultado no **Apêndice B**.

Figura 20 – Estrutura de diretórios dos testes E2E.



Fonte: Autoria própria.

- **tests/**: Este diretório contém as suítes de teste, que são arquivos `.robot` agrupados por funcionalidade. Cada arquivo define um ou mais cenários de teste (*Test Cases*) utilizando uma sintaxe de palavras-chave.
- **resources/**: Contém os arquivos de recursos, como o `common_keywords.robot`. Este arquivo centraliza palavras-chave reutilizáveis em múltiplas suítes, como "Abrir Navegador" ou "Gerar Dados De Registro". Essa abordagem, similar ao padrão *Page Object Model (POM)*, reduz a duplicação de código e facilita a manutenção.
- **results/**: Diretório onde o *Robot Framework* armazena automaticamente os resultados de cada execução, incluindo relatórios detalhados (`report.html`) e logs de execução (`log.html`).
- **run_all_tests.bat**: Um script de execução que simplifica o processo de rodar todas as suítes de teste de uma só vez, garantindo que a cobertura completa seja verificada de forma consistente.

4.4.2 Cenários de Teste Implementados

Para garantir a estabilidade dos fluxos de entrada e das funcionalidades iniciais do sistema, que representam o primeiro contato do usuário com a plataforma, foram definidos e automatizados alguns cenários de teste estratégicos

4.4.2.1 Cenário 1: Registro de Novo Aluno

Este conjunto de testes valida o fluxo de criação de uma nova conta de aluno. Foram implementados múltiplos casos de teste para assegurar a robustez do formulário de registro, incluindo:

- **Caminho de Sucesso (CT01):** Verifica se um usuário consegue se registrar com sucesso ao preencher todos os campos do formulário com dados válidos, sendo redirecionado para a tela de login ao final.
- **Validação de Campos Obrigatórios (CT02 a CT07, CT09):** Foram criados testes específicos para garantir que o sistema exiba uma mensagem de erro apropriada caso o usuário tente submeter o formulário sem preencher cada um dos campos obrigatórios, como Nome, E-mail, Matrícula e Senha.
- **Validação de Confirmação de Senha (CT08):** Um cenário específico foi criado para verificar se o sistema impede o registro e informa o usuário quando os campos "Senha" e "Confirme a Senha" não coincidem.

4.4.2.2 Cenário 2: Autenticação de Aluno

Este conjunto de testes valida o processo de login e o fluxo de recuperação de acesso, cobrindo os seguintes casos:

- **Login com Sucesso (CT01):** Testa o fluxo principal, onde um usuário com credenciais válidas consegue acessar o sistema e ser redirecionado para a página inicial.
- **Login com Credenciais Inválidas (CT04, CT05):** Cenários que verificam se o sistema nega o acesso e exibe uma mensagem de erro quando o usuário informa uma matrícula ou senha incorreta.
- **Validação de Campos Vazios (CT02, CT03, CT06):** Garante que o sistema apresente mensagens de erro se o usuário tentar fazer login sem preencher o campo de matrícula, o de senha ou ambos.
- **Recuperação de Senha (CT07, CT08):** Valida a funcionalidade "Esqueceu a senha", verificando se o sistema responde adequadamente tanto para e-mails válidos quanto inválidos, exibindo uma mensagem de confirmação genérica para proteger a privacidade dos dados.

4.4.2.3 Cenário 3: Anexar Histórico Escolar (Primeiro Acesso)

Este conjunto de testes valida a etapa inicial do fluxo de primeiro acesso de um novo aluno, focando na funcionalidade de upload do histórico escolar. Os casos de teste garantem a correta manipulação de diferentes tipos de arquivos e o feedback adequado ao usuário:

- **Upload com Sucesso (CT01):** Verifica se um usuário recém-cadastrado consegue realizar o login e anexar um arquivo de histórico escolar válido em formato PDF, resultando em uma mensagem de sucesso e o avanço para a próxima etapa do primeiro acesso (implícito no CT).
- **Validação de Formato de Arquivo (CT02):** Garante que o sistema impede o upload e exibe uma mensagem de erro apropriada quando o usuário tenta anexar um arquivo que não está no formato PDF esperado.
- **Validação de Arquivo Vazio (CT03):** Testa se o sistema detecta e rejeita o upload de um arquivo PDF válido, mas que está vazio, exibindo a mensagem de erro correspondente.
- **Validação de Conteúdo Não-Histórico (CT04):** Assegura que, mesmo que o arquivo seja um PDF válido, o sistema valida seu conteúdo e rejeita o upload se o documento não for reconhecido como um histórico escolar relevante, informando o usuário sobre o problema.

4.4.2.4 Cenário 4: Gestão de Disponibilidade de Horários

Este conjunto de testes visa validar a funcionalidade de gestão da disponibilidade de horários, uma etapa crucial para a montagem da grade do aluno. Os casos de teste cobrem desde o envio bem-sucedido de horários até cenários de validação e interação com a interface:

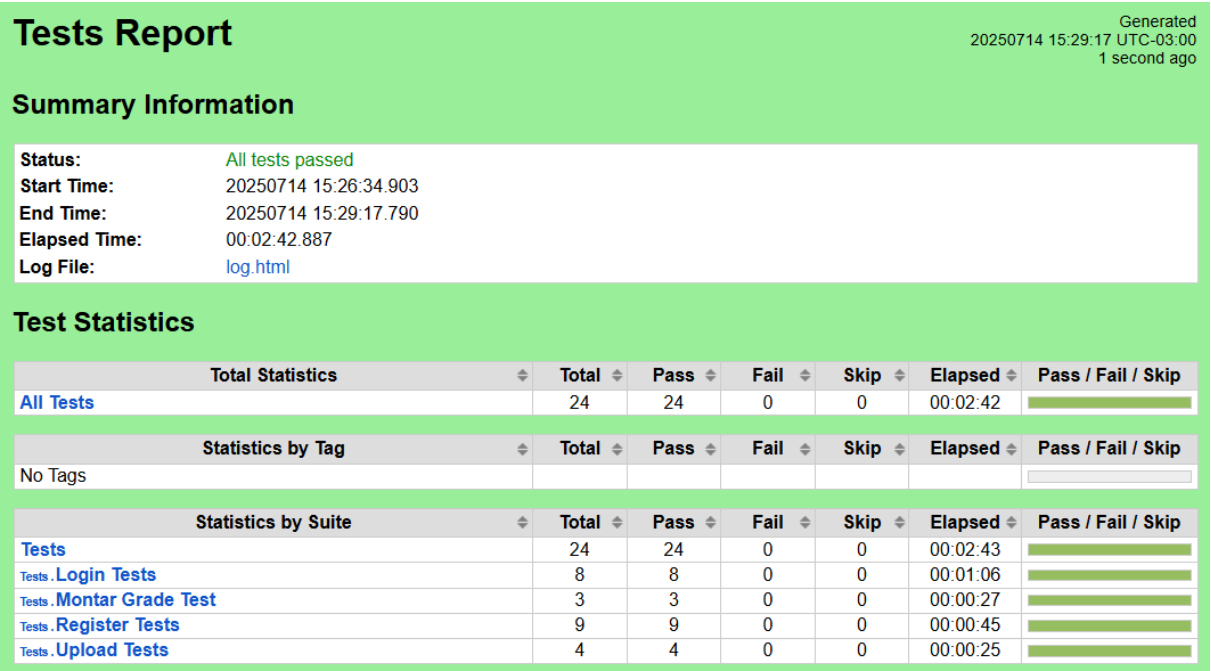
- **Envio de Disponibilidade com Sucesso (CT01):** Testa o fluxo principal onde um usuário, após login, acessa a tela de grade de horários, seleciona múltiplos blocos de tempo e submete sua disponibilidade com êxito, recebendo a mensagem de confirmação esperada.
- **Envio de Disponibilidade Sem Seleção (CT02):** Verifica se o sistema impede o envio e exibe uma mensagem de erro apropriada quando o usuário tenta submeter a disponibilidade sem selecionar nenhum horário na grade.
- **Limpeza de Seleção de Horários (CT03):** Garante que, ao selecionar horários

na grade e posteriormente acionar o botão "Limpar Seleção", todas as marcações de disponibilidade previamente feitas são removidas da interface, restaurando o estado inicial dos elementos.

4.4.3 Resultados e Análise

A execução dos testes *E2E* é realizada a cada ciclo de desenvolvimento ou antes de uma nova versão ser implantada. Os resultados são gerados pelo *Robot Framework* em formato HTML, como exemplificado na Figura 21. O relatório apresenta uma visão geral do status de cada suíte e de cada caso de teste (aprovado ou reprovado) e permite uma análise detalhada de cada passo executado através dos logs.

Figura 21 – Relatório de execução gerado pelo Robot Framework.



Fonte: Autoria própria.

A suíte de testes automatizados se mostrou eficaz na detecção de regressões, garantindo que alterações no código-fonte não introduzissem erros nas funcionalidades existentes. O relatório de execução mais recente indica que todos os 24 testes executados foram aprovados, resultando em uma taxa de aprovação de 100% para o ciclo de testes atual. Especificamente, as suítes "*Login Tests*", "*Montar Grade Tests*", "*Register Tests*" e "*Upload Tests*" obtiveram 100% de aprovação em seus respectivos 8, 3, 9 e 4 casos de teste. Este resultado demonstra a estabilidade e qualidade da aplicação no momento da execução.

5 Sistema de Perfis da Aplicação SPAA

Este capítulo descreve o funcionamento do sistema de perfis utilizado na aplicação SPAA (Sistema de Planejamento Acadêmico Assistido), com foco na classificação de estudantes com base em seus interesses e afinidades acadêmicas. O objetivo é fornecer recomendações personalizadas de disciplinas optativas, alinhadas ao perfil individual do estudante dentro do curso de Engenharia de Software da Universidade de Brasília.

5.1 Perfis Derivados do PPC do Curso

A construção dos perfis utilizados pela aplicação SPAA baseia-se na análise do Projeto Pedagógico de Curso (PPC) vigente do curso de Engenharia de Software da UnB ([Universidade de Brasília, 2023](#)). A partir da estrutura curricular proposta, foram identificados cinco grandes eixos de especialização ou afinidade, que representam agrupamentos coerentes de disciplinas e competências.

- **Perfil 1: Programação e Infraestrutura Técnica** – Engloba alunos com maior afinidade por lógica, estruturas de dados, sistemas operacionais, redes e tecnologias de *backend*. São estudantes que tendem a preferir componentes como Desenvolvimento *Web* Avançado, Computação em Nuvem e Segurança da Informação.
- **Perfil 2: Projeto e Análise de Software** – Foca em estudantes interessados em engenharia de requisitos, modelagem, arquitetura e *design* de *software*. Disciplinas como Arquitetura de *Software*, Engenharia de Produto e Métodos Ágeis são mais atraentes para esse grupo.
- **Perfil 3: Qualidade, Testes e Manutenção** – Representa estudantes voltados à confiabilidade do *software*, com interesse por processos de validação, testes automatizados, manutenção evolutiva e engenharia de *software* experimental.
- **Perfil 4: Gestão de Produtos e Processos** – Perfil para alunos com maior inclinação para liderança, visão de produto, métricas de desempenho e coordenação de equipes.
- **Perfil 5: Fundamentos Matemáticos e Computacionais** – Reúne estudantes com gosto por algoritmos, teoria da computação, ciência de dados e áreas mais formais. Preferem disciplinas como Computação Quântica, Otimização Combinatória e Teoria dos Grafos.

Esses perfis não são excludentes, mas funcionam como pontos de referência. A proposta não é rotular estudantes, mas identificar, com base em suas preferências declaradas, qual eixo está mais alinhado ao seu momento acadêmico. Essa abordagem facilita o planejamento de trajetórias coerentes e motivadoras.

5.2 Aplicação do Algoritmo de Mapeamento de Perfil

A identificação do perfil de interesse mais adequado para o estudante é realizada por meio de um algoritmo de classificação baseado em afinidade de respostas. Para uma análise mais aprofundada sobre a teoria e o funcionamento do algoritmo *K-Nearest Neighbors* (*KNN*), que fundamenta conceitualmente a abordagem aqui empregada, consulte a Seção 2.11. Diferentemente de uma implementação clássica do *KNN*, que utiliza distâncias euclidianas e votação entre vizinhos, a abordagem adotada neste trabalho foca na contagem direta das afinidades manifestadas pelo estudante. Cada perfil-base foi estabelecido a partir de médias ponderadas de respostas de especialistas e de alunos que demonstraram desempenho compatível com eixos temáticos específicos em disciplinas optativas. Os vetores resultantes são concebidos em um espaço vetorial de múltiplas dimensões, onde cada dimensão corresponde a uma questão do formulário de mapeamento de perfil.

A classificação do perfil do estudante ocorre através da consolidação das respostas fornecidas no formulário. As perguntas que compõem este formulário estão detalhadas no **Apêndice E**. Para cada questão, a escolha do estudante incrementa a contagem de afinidade para um ou mais perfis-base correspondentes, conforme previamente configurado no sistema. A opção classificada como "Nada", que denota uma resposta neutra ou ambígua, não contribui para a afinidade com nenhum perfil específico.

Ao final do preenchimento, o sistema calcula a porcentagem de afinidade de cada perfil-base com base no total de respostas válidas (excluindo as respostas "Nada"). O perfil principal é então identificado como aquele que atinge ou supera um limiar mínimo de 22% de afinidade. Perfis adicionais que também superem esse limiar são classificados como secundários, indicando áreas de interesse complementares do estudante. Esse método permite uma classificação direta e interpretável das preferências do estudante.

A Figura 22 ilustra a distribuição dos cinco perfis profissionais em um espaço conceitual bidimensional, refletindo as afinidades demonstradas pelas respostas fornecidas pelos estudantes. Cada perfil é representado por uma cor específica na malha hexagonal: cinza para Fundamentos Matemáticos e Computacionais, verde para Programação e Infraestrutura Técnica, roxo para Gestão de Produtos e Processos, azul para Projeto e Análise de *Software*, e vermelho para Qualidade, Testes e Manutenção. Os hexágonos na malha correspondem a diferentes regiões do espaço de afinidade de perfis, e o posicionamento

de um estudante nesse mapa é conceitualmente determinado pelas suas respostas ao formulário.

Figura 22 – Representação Gráfica dos Perfis Profissionais e Suas Afinidades.



Fonte: Autoria própria.

A escolha da representação hexagonal para a visualização dos perfis é fundamentada em suas propriedades geométricas e simbólicas. Os hexágonos permitem um preenchimento eficiente do espaço em uma grade bidimensional, facilitando a representação de adjacências e transições entre diferentes perfis. Além disso, a forma hexagonal, com suas seis arestas, pode ser interpretada como um símbolo de interconexão e equilíbrio, características relevantes para a multidisciplinaridade frequentemente presente nas áreas de Tecnologia da Informação. Visualmente, essa estrutura em colmeia torna a identificação de *clusters* (perfis puros) e de zonas de transição (perfis híbridos) mais intuitiva, contribuindo para a clareza na interpretação das afinidades do estudante.

Esta visualização permite identificar distintos padrões de afinidade. Hexágonos de cores puras, rodeados exclusivamente por outros hexágonos da mesma tonalidade, representam perfis com forte alinhamento a uma única área específica, indicando que o estudante demonstra uma afinidade predominante com aquela competência. Zonas de transição, caracterizadas por hexágonos que se encontram próximos a diferentes cores ou que exibem uma mistura, sugerem características interdisciplinares, denotando que o estudante pode se beneficiar de competências e conhecimentos oriundos de múltiplas áreas. Hexágonos cinza representam áreas neutras ou de baixa afinidade, onde não há uma correlação significativa com nenhum dos perfis profissionais definidos. A proximidade

entre hexágonos de cores distintas, nesse contexto visual, indica uma afinidade conceitual entre os perfis correspondentes, facilitando uma interpretação visual direta da classificação gerada pelo sistema.

Atualmente, o algoritmo opera com base em um conjunto fixo de vetores de perfil previamente definidos, cujas afinidades são calculadas a partir das respostas do estudante. As escolhas do formulário geram um vetor de interesse que é comparado diretamente a esses perfis estabelecidos, sem dependência de um banco de dados de respostas anteriores dos alunos. Essa abordagem inicial viabiliza a recomendação de perfis mesmo em contextos com volume de dados reduzido. Em etapas futuras, com a ampliação da base de usuários e o enriquecimento das respostas, será possível refinar os vetores de perfil, possivelmente integrando modelos de classificação mais complexos ou algoritmos de aprendizado de máquina que se beneficiem de grandes volumes de dados.

6 Conclusão

Este capítulo sintetiza os principais resultados e contribuições do presente trabalho, revisita os objetivos propostos e reafirma a viabilidade da solução desenvolvida. Adicionalmente, são discutidas as adaptações e o rigor metodológico empregados para otimizar a experiência do usuário e garantir a robustez do sistema, destacando a abrangente estratégia de testes adotada. Por fim, são delineados os caminhos para trabalhos futuros, evidenciando o potencial de expansão e sustentabilidade do sistema proposto.

6.1 Reafirmação da Viabilidade e Contribuições do SPAA

Este trabalho concretizou o desenvolvimento e a validação do Sistema de Planejamento Acadêmico Assistido (SPAA), uma ferramenta inovadora projetada para auxiliar estudantes da Universidade de Brasília, com foco no campus Gama (FCTE), na organização eficiente de suas grades curriculares. O estudo validou de forma inequívoca a hipótese de que a ausência de ferramentas automatizadas para o planejamento acadêmico é uma lacuna que contribui para problemas como conflitos de horário, sobrecarga e, em última instância, a evasão universitária.

A metodologia adotada, com destaque para a pesquisa de campo centrada no estudante (Seção 3.2), foi fundamental para compreender as reais dores e necessidades do público-alvo. Os dados coletados confirmaram a predominância de métodos manuais de planejamento e a clara demanda por uma solução tecnológica que otimize esse processo. Através do SPAA, demonstrou-se a viabilidade de automatizar a montagem da grade curricular (RF07), considerando pré-requisitos e horários, e de identificar conflitos (RF08), cumprindo os objetivos gerais e específicos definidos.

Além de sua funcionalidade principal, o SPAA se destaca por uma contribuição substancial: a implementação de um sistema de perfis inovador. Este sistema, embora conceitualmente fundamentado no algoritmo *K-Nearest Neighbors* (KNN), emprega uma abordagem adaptada para mapear as afinidades acadêmicas do estudante a um dos perfis pré-definidos do curso, conforme detalhado na Seção 5.2. Essa capacidade de adaptação e personalização, enraizada diretamente no Projeto Pedagógico de Curso (PPC) da UnB, diferencia o SPAA das abordagens genéricas, estabelecendo a base para futuras recomendações personalizadas de disciplinas optativas (RF11), um objetivo a ser plenamente desenvolvido em trabalhos posteriores.

6.2 Qualidade e Confiabilidade da Solução

A robustez e a qualidade do SPAA foram asseguradas por uma estratégia de testes abrangente e orientada, aplicando tanto técnicas de caixa-branca quanto de caixa-preta. Notadamente, a adoção de uma **abordagem adaptada de *Design Orientado a Testes (TDD)*** no desenvolvimento das camadas de negócio (*SPAA.Business*) e de acesso a dados (*SPAA.Data*) foi crucial. Essa prática, focada nas áreas mais críticas do sistema devido a restrições de prazo, garantiu uma elevada cobertura de testes (85.4% e 97% respectivamente), solidificando o núcleo do sistema contra regressões e assegurando a consistência das regras de negócio. Essa escolha não apenas validou o funcionamento das menores unidades de código, mas também guiou o *design*, resultando em um *software* modular e de fácil manutenção.

Na camada de apresentação (*SPAA.App*), os testes unitários em *controllers* e os testes de integração complementaram a validação, assegurando que a orquestração das requisições e a colaboração entre as camadas funcionassem conforme o esperado, com uma cobertura de 62.5%. Adicionalmente, a implementação de testes de ponta a ponta (*E2E*) com *Robot Framework*, simulando a interação do usuário final em cenários críticos como registro, autenticação e gestão de horários, confirmou a fluidez e a estabilidade da aplicação em um ambiente integrado. A aprovação de 100% dos 24 casos de teste implementados neste ciclo atesta a eficácia da estratégia mista e a qualidade do sistema em sua versão atual.

6.3 Caminhos para o Futuro: Expansão e Sustentabilidade do SPAA

Embora a versão atual do SPAA, concebida como *Produto Mínimo Viável (MVP)*, demonstre plena funcionalidade e eficácia na resolução dos problemas centrais previamente identificados, o trabalho desenvolvido estabeleceu uma arquitetura robusta e modular em camadas. Tal fundamentação arquitetônica facilita substancialmente a incorporação de funcionalidades adicionais em iterações subsequentes do sistema. As diretrizes para Trabalhos Futuros, delineadas nesta seção, evidenciam a inerente escalabilidade e o considerável potencial de aprimoramento da ferramenta.

6.3.1 Implementação do MVP2 e Expansão de Funcionalidades

Conforme delineado no escopo inicial do projeto e detalhadamente documentado nas fases primárias de desenvolvimento, uma parcela dos requisitos levantados não foi integralmente contemplada na versão corrente do sistema, sendo categorizada como de

implementação futura e, portanto, excluída do escopo do *MVP*. Estes pontos constituem a base para a próxima fase de desenvolvimento, designada como *MVP2*.

Dentre as funcionalidades cruciais para a evolução do SPAA, destaca-se a **projeção visual da grade curricular para semestres futuros (RF13)**. Embora os dados inerentes a esta funcionalidade já estejam disponíveis internamente – uma vez que o sistema detém informações sobre as disciplinas cursadas, pendentes e o perfil de interesse do usuário –, os aspectos visuais e a *interface* para sua apresentação ainda não foram desenvolvidos. A concretização desta funcionalidade é tecnicamente viável e promete agregar valor substancial à experiência do usuário, ao propiciar uma visualização clara e preditiva de sua trajetória acadêmica.

Outro pilar para o avanço do sistema reside na **implementação de funcionalidades administrativas (RF17)**. Esta proposta visa o desenvolvimento de um painel exclusivo para usuários com prerrogativas administrativas, conferindo-lhes capacidades de gestão e monitoramento, tais como acesso a dados agregados, controle de perfis de usuários e supervisão de padrões de uso da aplicação. Embora o SPAA já possua a capacidade de persistir dados individualizados em sua base de dados, a camada de administração correspondente ainda não foi desenvolvida. A construção de tal *interface* exigirá definições precisas de permissões, mecanismos de autenticação diferenciados e rigorosos controles de acesso. Esta expansão detém um potencial estratégico significativo em contextos institucionais, como sua adoção por coordenações de curso e núcleos docentes.

A **integração com o Sistema Integrado de Gestão de Atividades Acadêmicas (SIGAA) (RNF06)** representa uma otimização substancial na experiência do usuário. Este requisito não funcional preconiza a compatibilidade do SPAA com as principais tecnologias e ferramentas empregadas pela Universidade de Brasília. Embora, na configuração atual, o SPAA opere de maneira autônoma, esta integração possibilitaria a leitura direta do histórico escolar do discente, eliminando a necessidade de procedimentos de *upload* manual de documentos. Adicionalmente, simplificaria o processo de alimentação inicial das bases de dados do sistema, como currículos e informações de disciplinas, que presentemente dependem da execução de *scripts* em *Python* (o repositório de código-fonte contendo esses *scripts* pode ser acessado no Apêndice F). Em uma perspectiva futura, é altamente desejável que o SIGAA viabilize a integração via *Interface de Programação de Aplicações (API)*, promovendo assim uma experiência acadêmica mais abrangente e personalizável para os estudantes.

Aprimoramentos substanciais também são previstos para o **sistema de perfis e recomendação de disciplinas optativas (RF11)**. Atualmente, o mapeamento de afinidades é realizado a partir de respostas a um formulário estático. Em iterações futuras, a intenção é enriquecer e aprimorar o algoritmo *K-Nearest Neighbors (KNN)* utilizado, incorporando *feedback* contínuo dos usuários. Com a ampliação da base de dados proveniente

das interações e escolhas reais dos estudantes, será possível refinar os vetores de perfil e tornar as recomendações de disciplinas optativas cada vez mais precisas e adaptadas às tendências e preferências coletivas. Esse processo de aprendizado contínuo, impulsionado pelo *feedback* dos usuários, é fundamental para maximizar a relevância e a utilidade das sugestões oferecidas pelo SPAA, consolidando-o como uma ferramenta de planejamento acadêmico dinâmico e inteligente.

Por fim, foram identificadas diversas oportunidades para o aprimoramento da **Experiência do Usuário (UX)**. Notavelmente, resalta-se a necessidade de ajustes no *layout* e no comportamento de certas janelas *pop-up* – como a de exclusão de conta –, que poderiam ser redesenhadas para oferecer uma interação visual mais clara e responsiva. Outra melhoria pertinente envolve a funcionalidade de edição da grade horária: atualmente, ao acessar essa área, o usuário não visualiza as disciplinas e tarefas previamente cadastradas. A implementação futura prevê que esses dados sejam automaticamente pré-carregados, otimizando o processo de edição e mitigando o atrito durante o planejamento.

6.3.2 Expansão da Cobertura e Abrangência dos Testes

Apesar de a aplicação SPAA ter empregado uma metodologia consistente de testes ao longo de seu ciclo de desenvolvimento, vislumbra-se um considerável espaço para a expansão da profundidade e abrangência dos testes automatizados. Esta expansão é particularmente relevante no que concerne à validação da integração entre os módulos e à simulação de fluxos completos de interação do usuário.

Como trabalho futuro, propõe-se um incremento substancial na porcentagem de **testes de integração**, com o intuito de validar de maneira mais exaustiva a comunicação e a interação entre os distintos componentes do sistema, notadamente as camadas de aplicação, de negócio e de dados. A ampliação desses testes é crucial para a detecção de inconsistências que não seriam facilmente identificadas em testes unitários isolados, garantindo a robustez e a confiabilidade do sistema em fluxos que envolvem múltiplas operações encadeadas.

Adicionalmente, é fortemente recomendada a expansão do conjunto de **testes de ponta a ponta (E2E)**, mediante o aumento do volume de casos de teste (*CTs*) e a diversificação de cenários. Esta expansão permitirá uma cobertura mais fidedigna dos contextos reais de uso enfrentados pelos usuários, incluindo variações de comportamento, entradas inválidas, uso prolongado do sistema e manipulação de dados personalizados. Com isso, busca-se assegurar que as funcionalidades mais críticas da aplicação permaneçam operacionais conforme o esperado, mesmo após a implementação de modificações futuras no código-fonte. Tais ações são de particular relevância à medida que a aplicação se move em direção a um cenário de adoção institucional, onde a estabilidade, a previsibilidade e a

robustez constituem fatores imprescindíveis para a confiabilidade da ferramenta.

6.3.3 Considerações Finais sobre o Potencial do SPAA

Em síntese, este trabalho não apenas validou a premissa fundamental de que uma ferramenta automatizada pode otimizar significativamente o planejamento acadêmico, mas também resultou na entrega de um sistema intrinsecamente robusto e escalável. O SPAA, portanto, configura-se como uma contribuição valiosa para a comunidade acadêmica da Universidade de Brasília, oferecendo um potencial transformador para a experiência do estudante e servindo como um modelo paradigmático para futuras inovações no suporte à trajetória universitária. As implementações propostas nesta seção representam trajetórias viáveis para a evolução contínua do sistema, baseadas tanto em funcionalidades preexistentes quanto em requisitos acordados como fora do escopo mínimo inicial. A continuidade do desenvolvimento, especialmente com o crescimento da base de usuários, poderá viabilizar a incorporação dessas funcionalidades em versões futuras, contribuindo para o amadurecimento progressivo da ferramenta e sua eventual adoção em escala institucional.

Referências

APRENDER3. *Aprender 3 - Moodle*. 2001. Disponível em: <<https://aprender3.unb.br/>>. Citado na página 22.

BECCENERI, J. C. Meta-heurísticas e otimização combinatória: Aplicações em problemas ambientais. In: *Meta-Heurísticas e Otimização Combinatória: Aplicações em Problemas Ambientais*. São José dos Campos: INPE, 2008. Citado na página 23.

BECK, K. *Test-Driven Development by Example*. Boston, MA: Addison-Wesley Professional, 2002. Citado na página 63.

BURKE, E.; NEWALL, J. Solving examination timetabling problems through adaption of heuristic orderings. *Annals of Operations Research*, Springer, v. 129, p. 107–134, 2004. Disponível em: <<https://doi.org/10.1023/B:ANOR.0000030684.30824.08>>. Citado na página 23.

CABRERA, A. F. et al. *The Convergence Between Two Theories of College Persistence*. 1992. ResearchGate. Disponível em: <https://www.researchgate.net/publication/261699782_The_Convergence_Between_Two_Theories_of_College_Persistence>. Citado na página 17.

CARVALHO, N. et al. Avaliação da eficácia e utilização do sigaa na formação discente: Estudo de caso nos cursos de licenciatura da ufpi, campus de parnaíba evaluation of the effectiveness and use of sigaa in the discipline training: Case study in the courses of licensing of the ufpi, parnaíba campus. *CONGRESSO INTERNACIONAL DE EDUCAÇÃO E TECNOLOGIAS*, 2018. Citado na página 21.

DevMedia. *Introdução ao Padrão MVC*. 2023. Acesso em: 13 dez. 2024. Disponível em: <<https://www.devmedia.com.br/introducao-ao-padrao-mvc/29308>>. Citado na página 24.

ELMASRI, R.; NAVATHE, S. B. *Sistemas de Banco de Dados*. [S.l.]: Editora, 2008. Revisão Técnica: Luis Ricardo de Figueiredo, Mestre em Ciências da Computação e Doutorando pela USP-Ribeirão Preto. Citado na página 25.

FERRÃO, A. D. L. HorÁriosufcg: Ferramenta para o auxílio no planejamento de horÁrios e escolha de disciplinas para a matrícula. 2023. Disponível em: <<http://dspace.sti.ufcg.edu.br:8080/xmlui/bitstream/handle/riufcg/29308/ARTHUR%20DE%20LIMA%20FERR%c3%83O%20-%20TCC%20ARTIGO%20CI%c3%8aNCIA%20DA%20COMPUTA%c3%87%c3%83O%20CEEI%202023.pdf?sequence=1&isAllowed=y>>. Citado na página 37.

GOMES, V. Controle universitário: Um aplicativo de controle dos estudos e do andamento do graduando ao longo do curso superior. 3 2021. Disponível em: <https://monografias.ufop.br/bitstream/35400000/3464/6/MONOGRRAFIA_Contr%20leUniversit%c3%a1rioAplicativo.pdf>. Citado 2 vezes nas páginas 38 e 39.

LOCK, A. *ASP.NET Core in Action*. New York, NY: Manning Publications, 2018. Citado na página 30.

MARTIN, R. C. *Código Limpo: Habilidades Práticas do Agile Software*. [S.l.]: Alta Books, 2009. Tradução da obra original *Clean Code: A Handbook of Agile Software Craftsmanship*. Citado na página 63.

Microsoft Learn. *Common web application architectures*. s.d. Acesso em: 24 jan. 2025. Disponível em: <<https://learn.microsoft.com/pt-br/dotnet/architecture/modern-web-apps-azure/common-web-application-architectures>>. Citado na página 56.

Ministério da Educação (MEC). *Censo da Educação Superior 2018*. 2019. Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira (INEP). Acesso em: 5 dez. 2024. Disponível em: <http://download.inep.gov.br/educacao_superior/censo_superior/documentos/2019/apresentacao_censo_superior2018.pdf>. Citado na página 16.

Ministério da Educação (MEC). *Política Nacional de Assistência Estudantil (PNAES)*. 2024. Ministério da Educação, Governo Federal. Acesso em: 5 dez. 2024. Disponível em: <<https://www.gov.br/mec/pt-br/areas-de-atuacao/es/pnaes>>. Citado na página 17.

MIRANDA, E. Moscow rules: A quantitative exposé. *Agile Processes in Software Engineering and Extreme Programming*, p. 28–43, 6 2022. Disponível em: <<https://link.springer.com/bookseries/7911>>. Citado 2 vezes nas páginas 29 e 30.

MOROSINI, M. C. et al. *A Evasão na Educação Superior no Brasil: Uma Análise da Produção de Conhecimento nos Periódicos Qualis entre 2000-2011*. 2012. Repositório PUCRS. Disponível em: <<https://repositorio.pucrs.br/dspace/handle/10923/8762>>. Citado na página 17.

MYERS, G.; BADGETT, T.; SANDLER, C. *THE ART OF SOFTWARE TESTING*. Third edition. [S.l.]: John Wiley Sons, Inc, 2012. v. 3. 85-112 p. ISBN 978-1-118-03196-4. Citado na página 31.

ngrok. *ngrok Documentation*. 2025. Acesso em: 15 jul. 2025. Disponível em: <<https://ngrok.com/docs>>. Citado na página 33.

ngrok. *ngrok: Secure tunnels to localhost*. 2025. Acesso em: 15 jul. 2025. Disponível em: <<https://ngrok.com/>>. Citado na página 33.

NORMANDO, C. *Arquitetura MVC e Princípios de Projeto*. 2023. Acesso em: 13 dez. 2024. Disponível em: <<https://medium.com/@celionormando/arquitetura-mvc-e-princ%C3%ADpios-de-projeto-3d0b278ef910>>. Citado na página 23.

Python Software Foundation. *Python Programming Language*. 2025. Acesso em: 4 jul. 2025. Disponível em: <<https://www.python.org/>>. Citado na página 31.

ROBERTS, D.; JOHNSON, R. et al. Evolving frameworks: A pattern language for developing object-oriented frameworks. *Pattern languages of program design*, v. 3, p. 471–486, 1996. Citado na página 30.

Robot Framework Foundation. *Robot Framework*. 2025. Acesso em: 4 jul. 2025. Disponível em: <<https://robotframework.org/>>. Citado na página 32.

SCHWABER, K.; SUTHERLAND, J. *Um Guia Definitivo para o Scrum: As Regras do Jogo*. 2016. Scrum Guides. Disponível em: <<https://scrumguides.org/docs/scrumguide/v2016/2016-Scrum-Guide-Portuguese-Brazilian.pdf>>. Citado 2 vezes nas páginas 19 e 20.

- SEI. *SEI - Sistema Eletrônico de Informações*. 2009. Disponível em: <<https://www.portalsei.unb.br/>>. Citado na página 22.
- SIGAA. *SIGAA - Sistema Integrado de Gestão de Atividades Acadêmicas*. 2009. Disponível em: <<https://sigaa.unb.br/sigaa/portais/discente/discente.jsf>>. Citado na página 21.
- SOARES, M. Metodologias Ágeis extreme programming e scrum para o desenvolvimento de software. *Revista Eletrônica de Sistemas de Informação*, v. 3, n. 1, 2004. ISSN 1677-3071. Disponível em: <<https://periodicosibepes.org.br/index.php/reinfo/article/view/146>>. Citado na página 19.
- STICKDORN, M. et al. *Isto é Design de Serviço na Prática: Como*. 1ª edição. ed. [S.l.]: Manning Publications, 2018. ISBN 978-8582605271. Citado 2 vezes nas páginas 34 e 42.
- TINTO, V. *Dropout from Higher Education: A Theoretical Synthesis of Recent Research*. 1975. JSTOR. Disponível em: <<https://doi.org/10.2307/1170024>>. Citado na página 16.
- TORRES, J. M. Gradeufop: Uma aplicação web para ajuste de grades curriculares. 2024. Disponível em: <https://www.monografias.ufop.br/bitstream/35400000/7170/4/MONOGRAFIA_GradeUFOPAplica%c3%a7%c3%a3oWeb.pdf>. Citado na página 39.
- Universidade de Brasília. *Projeto Pedagógico do Curso de Engenharia de Software*. 2023. Disponível em: <https://software.unb.br/images/ESW/PPC_Software_2024_12_16.pdf>. Citado na página 71.
- Universidade de Brasília. *Modelagem de Dados*. s.d. Acesso em: 21 jan. 2025. Disponível em: <https://sae.unb.br/cae/conteudo/unbfga/lbd/banco2_modelagemDeDados.html>. Citado 2 vezes nas páginas 25 e 28.
- VASCONCELOS, N. et al. Contribuição dos sistemas integrados de gestão para as práticas de ensino e aprendizagem. *Augusto Guzzo Revista Acadêmica*, v. 9, p. 45–52, 07 2012. Citado na página 21.
- xUnit Community. *xUnit Documentation*. 2023. Acesso em: 13 dez. 2024. Disponível em: <<https://xunit.net/#documentation>>. Citado na página 31.

A FORMULÁRIO DE COLETA DE DADOS

Este apêndice apresenta o formulário utilizado para a coleta de dados com os estudantes da Universidade de Brasília (UnB). O questionário foi aplicado online e teve como objetivo compreender as dificuldades no planejamento acadêmico e as expectativas em relação a uma ferramenta de apoio. As imagens a seguir ilustram a estrutura e o conteúdo das perguntas do formulário.

Figura 23 – Tela inicial do formulário de coleta de dados

A imagem mostra a tela inicial de um formulário online. No topo, há uma barra decorativa com uma fotografia de uma paisagem urbana com plantas e um edifício, sobreposta a um logotipo verde e azul. Abaixo, o título 'Formulário para Coleta de Dados sobre Planejamento Acadêmico' é exibido em uma caixa branca com bordas verdes. O texto principal descreve o objetivo do formulário e a importância dos dados coletados. Abaixo, há uma seção de consentimento e uma data de validade. No rodapé, uma legenda indica que o asterisco (*) marca perguntas obrigatórias.

Formulário para Coleta de Dados sobre Planejamento Acadêmico

Este formulário tem como objetivo coletar informações para compreender as dificuldades enfrentadas no planejamento acadêmico e avaliar a eficácia de um sistema automatizado desenvolvido para esse fim. Os dados coletados serão de grande relevância para embasar as análises e conclusões deste Trabalho de Conclusão de Curso (TCC), contribuindo diretamente para a construção de soluções que atendam às necessidades identificadas.

Ao prosseguir, você concorda em participar da pesquisa e autoriza o uso dos dados fornecidos exclusivamente para este estudo.

O formulário ficará disponível dos dias 09/12 até 17/12.

*** Indica uma pergunta obrigatória**

Fonte: Autoria própria

Figura 24 – Seção do formulário: Perguntas de perfil dos respondentes (1 e 2)

1. Você é/foi estudante da UnB? *

☐ Sim, UnB Campus Darcy Ribeiro

☐ Sim, UnB Campus Gama

☐ Sim, UnB Campus Ceilândia

☐ Sim, UnB Campus Planaltina

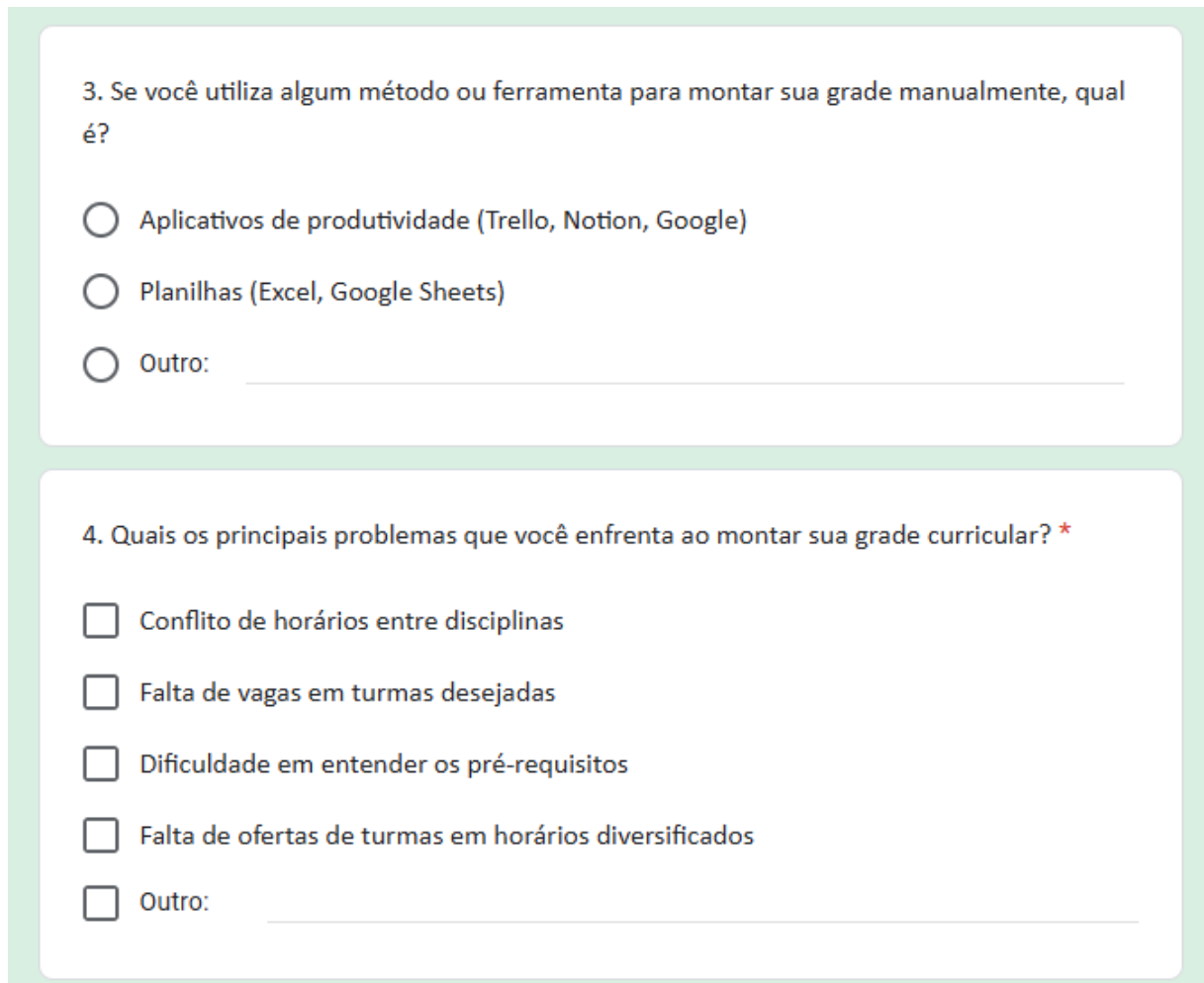
☐ Não

2. Como você monta sua grade horária atualmente? *

Sua resposta _____

Fonte: Autoria própria

Figura 25 – Seção do formulário: Métodos atuais de planejamento (Perguntas 3 e 4)



3. Se você utiliza algum método ou ferramenta para montar sua grade manualmente, qual é?

- ☐ Aplicativos de produtividade (Trello, Notion, Google)
- ☐ Planilhas (Excel, Google Sheets)
- ☐ Outro: _____

4. Quais os principais problemas que você enfrenta ao montar sua grade curricular? *

- ☐ Conflito de horários entre disciplinas
- ☐ Falta de vagas em turmas desejadas
- ☐ Dificuldade em entender os pré-requisitos
- ☐ Falta de ofertas de turmas em horários diversificados
- ☐ Outro: _____

Fonte: Autoria própria

Figura 26 – Seção do formulário: Dificuldades no planejamento acadêmico (Perguntas 5 e 6)

5. Você já deixou de cursar disciplinas importantes devido a conflitos ou indisponibilidade? *

☐ Sim, frequentemente

☐ Sim, algumas vezes

☐ Não, nunca

6. Qual a sua principal dificuldade em relação aos pré-requisitos das disciplinas? *

☐ Dificuldade em identificar quais são os pré-requisitos de uma matéria

☐ Esquecer de cursar disciplinas que são pré-requisito

☐ Pré-requisitos confusos ou mal definidos

☐ Nenhuma dificuldade

☐ Outro: _____

Fonte: Autoria própria

Figura 27 – Seção do formulário: Opiniões sobre planejamento estratégico (Perguntas 7 e 8)

7. Como a falta de organização da grade curricular impacta seu progresso acadêmico? *

- ☐ Atraso na conclusão do curso
- ☐ Sobrecarga de disciplinas em um único semestre
- ☐ Necessidade de adaptar a grade constantemente
- ☐ Falta de cumprir requisitos necessários para seguir no fluxo
- ☐ Outro: _____

8. Você acredita que um planejamento de longo prazo, que organize suas disciplinas para os próximos semestres de forma estratégica e equilibrada, seria útil para você? *

- ☐ Sim
- ☐ Não, prefiro planejar semestre a semestre

Fonte: Autoria própria

Figura 28 – Seção do formulário: Ferramentas existentes (Pergunta 9)

Continuação

9. Você utiliza ou conhece alguma plataforma que facilite a montagem da grade curricular? *

- ☐ Sim
- ☐ Não

Fonte: Autoria própria

Figura 29 – Seção do formulário: Necessidade de sistema automatizado (Perguntas 10 e 11)

10. Você sente falta de uma ferramenta para o planejamento de grade curricular? *

☐ Sim, seria muito útil

☐ Não, consigo me organizar com métodos atuais

11. O que mais te motivaria a usar um sistema que ajuda a planejar a grade curricular? *

☐ Recomendações personalizadas

☐ Interface intuitiva

☐ Economia de tempo

☐ Cumprimento eficiente dos pré-requisitos

☐ Outro: _____

Fonte: Autoria própria

Figura 30 – Seção do formulário: Expectativas e funcionalidades (Perguntas 12 e 13)

12. Quais funcionalidades você considera indispensáveis em um sistema de gestão acadêmica assistido? *

- ☐ Recomendações automáticas de grade horaria
- ☐ Visualização de conflitos de horário
- ☐ Identificação de matérias que trancam outras
- ☐ Sugestão de matérias optativas com base no meu perfil
- ☐ Planejamento de longo prazo
- ☐ Planejamento baseado no histórico acadêmico do aluno
- ☐ Feedback e relatórios sobre o progresso acadêmico
- ☐ Outro: _____

13. O quão disposto você está a utilizar um sistema que auxilia no planejamento de sua grade curricular? *

- ☐ Muito disposto
- ☐ Disposto
- ☐ Indiferente
- ☐ Não muito disposto
- ☐ Não disposto

Fonte: Autoria própria

Figura 31 – Seção do formulário: Interesse em participação e feedback (Perguntas 14, 15 e 16)

14. Quais seriam suas principais preocupações ao usar um sistema assistido para montar sua grade curricular? *

- ☐ Segurança de dados
- ☐ Complexidade do sistema
- ☐ Custo ou acessibilidade
- ☐ Falta de personalização
- ☐ Outro: _____

15. Quais outras funcionalidades ou melhorias você gostaria de ver em um sistema de planejamento de grade curricular? *

Sua resposta _____

16. Você gostaria de receber mais informações ou participar de testes para o desenvolvimento do sistema? *

- ☐ Sim
- ☐ Não

Fonte: Autoria própria

A detalhada estrutura do formulário, conforme ilustrado, foi fundamental para coletar dados que subsidiaram a análise das necessidades dos estudantes e a posterior concepção do sistema proposto.

B REPOSITÓRIO DOS TESTES DE PONTA A PONTA (E2E)

Este apêndice fornece o link para o repositório *online* que contém os scripts completos dos testes de Ponta a Ponta (E2E) desenvolvidos para o sistema. O acesso a este repositório permite a consulta detalhada de todos os cenários de teste, palavras-chave e arquivos de recursos utilizados na automação, conforme descrito na Seção 4.4.

URL do Repositório: <<https://github.com/Hellen159/TCC-E2E>>

O repositório está organizado para facilitar a navegação e a compreensão da estrutura dos testes, incluindo os diretórios para suítes, recursos e resultados, bem como o script de execução principal.

C REPOSITÓRIO DO CÓDIGO-FONTE DA APLICAÇÃO

Este apêndice fornece o link para o repositório *online* que contém o código-fonte completo da aplicação desenvolvida. O acesso a este repositório permite a consulta detalhada de toda a implementação do sistema, incluindo as camadas de negócio, acesso a dados, interface de usuário e demais componentes que compõem a arquitetura da solução proposta.

URL do Repositório: <<https://github.com/Hellen159/TCC-Projeto>>

O repositório está organizado de forma a facilitar a navegação pelo projeto, permitindo a análise da estrutura de diretórios e dos arquivos fonte que foram utilizados no desenvolvimento do sistema.

D CRONOGRAMA DE DESENVOLVIMENTO DO MVP (SPRINTS)

Este apêndice detalha o cronograma das *sprints* adotadas para o desenvolvimento do Produto Mínimo Viável (MVP) do sistema. O projeto seguiu uma metodologia ágil, com *sprints* quinzenais focadas em entregas incrementais e validações contínuas, conforme descrito na Seção 3.4.

O planejamento busca garantir a evolução progressiva do sistema até a entrega do MVP e a preparação para a apresentação final do Trabalho de Conclusão de Curso (TCC).

Tabela 9 – Planejamento das Sprints de Desenvolvimento do MVP

Sprint	Período	Atividades Planejadas
Sprint 1	04/03 a 17/03	Levantamento técnico, modelagem inicial do banco de dados, criação da estrutura do projeto e implementação da autenticação básica (cadastro, login e logout).
Sprint 2	18/03 a 31/03	Implementação da funcionalidade de recuperação de senha, exclusão de conta e estruturação da base para gestão de perfil. Testes iniciais da autenticação.
Sprint 3	01/04 a 14/04	Integração com sistema simulado (<i>mock</i>) de histórico acadêmico, exibição de disciplinas cursadas e pendentes, e início da construção da interface do usuário.
Sprint 4	15/04 a 28/04	Desenvolvimento do formulário de definição de preferências e restrições de horários. Refinamento da modelagem acadêmica com base nos dados históricos.
Sprint 5	29/04 a 12/05	Implementação do algoritmo de montagem automática da grade, considerando histórico acadêmico e preferências do usuário. Realização de testes de consistência.
Sprint 6	13/05 a 26/05	Desenvolvimento da funcionalidade de verificação e resolução automática de conflitos de horários. Integração da interface com o gerador de grade.
Sprint 7	27/05 a 09/06	Implementação da funcionalidade de edição manual da grade pelo aluno e conclusão do módulo de gerenciamento de perfil.
Sprint 8	10/06 a 23/06	Desenvolvimento da funcionalidade de sugestão de disciplinas optativas com base no perfil do aluno. Aplicação de testes com usuários.
Sprint 9	24/06 a 07/07	Refatoração do código, realização de testes de usabilidade, correções de bugs e otimizações visuais com base no <i>feedback</i> dos testes.
Sprint 10	08/07 a 21/07	Congelamento da versão final, execução de testes finais, finalização da documentação técnica, redação das conclusões e preparação da apresentação do TCC.

Fonte: Autoria própria.

Como pode ser observado na Tabela 9, a estruturação das *sprints* prioriza inicialmente a implementação das funcionalidades essenciais, como a autenticação de usuários e a integração com o histórico acadêmico. Em seguida, são desenvolvidas as funcionalidades relacionadas ao planejamento acadêmico e ao algoritmo de geração de grades, culminando com fases de ajustes, testes de usabilidade e otimizações.

O planejamento foi elaborado de forma a permitir a construção gradual e consistente do sistema, assegurando que cada ciclo de desenvolvimento contemplasse não apenas a implementação de novas funcionalidades, mas também a realização de testes e ajustes baseados no feedback obtido, garantindo uma evolução contínua e centrada nas necessidades dos usuários finais.

E FORMULÁRIO DE MAPEAMENTO DE PERFIL DO ESTUDANTE

Este apêndice apresenta as dez questões que compõem o formulário utilizado para o mapeamento do perfil de interesse do estudante, juntamente com suas respectivas opções de resposta. Cada figura corresponde a uma pergunta do formulário, conforme implementado no sistema.

Figura 32 – Questão 1 do Formulário de Mapeamento de Perfil.

Etapa 2: Suas áreas de interesse

Nos diga mais sobre suas preferências acadêmicas.



1. Durante o desenvolvimento de um sistema para uma organização, a equipe precisa tomar decisões críticas. Uma delas é escolher entre priorizar a estrutura robusta do backend ou focar primeiro na prototipação rápida da interface. O prazo é apertado, e ambas são importantes. Diante disso, qual decisão você defenderia com mais convicção?

A) Eu faria qualquer uma das duas, desde que alguém me dissesse claramente o que fazer.

B) Começar pela base do sistema, garantindo uma arquitetura sólida desde o início.

C) Focar primeiro no que o usuário verá e interagirá, mesmo que depois precise refatorar.

Figura 33 – Questão 2 do Formulário de Mapeamento de Perfil.



2. Uma funcionalidade lançada recentemente começou a apresentar falhas intermitentes. A equipe precisa resolver a situação rapidamente. Diante da pressão do tempo, qual seria seu primeiro passo?

A) Investigar causas profundas, revisar as decisões anteriores e entender o impacto no design.

B) Eu começaria por qualquer um dos dois caminhos, dependendo de quem me pedisse para agir primeiro.

C) Aplicar testes automatizados, revisar logs e garantir estabilidade antes de investigar causas profundas.

Fonte: Autoria própria

Figura 34 – Questão 3 do Formulário de Mapeamento de Perfil.



3. Um sistema antigo precisa ser mantido funcionando, mas a documentação está desatualizada e há dívidas técnicas. A equipe deve dividir as tarefas urgentemente. Qual papel você assumiria nessa situação?

A) Atualizar a documentação, entender o funcionamento e propor melhorias no processo.

B) Refatorar partes do código e automatizar testes para garantir que nada se quebre.

C) Eu faria qualquer uma das duas, desde que alguém me dissesse claramente o que fazer.

Fonte: Autoria própria

Figura 35 – Questão 4 do Formulário de Mapeamento de Perfil.



4. Seu grupo vai participar de uma competição onde o desafio é desenvolver uma solução para otimizar rotas de entrega. Com pouco tempo disponível, como você se sentiria mais útil nesse cenário?

A) Trabalhando nos cálculos, buscando uma solução matemática eficaz.

B) Eu ajudaria no que fosse pedido, sem me importar tanto com a natureza da tarefa.

C) Criando interfaces intuitivas e visualizações claras para mostrar os resultados aos avaliadores.

Fonte: Autoria própria

Figura 36 – Questão 5 do Formulário de Mapeamento de Perfil.



5. Uma startup quer desenvolver um sistema que funcione para milhões de usuários, com alta performance e segurança. Antes de começar, é preciso decidir qual aspecto discutir primeiro. Qual você priorizaria?

A) Arquitetura escalável, banco de dados eficiente e servidores bem distribuídos.

B) Eu simplesmente seguiria a decisão da maioria do time ou do líder do projeto.

C) Entender o usuário final e mapear suas necessidades reais antes de projetar a infra.

Fonte: Autoria própria

Figura 37 – Questão 6 do Formulário de Mapeamento de Perfil.



6. Um cliente não técnico quer saber se o sistema que sua equipe está desenvolvendo está 'funcionando direito'. Você tem poucas horas para preparar uma demonstração. Qual abordagem você usaria?

A) Explicaria o funcionamento com fluxos simples e mostraria protótipos navegáveis.

B) Mostraria os testes automatizados, os indicadores de cobertura e estabilidade.

C) Eu faria algo misto, sem me comprometer com uma abordagem específica.

Fonte: Autoria própria

Figura 38 – Questão 7 do Formulário de Mapeamento de Perfil.



7. Durante a faculdade, você participa de um projeto onde todos vão programar, mas os papéis são flexíveis. Sem combinar previamente, qual tipo de tarefa você assumiria espontaneamente?

A) Organizar o planejamento, acompanhar o progresso e garantir comunicação eficaz entre os membros.

B) Resolver problemas de lógica e estruturar o sistema para escalar com facilidade.

C) Eu esperaria outra pessoa me atribuir uma tarefa antes de decidir sozinho.

Fonte: Autoria própria

Figura 39 – Questão 8 do Formulário de Mapeamento de Perfil.



8. Sua equipe tem um sistema funcionando bem, mas as prioridades mudaram: agora é preciso manter e evoluir o software existente. Diante dessa mudança, qual seria sua prioridade?

A) Analisar dados de uso, coletar feedback e planejar evoluções com base nas necessidades reais dos usuários.

B) Criar estratégias de versionamento, testes de regressão e monitoramento de erros.

C) Eu seguiria o que já estava sendo feito até alguém solicitar algo diferente.

Fonte: Autoria própria

Figura 40 – Questão 9 do Formulário de Mapeamento de Perfil.



9. Diante de um problema complexo, sua primeira abordagem costuma ser...

A) Quebrar o problema em partes menores, usar modelos matemáticos e resolver passo a passo.

B) Eu não teria um método definido — dependeria do dia ou da inspiração.

C) Testar diferentes caminhos, observar padrões e ajustar conforme os resultados.

Fonte: Autoria própria

Figura 41 – Questão 10 do Formulário de Mapeamento de Perfil.



10. Em um projeto com prazo apertado, onde há várias formas de atingir o objetivo, qual caminho você tende a escolher?

A) Buscar soluções com base em fundamentos sólidos, mesmo que exijam mais tempo de desenvolvimento.

B) Escolher a solução prática que funcione rápido e possa ser ajustada depois.

C) Eu faria o que parecesse mais fácil ou o que outro colega sugerisse primeiro.

Fonte: Autoria própria

F REPOSITÓRIO DE CÓDIGO-FONTE DOS SCRIPTS DE EXTRAÇÃO DE DADOS DO SIGAA

Este apêndice fornece o link para o repositório de código-fonte que contém os *scripts* desenvolvidos em Python, responsáveis pela extração e tratamento dos dados da plataforma SIGAA. Esses *scripts* são essenciais para o processo de alimentação e popularização inicial da base de dados do sistema SPAA com informações curriculares e de disciplinas.

Link para o Repositório: <<https://github.com/Hellen159/TCC-Popula-Banco>>

O repositório está organizado de forma a permitir a fácil navegação e compreensão da estrutura do projeto e das suas diversas funcionalidades relacionadas à gestão de dados.