



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

Andressa Evelyn de Sa Silva
Victor Hugo Seiji Ueyama

ESTUDO 360
Um aplicativo acadêmico

Americana, SP
2024

Andressa Evelyn de Sa Silva
Victor Hugo Seiji Ueyama

ESTUDO 360
Um aplicativo acadêmico

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

Orientador: Prof. Me. Jonas Bodê

Este trabalho corresponde à versão final do Trabalho de Conclusão de Curso apresentado por Andressa Evelyn de Sa Silva, Victor Hugo Seiji Ueyama e orientado pelo Prof. Me. Jonas Bodê.

Americana, SP
2024

FICHA CATALOGRÁFICA

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

SILVA, Andressa Evelyn de Sa

Estudo 360: um aplicativo acadêmico. / Andressa Evelyn de Sa Silva, Victor Hugo Seiji Ueyama – Americana, 2024.

136f.

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

Orientador: Prof. Ms. Jonas Bodê

1. Banco de dados 2. Engenharia de software 3. Gestão do conhecimento. I. SILVA, Andressa Evelyn de Sa, II. UYAMA, Victor Hugo Seiji III. BODÊ, Jonas IV. Centro Estadual de Educação Tecnológica Paula Souza – Faculdade de Tecnologia de Americana Ministro Ralph Biasi

CDU: 681.3.07

681.3.05

316.77

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

Victor Hugo Seiji Ueyama
Andressa Evelyn de Sá Silva

Estudo 360

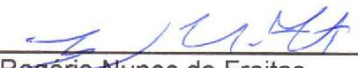
Trabalho de graduação apresentado como exigência parcial para obtenção do título de Tecnólogo em Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas pelo Centro Paula Souza – FATEC Faculdade de Tecnologia de Americana Ministro Ralph Biasi.
Área de concentração: Análise e Desenvolvimento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

Americana, 2 de dezembro de 2024.

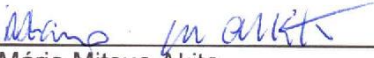
Banca Examinadora:



Jonas Bodê
Mestre
Fatec Americana "Ministro Ralph Biasi"



Rogério Nunes de Freitas
Mestre
Fatec Americana "Ministro Ralph Biasi"



Mário Mitsuo Akita
Especialista
Fatec Americana "Ministro Ralph Biasi"

RESUMO

O presente trabalho descreve o desenvolvimento do aplicativo móvel Estudo360, concebido para auxiliar estudantes universitários no gerenciamento eficiente do tempo e na organização das atividades acadêmicas. O aplicativo integra técnicas comprovadas, como o Método Pomodoro, e elementos de gamificação para aumentar a produtividade e o engajamento dos usuários. A metodologia adotada para a elaboração do Estudo360 baseou-se em práticas ágeis, especificamente na combinação das metodologias Scrum e Kanban, que permitiram uma gestão flexível e eficiente do ciclo de desenvolvimento, facilitando ajustes contínuos e entregas incrementais. Inicialmente, foi conduzida uma pesquisa exploratória para identificar aplicativos similares no mercado, como Forest, Habitica, MyStudyLife e Focus To-Do, com o objetivo de mapear funcionalidades essenciais e identificar oportunidades de melhoria. A análise das necessidades dos usuários foi realizada por meio de entrevistas e questionários, focando em estudantes universitários e suas demandas específicas relacionadas ao gerenciamento de tempo e organização acadêmica. Com base nos dados coletados, foram definidos os requisitos funcionais e não funcionais do aplicativo, utilizando técnicas de engenharia de requisitos para garantir a clareza e completude das funcionalidades propostas. A modelagem do sistema foi realizada utilizando a linguagem UML (Unified Modeling Language), facilitando a visualização da estrutura do aplicativo e a interação entre seus componentes. A escolha do Firebase como banco de dados não relacional permitiu uma modelagem de dados flexível e escalável, essencial para suportar as diversas funcionalidades do Estudo360, como gerenciamento de tarefas, sessões de Pomodoro, sistema de pontuação e notificações em tempo real. A etapa de prototipagem, realizada com a ferramenta Figma, foi fundamental para validar o design e a usabilidade do aplicativo, garantindo uma interface intuitiva e agradável para os usuários. Após a validação dos protótipos, iniciou-se o desenvolvimento das funcionalidades principais, seguido por uma fase de testes rigorosos que incluíram testes unitários, de integração e de usabilidade, assegurando a qualidade e a performance do aplicativo. Os resultados alcançados incluem a implementação bem-sucedida das funcionalidades previstas, proporcionando aos estudantes uma ferramenta eficaz para a gestão de suas atividades acadêmicas e pessoais. A conclusão do trabalho evidencia que o Estudo360 atende às necessidades dos usuários, contribuindo significativamente para a melhoria da produtividade e organização acadêmica dos estudantes universitários.

Palavras Chave: Gerenciamento Acadêmico; Engenharia de Software; Banco de Dados Não Relacional.

ABSTRACT

This study describes the development of the mobile application Estudio360, designed to assist university students in efficiently managing their time and organizing academic activities. The application integrates proven techniques, such as the Pomodoro Method, and gamification elements to enhance user productivity and engagement. The methodology adopted for the creation of Estudio360 was based on agile practices, specifically the combination of Scrum and Kanban methodologies, which allowed for flexible and efficient management of the development cycle, facilitating continuous adjustments and incremental deliveries. Initially, an exploratory research was conducted to identify similar applications available in the market, such as Forest, Habitica, MyStudyLife, and Focus To-Do, aiming to map essential functionalities and identify opportunities for improvement. User needs analysis was performed through interviews and questionnaires, focusing on university students and their specific demands related to time management and academic organization. Based on the collected data, both functional and non-functional requirements of the application were defined using requirements engineering techniques to ensure clarity and completeness of the proposed functionalities. System modeling was carried out using Unified Modeling Language (UML), facilitating the visualization of the application's structure and the interaction between its components. The choice of Firebase as a non-relational database enabled flexible and scalable data modeling, essential to support the diverse functionalities of Estudio360, such as task management, Pomodoro sessions, scoring system, and real-time notifications. The prototyping phase, conducted with the Figma tool, was fundamental in validating the application's design and usability, ensuring an intuitive and user-friendly interface. After prototype validation, the development of the main functionalities began, followed by a rigorous testing phase that included unit tests, integration tests, and usability tests, ensuring the application's quality and performance. The achieved results include the successful implementation of the planned functionalities, providing students with an effective tool for managing their academic and personal activities. The conclusion of the study demonstrates that Estudio360 meets user needs, significantly contributing to the improvement of productivity and academic organization among university students.

Keywords: Academic Management; Software Engineering; Non-Relational Database.

LISTA DE FIGURAS

Figura 1 – Diagrama de caso de uso geral do aplicativo.....	44
Figura 2 - Diagrama de caso de uso de autenticação e gestão de contas.	45
Figura 3 – Diagrama de caso de uso da Home.	48
Figura 4 - Diagrama de caso de uso de gerenciamento de agenda.....	51
Figura 5 - Diagrama de caso de uso do calendário acadêmico.....	54
Figura 6 - Diagrama de caso de uso de personalização de horário da aula.	56
Figura 7 - Diagrama de caso de uso da função Notas e Faltas.	58
Figura 8 - Diagrama de caso de uso do Pomodoro.....	61
Figura 9 - Diagrama de caso de uso da Pontuação e Recompensa.	64
Figura 10 - Diagrama de caso de uso da função Ajuda.	67
Figura 11 - Diagrama de Classes do Login.	71
Figura 12 - Diagrama de Classes da Home.	72
Figura 13 - Diagrama de Classes da Agenda.....	74
Figura 14 - Diagrama de Classes do Calendário Acadêmico.	75
Figura 15 - Diagrama de Classes do Horário de Aula.	76
Figura 16 - Diagrama de Classes da função Notas e Faltas.	77
Figura 17 - Diagrama de Classes da função Pomodoro.....	78
Figura 18 - Diagrama de Classes da função Pontuação e Recompensas.	79
Figura 19 - Diagrama de Classes da função Ajuda.	80
Figura 20 - Diagrama de fluxo de tela do Aplicativo Estudo360.	106
Figura 21 - Captura das telas de Login, Redefinição de Senha e Cadastro.....	108
Figura 22 - Captura da tela de Cadastro e de Termos.	109
Figura 23 - Captura das telas da Home, Notificação e Configuração.....	111
Figura 24 - Captura das telas de Ajuda, Tutorial e Dúvidas.	113
Figura 25 - Captura da tela de Menu.....	115
Figura 26 - Captura das telas da função Agenda.	117
Figura 27 - Captura das telas da função Calendário Acadêmico.	119
Figura 28 - Captura das telas da função Horário das Aulas.....	121
Figura 29 - Captura das telas da função Notas e Faltas e do Gráfico.....	123
Figura 30 - Captura das telas de Notas e Faltas.	124
Figura 31 - Captura das telas da função Pomodoro.....	126
Figura 32 - Captura das telas da função de Início do Pomodoro.	127

Figura 33 - Captura das telas das Mensagens exibidas no Pomodoro.	129
Figura 34 - Captura das telas do Timer e da mensagem exibida.	130
Figura 35 - Captura das telas do Cronômetro.	131
Figura 36 - Captura das telas da função Pontuação e Recompensas.	132

LISTA DE TABELAS

Tabela 1 - Comparativo de funcionalidades entre aplicativos.	25
Tabela 2 - Requisitos funcionais da função Login.	28
Tabela 3 - Requisitos funcionais da função Home.	29
Tabela 4 - Requisitos funcionais da função Agenda.	30
Tabela 5 - Requisitos funcionais da função Calendário Acadêmico.	31
Tabela 6 - Requisitos funcionais da função Horário de aula.	31
Tabela 7 - Requisitos funcionais da função Notas e Faltas.	32
Tabela 8 - Requisitos funcionais da função Pomodoro.	33
Tabela 9 - Requisitos funcionais da função Pontuação.	34
Tabela 10 - Requisitos funcionais da função Ajuda.	35
Tabela 11 - Requisitos Não Funcionais do Projeto.	37
Tabela 12 - Caso de Uso CA001: Autenticação e Gestão de Contas.	45
Tabela 13 - Caso de Uso CA002: Tela Inicial (Home).	48
Tabela 14 - Caso de Uso CA003: Gerenciamento de Agenda.	52
Tabela 15 - Caso de Uso CA004: Calendário Acadêmico.	54
Tabela 16 - Caso de Uso CA005: Horário de Aula.	56
Tabela 17 - Caso de Uso CA006: Gerenciamento de Notas e Faltas.	58
Tabela 18 - Caso de Uso CA007: Utilização do Temporizador Pomodoro.	61
Tabela 19 - Caso de Uso CA008: Sistema de Pontuação e Recompensas.	64
Tabela 20 - Caso de Uso CA009: Funcionalidade de Ajuda.	67
Tabela 21 - Dicionário de Dados - Calendário Acadêmico.	89
Tabela 22 - Dicionário de Dados - Horário das Aulas.	90
Tabela 23 - Dicionário de Dados - Informações Pessoais e Acadêmicas.	90
Tabela 24 - Dicionário de Dados - Configurações.	91
Tabela 25 - Dicionário de Dados - Eventos (events).	92
Tabela 26 - Dicionário de Dados - Materiais (materias).	92
Tabela 27 - Dicionário de Dados - Notas (notas).	93
Tabela 28 - Dicionário de Dados - Nível e Pontuação (level).	93
Tabela 29 - Dicionário de Dados - Histórico (historico).	93
Tabela 30 - Dicionário de Dados - Pomodoro Ativo (pomodoro).	94
Tabela 31 - Dicionário de Dados - Notificações (notificacoes).	95
Tabela 32 - Dicionário de Dados - Configurações Gerais (configuracao).	95

Tabela 33 - Plano de Sprints do Projeto.....98

LISTA DE ABREVIATURAS E SIGLAS

ABNT	Associação Brasileira de Normas Técnicas
API	Application Programming Interface
BR	Brazil
CA	Caso de Uso
CPU	Central Processing Unit
DB	Database
ER	Entity-Relationship
FAQ	Frequently Asked Questions
IDE	Integrated Development Environment
ID	Identifier
LGPD	Lei Geral de Proteção de Dados
NoSQL	Not Only SQL
PO	Product Owner
RA	Registro Acadêmico
RPG	Role-Playing Game
RF	Requisito Funcional
RNF	Requisito Não Funcional
SDK	Software Development Kit
SM	Scrum Master
SQL	Structured Query Language
TCC	Trabalho de Conclusão de Curso
UI	User Interface
UX	User Experience
UML	Unified Modeling Language
XP	Extreme Programming

SUMÁRIO

1 INTRODUÇÃO	14
1.1 Contextualização do Problema	14
1.2 Objetivos	15
1.3 Justificativa	16
1.4 Estrutura do Trabalho	17
2 METODOLOGIA	18
2.1 Natureza da Pesquisa	18
2.2 Metodologia de Desenvolvimento	18
2.3 Planejamento e Pesquisa Inicial	19
2.4 Definição de Requisitos	19
2.5 Modelagem do Sistema	20
2.6 Prototipagem	21
2.7 Desenvolvimento	21
3 PROJETO DO SISTEMA	24
3.1 Softwares Similares	24
3.2 Levantamento de Requisitos	26
3.3 Recursos e Ferramentas	40
4 MODELAGEM.....	42
4.1 Casos de uso	42
4.2 Diagrama de Classes	69
5 BANCO DE DADOS	81
5.1 Bancos de Dados Não Relacionais	81
5.2 Representação da Estrutura de Dados no Firebase	82
5.3 Dicionário de Dados	88
6 DESENVOLVIMENTO	96
6.1 Metodologias Ágeis	96
7 PROTÓTIPO.....	101
7.1 Design da Interface	101
7.2 Interfaces de Usuário	106

8 CONSIDERAÇÕES FINAIS.....	133
9 REFERÊNCIAS BIBLIOGRÁFICAS.....	135

1 INTRODUÇÃO

1.1 Contextualização do Problema

O ambiente universitário é repleto de desafios que exigem dos estudantes habilidades de organização, disciplina e gerenciamento eficaz do tempo. Esses são fatores muito importantes para lidar com a vasta quantidade de atividades acadêmicas, como leituras, pesquisas, trabalhos em grupo e provas, além de outras responsabilidades extracurriculares e pessoais. No entanto, muitos estudantes enfrentam dificuldades em equilibrar suas demandas acadêmicas com a vida pessoal, o que frequentemente resulta em procrastinação, ansiedade e sobrecarga de trabalho.

Diversos estudos apontam para a importância de uma boa gestão do tempo para o sucesso acadêmico. Segundo Britton e Tesser (1991) destacam que estudantes que utilizam práticas eficientes de gerenciamento do tempo tendem a apresentar melhor desempenho acadêmico e menores níveis de estresse. Por outro lado, a ausência de uma estrutura organizada para lidar com múltiplas tarefas aumenta a probabilidade de procrastinação e impacta negativamente o desempenho.

Ao longo do tempo, o uso de tecnologias digitais para auxiliar no gerenciamento de atividades tem se tornado uma solução cada vez mais popular entre os estudantes. Ferramentas como aplicativos móveis voltados para a organização e o planejamento permitem que os usuários planejem suas tarefas, monitorem seu progresso e mantenham um controle mais eficiente de seus compromissos. É nesse contexto que o “Estudo360” surge como um recurso tecnológico voltada para estudantes universitários, proporcionando funcionalidades que otimizam a gestão do tempo e a produtividade acadêmica por meio do Método Pomodoro, da gamificação e de outras ferramentas essenciais, como o calendário acadêmico e o horário das aulas.

Além disso, em um cenário acadêmico marcado por um volume crescente de informações e demandas, a falta de organização adequada pode gerar altos níveis de estresse e prejudicar o desempenho acadêmico. Segundo Schraw, Crippen e Hartley (2007) identificam que uma das maiores fontes de ansiedade entre os estudantes é a falta de organização e o acúmulo de tarefas, o que reforça a necessidade de soluções tecnológicas que auxiliem os alunos a gerenciar suas atividades de forma eficiente e eficaz.

1.2 Objetivos

1.2.1 Objetivo Geral

Desenvolver o aplicativo móvel Estudo360 para auxiliar estudantes universitários no gerenciamento eficiente do tempo e na organização das atividades acadêmicas, promovendo aumento de produtividade e redução do estresse.

1.2.2 Objetivos Específicos

Os objetivos específicos deste projeto incluem melhorar o gerenciamento de tempo, disponibilizando uma agenda digital que facilite o planejamento das atividades acadêmicas e permita que os estudantes visualizem suas tarefas e compromissos de forma organizada e clara.

Além disso, busca-se aumentar a produtividade acadêmica através da integração do Método Pomodoro, incentivando os usuários a dividir suas tarefas em intervalos de tempo focados e a realizar pausas regulares, promovendo maior concentração e evitando a fadiga mental.

Outro objetivo é incentivar o engajamento e a motivação dos estudantes, utilizando elementos de gamificação para mantê-los engajados e motivados a completar suas tarefas por meio de um sistema de pontuação e recompensas.

Também pretende-se reduzir o estresse relacionado às atividades acadêmicas, fornecendo ferramentas que promovam a organização e o planejamento, contribuindo para a diminuição do estresse acadêmico e proporcionando uma sensação de controle sobre as atividades diárias.

Adicionalmente, visa-se facilitar o acompanhamento dos horários das aulas e a localização das salas, oferecendo recursos que permitam aos estudantes visualizar claramente seu cronograma acadêmico e a localização das salas de aula.

Outro objetivo é integrar o calendário acadêmico institucional, incorporando o calendário da instituição para que os estudantes possam acompanhar feriados, datas importantes e prazos institucionais de maneira prática e eficiente.

Por fim, pretende-se disponibilizar uma agenda para anotações pessoais, oferecendo um espaço dedicado para que os estudantes possam fazer anotações e registrar informações importantes para suas atividades acadêmicas.

1.3 Justificativa

A justificativa para o desenvolvimento do Estudo360 reside na crescente necessidade de estudantes universitários por ferramentas que os ajudem a organizar e gerenciar suas atividades acadêmicas. Com o aumento das exigências acadêmicas e a complexidade da vida universitária, os estudantes frequentemente enfrentam dificuldades para equilibrar suas responsabilidades, o que pode resultar em estresse, procrastinação e menor desempenho acadêmico.

O uso do Método Pomodoro no Estudo360 proporciona uma solução prática para o gerenciamento do tempo, permitindo que os estudantes alternem entre períodos de estudo focado e pausas regulares, maximizando a eficiência. Segundo Cirillo (2006), criador do método, argumenta que essa técnica, ao estruturar o tempo de estudo em blocos curtos, ajuda a manter o foco e a evitar a fadiga mental. Os estudos como os de Baumeister e Tierney (2011) e Van den Eynde, Hine e Hargreaves (2019) reforçam que pausas regulares são importantes para manter a energia mental ao longo do dia e que estudantes que utilizam o método sentem-se mais no controle de seu tempo e relatam menos estresse.

A gamificação, por sua vez, oferece uma abordagem inovadora para motivar os estudantes a concluir suas tarefas de forma eficiente. Segundo Kapp (2012), a gamificação é uma estratégia eficaz para aumentar o engajamento em ambientes educacionais, transformando atividades acadêmicas em desafios recompensadores. Além disso, estudos como os de Hamari, Koivisto e Sarsa (2014) confirmam que a gamificação não apenas aumenta a motivação, mas também pode levar a uma melhoria no desempenho acadêmico, ao criar um ambiente mais interativo e envolvente. Ao implementar sistemas de pontuação e recompensas, o Estudo360 motiva os estudantes a se manterem focados e a completarem suas tarefas de forma mais eficaz.

Por fim, a agenda digital integrada no Estudo360, juntamente com funcionalidades como o calendário acadêmico, o horário das aulas e a localização das salas, é uma ferramenta que auxilia o planejamento acadêmico. As ferramentas de planejamento digital, como observam Gagne et al. (2005), ajudam muito na administração do tempo e no cumprimento dos prazos. O uso de uma agenda ajuda os estudantes a organizar suas responsabilidades de forma prática, minimizando o estresse e a ansiedade, enquanto mantém o foco no cumprimento das tarefas diárias.

De acordo com Zimmerman (2002), a capacidade de tomar notas e organizar pensamentos é uma prática fundamental para a aprendizagem autorregulada, e a inclusão de uma agenda para anotações no Estudo360 permite que os alunos organizem suas ideias, lembretes e informações de maneira prática e acessível.

1.4 Estrutura do Trabalho

Este trabalho está organizado em oito capítulos. O Capítulo 2, Metodologia, apresenta os fundamentos teóricos e metodológicos utilizados no desenvolvimento do projeto, incluindo técnicas de gerenciamento do tempo e a escolha das metodologias ágeis, com destaque para o Scrum.

O Capítulo 3, Projeto do Sistema, detalha os requisitos funcionais e não funcionais do sistema, além dos recursos e ferramentas utilizadas no desenvolvimento do aplicativo Estudo360.

No Capítulo 4, Modelagem do Sistema, são descritos os artefatos de modelagem, como diagramas UML, casos de uso e diagramas de classes, que representam a estrutura e o comportamento do aplicativo, facilitando a compreensão de sua arquitetura.

O Capítulo 5, Banco de Dados, aborda a estrutura do banco de dados utilizada no aplicativo, destacando a escolha do Firebase como solução NoSQL, a organização dos dados em uma estrutura hierárquica e o dicionário de dados detalhado.

Em seguida, o Capítulo 6, Desenvolvimento, descreve o processo de desenvolvimento do aplicativo, com ênfase nas etapas realizadas, na implementação das funcionalidades principais, no planejamento dos sprints segundo o Scrum e nos desafios enfrentados ao longo do projeto.

O Capítulo 7, Protótipo, apresenta o protótipo do aplicativo, incluindo capturas de tela e descrições detalhadas das interfaces de usuário, com destaque para as escolhas de design, navegação, layout, padrões de design e elementos visuais utilizados.

Por fim, o Capítulo 8, Considerações Finais, conclui o trabalho apresentando as contribuições do projeto, as limitações encontradas, a avaliação dos resultados obtidos e sugestões para trabalhos futuros, enfatizando o impacto do Estudo360 no apoio ao gerenciamento acadêmico dos estudantes.

2 METODOLOGIA

Este capítulo descreve e detalha a metodologia utilizada para a elaboração e desenvolvimento deste trabalho.

2.1 Natureza da Pesquisa

A natureza da pesquisa realizada neste trabalho é de caráter aplicada. A pesquisa aplicada tem como objetivo solucionar problemas práticos utilizando conhecimentos teóricos existentes (GIL, 2017). Essa abordagem foi escolhida para desenvolver um aplicativo funcional que atenda às necessidades específicas de estudantes universitários no gerenciamento de tempo e organização acadêmica.

2.2 Metodologia de Desenvolvimento

Para o desenvolvimento do aplicativo Estudo360, foi adotada uma abordagem metodológica centrada em práticas ágeis, especificamente a metodologia Scrum. O Scrum é um framework que promove a gestão e o planejamento de projetos de software de forma incremental e iterativa, permitindo flexibilidade e adaptação às mudanças (SCHWABER; SUTHERLAND, 2017). A opção pelo Scrum visa garantir benefícios como:

- **Flexibilidade:** Capacidade de responder rapidamente a mudanças nos requisitos ou no ambiente de desenvolvimento.
- **Eficiência:** Entregas frequentes de incrementos funcionais do produto, aumentando a produtividade.
- **Colaboração:** Incentivo ao trabalho em equipe e comunicação contínua entre os membros.

As características descritas acima demonstram como o Scrum atende à necessidade de adaptação e agilidade no contexto do desenvolvimento do Estudo360.

2.3 Planejamento e Pesquisa Inicial

Para embasar o desenvolvimento do Estudo360, foi realizada uma pesquisa exploratória sobre aplicativos similares disponíveis no mercado, como *Forest*, *Habitica*, *MyStudyLife* e *Focus To-Do*. Essa análise comparativa teve como objetivo identificar funcionalidades essenciais, pontos de melhoria e lacunas que poderiam ser preenchidas pelo Estudo360.

Além disso, foi conduzida uma investigação focada nas necessidades do público-alvo principal: estudantes universitários. Essa etapa incluiu duas abordagens principais:

1. Entrevistas e questionários: Foram coletados dados diretamente dos estudantes para compreender as principais dificuldades enfrentadas no gerenciamento do tempo e na organização acadêmica.
2. Estudo de hábitos de estudo: Foram analisadas técnicas amplamente utilizadas, como o Método Pomodoro, e estratégias de motivação para identificar práticas que poderiam ser incorporadas ao aplicativo.

Essa fase inicial de planejamento e pesquisa foi essencial para alinhar as funcionalidades do Estudo360 às reais demandas dos usuários, garantindo que o aplicativo atendesse às suas necessidades de forma prática e eficiente.

2.4 Definição de Requisitos

A Engenharia de Requisitos foi utilizada para descrever e listar os serviços do aplicativo de maneira concisa e objetiva. De acordo com Sommerville (2011), a engenharia de requisitos é essencial para entender o que deve ser construído e garantir que o sistema atenda às expectativas dos usuários. As atividades realizadas nesta etapa incluem:

- Elicitação de Requisitos: Identificação das funcionalidades desejadas pelos usuários e stakeholders.
- Análise de Requisitos: Avaliação da viabilidade técnica e alinhamento com os objetivos do projeto.
- Especificação de Requisitos: Documentação detalhada dos requisitos funcionais e não funcionais.

- Validação de Requisitos: Revisão dos requisitos com os usuários para assegurar sua completude e correção.

Os requisitos foram categorizados em:

- Requisitos Funcionais: Descrevem as funcionalidades que o sistema deve oferecer, como gestão de tarefas, integração do Método Pomodoro, sistema de gamificação, entre outros.
- Requisitos Não Funcionais: Referem-se às restrições e qualidades do sistema, incluindo desempenho, segurança, usabilidade e portabilidade.

2.5 Modelagem do Sistema

A modelagem do sistema foi realizada utilizando a Unified Modeling Language (UML), que permitiu a criação de diagramas para facilitar a compreensão e visualização da estrutura do sistema e de seus componentes (BOOCH; RUMBAUGH; JACOBSON, 2005). Entre os diagramas desenvolvidos, destacam-se os casos de uso, as classes e a modelagem do banco de dados.

Os diagramas de casos de uso foram elaborados para representar as interações entre os usuários, chamados de atores, e o sistema. Esses diagramas destacam as principais funcionalidades do aplicativo, oferecendo uma visão clara das ações esperadas e dos requisitos que o sistema deve atender.

Já os diagramas de classes descrevem a estrutura estática do sistema, detalhando as classes, seus atributos, métodos e os relacionamentos entre elas. Essa modelagem foi fundamental para o design orientado a objetos e serviu como base para a implementação do código-fonte, garantindo maior organização e flexibilidade no desenvolvimento.

A modelagem do banco de dados foi realizada utilizando o Firebase Realtime Database, uma solução NoSQL em tempo real fornecida pela Google. Essa escolha se deu por diversas razões, incluindo a sincronização em tempo real, que garante atualizações instantâneas entre os usuários conectados; a facilidade de integração, com APIs amigáveis para aplicativos móveis; a escalabilidade, permitindo suportar um grande número de usuários simultaneamente; e os serviços adicionais oferecidos, como Firebase Authentication, Storage e Analytics. Os dados armazenados no banco incluem informações de usuários, tarefas, sessões de Pomodoro, pontuações e

recompensas, estruturados para garantir eficiência e uma experiência de usuário otimizada.

2.6 Prototipagem

A etapa de prototipagem teve um papel determinante na validação do design e usabilidade da aplicação móvel. Segundo Coutinho (2006, p. 16), a prototipagem permite identificar problemas de usabilidade e realizar ajustes antes da fase de desenvolvimento, tornando o processo mais eficiente.

Para essa etapa, foram utilizadas ferramentas e técnicas específicas. O Figma foi empregado para criar protótipos de alta fidelidade das interfaces de usuário, possibilitando a simulação da navegação e interação com o aplicativo. Além disso, testes de usabilidade foram realizados com usuários representativos, com o objetivo de coletar feedback valioso sobre o design e as funcionalidades propostas.

O processo de prototipagem incluiu diversas etapas bem definidas. Inicialmente, foi feita a definição dos fluxos de usuário, mapeando as jornadas que os usuários percorrem ao utilizar o aplicativo. Em seguida, foram criados wireframes, que serviram como esboços iniciais das telas, estruturando o layout e os elementos principais. Após essa etapa, foram desenvolvidos protótipos detalhados, incluindo o design das telas com cores, tipografia e elementos visuais consistentes. Por fim, esses protótipos foram apresentados a um grupo de usuários para validação, permitindo identificar melhorias necessárias.

As informações coletadas durante o processo de prototipagem influenciaram diretamente as decisões relacionadas ao design e à priorização das funcionalidades. Esse trabalho garantiu que o aplicativo fosse desenvolvido para atender às necessidades dos usuários de forma clara e intuitiva, contribuindo para uma experiência mais satisfatória.

2.7 Desenvolvimento

Após a validação dos protótipos, iniciou-se a fase de desenvolvimento do aplicativo, seguindo as práticas ágeis do Scrum. O projeto foi estruturado em sprints, cada uma com duração aproximada de quatro semanas, com atividades bem definidas para cada etapa. O planejamento das sprints incluiu a definição das tarefas a serem

realizadas, estimativas de tempo e os objetivos a serem alcançados. Durante o desenvolvimento, foram realizadas reuniões semanais breves, que permitiram acompanhar o progresso do projeto e identificar possíveis impedimentos.

A codificação das funcionalidades planejadas foi realizada ao longo de cada sprint, culminando com a revisão e demonstração das funcionalidades implementadas aos stakeholders ao final de cada ciclo. Além disso, foram realizadas retrospectivas ao término de cada sprint, promovendo reflexões sobre o processo e identificando melhorias para as etapas subsequentes.

O desenvolvimento utilizou diversas tecnologias e ferramentas. A linguagem de programação Kotlin foi empregada para o desenvolvimento nativo na plataforma Android, utilizando o Android Studio como ambiente de desenvolvimento integrado. Algumas bibliotecas e frameworks desempenharam papéis importantes nesse processo, incluindo o Firebase SDK, responsável por funcionalidades como autenticação, banco de dados em tempo real e armazenamento, e o Navigation Component, utilizado para o gerenciamento de navegação entre telas. Outras ferramentas relevantes foram o TapTargetView, para criação de tutoriais interativos; o RecyclerView, que permitiu a exibição eficiente de listas; e o ConstraintLayout, utilizado para criar layouts responsivos.

2.8 Testes e Validação

A qualidade do aplicativo foi assegurada por meio de uma abordagem abrangente de testes. Foram realizados testes unitários para verificar a lógica das unidades individuais de código, bem como testes de integração para garantir que os módulos funcionassem corretamente quando combinados. Além disso, testes funcionais foram conduzidos para validar que as funcionalidades atendiam aos requisitos especificados.

Testes de usabilidade foram aplicados para avaliar a facilidade de uso e a experiência geral dos usuários, permitindo identificar pontos de melhoria na interface e na navegação do aplicativo. Por fim, o aplicativo foi submetido a testes beta, distribuído para um grupo seletivo de usuários, a fim de identificar possíveis falhas e coletar feedback antes do lançamento oficial. Os resultados obtidos nesses testes foram fundamentais para corrigir falhas e otimizar o desempenho do sistema.

2.9 Documentação

Durante todas as etapas do projeto, foi mantida uma documentação detalhada com o objetivo de registrar informações essenciais e garantir a organização do desenvolvimento. Essa documentação incluiu a criação de documentos de requisitos funcionais e não funcionais, além de diagramas UML, como casos de uso e diagramas de classes, para representar a estrutura e o funcionamento do sistema.

Também foi elaborado um manual do usuário, contendo orientações detalhadas para a utilização do aplicativo. Todo o material produzido seguiu as normas da ABNT (Associação Brasileira de Normas Técnicas), assegurando a padronização e a facilidade de compreensão.

3 PROJETO DO SISTEMA

Este capítulo detalha o processo de desenvolvimento do aplicativo de rotina acadêmica, descrevendo cada etapa envolvida, desde o levantamento de requisitos até a criação das interfaces de usuário no Figma. O desenvolvimento foi conduzido utilizando a metodologia ágil Scrum, que proporcionou uma estrutura flexível e iterativa para a execução do projeto.

A escolha da metodologia Scrum foi baseada na necessidade de um processo dinâmico e colaborativo que permitisse a adaptação rápida às mudanças e garantisse a entrega contínua de valor. Segundo Schwaber e Sutherland (2017), "o Scrum é uma estrutura que permite às equipes abordarem problemas complexos adaptativos, ao mesmo tempo que entregam produtos de valor mais elevado de forma produtiva e criativa".

3.1 Softwares Similares

Atualmente, uma variedade de aplicativos voltados para a organização da rotina acadêmica está disponível no mercado, cada um oferecendo uma ampla diversidade de funcionalidades. Dentre os diversos aplicativos disponíveis, destacam-se quatro que foram selecionados para análise neste trabalho devido à sua popularidade e ampla aceitação pelos usuários: Forest, Habitica, MyStudyLife e Focus To-Do:

- Forest: é um aplicativo de foco que utiliza a técnica Pomodoro para ajudar os usuários a se concentrarem em suas tarefas. Ao iniciar uma sessão de foco, uma árvore virtual começa a crescer no aplicativo. Se o usuário conseguir manter o foco sem usar o telefone, a árvore prospera; caso contrário, ela morre. O aplicativo também incentiva a plantação de árvores reais por meio de parcerias com organizações de reflorestamento. As principais funcionalidades incluem a integração da técnica Pomodoro, gamificação por meio do crescimento de árvores virtuais, recompensas para incentivar a manutenção do foco e a geração de relatórios de produtividade.
- Habitica: é um aplicativo de produtividade que transforma tarefas diárias e hábitos em um jogo de RPG. Os usuários criam avatares, completam tarefas para ganhar pontos de experiência e recompensas dentro do jogo. A

gamificação visa aumentar a motivação e o engajamento, tornando o gerenciamento de tempo e tarefas mais divertidos. As principais funcionalidades incluem a transformação de tarefas em missões no estilo RPG, um sistema de recompensas e penalidades, criação e gerenciamento de hábitos, tarefas diárias e únicas, socialização por meio de grupos e guildas, além da personalização de avatares e equipamentos.

- **MyStudyLife:** é um aplicativo de gerenciamento acadêmico projetado especificamente para estudantes. Ele permite que os usuários organizem seus horários de aulas, tarefas, exames e prazos em um calendário unificado. O aplicativo sincroniza dados entre dispositivos e oferece notificações para ajudar os estudantes a manterem-se organizados e pontuais. As principais funcionalidades incluem um calendário acadêmico integrado, gerenciamento de tarefas e prazos, sincronização de dados entre dispositivos, notificações e lembretes personalizados, além de uma interface intuitiva e fácil de usar.
- **Focus To-Do:** combina gerenciamento de tarefas com a técnica Pomodoro, proporcionando uma abordagem estruturada para aumentar a produtividade. O aplicativo permite que os usuários criem listas de tarefas, definam prioridades e utilizem sessões Pomodoro para focar em cada atividade. Além disso, oferece gráficos e relatórios para monitorar o progresso. Este aplicativo destaca-se por integrar a técnica Pomodoro ao gerenciamento de tarefas, permitindo a definição de prioridades. Ele oferece relatórios e gráficos de produtividade, sincronização entre dispositivos, lembretes e notificações personalizadas, além de integração com calendários externos.

Considerando esses aspectos, foi elaborada a Tabela 1 destacando as principais diferenças do Estudo360 em relação aos aplicativos mencionados anteriormente.

Tabela 1 - Comparativo de funcionalidades entre aplicativos.

Funcionalidades	Forest	Habitica	My Study Life	Focus To-Do	Estudo360
Login	X	X	X	X	X
Home	X	X	X	X	X
Agenda	-	X	X	X	X

Calendário Acadêmico	-	-	X	-	X
Horário da Aulas	-	-	X	-	X
Notas e Faltas	-	-	X	-	X
Pomodoro	X	-	-	X	X
Pontuação	X	X	-	-	X
Ajuda	X	X	X	X	X

Fonte: Elaborado pelos autores (2024)

3.2 Levantamento de Requisitos

O levantamento de requisitos é uma etapa crucial no processo de desenvolvimento de software, sendo responsável por descobrir, analisar e documentar as necessidades dos stakeholders para um sistema. Conforme descrito por Sommerville (2011, p. 53), "O levantamento de requisitos é o processo de descobrir, analisar e documentar as necessidades dos stakeholders para um sistema". Esse processo visa entender o que o sistema deve fazer e quais são as expectativas dos stakeholders, como apontado por Kroll & Kruchten (2003, p. 27). Importância do Levantamento de Requisitos:

O sucesso de um projeto de software muitas vezes depende da qualidade do levantamento de requisitos. Requisitos mal levantados podem levar a erros de desenvolvimento, retrabalho, atrasos no projeto e aumento de custos. Como observam Robertson & Robertson (2009, p. 45), "O levantamento de requisitos é uma etapa fundamental para o sucesso de um projeto de software. Requisitos mal levantados podem levar a erros de desenvolvimento e retrabalho, atrasando o projeto e aumentando os custos". Além disso, requisitos bem elaborados servem como base para o planejamento, desenvolvimento e teste do sistema, conforme enfatizado por Sommerville & Sawyer (2009, p. 57).

Etapas do Levantamento de Requisitos

O processo de levantamento de requisitos envolve diversas etapas que variam conforme os modelos propostos. Goguen e Linde (1993, p. 117-125) destacam etapas como a identificação dos stakeholders, a elicitacão de requisitos, a análise de requisitos e a documentação de requisitos. Já Huse e Sommerville (2013, p. 83-104) apresentam um modelo mais detalhado, composto por seis etapas: identificação dos

stakeholders, compreensão das necessidades, definição dos requisitos, priorização dos requisitos, validação dos requisitos e gerenciamento dos requisitos.

Técnicas para Elicitação de Requisitos

A elicitação de requisitos pode ser realizada por meio de diversas técnicas amplamente descritas na literatura. Pressman (2010, p. 189-208) apresenta métodos como entrevistas, questionários, observação e análise de documentos. Complementando essa abordagem, Kotonya e Sommerville (1998, p. 113-144) oferecem um guia prático para a escolha e aplicação dessas técnicas, adaptando-as ao contexto do projeto.

Ferramentas para Levantamento de Requisitos

O uso de ferramentas específicas pode facilitar o processo de levantamento de requisitos, auxiliando na elicitação, análise, documentação e rastreabilidade. Entre as soluções disponíveis no mercado, destacam-se as ferramentas oferecidas pela Visure Solutions, como o Visure Requirements, Visure Requirements Lite e Visure Requirements for Agile. Além disso, o IBM Rational RequisitePro é amplamente utilizado por profissionais da área para gerenciar todas as fases do levantamento de requisitos.

Em resumo, o levantamento de requisitos é uma etapa chave no desenvolvimento de software, garantindo que as necessidades dos stakeholders sejam entendidas e documentadas de forma precisa e clara. O uso de técnicas adequadas de elicitação e ferramentas de apoio pode contribuir significativamente para o sucesso do projeto.

3.2.1 Requisitos Funcionais

Os requisitos funcionais são elementos críticos no desenvolvimento de sistemas de software, desempenhando um papel fundamental na definição das funcionalidades e operações que o sistema deve executar. Conforme destacado por Sommerville (2011, p. 53), "requisitos funcionais descrevem as funções que um sistema deve executar, definindo as funcionalidades do sistema e as restrições de como essas funcionalidades devem ser implementadas".

Importância dos Requisitos Funcionais:

Garantir que o sistema atenda às necessidades dos stakeholders:

Os requisitos funcionais garantem que o sistema seja desenvolvido em conformidade com as expectativas e necessidades dos stakeholders. Como afirmam Kroll e Kruchten (2003, p. 28), "os requisitos funcionais são essenciais para garantir que o sistema atenda às necessidades dos stakeholders, pois definem o que o sistema deve fazer e como ele deve se comportar".

Fornecer uma base para o planejamento e desenvolvimento do sistema:

Os requisitos funcionais fornecem uma base sólida para o planejamento e desenvolvimento do sistema. Robertson e Robertson (2009, p. 46) ressaltam que "a partir dos requisitos funcionais, é possível definir o escopo do projeto, estimar o tempo e o custo do desenvolvimento e planejar as atividades de desenvolvimento".

Servir como um contrato entre o cliente e a equipe de desenvolvimento:

Os requisitos funcionais documentam as expectativas do cliente em relação ao sistema e estabelecem um contrato entre o cliente e a equipe de desenvolvimento. Como enfatizam Sommerville e Sawyer (2009, p. 68), "os requisitos funcionais servem como um contrato entre o cliente e a equipe de desenvolvimento".

Medir o progresso do desenvolvimento e testar o sistema:

Os requisitos funcionais também servem como base para medir o progresso do desenvolvimento e testar o sistema. Conforme destacado por Goguen e Linde (1993, p. 118), "os requisitos funcionais servem como base para medir o progresso do desenvolvimento e testar o sistema, pois definem o que o sistema deve fazer e como ele deve se comportar".

Em resumo, os requisitos funcionais são elementos essenciais no desenvolvimento de sistemas de software, pois definem as funcionalidades e operações que o sistema deve realizar, garantindo que atenda às necessidades dos stakeholders, fornecendo uma base para o planejamento e desenvolvimento do sistema, estabelecendo um contrato entre o cliente e a equipe de desenvolvimento e servindo como base para medir o progresso do desenvolvimento e testar o sistema.

A Tabela 2 apresenta os requisitos funcionais relacionados à funcionalidade de login do aplicativo Estudo360. Esses requisitos detalham as principais funcionalidades necessárias para garantir o acesso seguro e eficiente dos usuários.

Tabela 2 - Requisitos funcionais da função Login.

Identificação	Requisito Funcional	Prioridade
---------------	---------------------	------------

RF0001	Os usuários devem fornecer credenciais válidas (como nome de usuário ou endereço de e-mail e senha) para acessar sua conta no aplicativo.	Essencial
RF0002	O sistema deve verificar as credenciais fornecidas pelo usuário para garantir a autenticidade e a segurança do login.	Essencial
RF0003	Os usuários que não têm uma conta devem poder criar uma nova fornecendo informações pessoais (como nome completo, RA, curso, semestre, período, endereço, e-mail, telefone, data, sexo).	Essencial
RF0004	O aplicativo deve verificar a disponibilidade do endereço de e-mail para evitar duplicatas durante a criação de uma nova conta.	Essencial
RF0005	Os usuários devem ter a opção de redefinir a senha em caso de esquecimento, com o sistema enviando um link de redefinição de senha para o e-mail registrado.	Essencial
RF0006	As informações de login e dados pessoais dos usuários devem ser armazenados de forma segura e protegidos por medidas de segurança, como criptografia.	Essencial

Fonte: Elaborado pelos autores (2024)

A Tabela 3 apresenta os requisitos funcionais relacionados à tela inicial (Home) do aplicativo Estudo360. Esses requisitos definem as principais funcionalidades necessárias para oferecer uma experiência personalizada.

Tabela 3 - Requisitos funcionais da função Home.

Identificação	Requisito Funcional	Prioridade
RF0007	O sistema deve exibir o nível atual do usuário na tela inicial, através de uma imagem representativa.	Essencial
RF0008	O sistema deve permitir que o usuário acesse e interaja com até quatro funções principais diretamente da tela inicial.	Essencial
RF0009	O sistema deve permitir que o usuário acesse as notificações a partir da tela inicial.	Essencial
RF0010	O sistema deve permitir que o usuário acesse as configurações a partir da tela inicial.	Essencial
RF0011	O sistema deve permitir que o usuário acesse a função Ajuda a partir da tela inicial.	Essencial
RF0012	O sistema deve permitir que o usuário personalize quais funções são exibidas na tela inicial.	Essencial
RF0013	O sistema deve permitir que o usuário visualize detalhadamente sua pontuação e nível ao clicar na área correspondente.	Essencial

RF0014	O sistema deve permitir que o usuário configure preferências de notificações para funcionalidades específicas.	Essencial
RF0015	O sistema deve permitir que o usuário altere sua senha mediante fornecimento da senha atual e nova senha.	Essencial
RF0016	O sistema deve permitir que o usuário recupere sua senha em caso de esquecimento, enviando um link de redefinição.	Essencial
RF0017	O sistema deve enviar notificações ao usuário de acordo com as preferências configuradas.	Essencial
RF0018	O sistema deve sincronizar automaticamente as informações exibidas na tela inicial com os dados mais recentes.	Essencial

Fonte: Elaborado pelos autores (2024)

A Tabela 4 apresenta os requisitos funcionais associados à funcionalidade de agenda do aplicativo Estudo360. Esses requisitos abrangem a criação, visualização, edição e exclusão de eventos, além de recursos adicionais, como integração com a função Pomodoro, compartilhamento externo e filtros de pesquisa, que tornam a gestão de compromissos mais eficiente e personalizada para os usuários.

Tabela 4 - Requisitos funcionais da função Agenda.

Identificação	Requisito Funcional	Prioridade
RF0019	O sistema deve permitir aos usuários adicionar eventos à agenda.	Essencial
RF0020	Os usuários devem visualizar eventos existentes.	Essencial
RF0021	Os usuários devem editar eventos existentes.	Essencial
RF0022	Os usuários devem excluir eventos existentes.	Essencial
RF0023	A função agenda deve permitir integração com a função Pomodoro.	Importante
RF0024	Os eventos podem ser compartilhados externamente.	Importante
RF0025	O sistema deve conter uma funcionalidade de pesquisa para encontrar eventos na agenda.	Essencial
RF0026	A pesquisa pode incluir um filtro de prioridades (alta, média, baixa).	Importante

Fonte: Elaborado pelos autores (2024)

A Tabela 5 apresenta os requisitos funcionais relacionados à funcionalidade de calendário acadêmico do aplicativo. Esses requisitos definem as principais características para exibição e gerenciamento de eventos acadêmicos, como a visualização por mês, pesquisa de eventos e atualizações em tempo real, além de recursos opcionais, como o compartilhamento externo de eventos.

Tabela 5 - Requisitos funcionais da função Calendário Acadêmico.

Identificação	Requisito Funcional	Prioridade
RF0027	O sistema deve exibir eventos importantes relacionados à instituição no calendário acadêmico.	Essencial
RF0028	O sistema deve exibir o calendário separado por mês.	Essencial
RF0029	O sistema pode permitir pesquisar pelo título e descrição dos eventos.	Importante
RF0030	O usuário pode compartilhar os eventos para um aplicativo externo.	Desejável
RF0031	As atualizações dos dados devem refletir imediatamente para os usuários.	Essencial

Fonte: Elaborado pelos autores (2024)

A Tabela 6 apresenta os requisitos funcionais associados à funcionalidade de horário de aula. Esses requisitos abrangem a visualização detalhada dos horários por dia da semana, informações específicas de disciplinas, localização de salas no mapa do campus e a possibilidade de pesquisar horários de outros cursos.

Tabela 6 - Requisitos funcionais da função Horário de aula.

Identificação	Requisito Funcional	Prioridade
RF0032	O sistema deve apresentar os horários de aula de cada semestre, de cada curso e de cada período.	Essencial
RF0033	Os usuários devem visualizar os horários de aula por dia da semana.	Essencial
RF0034	Os usuários devem visualizar as informações da disciplina (Nome, Professor, Sala e Horário).	Essencial
RF0035	Deve haver uma opção para mostrar a localização da sala de aula no mapa do campus.	Essencial
RF0036	O sistema exibe o horário de acordo com as informações do semestre, curso e período do usuário.	Essencial
RF0037	O usuário pode pesquisar pelos horários de outros cursos, períodos e semestres, se assim desejar.	Essencial
RF0038	O usuário pode compartilhar por aplicativo externo as informações das aulas do dia específico.	Importante
RF0039	As atualizações dos dados devem refletir imediatamente para os usuários.	Essencial

Fonte: Elaborado pelos autores (2024)

A Tabela 7 apresenta os requisitos funcionais da funcionalidade de notas e faltas. Esses requisitos abrangem o gerenciamento de disciplinas, registro de notas avaliativas e cálculo automático da média final. Além disso, destacam-se funcionalidades para análise de desempenho, exibição de médias em tempo real e registro de faltas, proporcionando aos usuários uma visão clara e detalhada de seu progresso acadêmico.

Tabela 7 - Requisitos funcionais da função Notas e Faltas.

Identificação	Requisito Funcional	Prioridade
RF0040	Os alunos devem adicionar disciplinas em uma lista, especificando o nome e o ícone de cada disciplina.	Essencial
RF0041	O usuário pode alterar as informações como nome e ícone da disciplina.	Essencial
RF0042	O usuário pode excluir a disciplina selecionada.	Essencial
RF0043	O usuário pode alterar o título, a nota e o peso de uma atividade avaliativa.	Essencial
RF0044	O usuário pode excluir as informações da atividade avaliativa selecionada.	Essencial
RF0045	Os alunos devem registrar as notas obtidas em cada avaliação de uma disciplina, assim como seu peso correspondente.	Essencial
RF0046	Com base nas notas inseridas, o sistema deve calcular automaticamente a média final do aluno em cada disciplina.	Essencial
RF0047	O sistema deve exibir a média parcial em tempo real, à medida que inserem novas notas.	Essencial
RF0048	Os alunos devem analisar seu desempenho em cada disciplina.	Essencial
RF0049	O sistema pode exibir um demonstrativo do desempenho em cada disciplina.	Importante
RF0050	O usuário pode acrescentar em um contador suas faltas em uma matéria específica.	Importante

Fonte: Elaborado pelos autores (2024)

A Tabela 8 detalha os requisitos funcionais da funcionalidade Pomodoro. Esses requisitos incluem a configuração e personalização do temporizador para sessões de estudo e intervalos, emissão de alertas visuais e sonoros, e a associação de pomodoros a tarefas específicas. Além disso, a função integra recursos para registro

e análise de produtividade, bem como a conversão de tempo dedicado em pontos, reforçando o engajamento e a motivação dos usuários.

Tabela 8 - Requisitos funcionais da função Pomodoro.

Identificação	Requisito Funcional	Prioridade
RF0051	O aplicativo deve oferecer um temporizador Pomodoro configurável para sessões de estudo.	Essencial
RF0052	O aplicativo deve permitir que o usuário defina o tempo dedicado ao foco (pomodoro).	Essencial
RF0053	O aplicativo deve permitir que o usuário defina o tempo de descanso entre os pomodoros.	Essencial
RF0054	O aplicativo deve permitir que o usuário personalize o número de pomodoros por ciclo.	Essencial
RF0055	O aplicativo deve permitir que o usuário inicie o período definido no temporizador rapidamente.	Essencial
RF0056	O aplicativo pode permitir que o usuário pause o temporizador com um único clique.	Essencial
RF0057	O aplicativo deve permitir que o usuário retome o temporizador previamente pausado com um único clique.	Essencial
RF0058	O temporizador deve exibir o tempo restante de dedicação à tarefa.	Essencial
RF0059	O temporizador deve exibir o tempo restante de intervalo.	Essencial
RF0060	O temporizador deve emitir alertas sonoros ao final de cada ciclo.	Essencial
RF0061	O aplicativo deve emitir alertas visuais ao final de cada ciclo.	Essencial
RF0062	O aplicativo pode permitir que os usuários associem pomodoros a tarefas específicas de sua agenda.	Desejável
RF0063	O aplicativo pode permitir o registro das atividades realizadas durante cada pomodoro.	Desejável
RF0064	O aplicativo pode permitir adicionar notas opcionais para referência futura sobre atividades realizadas.	Desejável
RF0065	O aplicativo pode fornecer a visualização de relatórios de produtividade sobre o tempo gasto em cada atividade.	Desejável
RF0066	O sistema converterá minutos dedicados no Pomodoro em pontos que serão utilizados na função Pontuação.	Essencial
RF0067	O usuário pode selecionar a função Timer.	Essencial
RF0068	O aplicativo deve permitir que o usuário inicie o Timer logo após ter definido um valor.	Essencial

RF0069	O aplicativo pode permitir que o usuário pause o Timer com um único clique.	Essencial
RF0070	O aplicativo deve permitir que o usuário retome o Timer previamente pausado com um único clique.	Essencial
RF0071	O temporizador deve exibir o tempo restante do Timer.	Essencial
RF0072	O temporizador deve emitir alertas sonoros no final do Timer.	Essencial
RF0073	O aplicativo deve emitir alertas visuais no final do Timer.	Essencial
RF0074	O usuário pode selecionar a função Cronômetro.	Essencial
RF0075	O aplicativo deve permitir que o usuário inicie o Cronômetro em um único clique.	Essencial
RF0076	O aplicativo pode permitir que o usuário pause o Cronômetro com um único clique.	Essencial
RF0077	O aplicativo deve permitir que o usuário retome o Cronômetro previamente pausado com um único clique.	Essencial
RF0078	O temporizador deve exibir o tempo do Cronômetro.	Essencial

Fonte: Elaborado pelos autores (2024)

A Tabela 9 apresenta os requisitos funcionais da funcionalidade de pontuação. Esses requisitos descrevem a mecânica de gamificação, que inclui a atribuição de pontos baseados em interações com recursos do aplicativo, progressão por níveis e desbloqueio de recompensas exclusivas.

Tabela 9 - Requisitos funcionais da função Pontuação.

Identificação	Requisito Funcional	Prioridade
RF0079	Os usuários devem ganhar pontos ao interagir com diferentes recursos do Pomodoro (como concluir sessões de estudo).	Essencial
RF0080	Os usuários devem gerar uma quantidade específica de pontos para cada tipo de interação, dependendo da relevância e complexidade da ação.	Essencial
RF0081	Os usuários devem ser atribuídos a um nível com base na quantidade de pontos acumulados.	Essencial
RF0082	À medida que os usuários acumulam mais pontos, eles devem subir de nível e desbloquear recompensas exclusivas.	Essencial
RF0083	No aplicativo, as recompensas devem ser proporcionais ao nível alcançado e aumentar em valor à medida que o usuário avança no sistema de pontuação.	Essencial

RF0084	Os usuários devem ter acesso a uma interface onde possam acompanhar seu progresso (como visualizar o número de pontos acumulados, o nível atual, próximas recompensas, histórico de interações).	Essencial
--------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------

Fonte: Elaborado pelos autores (2024).

A Tabela 10 apresenta os requisitos funcionais relacionados à função Ajuda do aplicativo Estudo360. Esses requisitos abrangem o acesso a tutoriais específicos de cada funcionalidade, a exibição de dúvidas frequentes e suas respostas, além da navegação intuitiva entre as páginas de suporte, garantindo que os usuários possam resolver suas dúvidas de forma prática e eficiente.

Tabela 10 - Requisitos funcionais da função Ajuda.

Identificação	Requisito Funcional	Prioridade
RF0085	O sistema deve permitir que o usuário acesse o tutorial/dúvidas frequentes a partir da action bar	Essencial
RF0086	O sistema deve exibir uma lista de opções de tutoriais separadas por funcionalidades do aplicativo.	Essencial
RF0087	O sistema deve permitir que o usuário selecione uma funcionalidade específica para visualizar o tutorial correspondente.	Essencial
RF0088	O sistema deve retornar à página de Tutorial/Dúvidas ao final de cada tutorial específico de funcionalidade.	Essencial
RF0089	O sistema deve exibir as principais dúvidas frequentes na parte inferior da página de Tutoriais/Dúvidas.	Essencial
RF0090	O sistema deve exibir as respostas das dúvidas frequentes na parte inferior da página de Tutoriais/Dúvidas.	Essencial

Fonte: Elaborado pelos autores (2024).

3.2.2 Requisitos Não Funcionais

Complementam os Requisitos Funcionais e são fundamentais no que envolve desempenho e qualidade do Desenvolvimento de Sistemas de softwares, abrangendo vertentes como: segurança, confiabilidade, usabilidade, entre outros. Como mencionado por Soni, Liang e Hsieh (2015, p. 208), "os requisitos não funcionais são fundamentais para avaliar a arquitetura do sistema, garantindo que ela atenda aos critérios de desempenho e escalabilidade exigidos".

Importância dos Requisitos Não Funcionais

Os requisitos não funcionais são fundamentais para garantir o atendimento às necessidades de qualidade do sistema para o usuário final. Conforme salientado por Bass, Clements e Kazman (2012, p. 121), "os requisitos não funcionais são essenciais para assegurar que o sistema atenda aos padrões de qualidade exigidos pelos stakeholders".

Além disso, esses requisitos fornecem uma base sólida para o desenvolvimento do sistema, estabelecendo restrições e limitações de qualidade que devem ser consideradas. Como observado por Leffingwell (2010, p. 93), "os requisitos não funcionais definem as restrições e limitações que o sistema deve respeitar, garantindo que ele opere dentro dos parâmetros desejados".

Outro aspecto importante é a melhoria da experiência do usuário. De acordo com Nielsen (1994, p. 112), "os requisitos não funcionais relacionados à usabilidade são essenciais para melhorar a experiência do usuário, tornando o sistema mais acessível e eficiente". Nesse sentido, um sistema que atenda aos requisitos de usabilidade tende a ser mais intuitivo e fácil de utilizar, contribuindo significativamente para a satisfação do usuário.

Exemplos de requisitos não funcionais que contribuem para a melhoria do sistema incluem atributos de desempenho, segurança, usabilidade e manutenibilidade. Esses aspectos garantem que o sistema opere de forma confiável e eficiente, atendendo às expectativas dos stakeholders.

O "Desempenho" trata-se do tempo de resposta do sistema para o usuário final, isto é, a taxa de transferência de dados, a qual não é recomendada ser demorada para não comprometer a experiência do usuário com o sistema. A "Segurança", por sua vez, envolve etapas como autenticação, autorização, criptografia, confidencialidade, integridade, disponibilidade. Já a "Usabilidade" visa fornecer uma interface amigável com uma navegação intuitiva. Na "Confiabilidade" é onde é pensado em alternativas em casos de falhas, como recuperação, backup e restauração. Na "Manutenibilidade" o sistema deve ser de fácil modificação e correção de erros. Com a "Portabilidade" o sistema deve ser capaz de ser executado em diferentes plataformas.

Sendo assim, os requisitos não funcionais abordam especificações que não se referem diretamente às funcionalidades, mas sim à sua estrutura, desempenhando

um papel na satisfação do usuário final e na eficácia do sistema como um todo. A Tabela 11 apresenta os requisitos não funcionais deste projeto.

Tabela 11 - Requisitos Não Funcionais do Projeto.

Identificação	Requisito Não Funcional	Prioridade
RNF001	Usabilidade: O aplicativo deve possuir uma interface intuitiva e amigável, facilitando a navegação e interação do usuário.	Essencial
RNF002	Performance: O aplicativo deve carregar as informações e responder às interações do usuário em até 2 segundos.	Essencial
RNF003	Compatibilidade: O aplicativo deve ser compatível com as versões mais recentes dos sistemas operacionais Android.	Essencial
RNF004	Segurança de Dados: Os dados pessoais e sensíveis dos usuários devem ser protegidos por criptografia durante o armazenamento e transmissão.	Essencial
RNF005	Confiabilidade: O sistema deve estar disponível 99.9% do tempo, minimizando falhas e interrupções de serviço.	Essencial
RNF006	Escalabilidade: O aplicativo deve suportar um aumento no número de usuários sem degradação significativa de performance.	Importante
RNF007	Manutenibilidade: O código do aplicativo deve ser modular e bem documentado para facilitar futuras manutenções e atualizações.	Essencial
RNF008	Eficiência Energética: O aplicativo deve ser otimizado para minimizar o consumo de bateria dos dispositivos móveis.	Importante
RNF009	Integração com Terceiros: O sistema deve permitir a integração com APIs e serviços externos de forma segura e eficiente.	Desejável
RNF010	Conformidade Legal: O sistema deve estar em conformidade com as leis e regulamentações aplicáveis, como LGPD.	Essencial
RNF011	Backup e Recuperação: O sistema deve realizar backups regulares dos dados e permitir recuperação rápida em caso de falhas.	Essencial
RNF012	Escalabilidade de Dados: O banco de dados deve suportar o crescimento do volume de dados sem perda de desempenho.	Importante
RNF013	Resiliência: O sistema deve ser capaz de se recuperar automaticamente de falhas inesperadas.	Importante
RNF014	Atualizações Automáticas: O aplicativo deve suportar atualizações automáticas sem perda de dados do usuário.	Desejável
RNF015	Tempo de Resposta: As funcionalidades críticas devem ter tempo de resposta inferior a 2 segundos.	Essencial

RNF016	Interface Consistente: O design da interface deve ser consistente em todas as telas e funcionalidades.	Essencial
RNF017	Feedback do Usuário: O sistema deve fornecer feedback imediato após cada ação do usuário.	Essencial
RNF018	Privacidade: As informações pessoais dos usuários não devem ser compartilhadas sem consentimento explícito.	Essencial
RNF019	Teste e Qualidade: O sistema deve ser rigorosamente testado para garantir qualidade e ausência de bugs críticos.	Essencial
RNF020	Logs de Erro: O sistema deve registrar erros e exceções para facilitar o diagnóstico e solução de problemas.	Importante
RNF021	Confiabilidade das Notificações: As notificações devem ser entregues ao usuário de forma consistente e sem atrasos significativos.	Importante
RNF022	Uso Eficiente de Recursos: O aplicativo deve gerenciar eficientemente o uso do processamento do dispositivo.	Essencial
RNF023	Segurança de Autenticação: O sistema deve proteger contra ataques e ameaças relacionadas ao login.	Essencial
RNF024	Criptografia de Senhas: As senhas devem ser armazenadas utilizando algoritmos de hash seguros.	Essencial
RNF025	Escalabilidade de Usuários Simultâneos: O sistema deve suportar múltiplos usuários acessando simultaneamente sem degradação.	Importante
RNF026	Documentação do Usuário: O aplicativo deve fornecer ajuda e documentação acessível ao usuário.	Desejável
RNF027	Compatibilidade com Dispositivos Antigos: O aplicativo deve funcionar em dispositivos com até 5 anos de fabricação.	Desejável
RNF028	Personalização: O sistema deve permitir certo nível de personalização sem comprometer a usabilidade geral.	Desejável
RNF029	Segurança na Comunicação: Todas as comunicações entre o aplicativo e o servidor devem ser feitas via protocolos seguros.	Essencial
RNF030	Não Repúdio: O sistema deve garantir que as ações do usuário sejam registradas de forma que não possam ser negadas posteriormente.	Desejável
RNF031	Integração Contínua: O desenvolvimento deve seguir práticas de integração e entrega contínuas para agilizar atualizações.	Desejável
RNF032	Modularidade: O sistema deve ser desenvolvido em módulos independentes para facilitar manutenção e escalabilidade.	Importante
RNF033	Consistência de Dados: O sistema deve garantir a integridade e consistência dos dados em todas as transações.	Essencial

RNF034	Proteção Contra Ataques: O sistema deve ser protegido contra injeção de código, cross-site scripting e outras vulnerabilidades comuns.	Essencial
RNF035	Facilidade de Instalação: O aplicativo deve ser fácil de instalar e configurar pelo usuário final.	Essencial
RNF036	Redução de Latência: O sistema deve minimizar a latência nas comunicações para melhorar a experiência do usuário.	Importante
RNF037	Atualizações Seguras: As atualizações do aplicativo devem ser verificadas quanto à autenticidade e integridade.	Essencial
RNF038	Controle de Versão: O sistema deve manter um controle de versão adequado para todos os componentes.	Importante
RNF039	Monitoramento: O sistema deve possuir mecanismos de monitoramento para desempenho e segurança em tempo real.	Desejável
RNF040	Interoperabilidade: O sistema deve ser capaz de interagir com outros sistemas e serviços quando necessário.	Desejável
RNF041	Facilidade de Uso: O aplicativo deve ser fácil de aprender e usar sem necessidade de treinamento extenso.	Essencial
RNF042	Tempo de Recuperação: Em caso de falha, o sistema deve ser capaz de retornar ao funcionamento normal em até 5 minutos.	Importante
RNF043	Consistência Visual: O design gráfico do aplicativo deve seguir uma identidade visual coerente.	Importante
RNF044	Análise de Uso: O sistema deve coletar dados anônimos de uso para melhoria contínua, respeitando a privacidade do usuário.	Desejável
RNF045	Tolerância a Falhas: O sistema deve continuar operando corretamente mesmo se ocorrerem falhas internas.	Importante
RNF046	Gestão de Recursos do Servidor: O backend deve gerenciar eficientemente recursos como CPU, memória e armazenamento.	Importante
RNF047	Experiência do Usuário: O aplicativo deve proporcionar uma experiência satisfatória, incentivando o uso contínuo.	Essencial
RNF048	Comunicação Offline: O aplicativo deve permitir acesso limitado a funcionalidades quando estiver sem conexão à internet.	Importante
RNF049	Compatibilidade com API's: O sistema deve ser compatível com versões futuras das APIs utilizadas.	Desejável
RNF050	Gerenciamento de Sessões: O sistema deve gerenciar sessões de usuário de forma segura e eficiente.	Essencial

Fonte: Elaborado pelos autores (2024)

3.3 Recursos e Ferramentas

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

- **Android Studio:** É o ambiente de desenvolvimento integrado (IDE) oficial para o desenvolvimento de aplicativos Android, baseado no IntelliJ IDEA. O Android Studio oferece um ambiente unificado para o desenvolvimento de aplicativos para todas as plataformas Android. Com ele, é possível desenvolver, depurar, testar e criar interfaces para smartphones e tablets Android, além de dispositivos Android Wear, Android TV e Android Auto. O IDE fornece recursos como um editor de código inteligente, ferramentas de performance, emuladores e um sistema de compilação flexível baseado no Gradle (ANDROID DEVELOPERS, 2024).
- **Kotlin:** Kotlin é uma linguagem de programação moderna e estaticamente tipada que compila para a Máquina Virtual Java (JVM) e também pode ser compilada para JavaScript ou código nativo. Desenvolvida pela JetBrains, Kotlin é totalmente interoperável com Java, permitindo que os desenvolvedores utilizem bibliotecas Java existentes. Ela foi projetada para ser concisa, segura e interoperável, oferecendo recursos como inferência de tipo, corrotinas para programação assíncrona e sintaxe simplificada. Em 2017, o Google anunciou o suporte oficial ao Kotlin para desenvolvimento Android, tornando-se uma das linguagens preferidas para essa plataforma (JETBRAINS, 2024).
- **Firebase:** O Firebase é uma plataforma de desenvolvimento de aplicativos móveis e web oferecida pelo Google. Ele fornece uma variedade de serviços, como autenticação de usuários, banco de dados em tempo real (Realtime Database), armazenamento em nuvem (Cloud Storage), hospedagem (Hosting) e funções em nuvem (Cloud Functions). O Firebase Realtime Database é um banco de dados NoSQL hospedado na nuvem que permite o armazenamento e sincronização de dados entre usuários em tempo real. A integração com o Firebase simplifica a implementação de funcionalidades complexas e permite que os desenvolvedores se concentrem na experiência do usuário (FIREBASE, 2024).
- **Astah UML:** O Astah UML é uma ferramenta de modelagem de software que suporta a criação de diversos diagramas baseados na Linguagem de Modelagem Unificada (UML). Desenvolvido pela Change Vision, o Astah UML permite que os desenvolvedores visualizem e documentem a arquitetura e o design de sistemas complexos por meio de diagramas de classes, casos de uso, sequência, atividades, entre outros. A ferramenta auxilia na comunicação entre membros da equipe e no

planejamento do desenvolvimento, garantindo uma melhor compreensão dos componentes e interações do sistema (CHANGE VISION, 2024).

- **CorelDRAW:** É um software de design gráfico vetorial desenvolvido pela Corel Corporation, lançado originalmente em 1989. Ele permite a criação de uma ampla gama de produtos de comunicação visual, como logotipos, ilustrações, materiais promocionais, layouts de página e tipografia. O CorelDRAW se destaca por suas ferramentas avançadas de desenho vetorial, interface personalizável e suporte a diversos formatos de arquivo. É amplamente utilizado por designers gráficos e profissionais de marketing para produzir trabalhos de alta qualidade e precisão (COREL, 2024).
- **Adobe Photoshop:** O Adobe Photoshop é um software de edição e manipulação de imagens raster, desenvolvido pela Adobe Inc. desde 1988. Reconhecido como um dos principais programas de edição de imagens, o Photoshop oferece uma vasta gama de ferramentas para retoque, composição, correção de cores, aplicação de efeitos especiais e criação de gráficos para web. Utilizado por fotógrafos, designers gráficos e artistas digitais, o Photoshop é essencial para a produção de imagens de alta qualidade em diversas indústrias (ADOBE, 2024).
- **Figma:** O Figma é uma ferramenta de design de interface de usuário (UI) e experiência do usuário (UX) baseada na nuvem. Lançado em 2016, o Figma permite que designers criem e colaborem em projetos de design em tempo real. Ele oferece recursos para criação de wireframes, protótipos interativos, design de interfaces e colaboração em equipe. A plataforma facilita o compartilhamento de projetos e feedback instantâneo, otimizando o fluxo de trabalho entre designers e desenvolvedores. Sua capacidade de funcionar diretamente no navegador e a integração com outros serviços tornam o Figma uma escolha popular entre profissionais de design (FIGMA, 2024).

4 MODELAGEM

No desenvolvimento de sistemas, a modelagem ajuda a transformar ideias em representações estruturadas e compreensíveis para toda a equipe. Neste trabalho, será utilizada a Unified Modeling Language (UML) para a modelagem do projeto. A UML é uma linguagem padronizada de modelagem que oferece uma série de diagramas e notações para representar diferentes aspectos de um sistema de forma visual e compreensível.

A UML foi desenvolvida para criar uma linguagem comum que pudesse ser utilizada por analistas, designers e desenvolvedores para descrever, visualizar e documentar sistemas de software complexos. De acordo com Ian Sommerville, "A UML foi criada para suportar o desenvolvimento de sistemas orientados a objetos, fornecendo uma notação padronizada que pode ser utilizada em todas as fases do desenvolvimento de software" (SOMMERVILLE, 2011).

A importância da UML na modelagem de projetos reside na sua capacidade de facilitar a comunicação entre todos os envolvidos no desenvolvimento do sistema.

Como Sommerville destaca, "Uma das principais vantagens da UML é que ela fornece uma linguagem comum para todos os participantes do projeto, permitindo que todos compartilhem uma visão comum do sistema" (SOMMERVILLE, 2011). Isso é particularmente relevante em projetos complexos, onde a clareza e a precisão na comunicação são essenciais para o sucesso.

Além disso, a UML permite uma melhor organização e planejamento do desenvolvimento, ajudando a identificar possíveis problemas e inconsistências no design antes mesmo da fase de implementação. "A utilização de modelos UML pode ajudar a reduzir riscos e aumentar a qualidade do software, proporcionando uma base sólida para a implementação" (SOMMERVILLE, 2011). Isso mostra como a modelagem com UML é uma prática valiosa e justificada no desenvolvimento de sistemas.

4.1 Casos de uso

Os casos de uso são uma das ferramentas mais importantes na modelagem de sistemas, especialmente quando se trata de capturar e definir os requisitos funcionais de um projeto. Um caso de uso descreve uma interação típica entre um usuário (ou

ator) e o sistema, detalhando como o sistema responde a uma solicitação específica do usuário para alcançar um objetivo particular.

O Diagrama de Casos de Uso é uma representação visual que ilustra os diferentes casos de uso do sistema e os atores envolvidos. Este diagrama ajuda a identificar todas as funcionalidades principais do sistema e como elas interagem com os usuários. Segundo Ivar Jacobson, "Os casos de uso são uma forma poderosa de capturar os requisitos funcionais de um sistema, concentrando-se nas interações entre os usuários e o sistema" (JACOBSON, 1992).

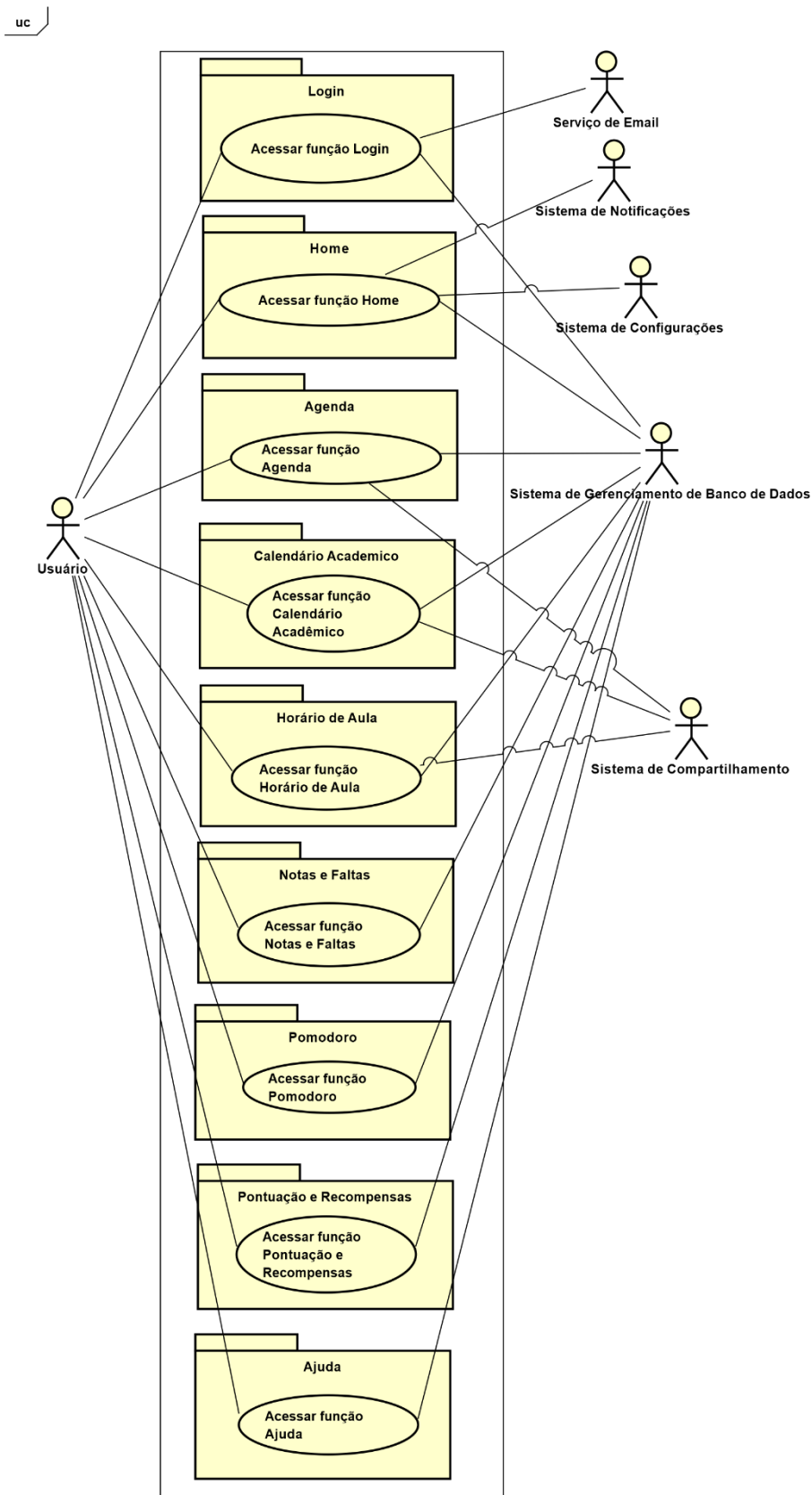
Além do diagrama, o Caso de Uso Textual é uma descrição detalhada de um caso de uso específico, que inclui o fluxo de eventos, as condições prévias, as condições de sucesso e as possíveis exceções. Este nível de detalhe é crucial para assegurar que todos os aspectos do requisito funcional sejam compreendidos e possam ser implementados corretamente. Alistair Cockburn destaca que "a clareza e a precisão na escrita de casos de uso textuais são essenciais para garantir que os desenvolvedores entendam plenamente os requisitos" (COCKBURN, 2000).

A utilização de casos de uso no desenvolvimento de sistemas oferece vários benefícios, incluindo a clareza na comunicação dos requisitos entre os stakeholders e a equipe de desenvolvimento, e a garantia de que todas as funcionalidades desejadas pelos usuários são contempladas. Como afirma Grady Booch, "Os casos de uso proporcionam uma abordagem estruturada para identificar e organizar os requisitos do sistema, facilitando a comunicação entre todos os envolvidos no projeto" (BOOCH et al., 2005).

Portanto, a modelagem de casos de uso, tanto através de diagramas quanto de descrições textuais, é essencial para a definição precisa e completa dos requisitos funcionais do sistema, contribuindo significativamente para o sucesso do projeto.

As próximas figuras apresentam os casos de uso para cada entrada do usuário no sistema. A Figura 1, em específico, ilustra o diagrama de caso de uso geral do aplicativo, fornecendo uma visão ampla das principais interações dos usuários com as funcionalidades do Estudo360.

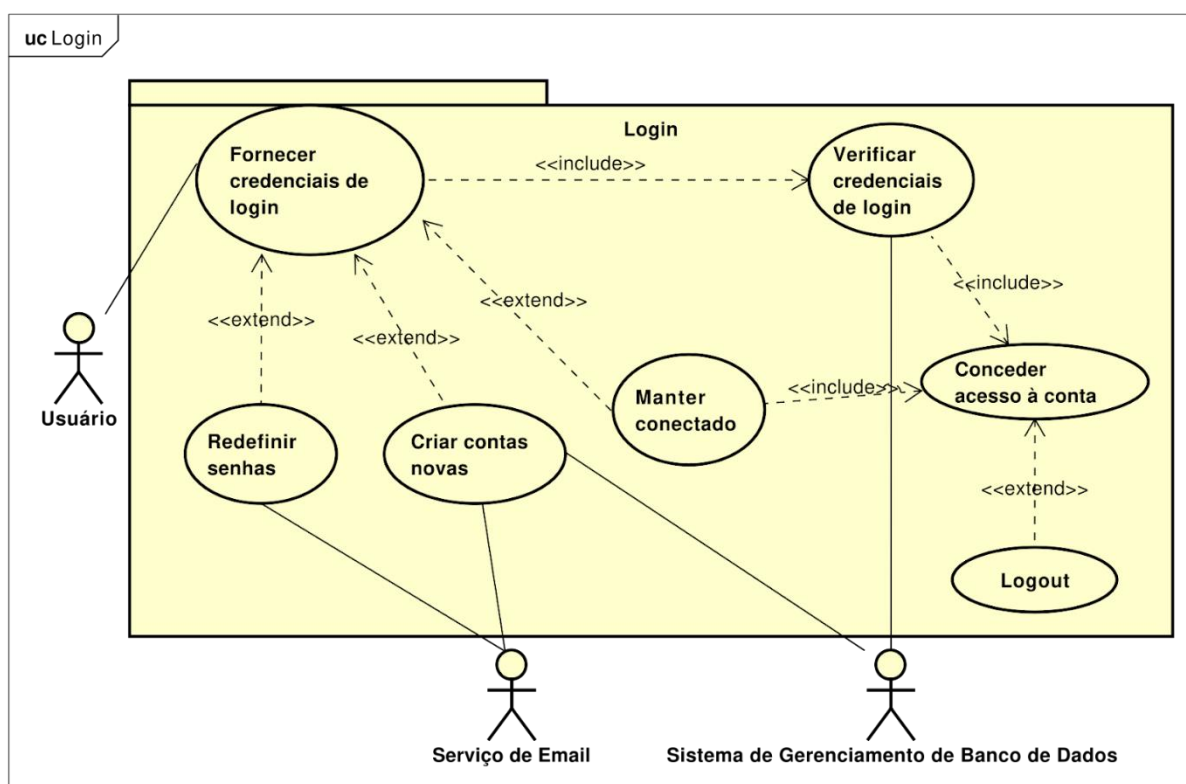
Figura 1 – Diagrama de caso de uso geral do aplicativo.



Fonte: Elaborado pelos autores (2024)

A Figura 2 apresenta o diagrama de caso de uso relacionado à funcionalidade de autenticação e gestão de contas no aplicativo Estudo360. Esse diagrama fornece uma visão detalhada das interações entre os usuários e o sistema, abordando operações como login, registro, redefinição de senha e segurança de credenciais. A Tabela 12 complementa a Figura 2 ao descrever o caso de uso CA001, especificando os elementos envolvidos, pré-condições, fluxo principal e pós-condições associados à funcionalidade.

Figura 2 - Diagrama de caso de uso de autenticação e gestão de contas.



Fonte: Elaborado pelos autores (2024).

Tabela 12 - Caso de Uso CA001: Autenticação e Gestão de Contas.

Elemento	Descrição
Nome	Autenticação e Gestão de Contas
Código	CA001
Ator Principal	Usuário

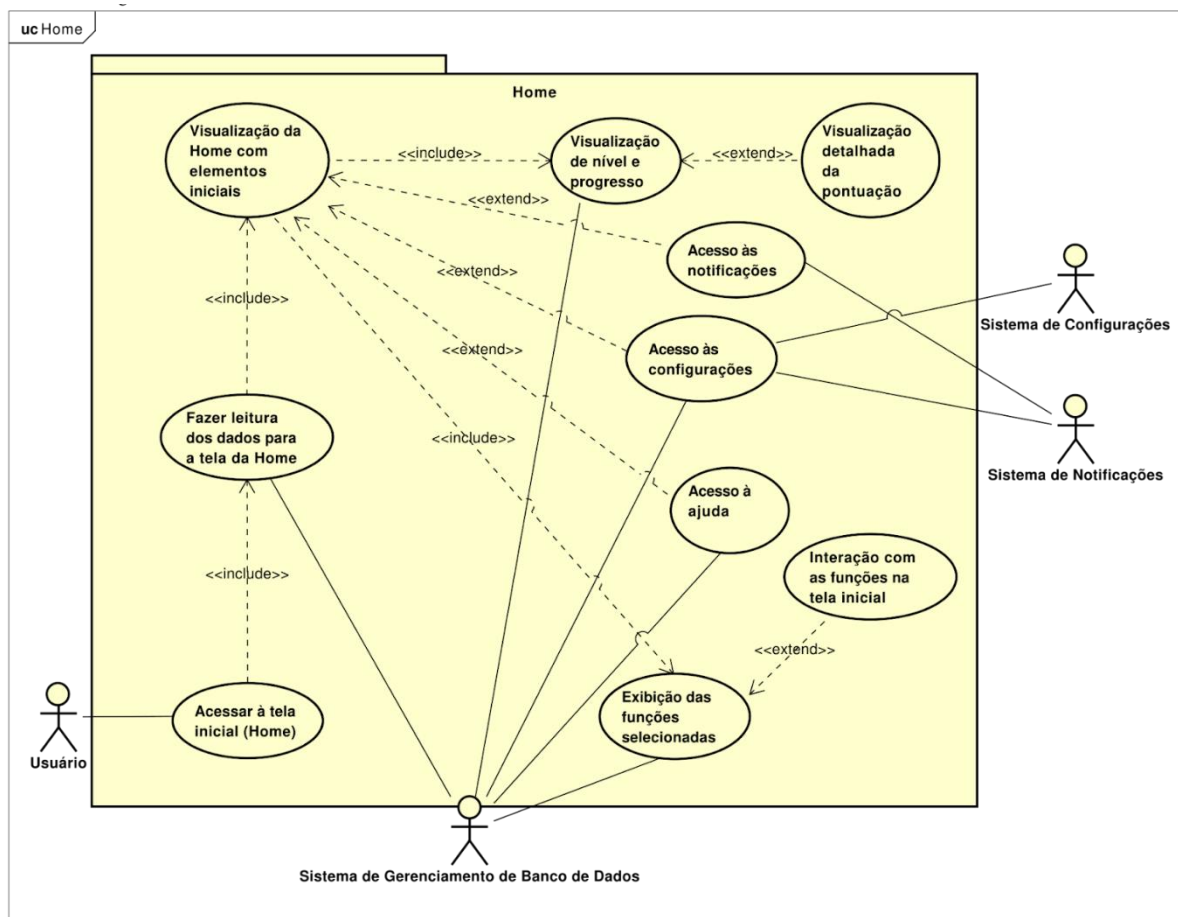
Atores Secundários	- Sistema de Gerenciamento de Banco de Dados: Armazena, recupera e gerencia credenciais.
	- Serviço de E-mail: Envia e-mails para redefinição de senha e comunicações relacionadas.
Descrição	Detalha como os estudantes interagem com o sistema para login, registro, redefinição de senha e manutenção das credenciais. Atores secundários suportam estas operações.
Pré-condições	- O aplicativo deve ter uma interface de login e registro.
	- O sistema de gerenciamento de usuários deve estar operacional.
Fluxo Principal	1. Fornecer e Verificar Credenciais de Login:
	- O estudante insere email e senha na tela de login.
	- O sistema verifica as credenciais e concede acesso se corretas.
	2. Criar Contas Novas:
	- O estudante insere Nome Completo, RA, E-mail, Celular, Sexo, Data de Nascimento, Curso, Semestre (1 a 6), Período, CEP, Rua, Número, Bairro, Cidade, Estado, Senha e Confirmar Senha.
	- Após validação, a conta é criada.
	3. Redefinir Senhas:
	- O estudante solicita redefinição de senha, insere email associado e segue instruções enviadas.
	4. Garantir Segurança:
	- O sistema usa criptografia e solicita atualizações periódicas de senha para segurança.
Pós-condições	- O estudante acessa o aplicativo após verificação de login.
	- Uma nova conta é criada e pronta para uso.
	- Senha redefinida com sucesso, garantindo acesso.

	- Dados do estudante permanecem seguros.
Requisitos Especiais	- O aplicativo deve seguir as melhores práticas de segurança de dados.
	- O banco de dados deve implementar medidas robustas contra ataques e vulnerabilidades.

Fonte: Elaborado pelos autores (2024)

A Figura 3 apresenta o diagrama de caso de uso da tela inicial (Home) do aplicativo Estudo360, oferecendo uma visão ampla das principais interações do usuário com essa funcionalidade. A Tabela 13 complementa a Figura 3 ao detalhar o caso de uso CA002, descrevendo os elementos, fluxos principais e alternativos, pré-condições e pós-condições associadas à Home. Essa funcionalidade centraliza informações como nível, progresso, notificações e acesso às funções principais, além de permitir personalizações e acesso rápido a ferramentas essenciais.

Figura 3 – Diagrama de caso de uso da Home.



Fonte: Elaborado pelos autores (2024)

Tabela 13 - Caso de Uso CA002: Tela Inicial (Home).

Elemento	Descrição
Nome	Tela Inicial (Home)
Código	CA002
Ator Principal	Usuário
Atores Secundários	- Sistema de Notificações: Responsável por exibir notificações relevantes ao usuário.
	- Sistema de Gerenciamento de Banco de Dados: Gerencia os dados do usuário.
	- Sistema de Configurações: Permite personalização das funções exibidas na tela inicial.

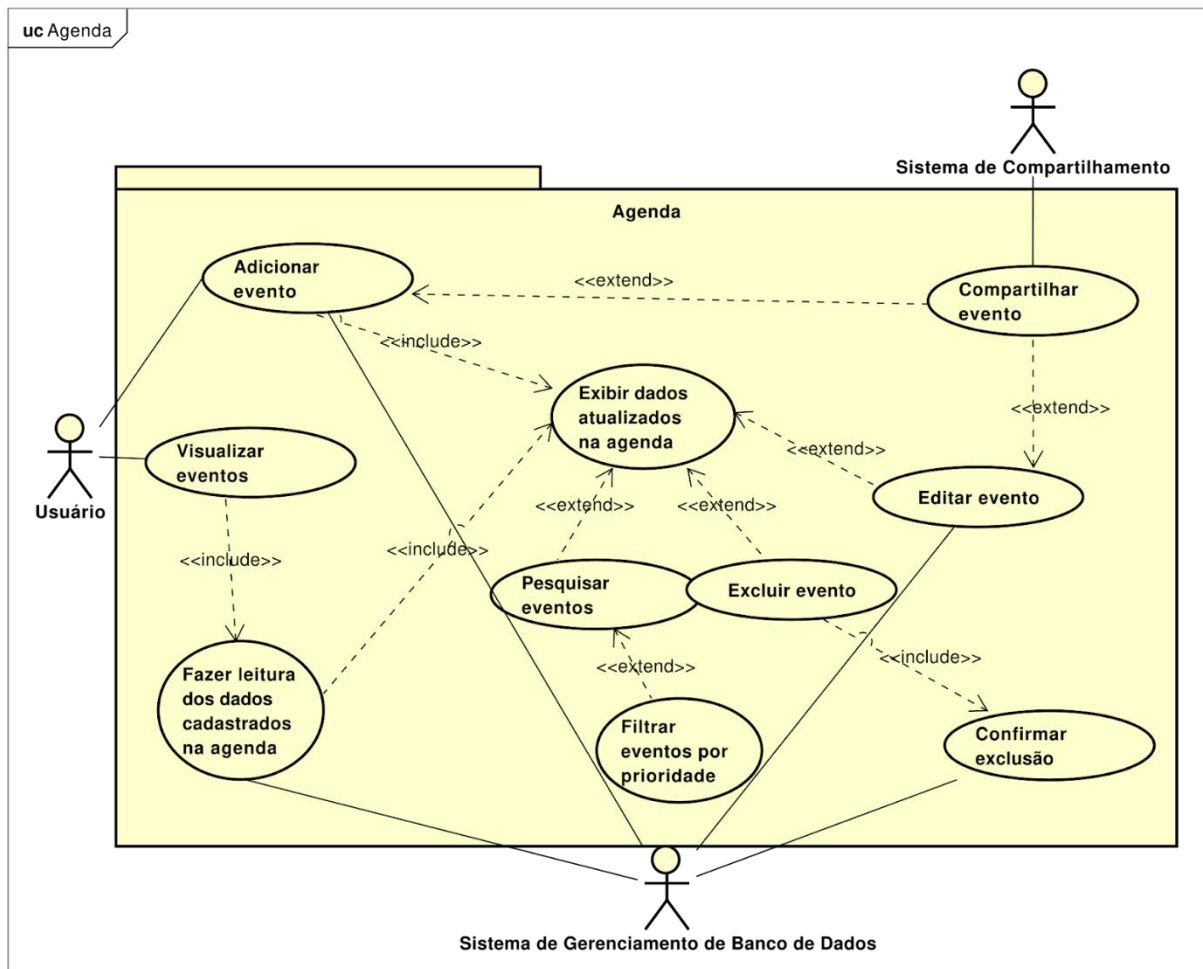
Descrição	Detalha como o usuário interage com a tela inicial do aplicativo, visualizando nível, progresso, funções principais e notificações, além de acessar configurações, ajuda e interagir com funcionalidades como Timer e Cronômetro.
Pré-condições	- O usuário deve estar autenticado no aplicativo.
	- As configurações das funções selecionadas pelo usuário devem estar armazenadas.
	- Dados como nível, pontos, eventos, horários e notas devem estar disponíveis no banco de dados.
Fluxo Principal	1. Acesso à Tela Inicial: O usuário abre o aplicativo e é direcionado para a tela inicial.
	2. Visualização de Nível e Progresso:
	- Nível atual do usuário exibido com imagem representativa.
	- Barra de progresso indica os pontos acumulados em relação ao próximo nível.
	3. Exibição das Funções Selecionadas:
	- Até quatro funções escolhidas pelo usuário são exibidas.
	- Funções disponíveis: Agenda, Calendário Acadêmico, Notas/Faltas, Horário das Aulas, Timer, Cronômetro.
	4. Interação com as Funções na Tela Inicial:
	- Agenda: Exibe eventos do dia atual e permite ver detalhes.
	- Calendário Acadêmico: Mostra eventos acadêmicos do dia com informações adicionais.
	- Notas/Faltas: Resumo das notas e faltas, com detalhes por disciplina.
	- Horário das Aulas: Exibe horários das aulas do dia, com detalhes adicionais.

	- Timer e Cronômetro: Permitem contagem regressiva e medição de atividades com opções de pausar, retomar e zerar.
	5. Acesso às Notificações e Configurações:
	- Botão de notificações exibe lista de notificações recebidas.
	- Botão de configurações permite personalizar o aplicativo.
	6. Acesso à Ajuda: Botão de ajuda leva à tela de dúvidas frequentes ou tutorial.
	7. Visualização Detalhada da Pontuação: Clicando na área de progresso, o usuário acessa estatísticas detalhadas e histórico.
Fluxos Alternativos	A1. Personalização das Funções na Tela Inicial:
	- Usuário seleciona até quatro funções na tela de configurações, e alterações são salvas.
	A2. Configuração de Notificações:
	- Usuário ajusta preferências de notificações na tela de configurações.
	A3. Alteração de Senha:
	- Usuário insere senha atual e nova senha na tela de configurações.
	A4. Recuperação de Senha:
	- Usuário solicita redefinição de senha, recebe email com instruções e redefine sua senha.
Pós-condições	- O usuário tem acesso a uma tela inicial personalizada.
	- As interações contribuem para o progresso de nível do usuário.
	- Configurações e preferências são salvas para uso futuro.

Fonte: Elaborado pelos autores (2024)

A Figura 4 apresenta o diagrama de caso de uso referente ao gerenciamento de agenda no aplicativo Estudo360, destacando as interações principais entre os usuários e o sistema. Complementando essa visão, a Tabela 14 descreve o caso de uso CA003, detalhando elementos como fluxos principais e alternativos, pré-condições e pós-condições, além das funcionalidades oferecidas, como adição, edição, exclusão, visualização e compartilhamento de eventos na agenda. Essa funcionalidade busca proporcionar uma gestão prática e eficiente das atividades acadêmicas do usuário.

Figura 4 - Diagrama de caso de uso de gerenciamento de agenda.



Fonte: Elaborado pelos autores (2024)

Tabela 14 - Caso de Uso CA003: Gerenciamento de Agenda.

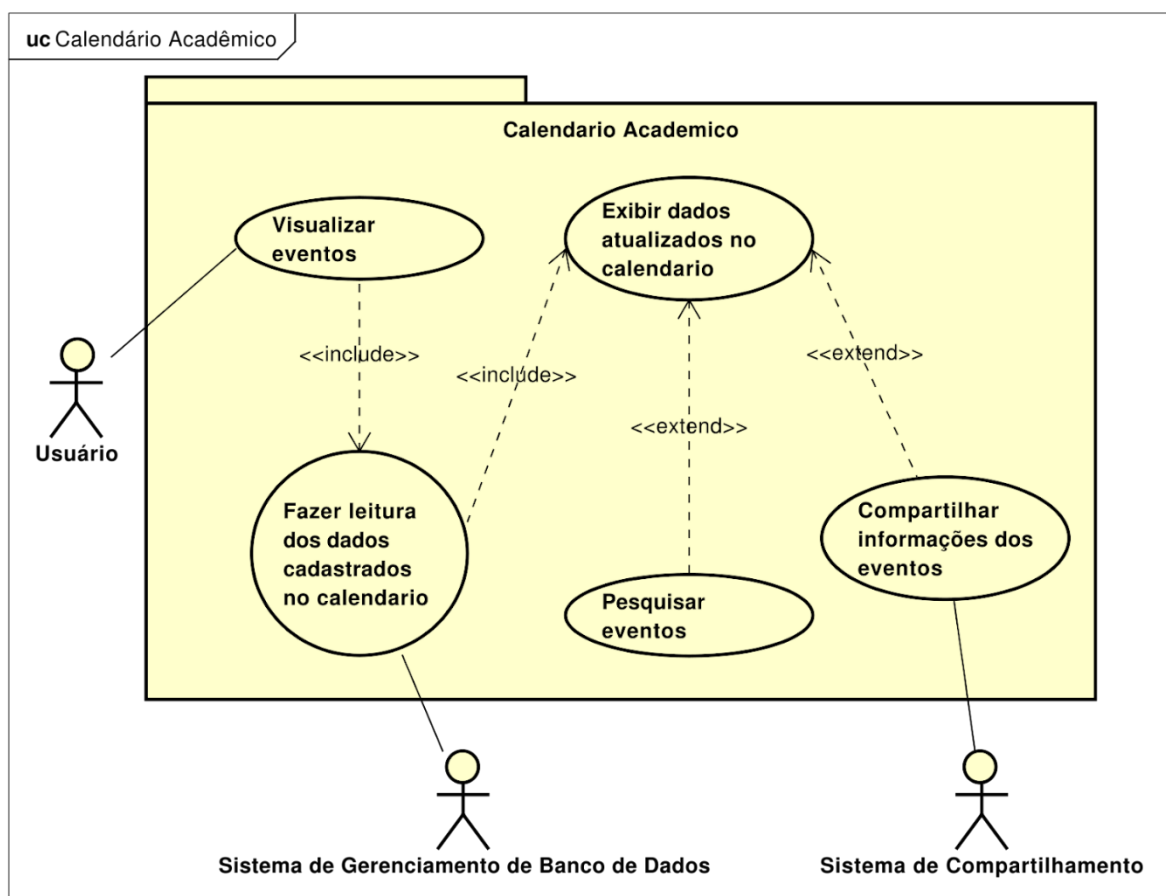
Elemento	Descrição
Nome	Gerenciamento de Agenda
Código	CA003
Ator Principal	Usuário
Atores Secundários	- Sistema de Gerenciamento de Banco de Dados: Gerencia o armazenamento e atualização dos eventos.
	- Sistema de Compartilhamento: Permite o envio de eventos para outros aplicativos.
Descrição	Permite ao estudante gerenciar sua agenda acadêmica, incluindo adicionar, visualizar, editar, excluir e compartilhar eventos.
Pré-condições	- O estudante deve estar autenticado no aplicativo.
Fluxo Principal	1. Acesso à Agenda:
	- O estudante acessa a funcionalidade de Agenda.
	- O aplicativo exibe um calendário com eventos listados para o dia selecionado.
	- Dias com eventos são marcados com um ponto vermelho; dias sem eventos mostram a mensagem: "Não há eventos para essa data."
	2. Informações dos Eventos:
	- Cada evento exibe título, descrição, data, hora, local, um checkbox para habilitar o Pomodoro e um dropdown para definir a prioridade do evento (alta ou normal).
	3. Adicionar Evento:
	- O estudante clica em "Adicionar Evento", preenche as informações do evento e salva.
	4. Visualizar Eventos:

	- O estudante filtra eventos por data ou prioridade (Todas, Alta, Normal ou Baixa).
	- O mecanismo de busca retorna eventos letra a letra conforme a pesquisa do usuário.
	5. Editar Evento:
	- O estudante pressiona o evento, clica em "Editar", faz alterações e salva.
	6. Excluir Evento:
	- O estudante pressiona o evento, clica em "Excluir", confirma a ação e o evento é removido.
	7. Compartilhar Evento:
	- O estudante seleciona "Compartilhar Evento", escolhe o método de compartilhamento e conclui a ação.
Pós-condições	- O evento é adicionado, atualizado ou excluído no banco de dados e refletido na interface.
Requisitos Especiais	- O aplicativo deve proteger os dados dos eventos, especialmente ao compartilhá-los.
	- A interface de gerenciamento de agenda deve ser intuitiva e facilitar a interação do usuário.

Fonte: Elaborado pelos autores (2024)

A Figura 5 apresenta o diagrama de caso de uso da funcionalidade de calendário acadêmico no aplicativo Estudo360, evidenciando as principais interações do usuário com os eventos exibidos. A Tabela 15 complementa a Figura 5 ao descrever o caso de uso CA004, detalhando fluxos principais e alternativos, além de elementos como pesquisa, compartilhamento e visualização offline. Essa funcionalidade proporciona ao usuário uma ferramenta eficiente para gerenciar prazos e eventos importantes relacionados à vida acadêmica.

Figura 5 - Diagrama de caso de uso do calendário acadêmico.



Fonte: Elaborado pelos autores (2024)

Tabela 15 - Caso de Uso CA004: Calendário Acadêmico.

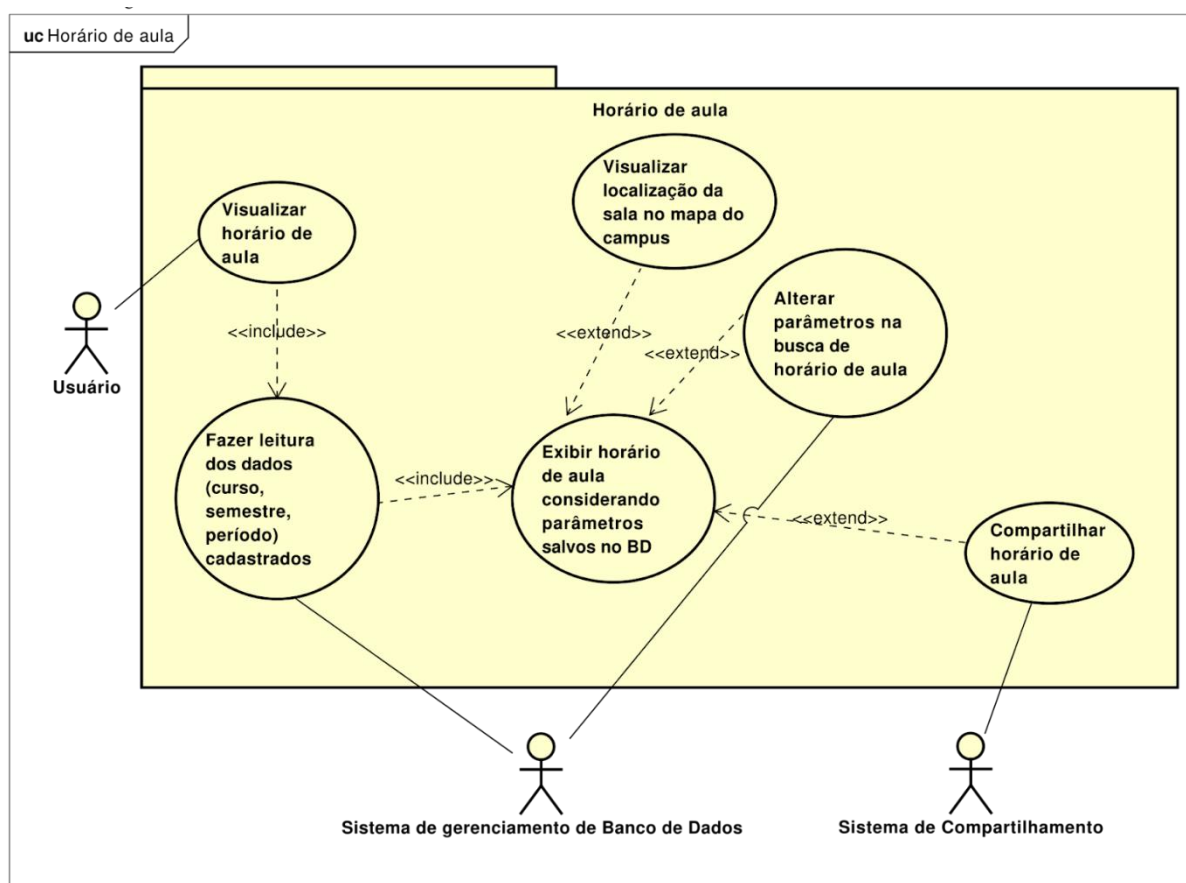
Elemento	Descrição
Nome	Calendário Acadêmico
Código	CA004
Ator Principal	Usuário
Atores Secundários	- Sistema de Gerenciamento de Banco de Dados: Gerencia os eventos do calendário acadêmico.
	- Sistema de Compartilhamento: Permite o envio de eventos para outros aplicativos.

Descrição	Permite ao usuário visualizar eventos do calendário acadêmico, filtrar por descrição ou data, e compartilhar eventos por aplicativos de mensagens.
Pré-condições	- O usuário deve estar autenticado no aplicativo.
	- O calendário acadêmico deve estar configurado e atualizado.
Fluxo Principal	1. Visualizar Eventos Importantes:
	- O usuário acessa o calendário acadêmico e visualiza eventos como datas de exames, prazos, feriados e outros eventos acadêmicos.
	2. Pesquisar Eventos:
	- O usuário pode pesquisar eventos utilizando palavras-chave para identificar eventos específicos.
	3. Compartilhar Eventos:
	- O aplicativo permite compartilhar informações de eventos por aplicativos de mensagens.
Fluxos Alternativos	A1. Alteração de Evento:
	- Quando ocorre uma alteração em um evento, o calendário acadêmico é atualizado automaticamente.
	A2. Visualização Sem Conexão:
	- Se estiver sem internet, o aplicativo exibe a última versão sincronizada do calendário.
Pós-condições	- O usuário visualiza os eventos acadêmicos, auxiliando no planejamento de estudos e atividades.
	- Notificações informam o usuário sobre eventos próximos.
Requisitos Especiais	- O aplicativo deve garantir atualização contínua e automática do calendário acadêmico.

Fonte: Elaborado pelos autores (2024)

A Figura 6 apresenta o diagrama de caso de uso relacionado à personalização de horários de aula no aplicativo Estudo360, destacando as interações do estudante com as funcionalidades de visualização e gerenciamento de horários. A Tabela 16 complementa a Figura 6 ao detalhar o caso de uso CA005, descrevendo os fluxos principais, como exibição de horários por dia da semana, detalhes de disciplinas e localização das salas no mapa do campus, além de funcionalidades adicionais, como o compartilhamento das informações. Essa funcionalidade visa oferecer praticidade e organização para o planejamento acadêmico do estudante.

Figura 6 - Diagrama de caso de uso de personalização de horário da aula.



Fonte: Elaborado pelos autores (2024).

Tabela 16 - Caso de Uso CA005: Horário de Aula.

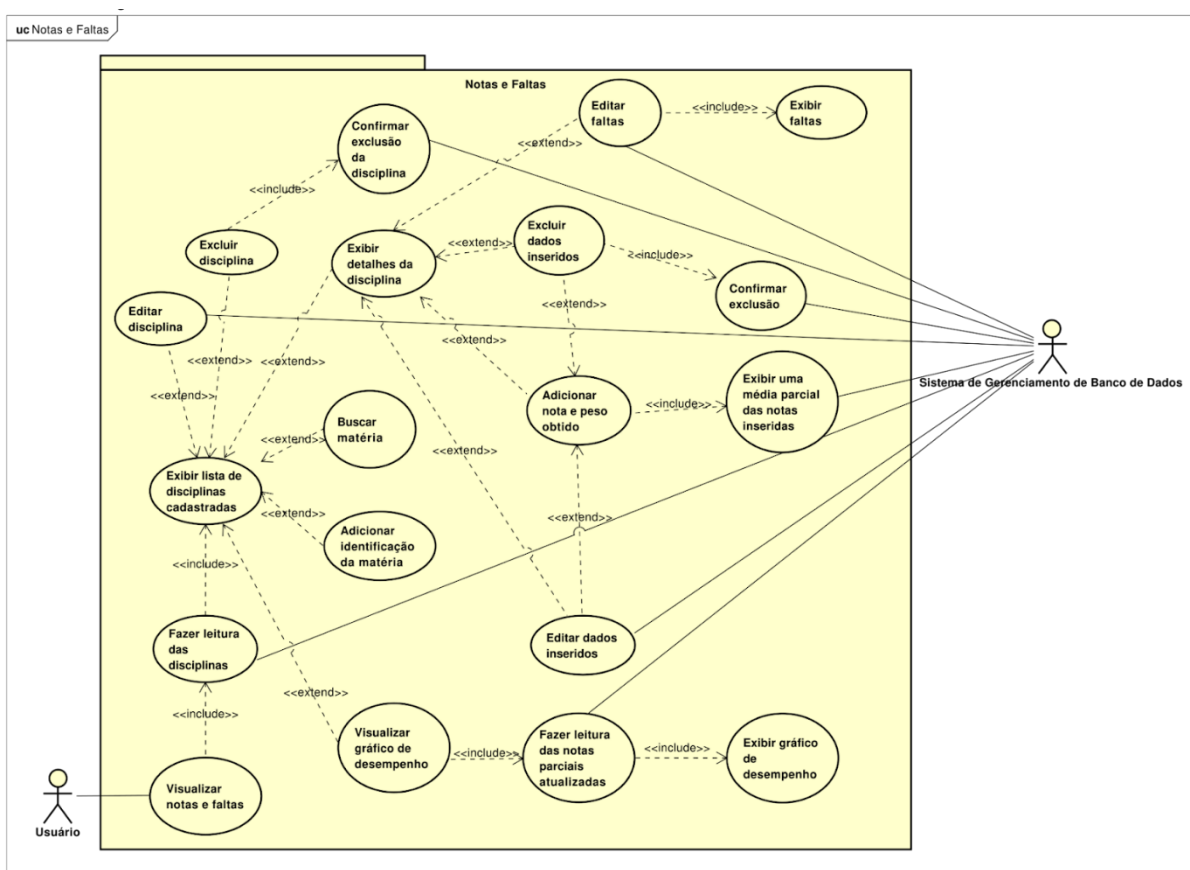
Elemento	Descrição
Nome	Horário de Aula
Código	CA005

Ator Principal	Usuário
Atores Secundários	- Sistema de Gerenciamento de Banco de Dados: Gerencia os horários e detalhes das aulas.
	- Sistema de Compartilhamento: Permite o envio das informações de horário por aplicativos.
Descrição	Permite ao estudante visualizar horários de aula por dia da semana, detalhes de cada disciplina, e acessar a localização das salas no mapa do campus. Também permite o compartilhamento dessas informações por aplicativos de mensagens.
Pré-condições	- O estudante deve estar autenticado no aplicativo.
	- O horário de aulas deve estar cadastrado no sistema pelo administrador.
Fluxo Principal	1. Acesso à Funcionalidade de Horários:
	- O estudante acessa a funcionalidade de horários no aplicativo.
	- O aplicativo exibe os horários organizados por dia da semana.
	2. Visualização de Detalhes das Aulas:
	- O estudante visualiza nome da disciplina, horário de início e término, professor e sala.
	3. Localização no Mapa:
	- O estudante clica na localização de uma aula para abrir o mapa do campus com a sala destacada.
	4. Compartilhamento de Horários:
	- O estudante clica no ícone de compartilhamento, escolhe o aplicativo de mensagem e envia as informações.
Fluxos Alternativos	A1. Aula sem Localização Definida:
	- Se a localização da aula não estiver definida, o aplicativo informa a localização da secretaria acadêmica para consulta.
Pós-condições	- Os horários são atualizados automaticamente após alterações feitas pelo administrador.
	- O estudante tem acesso a horários claros e atualizados, incluindo localizações precisas no mapa.
Requisitos Especiais	- O aplicativo deve incluir um mapa do campus integrado para facilitar a localização de salas.

Fonte: Elaborado pelos autores (2024)

A Figura 7 apresenta o diagrama de caso de uso referente à funcionalidade de gerenciamento de notas e faltas no aplicativo Estudo360, destacando as interações do usuário com o sistema para registro, cálculo e análise de desempenho acadêmico. A Tabela 17 complementa a Figura 7 ao descrever o caso de uso CA006, especificando fluxos principais e alternativos, como a adição de disciplinas, configuração de fórmulas de cálculo de médias, registro de notas e análise de desempenho. Essa funcionalidade visa oferecer uma gestão eficiente das notas e faltas, proporcionando ao usuário uma visão clara de seu progresso acadêmico.

Figura 7 - Diagrama de caso de uso da função Notas e Faltas.



Fonte: Elaborado pelos autores (2024)

Tabela 17 - Caso de Uso CA006: Gerenciamento de Notas e Faltas.

Elemento	Descrição
Nome	Gerenciamento de Notas e Faltas

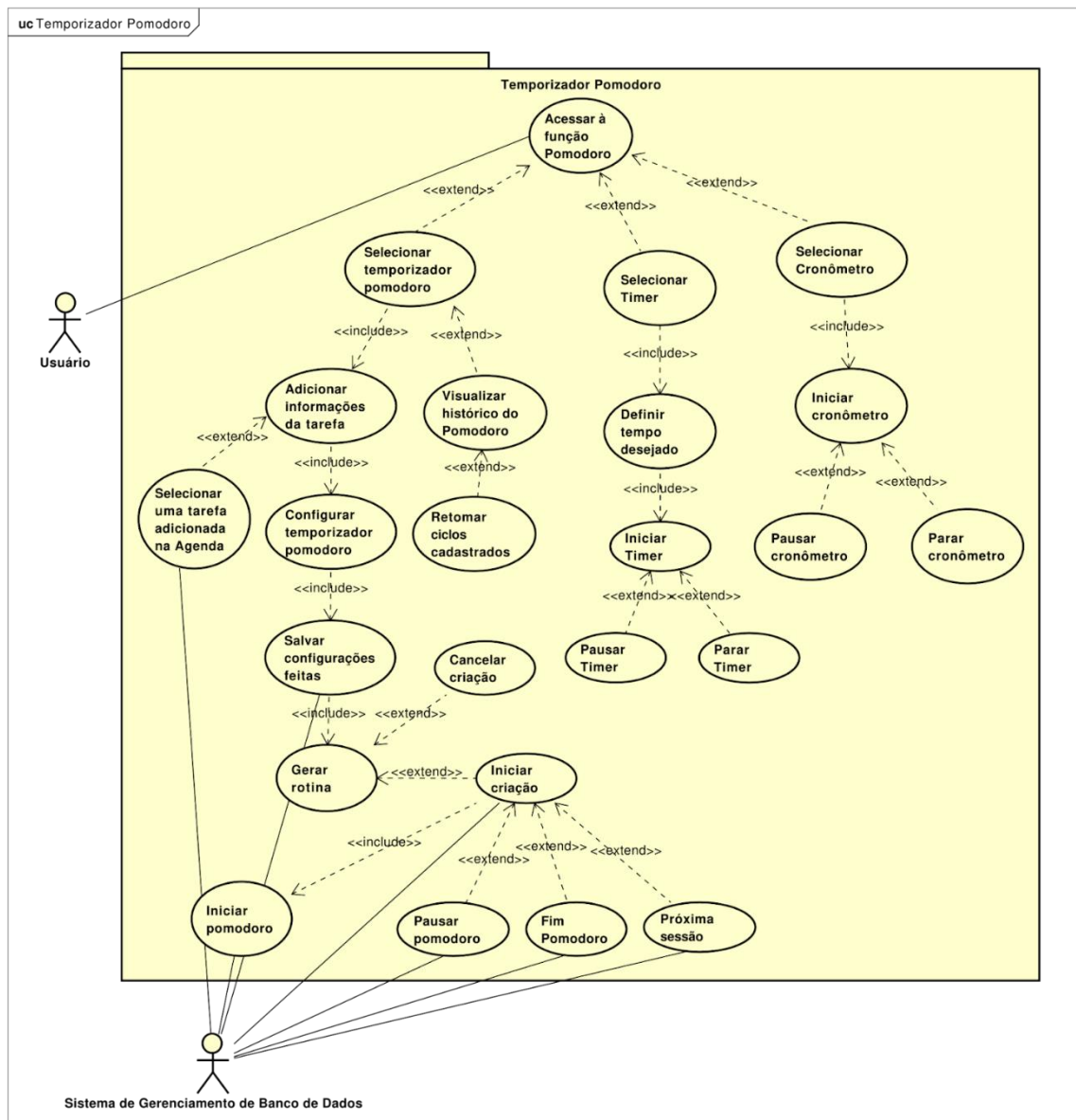
Código	CA006
Ator Principal	Usuário
Ator Secundário	Sistema de Gerenciamento de Banco de Dados: Gerencia o armazenamento de disciplinas, notas e fórmulas.
Descrição	Permite ao usuário adicionar disciplinas, cadastrar fórmulas de cálculo para médias, registrar notas, calcular médias automaticamente e analisar desempenho por disciplina.
Pré-condições	- O usuário deve estar autenticado no aplicativo.
	- O usuário deve criar disciplinas para registrar e gerenciar notas.
Fluxo Principal	1. Adicionar e Organizar Disciplinas:
	- O usuário seleciona a opção para adicionar uma nova disciplina, inserindo o nome e selecionando um ícone.
	2. Cadastrar Fórmulas de Cálculo de Médias:
	- O usuário define uma fórmula de cálculo para cada disciplina, baseada nos pesos de diferentes tipos de avaliações.
	3. Registrar e Calcular Médias de Notas:
	- O usuário registra as notas obtidas nas avaliações.
	- O aplicativo calcula automaticamente a média com base na fórmula cadastrada.
	4. Analisar Desempenho por Disciplina:
	- O usuário visualiza um resumo do desempenho acadêmico por disciplina.
Fluxos Alternativos	A1. Ajuste de Fórmula de Cálculo:
	- O usuário pode editar a fórmula de cálculo a qualquer momento, caso os critérios de avaliação mudem.

	A2. Exclusão de Disciplina:
	- O usuário pode remover uma disciplina e todas as notas associadas, caso ela seja cancelada ou não mais relevante.
	A3. Edição do Nome ou Ícone da Disciplina:
	- O usuário pode editar o nome ou ícone de uma disciplina, clicando em "Editar" e salvando as alterações.
Pós-condições	- Notas Registradas: As notas são armazenadas no sistema e associadas à disciplina.
	- Média Calculada: As médias são calculadas e exibidas conforme as fórmulas cadastradas.
	- Desempenho Analisado: O usuário tem uma visão clara de seu desempenho por disciplina, identificando áreas de melhoria.
Requisitos Especiais	- O aplicativo deve fornecer uma interface intuitiva para registro de notas, configuração de fórmulas e análise de desempenho.

Fonte: Elaborado pelos autores (2024)

A Figura 8 apresenta o diagrama de caso de uso relacionado à funcionalidade do Pomodoro no aplicativo Estudo360, ilustrando as interações do usuário com o sistema para gerenciar sessões de foco, descanso e produtividade. A Tabela 18 complementa a Figura 8 ao descrever o caso de uso CA007, detalhando os fluxos principais e alternativos, como configuração e utilização do temporizador Pomodoro, Timer e Cronômetro, além do monitoramento da produtividade por meio de relatórios. Essa funcionalidade é projetada para ajudar os usuários a gerenciar seu tempo de estudo de forma eficiente, incentivando a concentração e recompensando o esforço com pontos no sistema."

Figura 8 - Diagrama de caso de uso do Pomodoro.



Fonte: Elaborado pelos autores (2024)

Tabela 18 - Caso de Uso CA007: Utilização do Temporizador Pomodoro.

Elemento	Descrição
Nome	Utilização do Temporizador Pomodoro
Código	CA007
Ator Principal	Usuário
Atores Secundários	- Sistema de Gerenciamento de Banco de Dados: Armazena informações de sessões e relatórios.

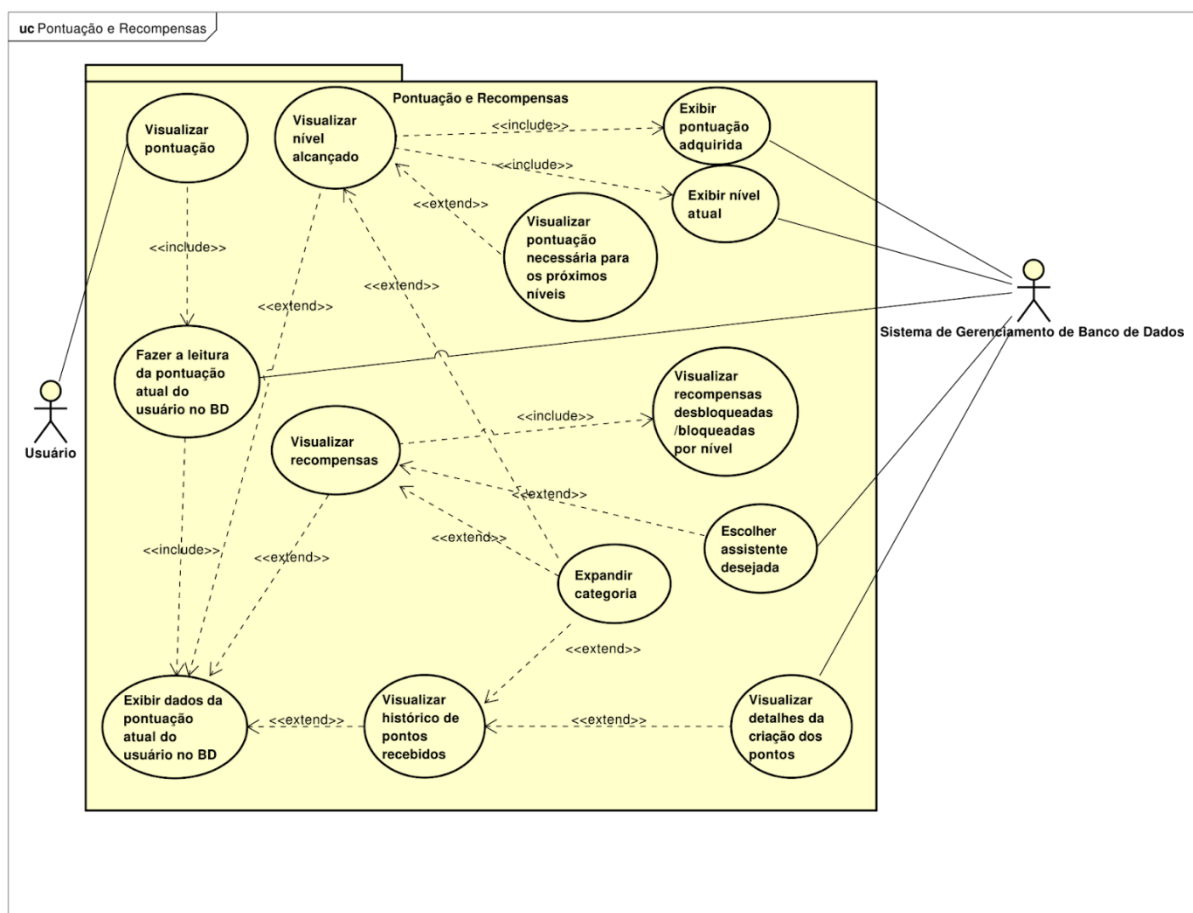
	- Sistema de Notificações: Envia alertas para início e fim das sessões de foco e descanso.
Descrição	Permite ao usuário utilizar a técnica Pomodoro para gerenciar sessões de estudo, alternar entre Pomodoro, Timer e Cronômetro, configurar temporizadores, iniciar, pausar e visualizar relatórios de produtividade. O tempo de estudo é recompensado com pontos que contribuem para a progressão de níveis no aplicativo.
Pré-condições	- O usuário deve estar autenticado no aplicativo.
	- O usuário deve ter acesso à funcionalidade de Temporizador Pomodoro no aplicativo.
Fluxo Principal	1. Seleção de Temporizador:
	- O usuário escolhe entre Pomodoro, Timer e Cronômetro na área superior da tela.
	2. Configuração e Utilização do Temporizador Pomodoro:
	- Configurar a duração das sessões de foco (ex.: 25 minutos), intervalos curtos (ex.: 5 minutos) e intervalos longos (ex.: 15 minutos).
	- Iniciar o temporizador, que conta regressivamente.
	- Receber notificações no final de períodos de foco e descanso.
	- Pausar e retomar o temporizador a qualquer momento.
	- Visualizar relatórios que mostram tempo dedicado ao estudo e informações de produtividade.
	3. Configuração e Utilização do Timer:
	- Configurar um tempo regressivo até 0.
	- Receber notificações ao final do tempo.
	- Pausar e retomar o Timer conforme necessário.

	4. Configuração e Utilização do Cronômetro:
	- Iniciar o cronômetro para medir a duração de atividades.
	- Pausar, retomar ou encerrar o cronômetro a qualquer momento.
Fluxos Alternativos	A1. Ajustes Durante o Estudo:
	- O usuário ajusta configurações do temporizador antes ou após ciclos de estudo para otimizar a rotina.
	A2. Notificações Personalizadas:
	- Configurar notificações personalizadas para alertas de início de foco e descanso, escolhendo a aparência do assistente virtual.
Pós-condições	- Temporizador Utilizado: Sessões de estudo gerenciadas com períodos de foco e descanso definidos.
	- Produtividade Monitorada: Relatórios detalham o tempo dedicado ao estudo e ajudam a analisar produtividade.
Requisitos Especiais	- O aplicativo deve fornecer uma interface intuitiva para configurar e operar os temporizadores.
	- As notificações devem ser eficazes e não intrusivas, garantindo concentração durante os estudos.

Fonte: Elaborado pelos autores (2024).

A Figura 9 apresenta o diagrama de caso de uso relacionado ao sistema de pontuação e recompensas do aplicativo Estudo360, destacando as interações do usuário com a funcionalidade de acumular pontos, subir de nível e desbloquear benefícios exclusivos. A Tabela 19 complementa a Figura 9 ao descrever o caso de uso CA008, detalhando fluxos principais e alternativos, como a atribuição de pontos por interações, progressão de níveis e acesso a recompensas. Essa funcionalidade visa engajar os usuários, promovendo maior envolvimento com as ferramentas do aplicativo por meio de uma mecânica de gamificação educativa.

Figura 9 - Diagrama de caso de uso da Pontuação e Recompensa.



Fonte: Elaborado pelos autores (2024).

Tabela 19 - Caso de Uso CA008: Sistema de Pontuação e Recompensas.

Elemento	Descrição
Nome	Sistema de Pontuação e Recompensas
Código	CA008
Ator Principal	Usuário
Ator Secundário	Sistema de Gerenciamento de Banco de Dados: Gerencia o registro de pontos, níveis e recompensas.
Descrição	Permite ao estudante acumular pontos por interações no aplicativo, como sessões de estudo usando o temporizador Pomodoro. Os pontos

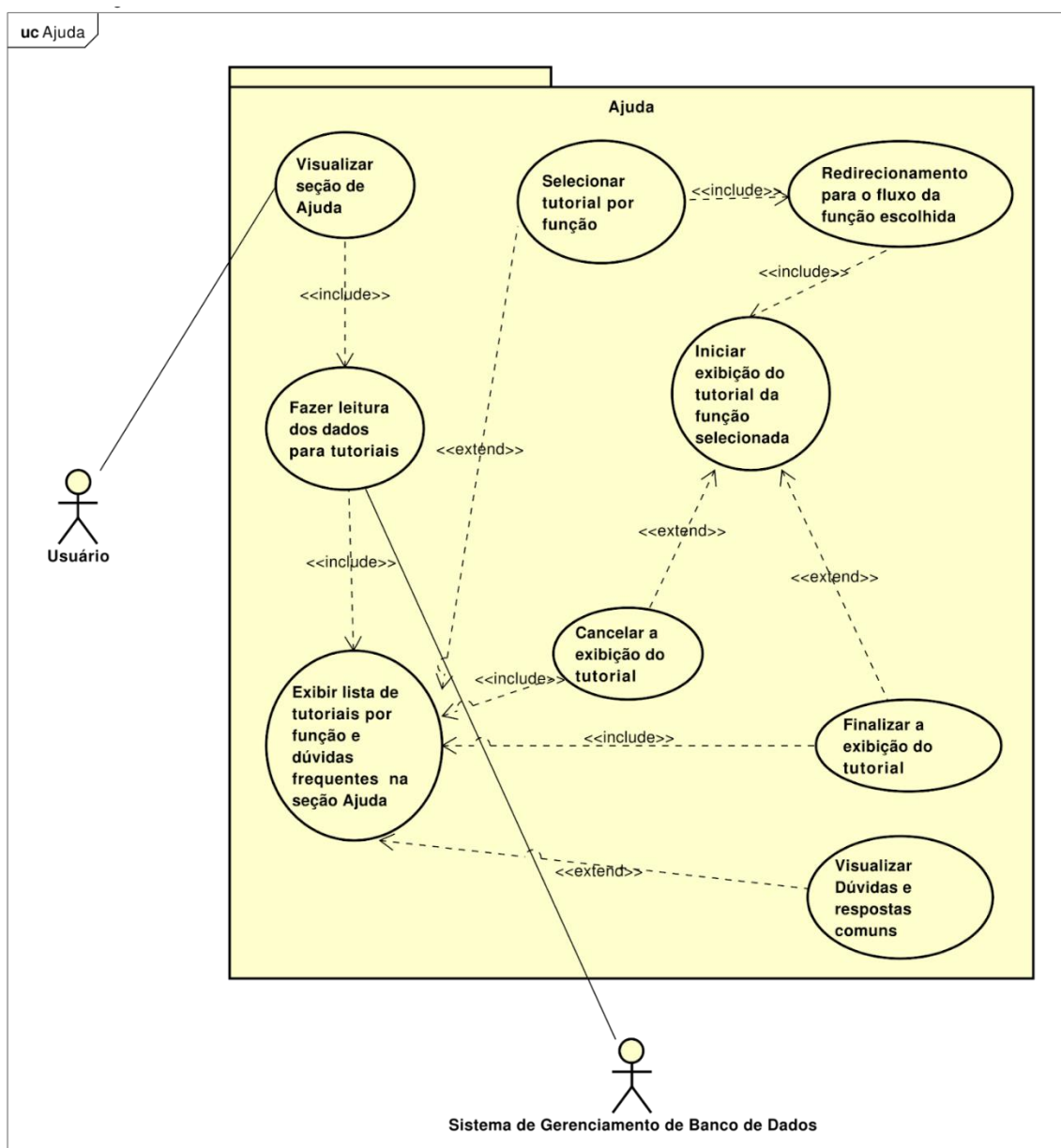
	acumulados permitem subir de nível e desbloquear recompensas como avatares personalizáveis.
Pré-condições	- O estudante deve estar autenticado no aplicativo.
	- O sistema de pontuação e recompensas deve estar implementado no aplicativo.
Fluxo Principal	1. Ganhar Pontos:
	- O estudante realiza atividades qualificadas, como completar sessões de estudo com o temporizador Pomodoro.
	- O aplicativo atribui automaticamente pontos pelas interações realizadas.
	2. Subir de Nível:
	- À medida que acumula pontos, o estudante avança para novos níveis.
	- Cada nível desbloqueia benefícios ou recompensas adicionais.
	3. Desbloquear Recompensas:
	- Recompensas, como avatares ou personalizações no Timer Pomodoro, são desbloqueadas em níveis específicos.
Fluxos Alternativos	A1. Eventos Temporários:
	- Eventos exclusivos podem oferecer recompensas temporárias para incentivar maior atividade no aplicativo.
Pós-condições	- Pontuação Atribuída: Pontos são registrados de forma justa e associada às interações do usuário.
	- Níveis e Recompensas Acessados: O estudante pode acessar níveis mais altos e desbloquear recompensas, melhorando sua experiência no aplicativo.

Requisitos Especiais	- O sistema de pontuação deve ser robusto e transparente, garantindo atribuição justa e consistente de pontos.
	- As recompensas devem incentivar participação significativa, sem comprometer o propósito educacional do aplicativo.

Fonte: Elaborado pelos autores (2024).

A Figura 10 apresenta o diagrama de caso de uso da funcionalidade de ajuda do aplicativo Estudo360, destacando as interações do usuário com as opções de tutoriais interativos e perguntas frequentes (FAQs). A Tabela 20 complementa a Figura 10 ao descrever o caso de uso CA009, detalhando os fluxos principais e alternativos para acessar guias detalhados e respostas a dúvidas comuns. Essa funcionalidade é projetada para oferecer suporte prático e rápido ao usuário, facilitando a compreensão e utilização das diversas ferramentas do aplicativo.

Figura 10 - Diagrama de caso de uso da função Ajuda.



Fonte: Elaborado pelos autores (2024).

Tabela 20 - Caso de Uso CA009: Funcionalidade de Ajuda.

Elemento	Descrição
Nome	Funcionalidade de Ajuda
Código	CA009
Ator Principal	Usuário (aluno ou estudante que utiliza o aplicativo)

Atores Secundários	Sistema de Gerenciamento de Banco de Dados: Gerencia os tutoriais e FAQs disponibilizados ao usuário.
Descrição	Permite que o usuário obtenha informações detalhadas sobre as funcionalidades do aplicativo por meio de tutoriais guiados ou respostas a perguntas frequentes (FAQs). A seção é dividida em dois modos principais:
	- Tutorial: Guia interativo que destaca e explica as principais funcionalidades do aplicativo.
	- Dúvidas: Lista de perguntas e respostas que esclarecem dúvidas comuns sobre o uso do aplicativo.
Pré-condições	- O usuário deve estar autenticado no aplicativo.
	- O usuário deve ter acesso à seção de Ajuda através da interface principal do aplicativo.
Fluxo Principal	1. Acessar a Funcionalidade de Ajuda:
	- O usuário abre o aplicativo e seleciona a funcionalidade de ajuda no menu ou na Action Bar.
	- O aplicativo exibe a tela de Ajuda com as opções "Tutorial" e "Dúvidas".
	2. Modo Tutorial:
	- O usuário seleciona uma funcionalidade na lista de tutoriais.
	- O aplicativo guia o usuário por um tutorial interativo, destacando e explicando as partes principais da funcionalidade selecionada.
	- Após o tutorial, o usuário é redirecionado para a tela de Ajuda.
	3. Modo Dúvidas:
	- O usuário seleciona uma pergunta na lista de FAQs (ex.: "O que é Pomodoro?").

	- O aplicativo exibe a resposta correspondente, proporcionando a informação desejada.
Pós-condições	- O usuário tem acesso a informações detalhadas sobre como utilizar as funcionalidades do aplicativo.
	- O usuário esclarece dúvidas comuns de forma rápida e prática.
Requisitos Especiais	- O aplicativo deve oferecer uma interface clara e organizada para navegação na seção de Ajuda.
	- O conteúdo dos tutoriais e FAQs deve ser atualizado para refletir as últimas funcionalidades do aplicativo.

Fonte: Elaborado pelos autores (2024).

4.2 Diagrama de Classes

O Diagrama de Classes é uma das principais ferramentas da UML (Unified Modeling Language) utilizada na modelagem de sistemas orientados a objetos. Ele representa a estrutura estática do sistema, mostrando as classes, seus atributos, métodos e os relacionamentos entre elas. Este diagrama é crucial para o design e a documentação do sistema, proporcionando uma visão clara da organização e das interações entre os componentes do sistema.

Segundo Ian Sommerville, "os diagramas de classes são fundamentais para a modelagem de sistemas orientados a objetos, pois eles permitem a visualização da estrutura estática do sistema, facilitando o entendimento e o planejamento do design do software" (SOMMERVILLE, 2011). Através do diagrama de classes, é possível identificar as principais entidades do sistema e como elas se relacionam, o que é essencial para a definição precisa da arquitetura do software.

Grady Booch, um dos criadores da UML, também destaca a importância dos diagramas de classes em sua obra, afirmando que "os diagramas de classes são um meio poderoso de representar os elementos de um sistema e seus relacionamentos, proporcionando uma base sólida para o desenvolvimento de sistemas orientados a objetos" (BOOCH et al., 2005). Booch enfatiza que esses diagramas ajudam a

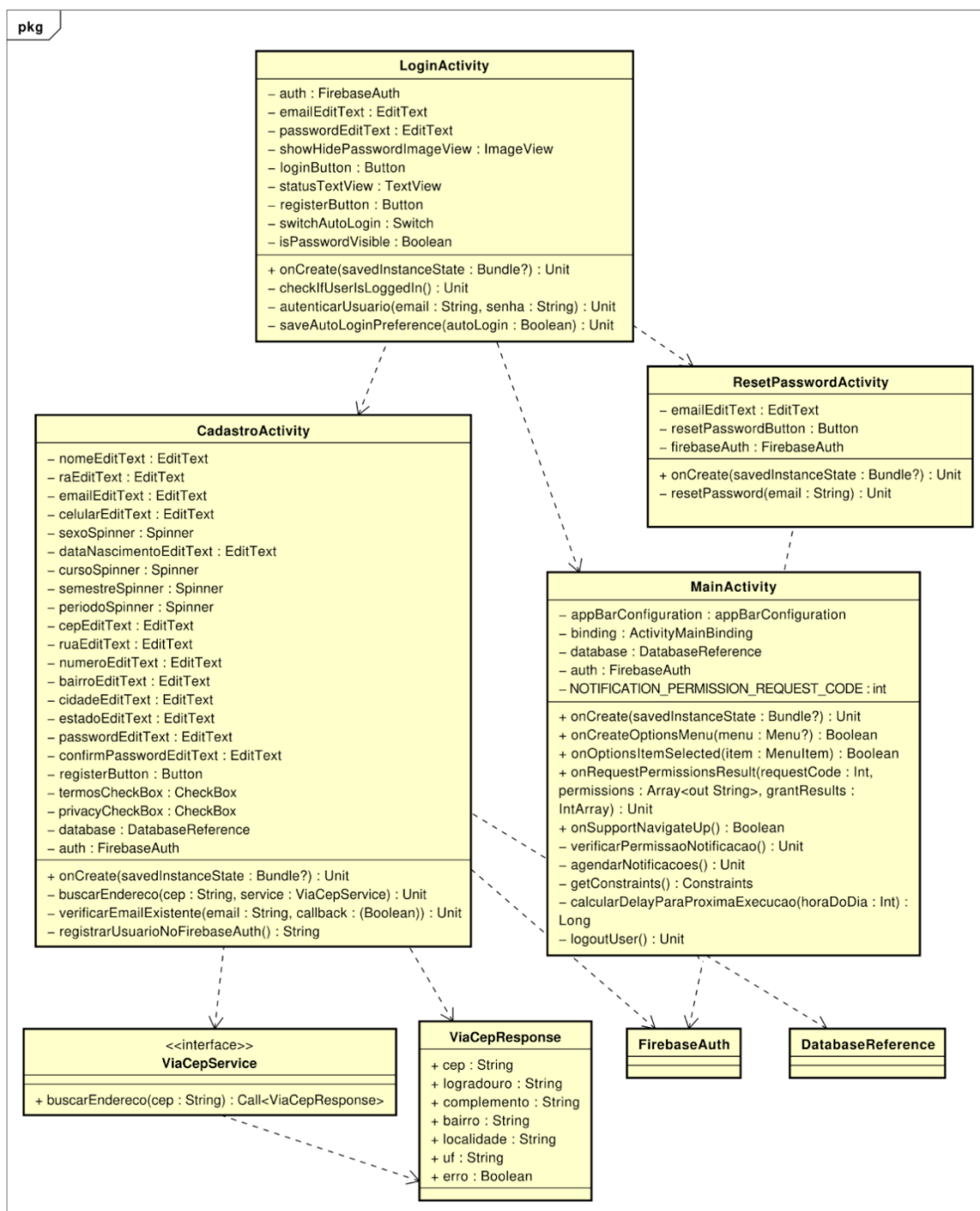
capturar tanto a estrutura estática quanto o comportamento dinâmico das classes e objetos, o que é vital para o design eficaz do sistema.

Ivar Jacobson, outro co-criador da UML, observa que "o diagrama de classes é uma ferramenta essencial para a modelagem de sistemas complexos, pois ele permite a visualização e a análise das dependências e associações entre as diferentes classes, facilitando a identificação de possíveis problemas e a melhoria do design" (JACOBSON, 1992). Jacobson destaca que o uso de diagramas de classes ajuda a assegurar que o sistema seja modular, flexível e fácil de manter.

Por fim, Alistair Cockburn comenta que "os diagramas de classes são fundamentais para a comunicação entre os membros da equipe de desenvolvimento, permitindo que todos compartilhem uma compreensão comum da estrutura do sistema e das responsabilidades de cada classe" (COCKBURN, 2000). Ele observa que a clareza proporcionada por esses diagramas é crucial para a colaboração eficaz e o desenvolvimento bem-sucedido de sistemas complexos.

A Figura 11 apresenta o diagrama de classes da funcionalidade de login do aplicativo Estudo360, destacando as classes, atributos, métodos e relacionamentos necessários para a autenticação.

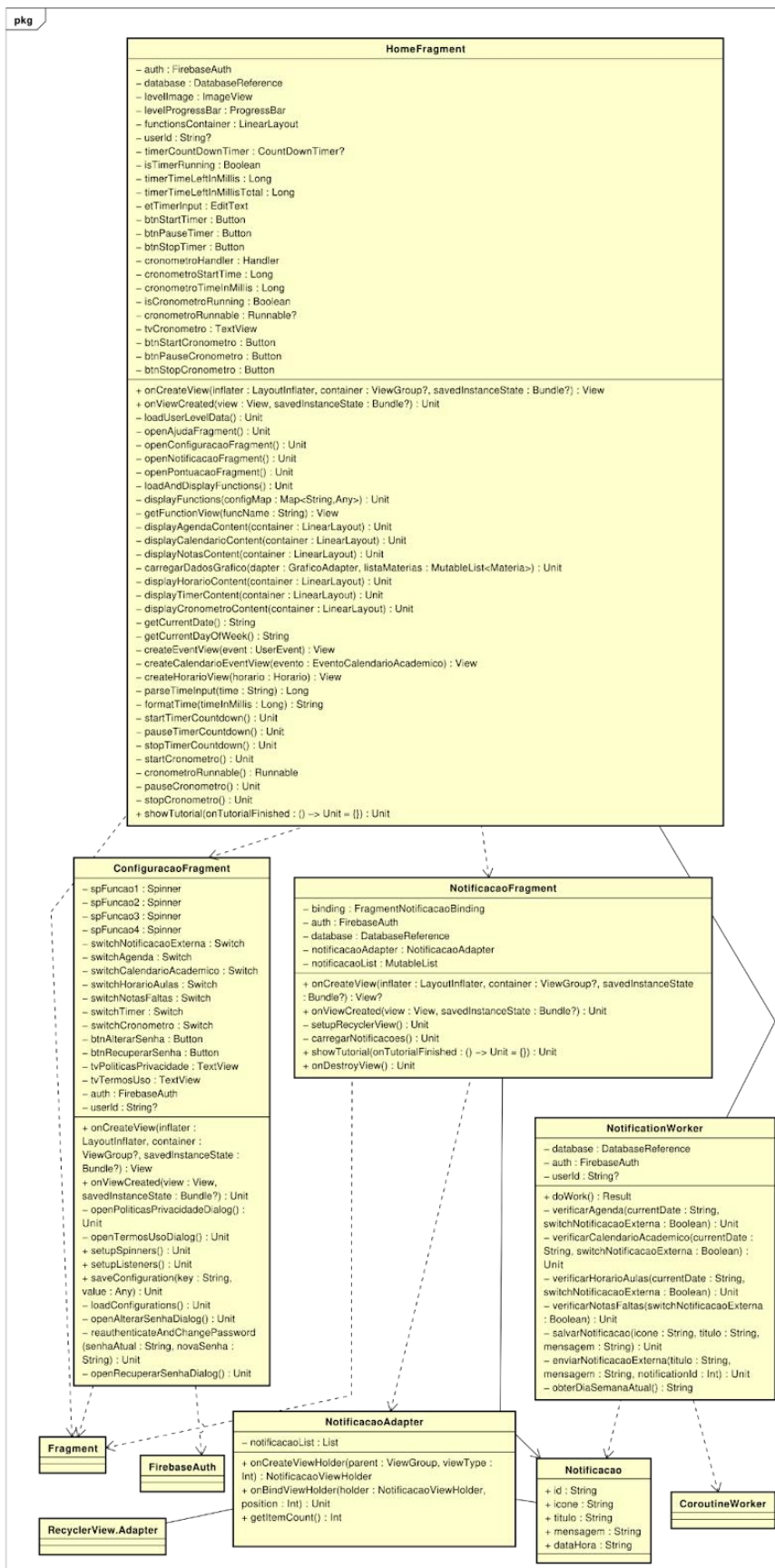
Figura 11 - Diagrama de Classes do Login.



Fonte: Elaborado pelos autores (2024).

A Figura 12 apresenta o diagrama de classes da funcionalidade Home do aplicativo Estudo360, detalhando a estrutura das classes responsáveis por exibir informações do usuário, acessar funcionalidades principais e gerenciar personalizações.

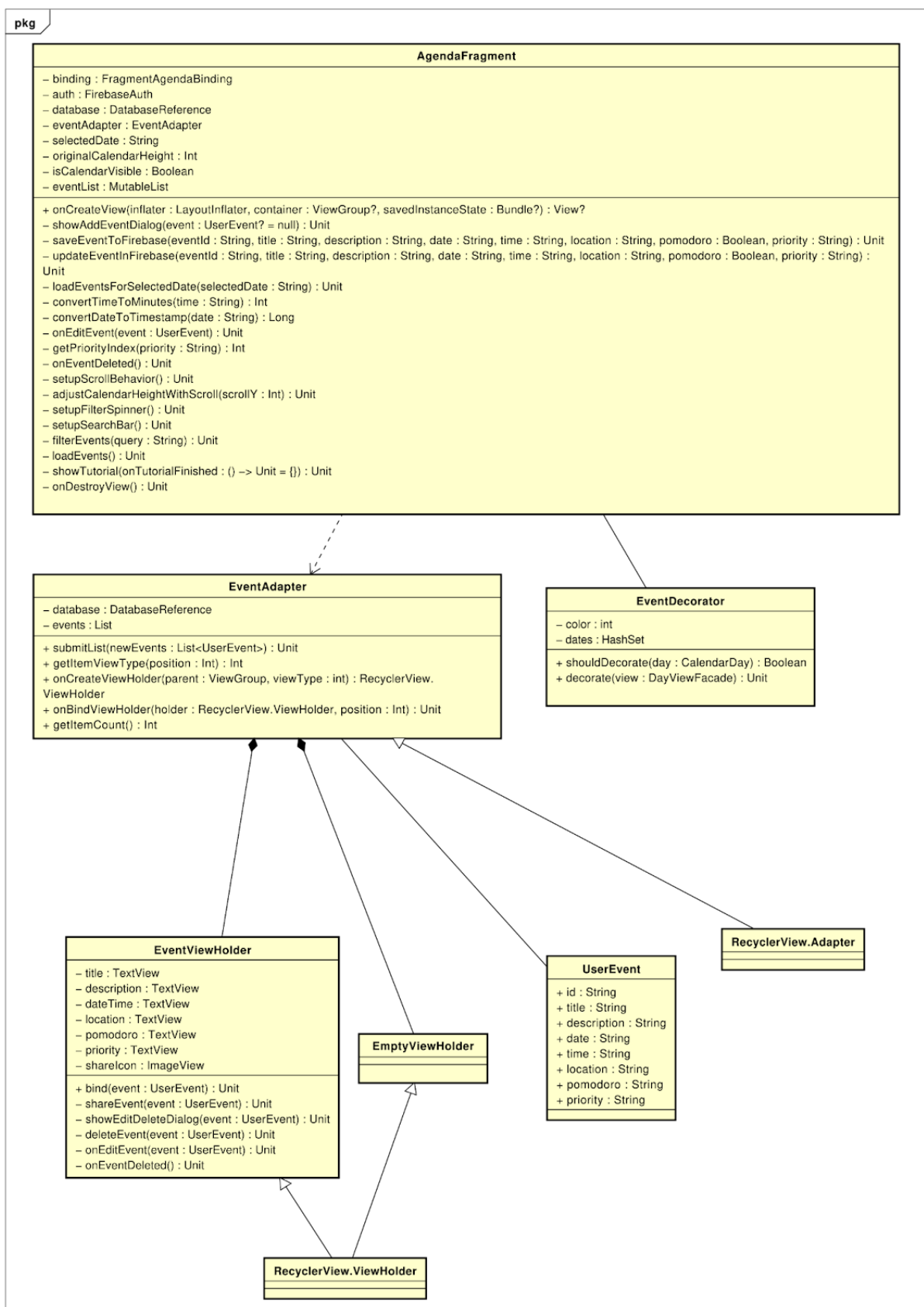
Figura 12 - Diagrama de Classes da Home.



Fonte: Elaborado pelos autores (2024)

A Figura 13 apresenta o diagrama de classes da funcionalidade de Agenda do aplicativo Estudo360, destacando as classes responsáveis pela gestão de eventos, visualização e manipulação de dados. Esse diagrama organiza as interações entre as classes, métodos e atributos necessários para o funcionamento da agenda.

Figura 13 - Diagrama de Classes da Agenda.

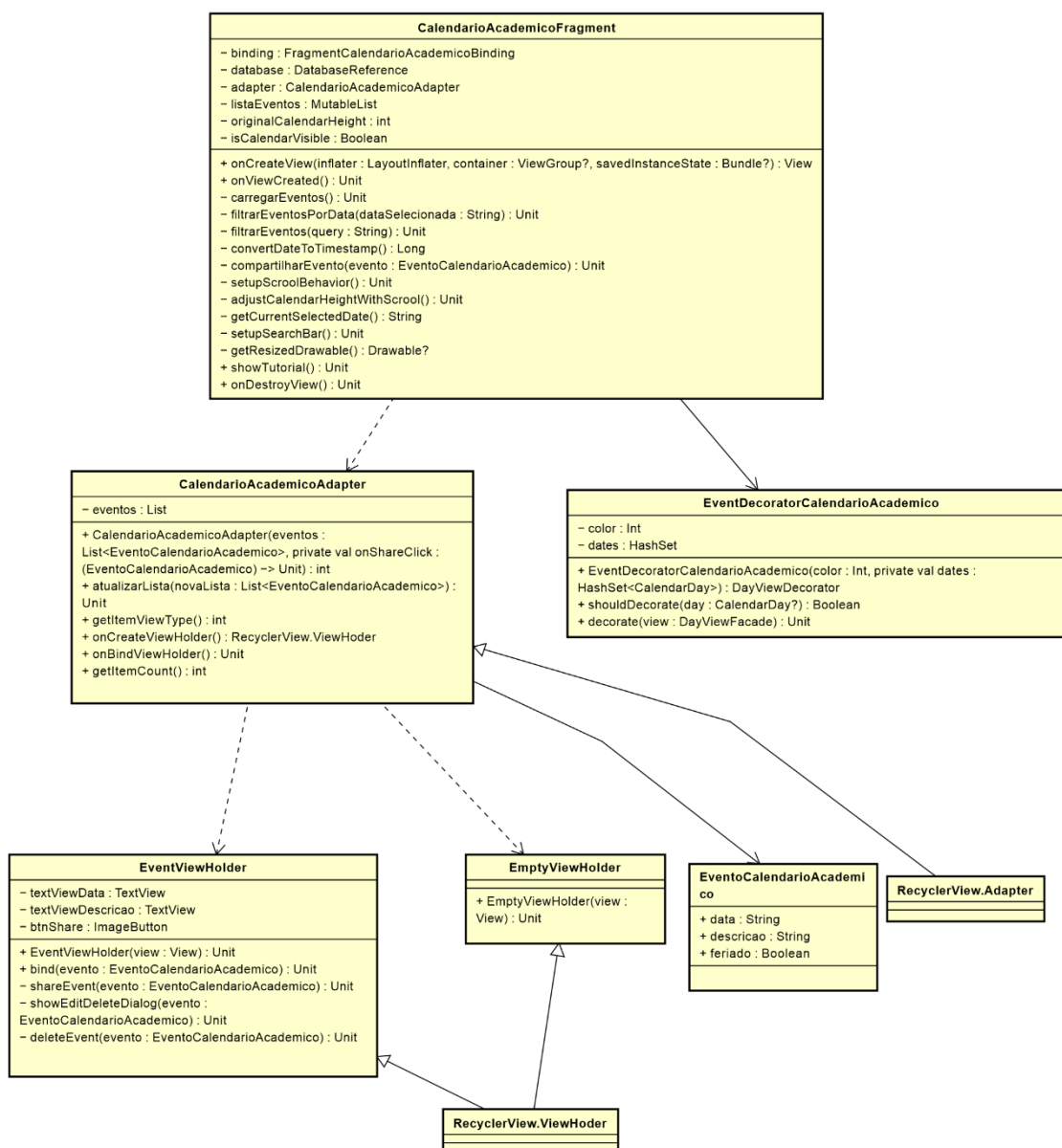


Fonte: Elaborado pelos autores (2024).

A Figura 14 apresenta o diagrama de classes da funcionalidade de Calendário Acadêmico do aplicativo Estudo360, destacando as classes que gerenciam a exibição e atualização de eventos acadêmicos. Esse diagrama detalha os atributos, métodos e relacionamentos necessários para o funcionamento dessa funcionalidade.

Figura 14 - Diagrama de Classes do Calendário Acadêmico.

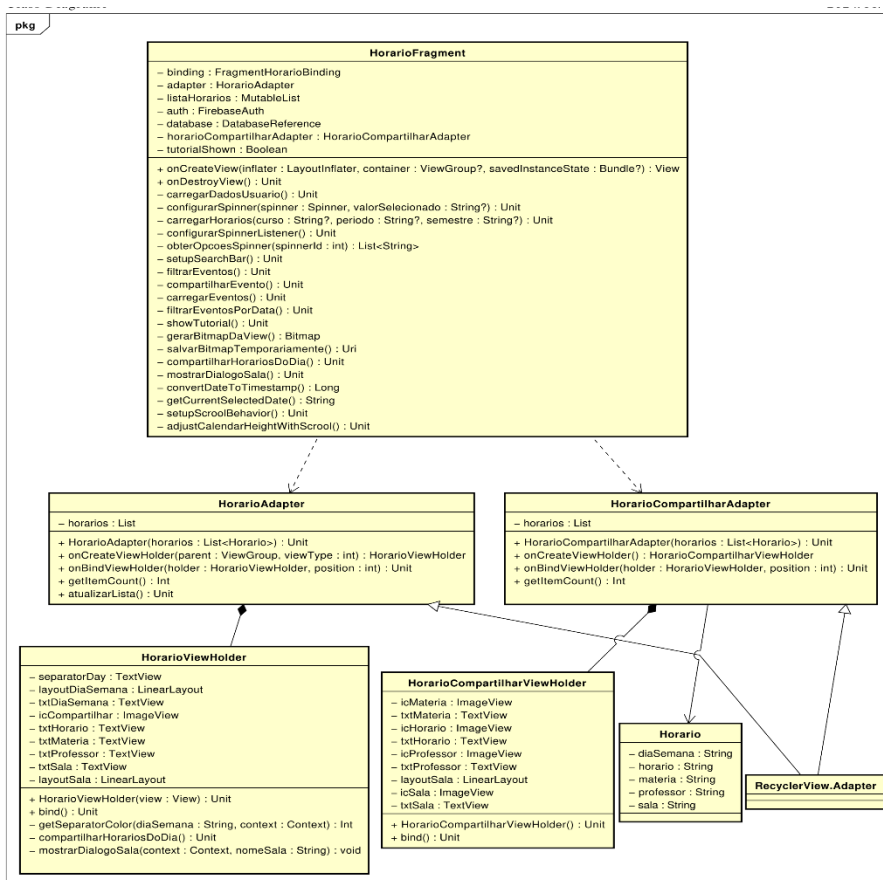
pkg



Fonte: Elaborado pelos autores (2024).

A Figura 15 apresenta o diagrama de classes da funcionalidade Horário de Aula do aplicativo Estudo360, detalhando as classes responsáveis por organizar e exibir os horários das disciplinas. Esse diagrama destaca os métodos, atributos e relacionamentos necessários para a gestão eficiente dos horários.

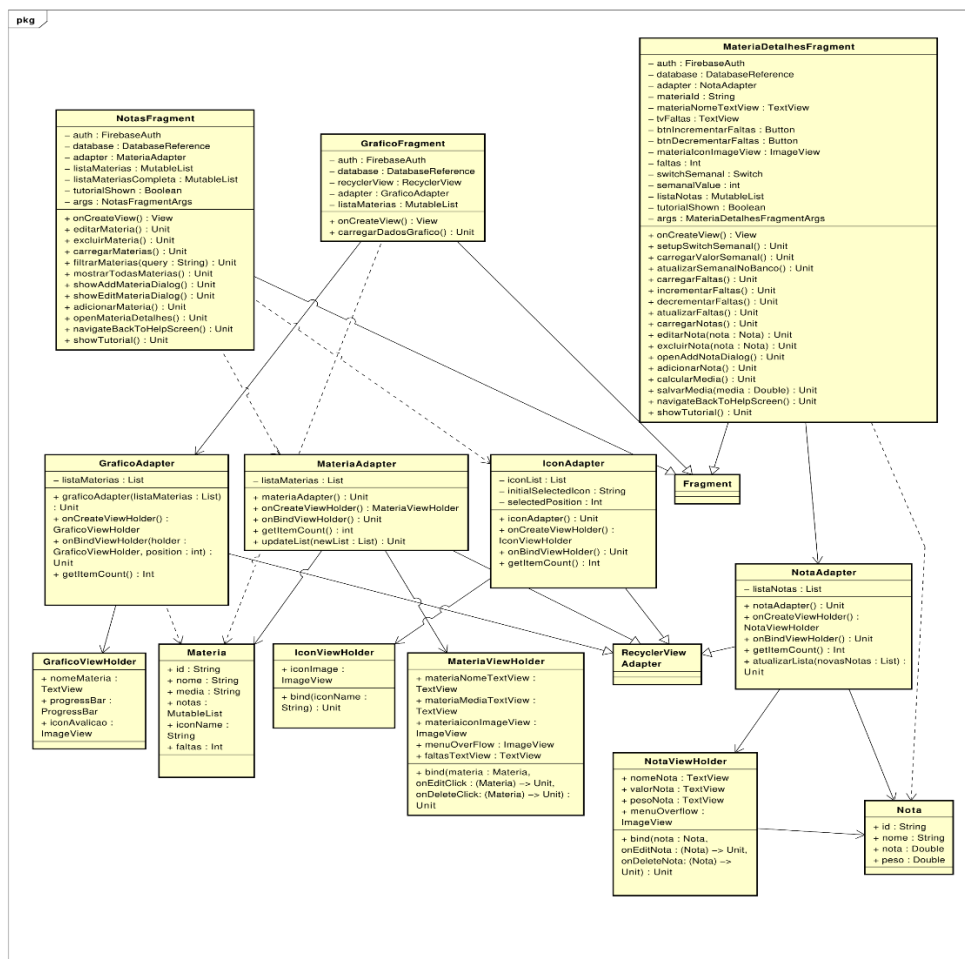
Figura 15 - Diagrama de Classes do Horário de Aula.



Fonte: Elaborado pelos autores (2024).

A Figura 16 apresenta o diagrama de classes da funcionalidade Notas e Faltas do aplicativo Estudo360, destacando as classes responsáveis pelo gerenciamento de disciplinas, registro de notas e cálculo de médias. Esse diagrama organiza os métodos, atributos e relacionamentos necessários para oferecer uma visão clara do desempenho acadêmico do usuário.

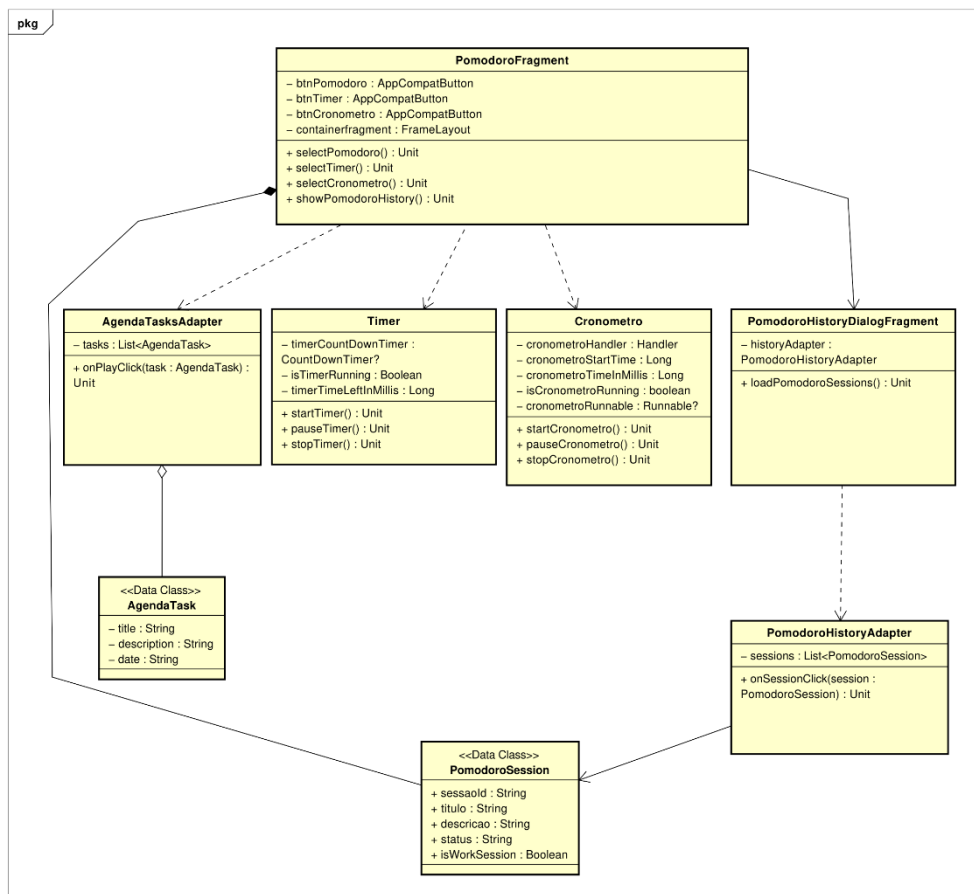
Figura 16 - Diagrama de Classes da função Notas e Faltas.



Fonte: Elaborado pelos autores (2024).

A Figura 17 apresenta o diagrama de classes da funcionalidade Pomodoro do aplicativo Estudo360, destacando as classes responsáveis pela configuração, controle de sessões de estudo e registro de produtividade. Esse diagrama detalha os métodos, atributos e relacionamentos necessários para gerenciar o temporizador e suas funcionalidades associadas.

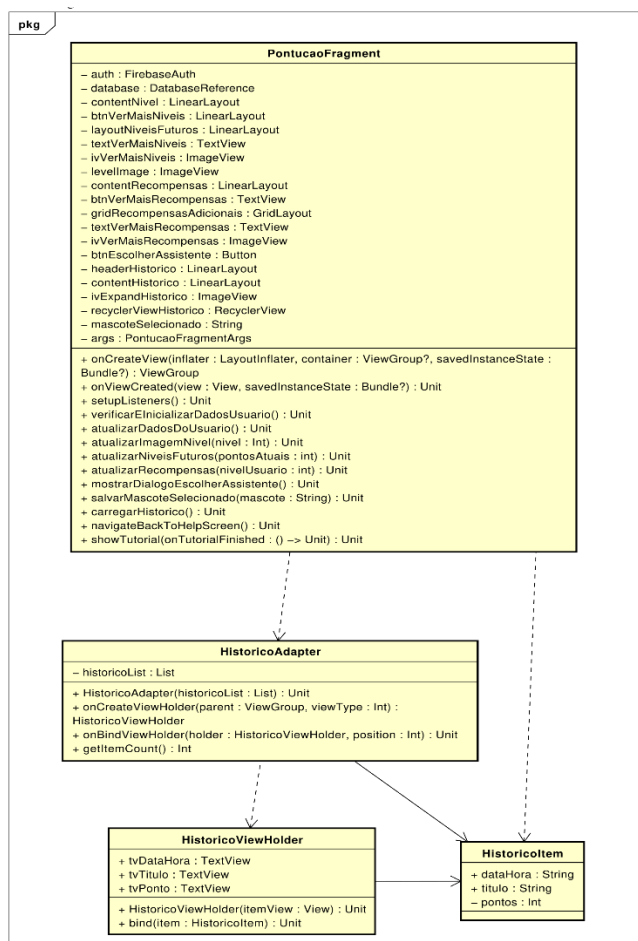
Figura 17 - Diagrama de Classes da função Pomodoro.



Fonte: Elaborado pelos autores (2024).

A Figura 18 apresenta o diagrama de classes da funcionalidade Pontuação e Recompensas do aplicativo Estudo360, detalhando as classes que gerenciam a atribuição de pontos, progressão de níveis e desbloqueio de recompensas. Esse diagrama organiza os atributos, métodos e relacionamentos necessários para incentivar o engajamento do usuário por meio de gamificação.

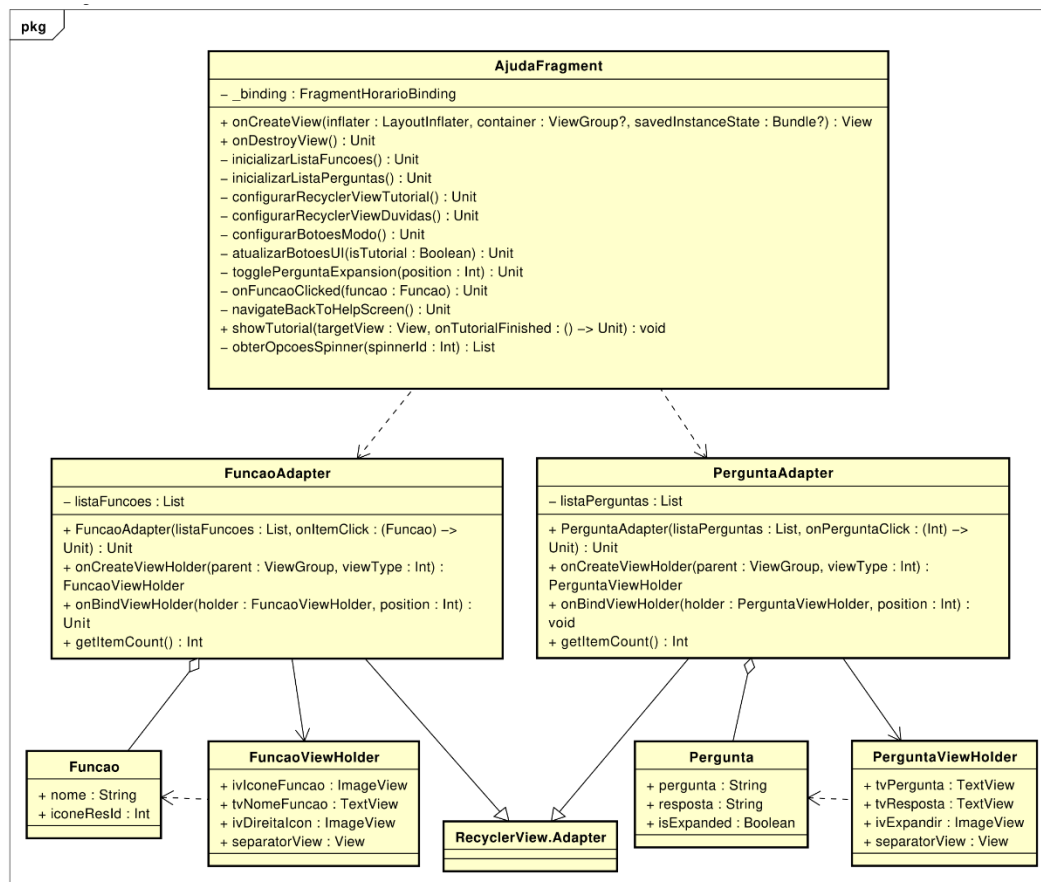
Figura 18 - Diagrama de Classes da função Pontuação e Recompensas.



Fonte: Elaborado pelos autores (2024)

A Figura 19 apresenta o diagrama de classes da funcionalidade Ajuda do aplicativo Estudo360, destacando as classes responsáveis por gerenciar tutoriais interativos e respostas às perguntas frequentes. Esse diagrama organiza os atributos, métodos e relacionamentos necessários para oferecer suporte eficiente e acessível ao usuário.

Figura 19 - Diagrama de Classes da função Ajuda.



Fonte: Elaborado pelos autores (2024)

5 BANCO DE DADOS

5.1 Bancos de Dados Não Relacionais

Os bancos de dados não relacionais, também conhecidos como NoSQL (Not Only SQL), representam uma categoria de sistemas de gerenciamento de banco de dados que diferem dos tradicionais bancos de dados relacionais em termos de estrutura, escalabilidade e flexibilidade. Diferentemente dos bancos de dados relacionais, que utilizam tabelas com esquemas rígidos e relações predefinidas, os bancos de dados NoSQL são projetados para armazenar e gerenciar grandes volumes de dados de forma distribuída, suportando variados tipos de dados e estruturas dinâmicas (CATTELL, 2011).

5.1.2 Características dos Bancos de Dados Não Relacionais

Os bancos de dados NoSQL apresentam diversas características que os tornam adequados para determinadas aplicações, especialmente aquelas que requerem alta escalabilidade, flexibilidade na modelagem de dados e desempenho otimizado para operações específicas. As principais características dos bancos de dados não relacionais incluem:

- **Modelagem Flexível de Dados:** Permitem a inserção de dados sem a necessidade de um esquema pré-definido, facilitando a adaptação a mudanças nos requisitos de dados.
- **Alta Escalabilidade:** Projetados para escalar horizontalmente, suportando a distribuição de dados em múltiplos servidores ou clusters.
- **Desempenho Otimizado:** Oferecem desempenho elevado para operações específicas, como leitura e escrita rápidas, essenciais para aplicações em tempo real.
- **Diversidade de Modelos de Dados:** Suportam diferentes modelos de dados, incluindo documentos, chave-valor, grafos e colunas largas (SADALAGE; FOWLER, 2012).

5.1.3 Firebase como Banco de Dados Não Relacional

O Firebase, uma plataforma de desenvolvimento de aplicativos móveis e web oferecida pela Google, utiliza um banco de dados não relacional denominado Firebase Realtime Database e Firestore. Esses bancos de dados são baseados em documentos e oferecem sincronização em tempo real, permitindo que os dados sejam atualizados instantaneamente em todos os dispositivos conectados (FIREBASE, 2023).

A escolha do Firebase como banco de dados para o projeto Estudo360 foi fundamentada em várias vantagens que ele oferece:

- Sincronização em Tempo Real: Atualizações de dados são refletidas instantaneamente em todos os clientes conectados, proporcionando uma experiência de usuário dinâmica e responsiva.
- Escalabilidade: Firebase é altamente escalável, suportando desde pequenas aplicações até grandes projetos com milhões de usuários.
- Facilidade de Integração: APIs amigáveis e integração simplificada com outras ferramentas e serviços da Google tornam o Firebase uma opção prática para desenvolvedores.
- Segurança: Regras de segurança flexíveis permitem o controle granular sobre o acesso aos dados, garantindo a proteção das informações dos usuários (FIREBASE, 2023).

5.2 Representação da Estrutura de Dados no Firebase

A estrutura de dados utilizada no Firebase para o aplicativo Estudo360 segue um modelo hierárquico em formato de árvore, permitindo uma organização lógica e eficiente das informações. Abaixo, apresenta-se a representação em árvore da estrutura do banco de dados:

```
Root
├── calendarioAcademico
│   ├── [0]
│   │   └── data
```

```

| | | └─ descricao
| | └─ feriado
| └─ ...
└─ horario
| └─ <Curso>
| | └─ Manhã
| | | └─ [1]
| | | | └─ diaSemana
| | | | └─ horario
| | | | └─ materia
| | | | └─ professor
| | | | └─ sala
| | | └─ ...
| | └─ Tarde
| | └─ Noite
| | └─ <Semestre>
| | | └─ [1]
| | | | └─ diaSemana
| | | | └─ horario
| | | | └─ materia
| | | | └─ professor
| | | | └─ sala
| | | └─ ...
| └─ ...
└─ usuarios
    └─ <ID do Usuário>
        └─ nome

```

```
| |— data_nascimento
| |— sexo
| |— email
| |— celular
| |— bairro
| |— rua
| |— numero
| |— cidade
| |— estado
| |— cep
| |— curso
| |— periodo
| |— semestre
| |— ra
| |— configuracao
| | |— Agenda
| | |— CalendarioAcademico
| | |— ...
| |— events
| | |— <ID do Evento>
| | | |— date
| | | |— time
| | | |— title
| | | |— description
| | | |— location
| | | |— pomodoro
| | | |— priority
```

```

| | └─ ...
| └─ materias
| | └─ <ID da Disciplina>
| | | └─ id
| | | └─ nome
| | | └─ iconName
| | | └─ media
| | | └─ faltas
| | | └─ notas
| | |   └─ <ID da Nota>
| | |     | └─ id
| | |     | └─ nome
| | |     | └─ nota
| | |     | └─ peso
| | |     └─ ...
| | └─ ...
| └─ level
| | └─ nivel
| | └─ pontos
| | └─ mascote
| | └─ historico
| |   └─ <ID do Registro>
| |     | └─ Titulo
| |     | └─ dataHora
| |     | └─ pontos
| |     └─ ...
| └─ pomodoro

```

```

| | | └─ <ID da Sessão>
| | | └─ sessaold
| | | └─ titulo
| | | └─ descricao
| | | └─ dataHora
| | | └─ status
| | | └─ cicloAtual
| | | └─ numeroCiclosEstudo
| | | └─ numeroCiclosDescanso
| | | └─ tempoTotalEstudo
| | | └─ pomodoroTimeLeftInMillis
| | | └─ isWorkSession
| | | └─ workSession
| | └─ ...
| └─ pomodoroHistory
| | └─ <ID da Sessão Histórica>
| | | └─ Mesmos campos de uma sessão de pomodoro
| | └─ ...
| └─ notificacoes
| | └─ <ID da Notificação>
| | | └─ id
| | | └─ titulo
| | | └─ mensagem
| | | └─ dataHora
| | | └─ icone
| | └─ ...
| └─ ...

```

└─ ...

5.2.1 Representação da Estrutura de Dados

A representação em árvore do banco de dados Firebase permite uma organização hierárquica eficiente, facilitando o acesso e a manipulação dos dados. Cada nó da árvore representa uma coleção ou documento, onde as coleções armazenam múltiplos documentos que, por sua vez, contêm campos e subcoleções.

- Root: É o ponto de partida da árvore, contendo todas as coleções principais do banco de dados.
- calendarioAcademico: Coleção que armazena os eventos acadêmicos, como datas importantes e feriados.
 - [0], [1], ...: Documentos individuais representando cada evento, contendo campos como data, descricao e feriado.
- horario: Coleção que organiza os horários das aulas por curso e período (Manhã, Tarde, Noite).
 - <Curso>: Documentos que representam cada curso, contendo subcoleções para diferentes períodos.
 - Manhã, Tarde, Noite: Subcoleções que armazenam os horários das aulas em cada período.
 - [1], [2], ...: Documentos individuais com informações sobre o dia da semana, horário, matéria, professor e sala.
- usuarios: Coleção que armazena os dados dos usuários do aplicativo.
 - <ID do Usuário>: Documentos únicos para cada usuário, contendo informações pessoais e acadêmicas.
 - configuracao: Subcoleção com preferências do usuário, como configurações de agenda e calendário acadêmico.
 - events: Subcoleção que registra eventos personalizados do usuário.
 - materias: Subcoleção que gerencia as disciplinas do usuário, incluindo notas e faltas.
 - level: Subcoleção que monitora o nível, pontos, mascote e histórico de pontuação do usuário.

- pomodoro: Subcoleção que armazena sessões de Pomodoro realizadas pelo usuário.
- pomodoroHistory: Subcoleção que mantém um histórico das sessões de Pomodoro.
- notificacoes: Subcoleção que gerencia as notificações enviadas ao usuário.

Vantagens da Representação em Árvore no Firebase

A estrutura hierárquica em árvore oferecida pelo Firebase apresenta diversas vantagens significativas para o gerenciamento de dados. Primeiramente, proporciona facilidade de navegação, permitindo acesso rápido e direto às informações por meio de caminhos específicos. Além disso, destaca-se pela escalabilidade, uma vez que facilita a expansão do banco de dados conforme novas funcionalidades são incorporadas ao sistema.

Outro benefício é o desempenho, que é otimizado durante consultas de dados, especialmente quando índices apropriados são implementados. Por fim, a representação em árvore assegura uma organização lógica, mantendo os dados bem-estruturados e alinhados à lógica do aplicativo, o que contribui para uma gestão eficiente das informações.

5.3 Dicionário de Dados

O Dicionário de Dados é uma ferramenta fundamental no desenvolvimento e manutenção de sistemas de informação, fornecendo uma descrição detalhada dos dados utilizados no sistema. Ele documenta cada elemento de dado, incluindo seu nome, tipo, formato, origem, restrições e relacionamento com outros dados. O Dicionário de Dados serve como uma referência central para desenvolvedores, analistas e outros stakeholders, assegurando uma compreensão compartilhada e precisa dos dados do sistema.

Ian Sommerville destaca a importância do Dicionário de Dados ao afirmar que "um dicionário de dados é essencial para manter a integridade e a consistência dos dados em um sistema, pois ele fornece uma definição única e precisa para cada elemento de dado" (SOMMERVILLE, 2011). Ele observa que a utilização de um

Dicionário de Dados ajuda a evitar ambiguidades e erros que podem surgir da interpretação inconsistente dos dados.

Barbara von Halle, uma especialista em modelagem de dados, ressalta que "o Dicionário de Dados desempenha um papel crítico na documentação de sistemas de informação, facilitando a comunicação entre os membros da equipe de desenvolvimento e assegurando que todos entendam exatamente como os dados devem ser utilizados e manipulados" (VON HALLE, 2002). Ela enfatiza que um Dicionário de Dados bem estruturado é fundamental para a qualidade e a eficácia do desenvolvimento de software.

Grady Booch, um dos criadores da UML, também reconhece a importância do Dicionário de Dados, afirmando que "o Dicionário de Dados é uma ferramenta valiosa para a modelagem de sistemas, pois ele detalha a estrutura dos dados em um nível granular, permitindo uma melhor análise e design do sistema" (BOOCH et al., 2005). Booch observa que o Dicionário de Dados ajuda a garantir que os desenvolvedores tenham uma compreensão clara dos requisitos de dados e das especificações.

Alistair Cockburn, em seu trabalho sobre design de sistemas, adiciona que "um Dicionário de Dados completo e bem mantido é essencial para a consistência e a precisão dos dados no sistema, além de servir como uma referência útil durante todo o ciclo de vida do desenvolvimento de software" (COCKBURN, 2000). Cockburn destaca que o Dicionário de Dados facilita a manutenção do sistema, permitindo uma rápida identificação e correção de problemas relacionados aos dados.

A Tabela 21 descreve o dicionário de dados do Calendário Acadêmico, incluindo informações sobre eventos e feriados.

Tabela 21 - Dicionário de Dados - Calendário Acadêmico.

Campo	Tipo de Dados	Descrição
data	String	Data do evento no formato "DD/MM/AAAA".
descricao	String	Descrição detalhada do evento acadêmico.
feriado	Boolean	Indica se o dia é um feriado (true) ou não (false).

Fonte: Elaborado pelos autores (2024).

A Tabela 22 apresenta o dicionário de dados da função Horário das Aulas, contendo informações sobre dias, horários e detalhes das aulas.

Tabela 22 - Dicionário de Dados - Horário das Aulas.

Campo	Tipo de Dados	Descrição
diaSemana	String	Dia da semana (e.g., "Segunda", "Terça").
horario	String	Horário da aula (e.g., "07:40/08:30 - 08:30/09:20").
materia	String	Nome da disciplina.
professor	String	Nome do professor responsável.
sala	String	Identificação da sala de aula.

Fonte: Elaborado pelos autores (2024).

A Tabela 23 apresenta o dicionário de dados das Informações Pessoais e Acadêmicas, abrangendo dados de contato, endereço e detalhes acadêmicos do usuário.

Tabela 23 - Dicionário de Dados - Informações Pessoais e Acadêmicas.

Campo	Tipo de Dados	Descrição
nome	String	Nome completo do usuário.
data_nascimento	String	Data de nascimento no formato "DD/MM/AAAA".
sexo	String	Gênero do usuário (e.g., "Masculino", "Feminino").
email	String	Endereço de email do usuário.
celular	String	Número de telefone celular.
bairro	String	Bairro de residência.
rua	String	Nome da rua.
numero	String	Número da residência.
cidade	String	Cidade de residência.
estado	String	Estado de residência.
cep	String	Código postal.

Campo	Tipo de Dados	Descrição
curso	String	Nome do curso matriculado.
periodo	String	Período de estudo (e.g., "Manhã", "Tarde", "Noite").
semestre	String	Semestre atual do curso.
ra	String	Registro Acadêmico do aluno.

Fonte: Elaborado pelos autores (2024).

A Tabela 24 apresenta o dicionário de dados das Configurações, detalhando opções de ativação e personalização de funcionalidades do sistema.

Tabela 24 - Dicionário de Dados - Configurações.

Campo	Tipo de Dados	Descrição
Agenda	Boolean	Ativa ou desativa a funcionalidade de agenda.
CalendarioAcademico	Boolean	Ativa ou desativa o calendário acadêmico.
HorarioAulas	Boolean	Ativa ou desativa o horário das aulas.
NotasFaltas	Boolean	Ativa ou desativa a visualização de notas e faltas.
Pomodoro	Boolean	Ativa ou desativa a funcionalidade de Pomodoro.
Pontuacao	Boolean	Ativa ou desativa o sistema de pontuação.
Cronometro	Boolean	Ativa ou desativa a funcionalidade de cronômetro.
Timer	Boolean	Ativa ou desativa a funcionalidade de timer.
NotificacaoExterna	Boolean	Ativa ou desativa notificações externas.
NotificacaoInterna	Boolean	Ativa ou desativa notificações internas.
Funcao1	String	Personalização da função principal 1.
Funcao2	String	Personalização da função principal 2.
Funcao3	String	Personalização da função principal 3.
Funcao4	String	Personalização da função principal 4.

Fonte: Elaborado pelos autores (2024).

A Tabela 25 apresenta o dicionário de dados dos Eventos, incluindo informações como data, horário, título, descrição, local e prioridade.

Tabela 25 - Dicionário de Dados - Eventos (events).

Campo	Tipo de Dados	Descrição
id	String	Identificador único do evento.
date	String	Data do evento no formato "DD/MM/AAAA".
time	String	Hora do evento (e.g., "14:30").
title	String	Título do evento.
description	String	Descrição detalhada do evento.
location	String	Local onde ocorrerá o evento.
pomodoro	String	Indica se o evento está associado a uma sessão de Pomodoro ("true" ou "false").
priority	String	Nível de prioridade do evento (e.g., "Alta", "Normal", "Baixa").

Fonte: Elaborado pelos autores (2024).

A Tabela 26 apresenta o dicionário de dados dos Materiais, contendo informações sobre disciplinas, como nome, ícone, médias, faltas e carga horária semanal.

Tabela 26 - Dicionário de Dados - Materiais (materias).

Campo	Tipo de Dados	Descrição
id	String	Identificador único da disciplina.
nome	String	Nome da disciplina.
iconName	String	Nome do ícone associado à disciplina.
media	String	Média das notas na disciplina.
faltas	Integer	Número de faltas acumuladas.

Campo	Tipo de Dados	Descrição
semanal	Integer	Número de aulas semanais da disciplina (se aplicável).

Fonte: Elaborado pelos autores (2024)

A Tabela 27 apresenta o dicionário de dados das Notas, com informações sobre avaliações, como descrição, nota obtida e peso na média final.

Tabela 27 - Dicionário de Dados - Notas (notas).

Campo	Tipo de Dados	Descrição
id	String	Identificador único da nota.
nome	String	Nome ou descrição da avaliação (e.g., "P1").
nota	Float	Nota obtida na avaliação.
peso	Float	Peso da avaliação na composição da média final.

Fonte: Elaborado pelos autores (2024)

A Tabela 28 apresenta o dicionário de dados do Nível e Pontuação, contendo informações sobre o nível, pontos acumulados e mascote associado ao usuário.

Tabela 28 - Dicionário de Dados - Nível e Pontuação (level).

Campo	Tipo de Dados	Descrição
nivel	Integer	Nível atual do usuário.
pontos	Integer	Pontos acumulados pelo usuário.
mascote	String	Nível ou tipo de mascote associado ao usuário.

Fonte: Elaborado pelos autores (2024).

A Tabela 29 apresenta o dicionário de dados do Histórico, com informações sobre o título, data e hora das atividades realizadas, além dos pontos obtidos.

Tabela 29 - Dicionário de Dados - Histórico (historico).

Campo	Tipo de Dados	Descrição
Título	String	Título da atividade realizada.
dataHora	String	Data e hora da atividade (e.g., "DD/MM/AAAA HH").

Campo	Tipo de Dados	Descrição
pontos	Integer	Pontos obtidos na atividade.

Fonte: Elaborado pelos autores (2024).

A Tabela 30 apresenta o dicionário de dados do Pomodoro Ativo, detalhando informações sobre a sessão, como título, status, ciclos de estudo e descanso, tempo restante e tipo de sessão.

Tabela 30 - Dicionário de Dados - Pomodoro Ativo (pomodoro).

Campo	Tipo de Dados	Descrição
sessaoid	String	Identificador único da sessão de Pomodoro.
titulo	String	Título da sessão de Pomodoro.
descricao	String	Descrição da sessão.
dataHora	String	Data e hora de início (e.g., "DD/MM/AAAA HH").
status	String	Status atual da sessão (e.g., "Ativo", "Pausado", "Concluído").
cicloAtual	Integer	Número do ciclo atual.
numeroCiclosEstudo	Integer	Número total de ciclos de estudo planejados.
numeroCiclosDescanso	Integer	Número total de ciclos de descanso planejados.
tempoTotalEstudo	Integer	Tempo total de estudo planejado em minutos.
pomodoroTimeLeftInMillis	Integer	Tempo restante da sessão em milissegundos.
isWorkSession	Boolean	Indica se a sessão atual é de trabalho (true) ou descanso (false).
workSession	Boolean	Indica se é uma sessão de trabalho.

Fonte: Elaborado pelos autores (2024).

A Tabela 31 apresenta o dicionário de dados das Notificações, com informações sobre o identificador, título, mensagem, data e hora de envio, e ícone associado.

Tabela 31 - Dicionário de Dados - Notificações (notificacoes).

Campo	Tipo de Dados	Descrição
id	String	Identificador único da notificação.
titulo	String	Título da notificação.
mensagem	String	Conteúdo da notificação.
dataHora	String	Data e hora de envio (e.g., "DD/MM/AAAA HH").
icone	String	Nome do ícone associado à notificação.

Fonte: Elaborado pelos autores (2024).

A Tabela 32 apresenta o dicionário de dados das Configurações Gerais, detalhando as opções para ativar ou desativar notificações e funcionalidades como cronômetro e timer.

Tabela 32 - Dicionário de Dados - Configurações Gerais (configuracao).

Campo	Tipo de Dados	Descrição
NotificacaoExterna	Boolean	Ativa ou desativa notificações externas.
NotificacaoInterna	Boolean	Ativa ou desativa notificações internas.
Cronometro	Boolean	Ativa ou desativa a funcionalidade de cronômetro.
Timer	Boolean	Ativa ou desativa a funcionalidade de timer.

Fonte: Elaborado pelos autores (2024).

6 DESENVOLVIMENTO

As metodologias de desenvolvimento de software são abordagens estruturadas que organizam e facilitam o processo de criação de sistemas de software. Elas fornecem diretrizes, práticas e princípios para a execução de projetos, ajudando as equipes a alcançarem seus objetivos de maneira eficiente e eficaz. Existem várias metodologias de desenvolvimento, cada uma com suas características, vantagens e desvantagens, adequadas a diferentes tipos de projetos e equipes.

Segundo Ian Sommerville, "as metodologias de desenvolvimento de software são essenciais para organizar o trabalho de desenvolvimento, minimizar riscos e assegurar a entrega de software de alta qualidade" (SOMMERVILLE, 2011). Ele destaca que a escolha da metodologia adequada pode influenciar significativamente o sucesso de um projeto.

6.1 Metodologias Ágeis

As metodologias ágeis, como o Scrum, surgiram como uma resposta às limitações do modelo cascata, especialmente em projetos onde os requisitos podem mudar rapidamente. Essas metodologias enfatizam a colaboração, a flexibilidade e a entrega contínua de pequenas partes funcionais do software. Alistair Cockburn, um dos signatários do Manifesto Ágil, afirma que "as metodologias ágeis promovem um desenvolvimento adaptativo e iterativo, permitindo às equipes responder rapidamente a mudanças" (COCKBURN, 2001).

6.1.2 Metodologia Scrum

O Scrum é uma metodologia ágil de gerenciamento de projetos de software, criada para melhorar a capacidade de resposta a mudanças e aumentar a eficiência das equipes de desenvolvimento. Ele é baseado em princípios de flexibilidade, colaboração e entrega contínua, permitindo que as equipes desenvolvam produtos incrementais de alta qualidade.

O Scrum é estruturado em ciclos curtos e repetitivos chamados sprints, que geralmente duram de duas a quatro semanas. Cada sprint resulta em um incremento do produto, que é uma versão funcional e potencialmente utilizável do software. Isso permite que as equipes entreguem valor ao cliente de forma contínua e regular, ajustando o trabalho conforme necessário.

Ken Schwaber e Jeff Sutherland, os criadores do Scrum, definem a metodologia como "uma estrutura que permite às equipes abordar problemas complexos adaptativos, enquanto entregam produtos de valor mais elevado de forma produtiva e criativa" (SCHWABER; SUTHERLAND, 2017).

6.1.3 Papéis no Scrum

Product Owner é responsável por maximizar o valor do produto e gerenciar o backlog do produto. O Product Owner prioriza as tarefas e garante que a equipe esteja trabalhando nas funcionalidades mais importantes.

Scrum Master: Facilita o processo Scrum e ajuda a remover impedimentos que possam atrapalhar a equipe. O Scrum Master garante que a equipe siga as práticas e princípios do Scrum.

Equipe de Desenvolvimento é composto por profissionais que realizam o trabalho de entregar incrementos do produto a cada sprint. A equipe de desenvolvimento é auto-organizada e multifuncional, composta por desenvolvedores, designers, testadores, entre outros.

6.1.4 Etapas de Desenvolvimento

- Data de Início: 1º de março de 2024
- Data de Término: 18 de novembro de 2024
- Total de Sprints: 8 sprints
- Duração de cada Sprint: Aproximadamente 4 semanas
- Período Total do Projeto: Cerca de 8 meses

Equipe do Projeto

- Victor: Product Owner e Desenvolvedor

- Andressa: Scrum Master e Desenvolvedora

A Tabela 33 apresenta o plano de sprints do projeto, detalhando os períodos, objetivos, principais tarefas, responsáveis e entregáveis para cada etapa. O planejamento abrange desde a definição inicial do backlog até a entrega final do projeto e sua documentação.

Tabela 33 - Plano de Sprints do Projeto.

Sprint	Período	Objetivo	Principais Tarefas	Responsáveis	Entregáveis
Sprint 1	1º de março a 31 de março de 2024	Preparar o projeto, levantar requisitos e definir o backlog do produto.	<ul style="list-style-type: none"> - Definir a visão do produto e objetivos gerais - Levantamento de requisitos funcionais - Levantamento de requisitos não funcionais - Análise de softwares similares - Criação do backlog - Reuniões semanais de alinhamento 	Victor (Product Owner) e Andressa (Scrum Master)	<ul style="list-style-type: none"> - Documento de Requisitos Funcionais - Documento de Requisitos Não Funcionais - Análise de Softwares Similares com tabela de comparação - Backlog inicial do produto priorizado
Sprint 2	1º de abril a 30 de abril de 2024	Criar e documentar os casos de uso, iniciar a modelagem do sistema.	<ul style="list-style-type: none"> - Criação dos casos de uso - Elaboração do diagrama de casos de uso - Documentação textual dos casos de uso - Início do diagrama de classes - Revisão e ajustes 	Victor e Andressa	<ul style="list-style-type: none"> - Diagrama de Casos de Uso - Documentação Textual dos Casos de Uso - Esboço inicial do Diagrama de Classes
Sprint 3	1º de maio a 31 de maio de 2024	Finalizar o diagrama de classes e modelar o banco de dados.	<ul style="list-style-type: none"> - Desenvolvimento do diagrama de classes - Criação do diagrama Entidade-Relacionamento (ER) - Revisão cruzada 	Victor e Andressa	<ul style="list-style-type: none"> - Diagrama de Classes finalizado - Diagrama Entidade-Relacionamento (ER) finalizado - Backlog atualizado

			<ul style="list-style-type: none"> - Ajustes conforme feedback - Atualização do backlog 		
Sprint 4	1º de junho a 30 de junho de 2024	Projetar as interfaces de usuário e criar protótipos navegáveis.	<ul style="list-style-type: none"> - Design das interfaces de usuário - Criação de protótipos navegáveis - Testes de usabilidade dos protótipos - Revisão e ajustes dos protótipos - Preparação para o desenvolvimento 	Victor e Andressa	<ul style="list-style-type: none"> - Protótipos de alta fidelidade das interfaces no Figma - Relatório de testes de usabilidade - Guia de estilo e padrões de design
Sprint 5	1º de julho a 31 de julho de 2024	Configurar o ambiente de desenvolvimento e iniciar a programação dos módulos principais.	<ul style="list-style-type: none"> - Configuração do ambiente de desenvolvimento - Desenvolvimento do módulo de autenticação - Desenvolvimento do módulo de agenda e calendário - Implementação da estrutura de banco de dados - Testes unitários iniciais 	Victor e Andressa	<ul style="list-style-type: none"> - Ambiente de desenvolvimento configurado - Módulos iniciais programados e testados - Repositório de código atualizado
Sprint 6	1º de agosto a 31 de agosto de 2024	Desenvolver as funcionalidades principais do aplicativo, como Método Pomodoro e gamificação.	<ul style="list-style-type: none"> - Desenvolvimento do módulo Método Pomodoro - Implementação do sistema de gamificação - Desenvolvimento do módulo de notificações - Integração entre módulos - Testes funcionais 	Victor e Andressa	<ul style="list-style-type: none"> - Módulo Método Pomodoro funcional - Sistema de gamificação implementado - Notificações configuradas - Aplicativo integrado e testado
Sprint 7	1º de setembro a 31 de outubro de 2024	Desenvolver funcionalidades adicionais e refinar o aplicativo.	<ul style="list-style-type: none"> - Desenvolvimento do módulo de anotações pessoais - Desenvolvimento do módulo de horário das aulas 	Victor e Andressa	<ul style="list-style-type: none"> - Funcionalidades adicionais implementadas - Interface refinada

			<ul style="list-style-type: none"> - Melhorias na interface do usuário - Implementação do calendário acadêmico - Correções de bugs e otimizações - Testes de integração e qualidade 		<ul style="list-style-type: none"> - Aplicativo otimizado e estável - Relatório de testes de integração
Sprint 8	1º de novembro a 18 de novembro de 2024	Realizar testes finais, finalizar a documentação e preparar a apresentação do projeto.	<ul style="list-style-type: none"> - Testes beta com usuários reais - Coleta e análise de feedback - Implementação de melhorias finais - Documentação técnica e do usuário - Preparação da apresentação final - Revisão final do trabalho acadêmico - Entrega do projeto 	Victor e Andressa	<ul style="list-style-type: none"> - Aplicativo finalizado e testado - Documentação completa (Técnica e do Usuário) - Apresentação preparada - Trabalho de Conclusão de Curso (TCC) finalizado

Fonte: Elaborado pelos autores (2024).

7 PROTÓTIPO

7.1 Design da Interface

O design de interface de um aplicativo é essencial para proporcionar uma experiência de usuário eficiente e intuitiva. No Estudo360, a prioridade foi criar uma interface clara e funcional, permitindo que os usuários acessem rapidamente as principais funcionalidades do aplicativo. A consistência visual e a usabilidade foram elementos centrais no desenvolvimento, com foco em facilitar a navegação e otimizar a organização das informações. Como afirmam Lidwell, Holden e Butler, "a consistência no design melhora a experiência do usuário ao criar uma sensação de familiaridade, reduzindo o esforço cognitivo necessário para aprender novas interações" (2019, p. 85). Esta seção descreve as escolhas de navegação, layout, design visual e elementos recorrentes, fundamentados em teorias de design e usabilidade.

7.1.1 Escolhas de Navegação

No desenvolvimento do Estudo360, foi fundamental garantir que a navegação fosse intuitiva e rápida. De acordo com Norman, "a navegação eficiente é um dos principais fatores que impactam diretamente a experiência do usuário, especialmente em interfaces digitais" (2018, p. 45). Com isso, optou-se por uma barra de navegação fixa na parte inferior do aplicativo, permitindo que o usuário acesse diretamente as principais funcionalidades, como Calendário, Tarefas, Material Didático e Configurações. Essa abordagem é amplamente utilizada em aplicativos como Instagram e Twitter, que utilizam barras de navegação fixa para manter uma navegação constante e fluida (JOHNSON, 2010).

Além disso, essa escolha permite que os usuários acessem funcionalidades sem precisar navegar por menus ocultos, como o menu hambúrguer, que pode sobrecarregar o usuário com opções não visíveis de imediato. Segundo Krug, "os usuários não gostam de pensar mais do que o necessário; eles preferem caminhos claros e ações rápidas" (2000, p. 19). Nesse sentido, a barra de navegação fixa facilita o acesso direto às funções mais importantes, otimizando o tempo de navegação e aumentando a produtividade dos estudantes, que geralmente possuem agendas apertadas.

7.1.2 Layout e Mapa do Site

O layout do Estudo360 foi cuidadosamente projetado para maximizar a organização das informações e proporcionar uma visão clara e objetiva das atividades dos usuários. Baxter afirma que "um layout com hierarquia bem definida permite que o usuário se concentre nas informações mais importantes, melhorando a clareza e a eficiência do design" (2015, p. 102). No aplicativo, essa hierarquia é refletida na tela inicial, que apresenta um resumo diário com eventos, tarefas e notificações em destaque, garantindo que o usuário tenha uma visão rápida de suas atividades principais.

O mapa do site segue uma estrutura lógica, facilitando a navegação entre as diversas funcionalidades do aplicativo. Na tela inicial, o usuário pode rapidamente adicionar eventos, visualizar suas tarefas ou acessar notificações importantes. De acordo com Norman, "os usuários tendem a buscar consistência e previsibilidade em interfaces digitais, o que pode ser obtido através de uma organização clara das informações e ações" (2018, p. 67). A disposição do calendário, por exemplo, oferece visões diária, semanal e mensal, o que facilita tanto o planejamento de curto quanto de longo prazo. A seção de Tarefas é organizada por prazos e prioridades, promovendo um controle eficiente sobre as responsabilidades acadêmicas.

7.1.3 Padrões de Design

O uso de padrões de design consistentes foi uma prioridade durante o desenvolvimento do Estudo360, visando garantir que a experiência do usuário seja homogênea e intuitiva em todas as telas. Lidwell, Holden e Butler apontam que "a repetição de padrões visuais ajuda o usuário a reconhecer e antecipar funções, reduzindo a carga cognitiva associada ao aprendizado de novas interações" (2019, p. 85). A consistência foi mantida em elementos como a paleta de cores, tipografia e ícones, que seguem um estilo homogêneo, tornando as interações dentro do aplicativo previsíveis e fáceis de entender.

- **Paleta de Cores:** A escolha da paleta de cores baseou-se em promover um ambiente tranquilo e focado. Foi adotada uma paleta composta por tons de azul claro e verde. Kandinsky, em sua obra *Ponto e Linha sobre o Plano*, afirma que "o azul está frequentemente associado à calma e introspecção, enquanto o verde representa equilíbrio e conclusão" (2011, p. 55). Com base nessas

propriedades, o azul foi utilizado em áreas que requerem concentração, como os fundos das telas, enquanto o verde é aplicado em ações conclusivas, como a marcação de tarefas concluídas.

- **Tipografia:** A escolha das fontes recaiu sobre as tipografias sans-serif, como Open Sans e Roboto, amplamente utilizadas em dispositivos móveis devido à sua clareza e legibilidade em diferentes tamanhos de tela. Segundo Lupton, "as fontes sans-serif são populares em interfaces digitais por serem visualmente limpas e objetivas, permitindo que o conteúdo seja lido com facilidade em qualquer dispositivo" (2010, p. 47).
- **Botões e Ícones:** No que se refere aos botões, seguiu-se o princípio da simplicidade e funcionalidade. Botões com bordas arredondadas e cores vibrantes foram utilizados para destacar ações importantes, como adicionar novos eventos ou concluir tarefas. "A simplicidade no design de botões é fundamental para evitar distrações e garantir que os usuários foquem nas ações desejadas" (JOHNSON, 2010, p. 61). Os ícones seguem um conjunto padronizado (Material Icons), garantindo que os usuários possam reconhecê-los imediatamente.

7.1.4 Elementos Recorrentes e Consistência

A consistência é um dos elementos mais críticos no design de interfaces, especialmente em aplicativos com múltiplas funcionalidades como o Estudo360. Nielsen argumenta que "a consistência visual e funcional é essencial para que os usuários se adaptem rapidamente ao aplicativo, sem necessidade de reaprender interações em cada tela" (1995, p. 98). No Estudo360, essa consistência foi mantida em elementos como botões de ação e links de navegação.

Os botões de ação primária são sempre destacados com a cor verde, enquanto os botões de ação secundária utilizam tons mais neutros, como o cinza. Esse esquema de cores cria uma hierarquia clara, que ajuda os usuários a identificar rapidamente as ações mais importantes. Além disso, a barra de navegação fixa garante que os usuários possam acessar as principais funcionalidades do aplicativo sem a necessidade de procurar por elas. "A previsibilidade nas interações é um fator crucial para a boa usabilidade; quando o usuário sabe o que esperar, ele realiza suas tarefas com mais eficiência" (NORMAN, 2018, p. 53).

7.1.5 Ajuda e Documentação

Para garantir que os usuários do Estudo360 possam utilizar todas as funcionalidades com facilidade, foi implementado um sistema de ajuda que é acessível e intuitivo. Krug destaca que "os usuários valorizam sistemas de ajuda que oferecem soluções rápidas e fáceis de entender, sem interromper seu fluxo de trabalho" (2000, p. 74). No Estudo360, isso foi alcançado com uma combinação de tooltips e uma seção de FAQ.

Os tooltips são dicas que aparecem automaticamente quando o usuário passa o cursor ou clica em determinados ícones ou campos de ação, oferecendo informações rápidas e relevantes sobre aquela função. "Tooltips fornecem assistência contextual sem sobrecarregar o usuário com informações desnecessárias" (BAXTER, 2015, p. 143). A seção de FAQ é organizada em categorias como Gerenciamento de Tarefas, Sincronização de Calendário e Configurações do Pomodoro, facilitando o acesso a respostas rápidas para questões frequentes. Para funcionalidades mais complexas, como a configuração do temporizador Pomodoro ou a integração com calendários externos, foram adicionados guias passo a passo e vídeos explicativos, garantindo que o usuário possa explorar todas as funcionalidades do aplicativo sem dificuldade.

7.1.6 Uso de Ícones Personalizados no Design

No Estudo360, o uso de ícones personalizados foi pensado com base na funcionalidade e clareza, elementos essenciais para uma experiência de usuário eficiente. Decidiu-se por utilizar ícones em preto e branco em áreas menores e mais discretas, enquanto os ícones coloridos são empregados em áreas maiores e de destaque. Essa estratégia de design visa garantir a hierarquia visual, facilitando a compreensão rápida e direta das funcionalidades, além de reduzir o excesso de informações visuais em espaços limitados.

- Ícones Preto e Branco em Áreas Menores: Em áreas menores, como botões secundários ou menus compactos, optou-se pelo uso de ícones em preto e branco. Lidwell, Holden e Butler afirmam que "a simplificação visual em elementos pequenos ajuda a evitar a sobrecarga cognitiva" (2019, p. 124). Ícones em preto e branco são mais simples e menos propensos a distrações, permitindo que o usuário foque na funcionalidade sem ser sobrecarregado com

excesso de cores. Essa escolha também facilita a legibilidade e garante que os ícones permaneçam nítidos em espaços compactos.

Krug reforça que "ícones mais simples são particularmente eficazes quando utilizados em menus ou áreas de navegação secundárias, onde o foco está em guiar o usuário de forma rápida e clara" (2000, p. 56). A simplicidade dos ícones em preto e branco nestes contextos contribui para uma navegação mais fluida e objetiva.

- Ícones Coloridos em Áreas Maiores: Em áreas maiores e de destaque, como a tela inicial, botões de ação principais e indicadores visuais de sucesso ou falha, os ícones foram projetados em cores vivas. Essa escolha está alinhada com a necessidade de destacar ações prioritárias e funções de maior importância. Como apontam Lupton e Baxter, "o uso de cores em elementos maiores e mais visíveis aumenta o impacto visual e melhora a capacidade de comunicação imediata de um ícone, tornando-o mais atraente e memorável" (2010, p. 89; 2015, p. 110).

As cores também desempenham um papel importante no feedback visual e no reconhecimento de padrões de comportamento do usuário. Kandinsky discute como "as cores têm a capacidade de evocar reações emocionais imediatas e influenciar a forma como interpretamos sinais visuais" (2011, p. 112). No Estudo360, ícones coloridos são usados para atrair atenção para ações que exigem foco ou que são centrais para a funcionalidade do aplicativo, como a marcação de tarefas concluídas ou a criação de novos eventos no calendário.

- Hierarquia Visual e Consistência: A combinação de ícones preto e branco para áreas menores e ícones coloridos em áreas de maior destaque estabelece uma hierarquia visual clara, essencial para uma navegação eficiente. Nielsen destaca a importância de "criar uma estrutura visual que guie o usuário de forma intuitiva", e o contraste entre ícones preto e branco e ícones coloridos ajuda a estabelecer essa hierarquia de forma natural (1995, p. 84). Os ícones coloridos atraem mais atenção onde necessário, enquanto os ícones preto e branco permitem uma experiência de uso mais fluida e sem distrações em áreas secundárias.

Essa abordagem também contribui para a consistência visual em todo o aplicativo. Lidwell, Holden e Butler afirmam que "a consistência no uso de cores e

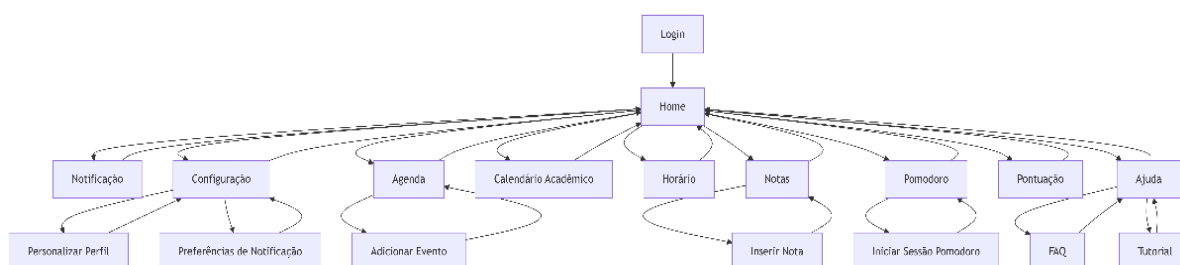
ícones reforça o entendimento do usuário sobre as funções e ajuda a criar previsibilidade nas interações, melhorando a usabilidade geral do aplicativo" (2019, p. 112).

7.2 Interfaces de Usuário

A necessidade da construção de uma interface amigável ao usuário é fundamental em um sistema. A interface faz parte do sistema computacional e determina como as pessoas operam e controlam o sistema. Quando uma interface é bem projetada, ela é compreensível, agradável e controlável. Neste contexto, estes protótipos têm como objetivo apresentar o aplicativo e os recursos da tela.

A figura 20 a seguir, apresenta um fluxo sugerido de navegação no aplicativo, começando a partir do login do usuário. Nela, estão organizadas as principais funcionalidades do sistema. Cada elemento está conectado de forma lógica, indicando caminhos de interação para ações como adicionar eventos, personalizar notificações, consultar horários ou gerenciar tarefas. Esse diagrama demonstra a estrutura hierárquica e a integração das funcionalidades, visando proporcionar uma experiência de uso acessível e eficiente.

Figura 20 - Diagrama de fluxo de tela do Aplicativo Estudo360.



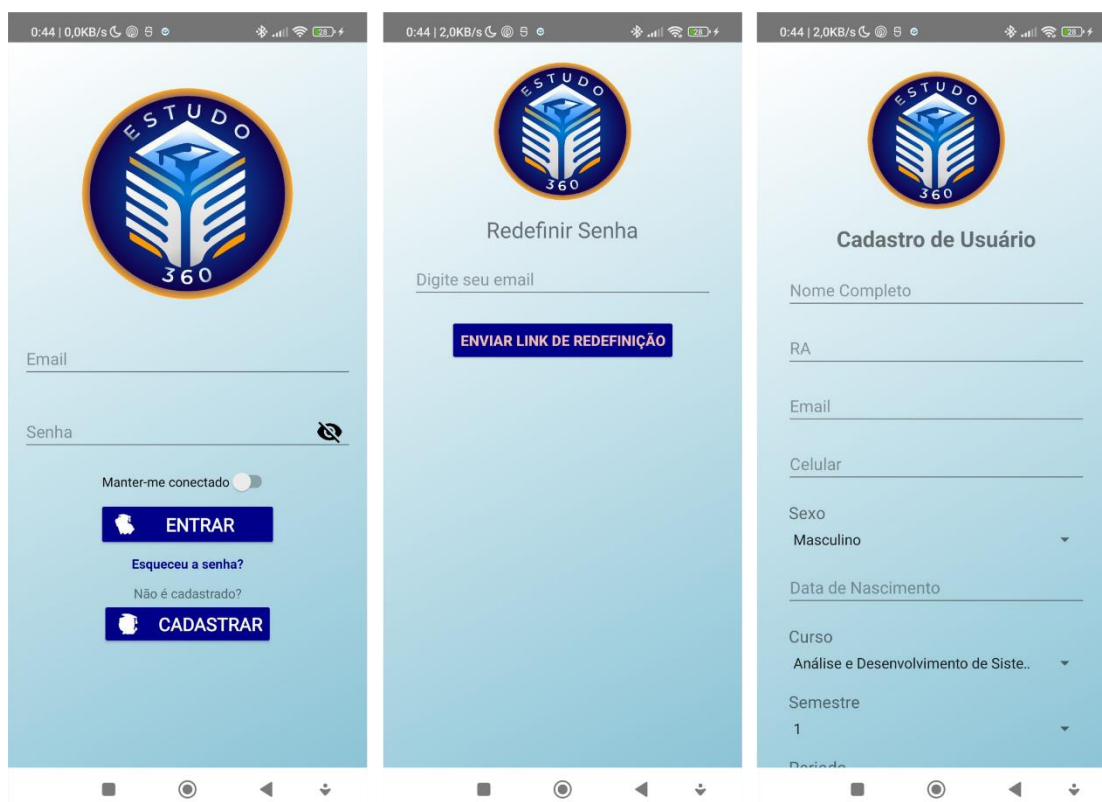
Fonte: Elaborado pelos autores (2024).

A primeira interação do usuário é pela Tela de Login, que é a porta de entrada principal do aplicativo. Nela, o usuário deve inserir seu e-mail cadastrado e senha nos campos correspondentes. O campo de senha oferece a opção de mostrar ou ocultar o que está sendo digitado, facilitando a inserção correta. Logo abaixo, há um botão "Manter-me conectado" que, quando ativado, permite que o usuário permaneça logado mesmo após fechar o aplicativo. Para acessar, basta clicar em "Entrar"; se as credenciais estiverem corretas, o usuário é direcionado à tela principal. Caso contrário, uma mensagem de erro aparece.

Se o usuário esquecer a senha, pode clicar em "Esqueceu a senha?", sendo levado à Tela de Redefinição de Senha. Ali, basta informar o e-mail associado à conta e clicar em "Enviar Link de Redefinição". O sistema enviará um e-mail com um link para redefinir a senha, garantindo segurança e facilidade na recuperação do acesso.

Para novos usuários, o aplicativo oferece a Tela de Cadastro, onde é possível criar uma conta preenchendo um formulário com informações pessoais e acadêmicas. Os campos incluem nome completo, registro acadêmico (RA), e-mail, celular, sexo, data de nascimento, curso, semestre e período. Depois de inserir os dados, o sistema verifica a disponibilidade do e-mail e valida os campos obrigatórios, garantindo que as informações estão corretas e evitando duplicidade de contas. Com o cadastro concluído, o usuário pode usar suas novas credenciais para acessar o aplicativo.

Figura 21 - Captura das telas de Login, Redefinição de Senha e Cadastro.



Fonte: Elaborado pelos autores (2024).

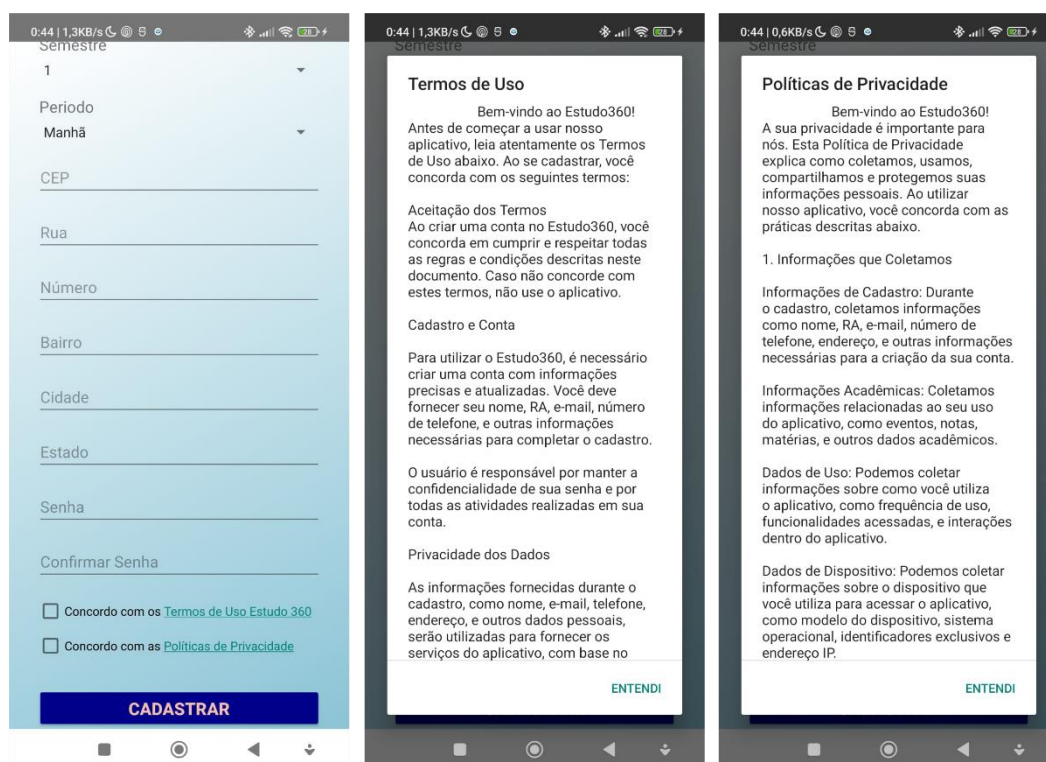
Antes de finalizar o cadastro, o aplicativo solicita que o usuário leia e concorde com os Termos de Uso e a Política de Privacidade. Essas opções aparecem como links clicáveis, permitindo que o usuário visualize os documentos diretamente na interface. A aceitação é confirmada por meio de duas caixas de seleção que precisam ser marcadas antes que o botão "Cadastrar" seja habilitado. Isso assegura que o usuário está ciente das condições de uso e práticas de privacidade, além de atender a requisitos legais como a LGPD.

A Tela de Termos de Uso fornece detalhes sobre as condições para utilizar o aplicativo. O documento explica que, ao criar uma conta, o usuário concorda em seguir as regras estabelecidas, como fornecer informações precisas e manter a confidencialidade das credenciais. Também aborda a responsabilidade do usuário em relação às atividades no aplicativo e esclarece como os dados fornecidos serão usados de forma ética e segura. Após a leitura, o usuário pode clicar em "Entendi" para voltar à Tela de Cadastro.

A Tela de Política de Privacidade detalha como as informações pessoais e acadêmicas são coletadas, armazenadas e protegidas. São coletados dados de cadastro, informações acadêmicas (como eventos e notas) e dados de uso do aplicativo. Também são coletados dados técnicos do dispositivo, como modelo, sistema operacional e endereço IP. O documento enfatiza o compromisso do aplicativo em usar essas informações apenas para os fins descritos, garantindo segurança e privacidade. Depois de ler, o usuário pode retornar à tela anterior clicando em "Entendi".

O fluxo de cadastro é concluído quando o usuário preenche todos os campos obrigatórios, aceita os termos e políticas e clica em "Cadastrar". O sistema valida as informações, verificando se não há duplicidades, como e-mails já cadastrados, e se todos os campos foram preenchidos corretamente. Se tudo estiver em ordem, o cadastro é finalizado com sucesso, e o usuário pode acessar o aplicativo pela Tela de Login.

Figura 22 - Captura da tela de Cadastro e de Termos.



Fonte Elaborado pelos autores (2024).

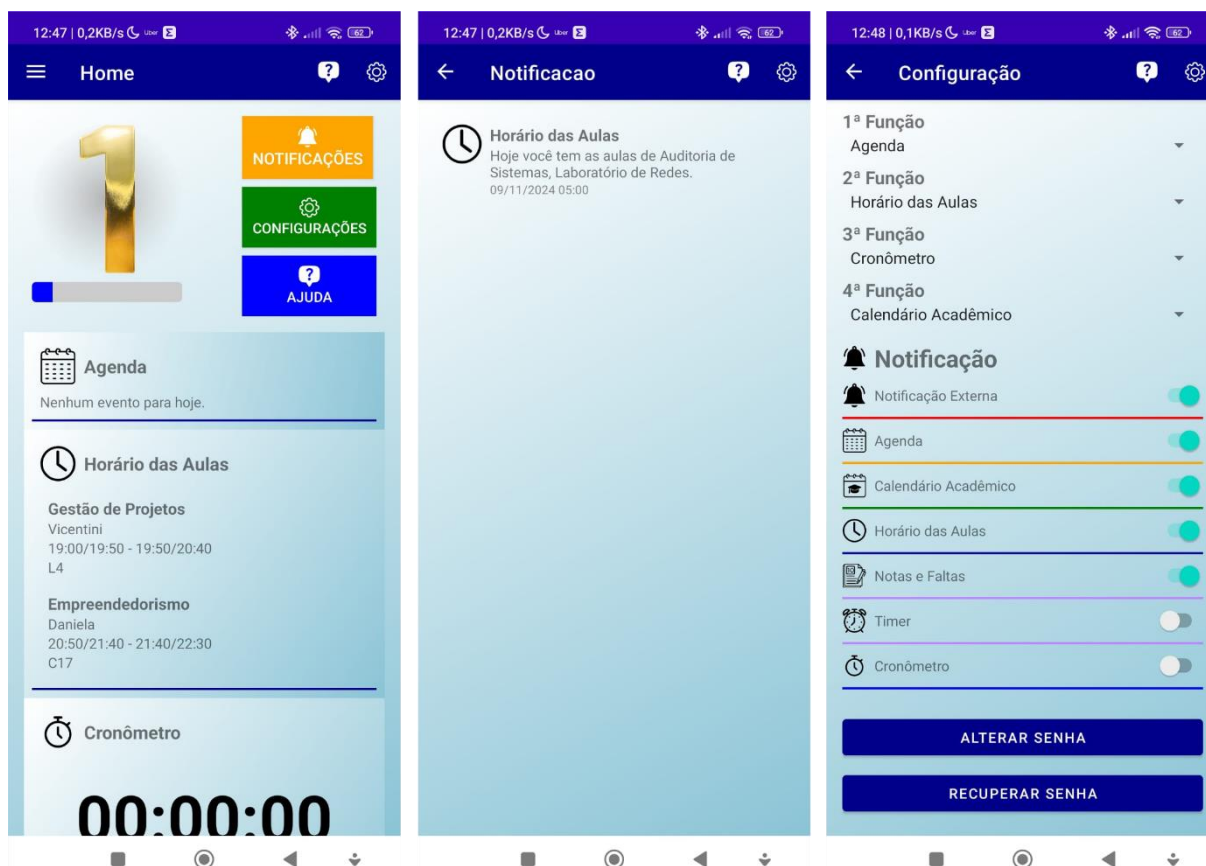
A Tela Home é o ponto de partida do aplicativo, onde o usuário encontra um resumo das principais informações e ferramentas acadêmicas. No topo, é exibido o nível atual do usuário com uma barra de progresso que mostra os pontos acumulados rumo ao próximo nível. Essa abordagem gamificada incentiva o uso contínuo. A tela também inclui botões de acesso rápido para Notificações, Configurações e Ajuda, facilitando a navegação. Logo abaixo, estão as ferramentas acadêmicas como a Agenda, que mostra os eventos programados para o dia ou a mensagem "Nenhum evento para hoje" se não houver atividades. O Horário das Aulas lista as disciplinas do dia com detalhes como nome da matéria, professor, horário e sala. Por fim, o Cronômetro está integrado diretamente na Home, permitindo que o usuário gerencie atividades em tempo real.

A Tela Notificações organiza e apresenta os alertas enviados pelo aplicativo. Cada notificação é listada com detalhes como título, descrição, data e horário, ajudando o usuário a acompanhar compromissos importantes. Por exemplo, uma notificação pode informar sobre aulas programadas, com informações completas de horário e local. A interface é simples e limpa, garantindo que o usuário identifique rapidamente as mensagens relevantes. Um ícone de ajuda está disponível para oferecer suporte adicional sobre o uso dessa funcionalidade.

Na Tela Configurações, o usuário encontra opções para personalizar a experiência no aplicativo. É possível selecionar até quatro funcionalidades principais que serão exibidas na Tela Home, como Agenda, Horário das Aulas, Cronômetro e Calendário Acadêmico. Essas opções são apresentadas em listas suspensas, permitindo fácil organização conforme as preferências individuais. Além disso, o usuário pode configurar notificações, ativando ou desativando alertas para diferentes funcionalidades como Agenda, Notas e Faltas ou Timer, por meio de botões deslizantes.

Outras opções importantes incluem os botões para Alterar Senha e Recuperar Senha, oferecendo segurança adicional e garantindo que o usuário possa gerenciar suas credenciais de forma simples. A combinação dessas funcionalidades faz da Tela Configurações um espaço central para personalizar e otimizar o uso do aplicativo.

Figura 23 - Captura das telas da Home, Notificação e Configuração.



Fonte: Elaborado pelos autores (2024).

Na aba Tutorial, os usuários têm acesso a guias interativos que explicam como utilizar cada funcionalidade do aplicativo. Ao selecionar opções como Home, Notificações, Agenda, Calendário Acadêmico ou Horário das Aulas, o sistema destaca visualmente a área correspondente na interface. Esse destaque, combinado com instruções claras, orienta o usuário sobre como utilizar a funcionalidade escolhida. Por exemplo, ao acessar o tutorial de Seleção de Curso, o aplicativo destaca a lista de cursos disponíveis e explica como escolher o desejado. O mesmo acontece com a Seleção de Período, orientando sobre a escolha do turno (manhã, tarde ou noite), e com a Seleção de Semestre, detalhando como configurar o semestre acadêmico. Essa abordagem visual facilita o aprendizado e promove a autonomia do usuário na navegação pelas ferramentas do aplicativo.

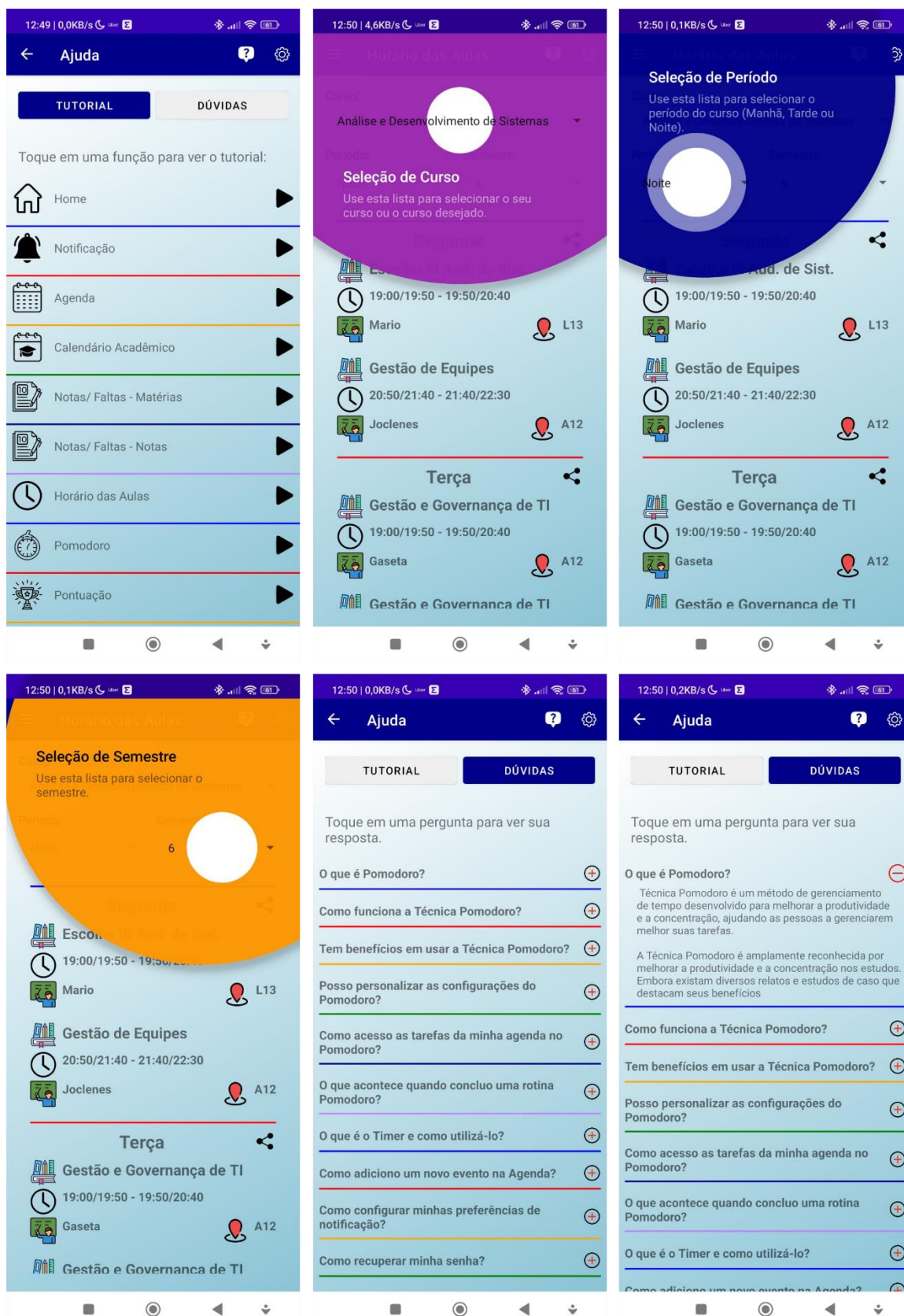
Na aba Dúvidas, o usuário encontra uma lista de perguntas frequentes organizadas de forma acessível. Cada pergunta pode ser expandida para exibir uma resposta detalhada. Exemplos de perguntas incluem: "O que é Pomodoro?", "Como

funciona a Técnica Pomodoro?", "Quais os benefícios em usar a Técnica Pomodoro?", "Como acesso as tarefas da minha agenda no aplicativo?", "Como configurar minhas preferências de notificações?" e "Como recuperar minha senha?". Ao expandir uma pergunta, o usuário obtém explicações claras e diretas. Essa funcionalidade é ideal para resolver dúvidas específicas de forma rápida, sem necessidade de suporte adicional.

A Tela de Ajuda combina suporte visual e textual para atender às diferentes formas de aprendizado dos usuários. No tutorial, os destaques visuais ajudam a localizar elementos na interface, enquanto as explicações textuais fornecem o contexto necessário para entendê-los. Na seção de dúvidas, as respostas estruturadas permitem que o usuário encontre rapidamente as informações que procura, promovendo uma experiência de uso eficiente.

Com essas duas seções integradas, o aplicativo oferece um suporte completo para todas as funcionalidades, incentivando os usuários a explorarem o sistema de forma independente. A interface intuitiva e a linguagem acessível tornam a experiência ainda mais prática.

Figura 24 - Captura das telas de Ajuda, Tutorial e Dúvidas.



Fonte: Elaborado pelos autores (2024).

O Menu Lateral do aplicativo Estudo360 é uma funcionalidade central que proporciona acesso rápido e organizado às principais ferramentas do sistema. Ele pode ser acessado pelo ícone de três linhas (conhecido como "menu hambúrguer") no topo da tela inicial. Esse menu foi projetado para oferecer uma navegação clara e eficiente, permitindo que o usuário encontre as funcionalidades de forma intuitiva.

Na parte superior do menu, são exibidas informações do usuário logado, incluindo nome completo, e-mail e número de identificação (RA). Por exemplo, "Victor Hugo Seiji Ueyama", "seiji.camp@gmail.com" e "0040482212035". Esses dados permitem uma experiência personalizada e ajudam o usuário a identificar rapidamente sua conta.

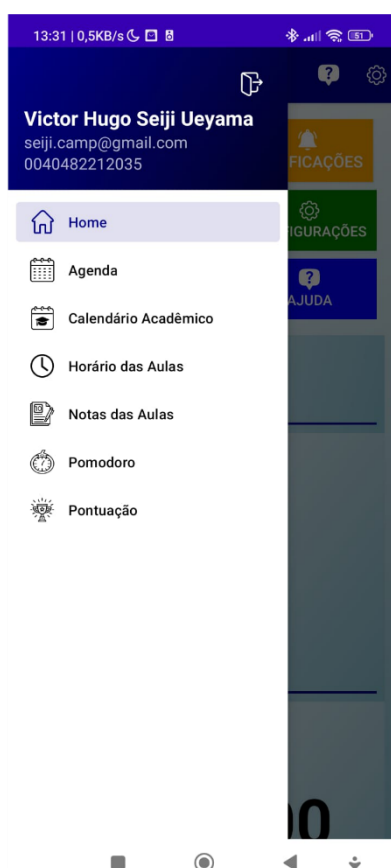
Logo abaixo, estão listadas as principais funcionalidades do aplicativo, cada uma com um ícone e uma descrição clara:

1. Home: Leva o usuário de volta à tela inicial, onde pode ver seu nível atual, barra de progresso, notificações, agenda e outras ferramentas como o cronômetro.
2. Agenda: Permite gerenciar compromissos e eventos acadêmicos ou pessoais. O usuário pode adicionar, visualizar, editar ou excluir eventos, além de integrar atividades à técnica Pomodoro.
3. Calendário Acadêmico: Apresenta os eventos da instituição organizados em um calendário mensal. O usuário pode ver prazos de trabalhos, datas de provas e feriados acadêmicos.
4. Horário das Aulas: Oferece uma visão detalhada do cronograma acadêmico, com informações como nome das disciplinas, horários, professores e salas de aula.
5. Notas das Aulas: Exibe um resumo do desempenho acadêmico, incluindo notas e faltas por disciplina. Ajuda a monitorar o progresso e calcular médias automaticamente.
6. Pomodoro: Acessa o temporizador integrado para sessões de estudo baseadas na técnica Pomodoro. O usuário pode configurar tempos de foco e descanso, acompanhar o progresso e ganhar pontos para o sistema de gamificação.

7. Pontuação: Permite acompanhar o progresso no sistema de gamificação. O usuário pode ver os pontos acumulados, níveis atingidos e recompensas desbloqueadas ao longo do tempo.

O design do Menu Lateral é funcional e intuitivo. Cada funcionalidade é acompanhada por um ícone que facilita a identificação visual, tornando o menu mais acessível. Essa estrutura organiza de forma centralizada os recursos do aplicativo, reduzindo o número de cliques necessários para acessar as principais ferramentas.

Figura 25 - Captura da tela de Menu.



Fonte: Elaborado pelos autores (2024).

Na tela inicial da Agenda, o usuário encontra um calendário mensal onde os dias com eventos são destacados. Ao selecionar uma data específica, a agenda exibe os eventos cadastrados para aquele dia. Se não houver compromissos, a mensagem "Não tem eventos para essa data" é exibida.

Para adicionar um novo evento, o usuário clica no botão de ação flutuante com um ícone de calendário e um "+". Isso abre um formulário com os seguintes campos:

- Título do Evento: Nome do compromisso (obrigatório).
- Descrição: Informações adicionais (opcional).
- Data: Data do evento (obrigatório).
- Hora: Horário de início (opcional).
- Local: Local do evento (opcional).
- Pomodoro: Um checkbox que, se ativado, integra o evento ao temporizador Pomodoro.
- Prioridade: Define a prioridade do evento (Normal ou Alta), destacada com cores para facilitar a identificação.

Após preencher os dados, o usuário pode salvar o evento clicando em "Salvar" ou cancelar a criação.

Os eventos cadastrados aparecem abaixo do calendário, mostrando informações como título, descrição, data, hora, local, indicação de integração com Pomodoro e prioridade destacada por cores. A interface clara facilita a visualização dos compromissos.

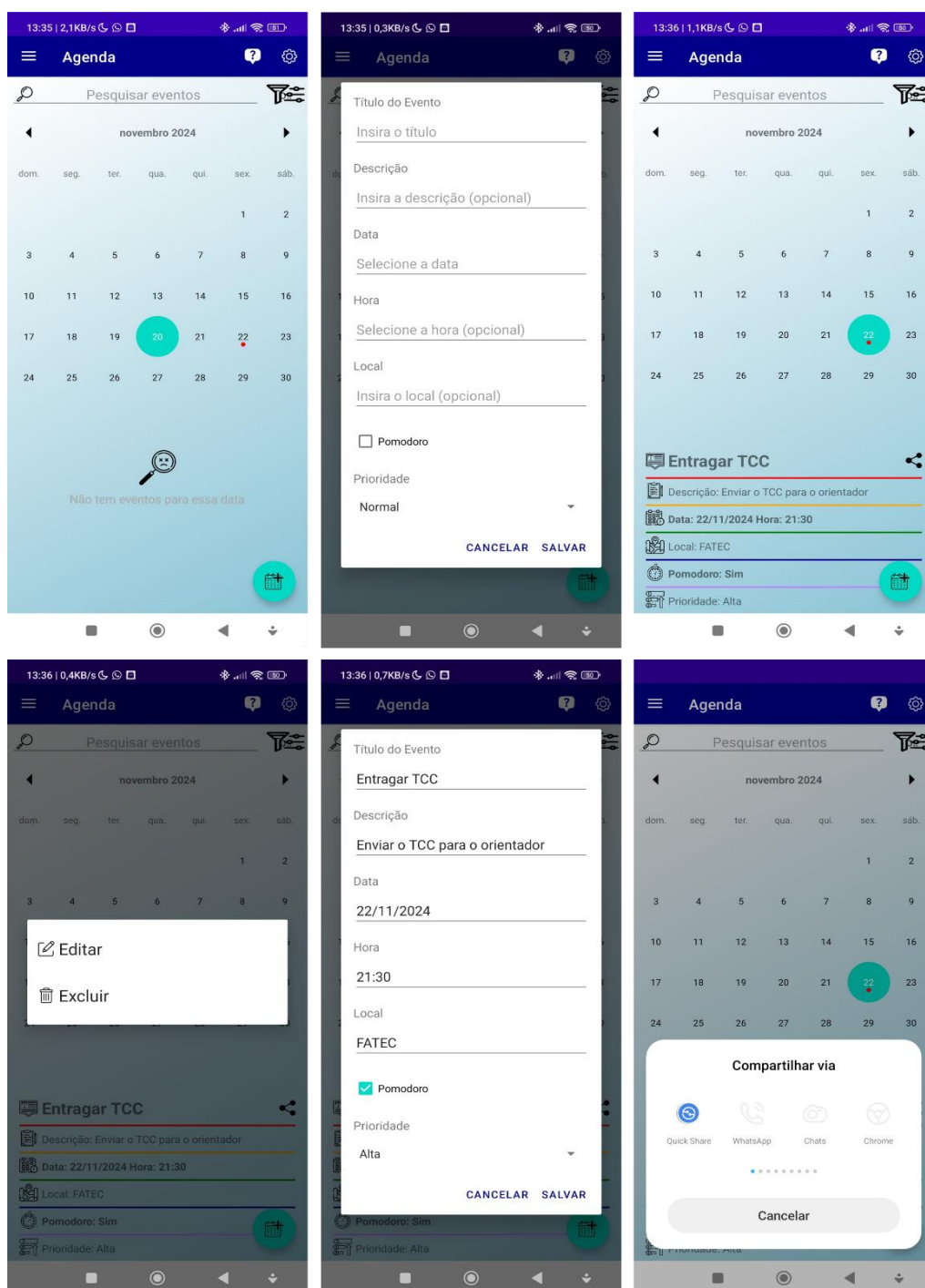
Para gerenciar eventos existentes, o usuário pode clicar em um evento para acessar as opções de Editar e Excluir. Na edição, o formulário é exibido com os dados já preenchidos, permitindo alterações rápidas. Para remover um evento, a opção de exclusão pede confirmação, evitando exclusões acidentais.

Além disso, é possível compartilhar eventos com outras pessoas via aplicativos de mensagens, e-mails ou outras plataformas, útil para compartilhar prazos acadêmicos, reuniões ou outros compromissos relevantes.

A integração com a técnica Pomodoro transforma a agenda em uma ferramenta poderosa para gerenciamento de tempo. Ao habilitar o Pomodoro para um evento, o usuário pode criar sessões de estudo estruturadas e acompanhar o progresso diretamente pela agenda.

A Agenda também permite personalização por prioridade, usando cores para destacar eventos mais importantes, e conta com uma interface amigável que combina o calendário com a lista de compromissos, tornando a navegação simples e eficiente.

Figura 26 - Captura das telas da função Agenda.



Fonte: Elaborado pelos autores (2024).

Na tela do Calendário Acadêmico, o usuário visualiza um calendário mensal onde os dias com eventos são marcados por pontos vermelhos. Ao selecionar uma data específica, o aplicativo exibe detalhes dos eventos associados na parte inferior da tela, com descrições como "Feriado - Dia da Consciência Negra" ou "Não haverá aula".

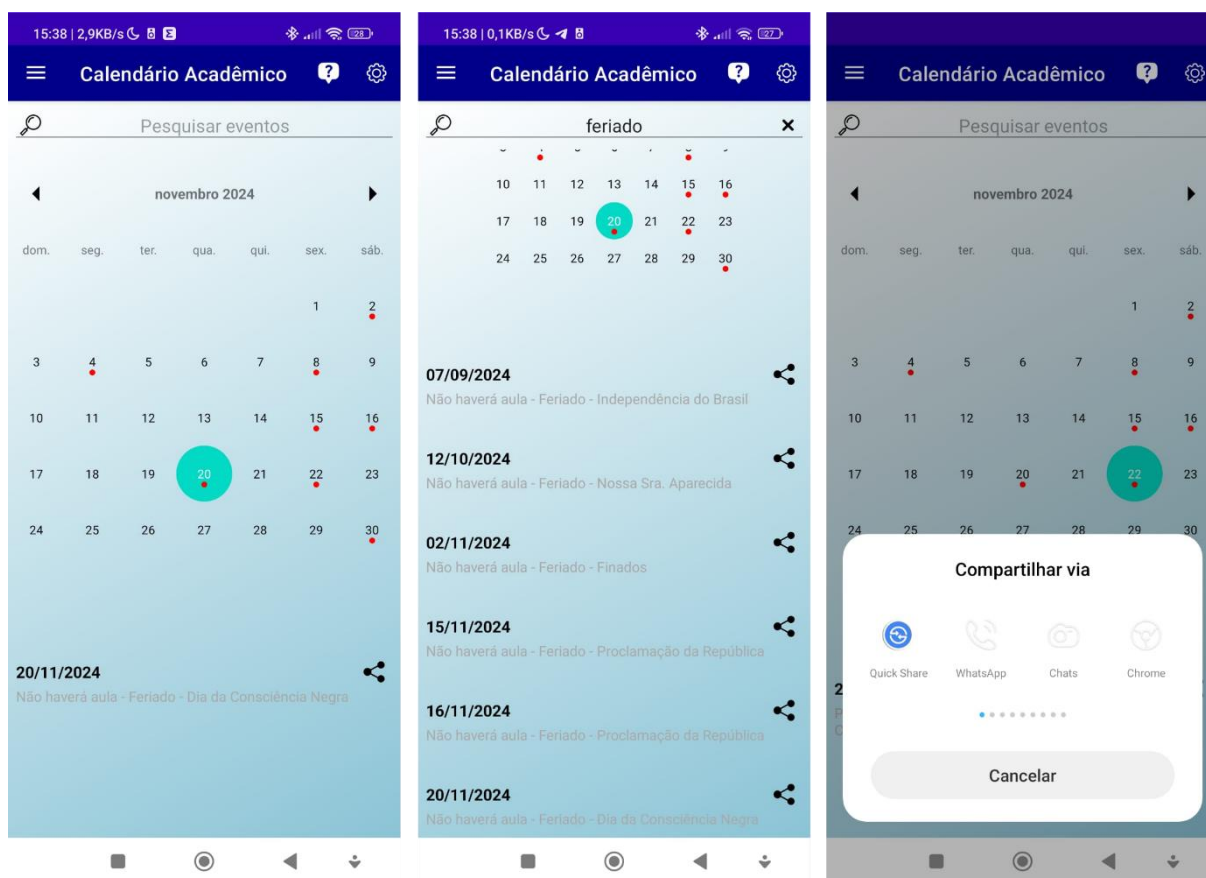
Há também uma barra de pesquisa que permite filtrar eventos por palavras-chave, como "feriado". Isso destaca as datas correspondentes no calendário e exibe apenas os eventos relacionados, facilitando a localização de informações específicas.

Além da visualização no calendário, o usuário pode ver uma lista completa de eventos organizados cronologicamente, baseada na busca. A lista detalha todas as datas importantes, como:

- 07/09/2024: Feriado - Independência do Brasil.
- 12/10/2024: Feriado - Nossa Senhora Aparecida.
- 02/11/2024: Feriado - Finados.
- 15/11/2024 e 16/11/2024: Feriado - Proclamação da República.
- 20/11/2024: Feriado - Dia da Consciência Negra.

Cada evento na lista vem com um ícone de compartilhamento, permitindo que o usuário envie essas informações para outras pessoas. Ao clicar no ícone, o sistema oferece opções de compartilhamento como WhatsApp, e-mail e outras plataformas, facilitando a comunicação de datas importantes com colegas ou professores.

Figura 27 - Captura das telas da função Calendário Acadêmico.



Fonte: Elaborado pelos autores (2024).

Na tela inicial do Horário das Aulas, o usuário pode selecionar o curso, o período (manhã, tarde ou noite) e o semestre em que está matriculado. Com base nessas informações, o aplicativo exibe uma grade organizada por dias da semana, como segunda-feira e terça-feira. Cada aula é apresentada com detalhes importantes:

- Nome da Disciplina: Exibe o nome completo da matéria, como "Escolha III Auditoria de Sistemas" ou "Gestão de Equipes".
- Horário da Aula: Indica o horário de início e término.
- Nome do Professor: Identifica o docente responsável, como "Mario" ou "Joclenes".
- Sala de Aula: Informa o número ou código da sala, como "L13" ou "A12".

Uma funcionalidade adicional é a possibilidade de visualizar a localização da sala de aula no mapa do campus. Ao clicar no ícone de localização ao lado do nome

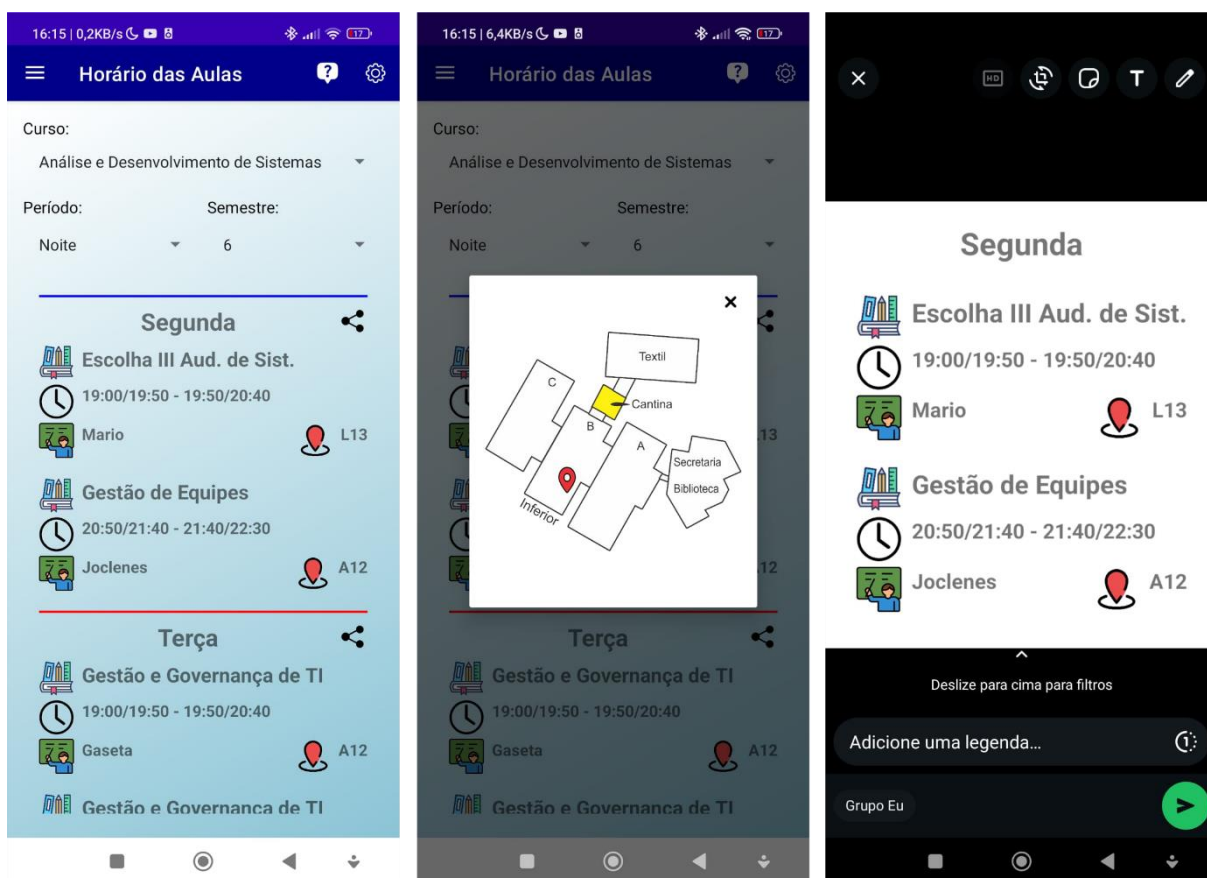
da sala, o aplicativo mostra um mapa detalhado com a sala destacada, facilitando a orientação dentro da instituição. Isso é especialmente útil para novos alunos ou em casos de mudanças nas salas.

O Horário das Aulas também permite o compartilhamento dos horários com colegas ou familiares. Clicando no ícone de compartilhamento, o usuário pode enviar os detalhes da grade horária via aplicativos como WhatsApp ou e-mail, facilitando a comunicação e o planejamento em grupo.

Entre os principais diferenciais dessa funcionalidade estão:

1. Organização Visual: Os horários são exibidos de forma clara e organizada por dias da semana.
2. Mapa Integrado: A opção de localizar as salas no mapa do campus ajuda na orientação.
3. Facilidade de Compartilhamento: O recurso de compartilhar horários torna a interação com outros usuários mais prática.
4. Personalização: A seleção de curso, período e semestre permite que o usuário veja apenas as informações relevantes.

Figura 28 - Captura das telas da função Horário das Aulas.



Fonte: Elaborado pelos autores (2024).

A funcionalidade Notas das Aulas foi projetada para ajudar os estudantes a gerenciar suas disciplinas, permitindo o acompanhamento detalhado de notas e faltas. Na tela principal, o usuário vê uma lista das matérias cadastradas, com informações claras sobre cada uma. Para cada disciplina, são exibidos o nome da matéria, a média calculada com base nas notas registradas e o número de faltas acumuladas, como em "Português" (Média: 10, Faltas: 8) ou "Logística" (Média: 7, Faltas: 2).

No topo da tela, há uma barra de busca para localizar rapidamente uma matéria específica. O botão "Adicionar Matéria", posicionado na parte inferior, permite que o usuário insira novas disciplinas. Ao clicar, uma janela se abre onde é possível:

1. Digitar o nome da matéria.
2. Escolher um ícone personalizado entre várias opções temáticas, facilitando a identificação visual.

Após preencher as informações, ao clicar em "Adicionar", a nova matéria é incluída na lista principal.

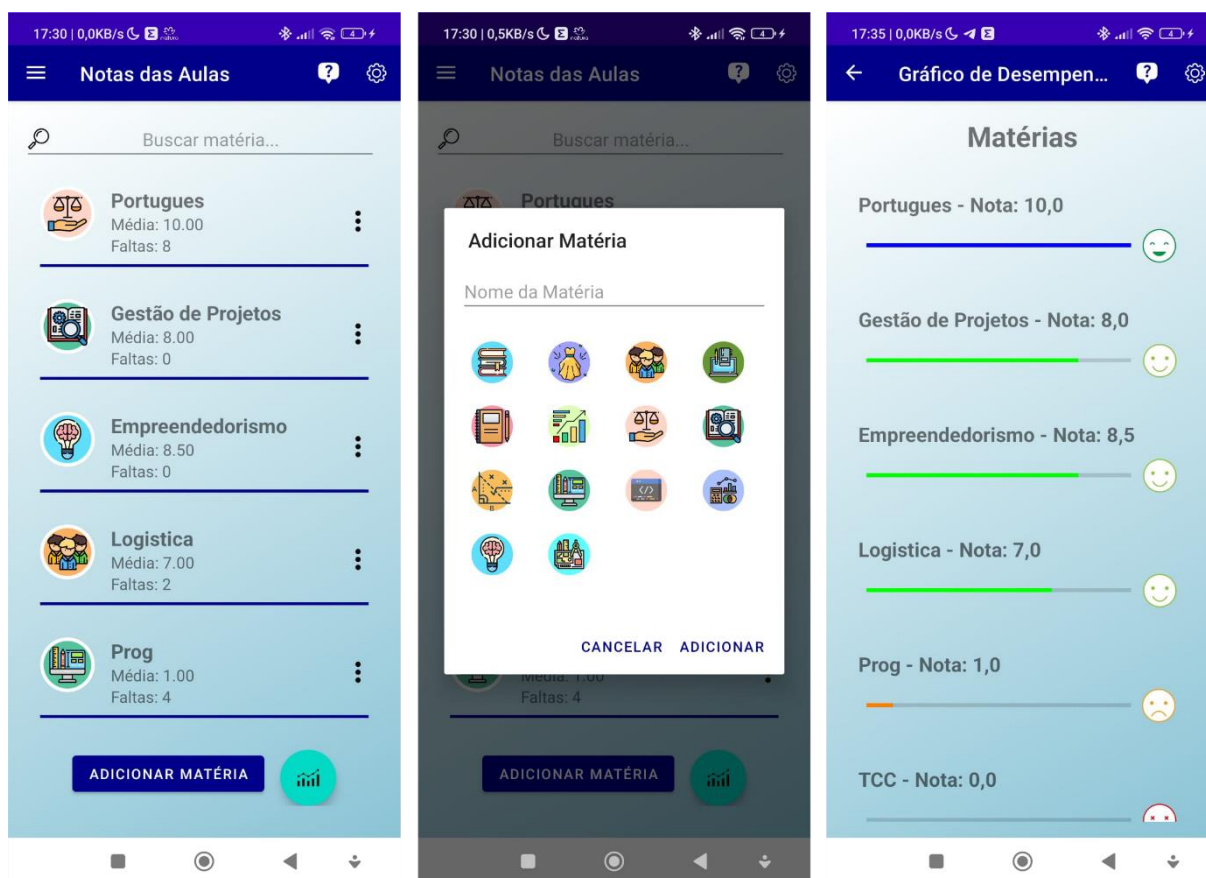
Outra funcionalidade é o Gráfico de Desempenho, que oferece uma visão geral do progresso do estudante em cada disciplina. Nesse gráfico, as matérias são apresentadas com suas respectivas notas, acompanhadas por barras de progresso coloridas que indicam o desempenho:

- Verde: Notas altas, bom desempenho.
- Amarelo: Desempenho mediano.
- Vermelho: Notas baixas, necessidade de atenção.

Ícones de expressões faciais são exibidos ao lado de cada barra, variando entre rostos sorridentes, neutros ou tristes, proporcionando uma interpretação rápida das informações.

Essa funcionalidade também permite controlar as faltas em cada disciplina, ajudando a monitorar a frequência e evitar problemas relacionados a excesso de ausências.

Figura 29 - Captura das telas da função Notas e Faltas e do Gráfico.



Fonte: Elaborado pelos autores (2024).

Ao acessar os detalhes de uma matéria específica, como "TCC", o usuário encontra informações essenciais para monitorar notas, faltas e carga horária semanal.

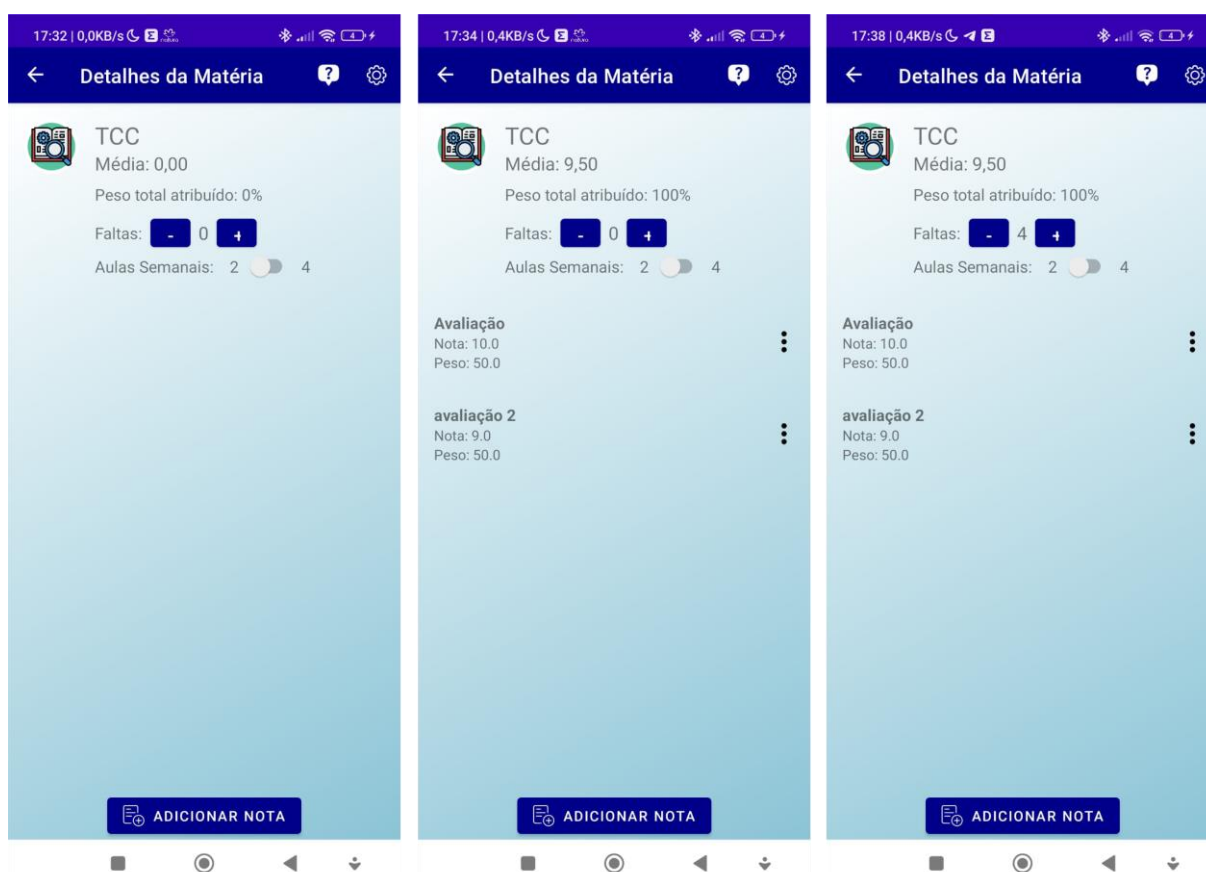
Na parte superior, o nome da matéria é exibido junto com a média atual, calculada automaticamente com base nas notas registradas. O sistema também informa o peso total atribuído, indicando quanto das avaliações já foi contabilizado na média final. Se o peso ainda não somar 100%, isso é claramente mostrado, incentivando o registro de novas avaliações.

A funcionalidade permite o controle de faltas. O estudante pode ajustar o número de ausências usando os botões "+" e "-", mantendo a informação sempre atualizada. Isso é útil para acompanhar a frequência e evitar problemas com excesso de faltas. Além disso, o aplicativo oferece a opção de configurar o número de aulas semanais, permitindo selecionar entre 2 ou 4 aulas, conforme a carga horária da disciplina.

Na parte inferior, é apresentada uma lista das avaliações cadastradas para a matéria. Cada avaliação mostra o nome, a nota obtida e o peso no cálculo da média final. O sistema recalcula e atualiza a média automaticamente com base nas informações inseridas.

O botão "Adicionar Nota", localizado na parte inferior, permite registrar novas avaliações. Ao clicar, uma interface é aberta para inserir o nome da avaliação, a nota e o peso correspondente. Após salvar, a nova avaliação é adicionada à lista, e a média da matéria é recalculada automaticamente.

Figura 30 - Captura das telas de Notas e Faltas.



Fonte: Elaborado pelos autores (2024).

A funcionalidade Pomodoro foi criada para ajudar os estudantes a gerenciar o tempo de forma eficiente durante as sessões de estudo. Baseada na técnica Pomodoro, permite alternar períodos de trabalho concentrado com intervalos de descanso, otimizando a produtividade e o foco.

Na primeira tela, os usuários podem configurar o Timer Pomodoro para personalizar a rotina de estudo. É possível adicionar informações detalhadas sobre a

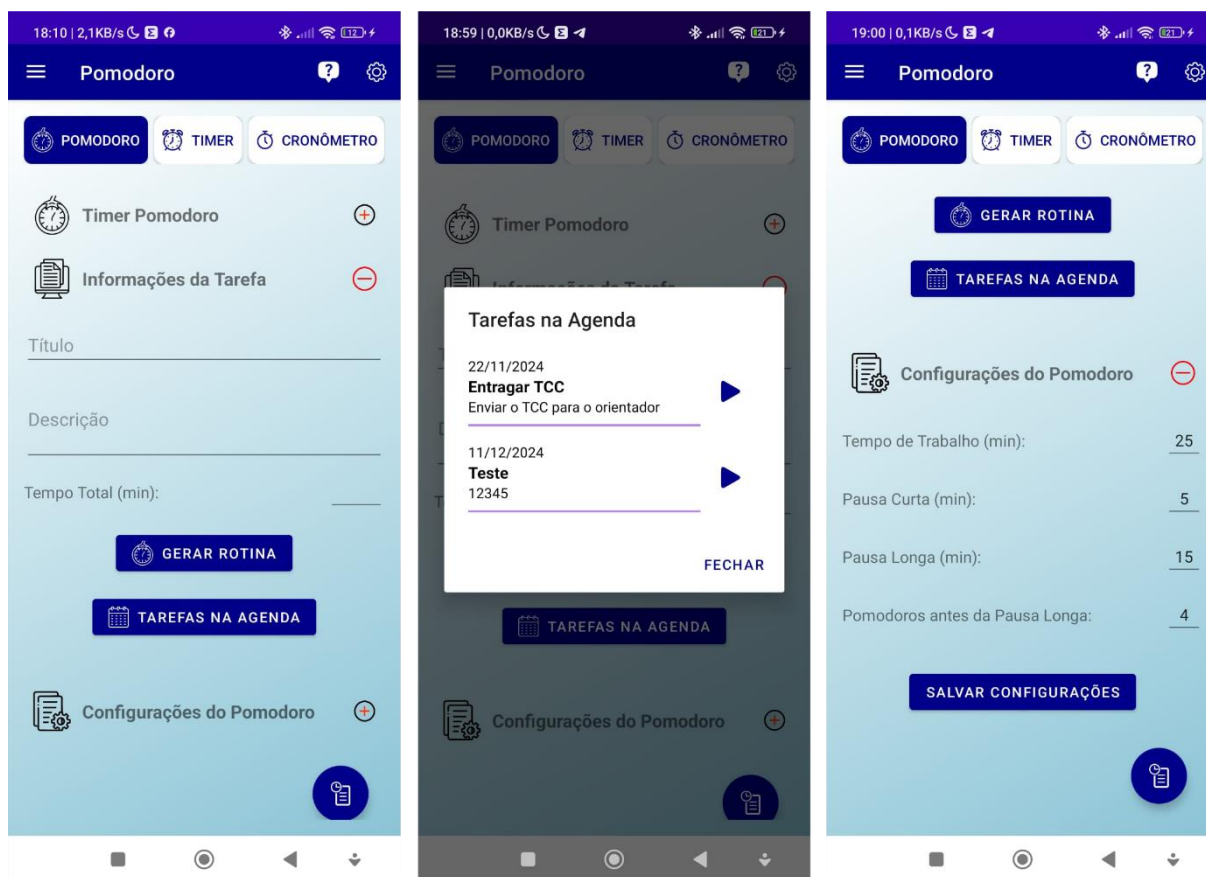
tarefa, como título, descrição e tempo total estimado em minutos. Após preencher esses campos, o botão "Gerar Rotina" cria automaticamente um ciclo de estudo baseado nas configurações padrão ou personalizadas. Além disso, o botão "Tarefas na Agenda" exibe uma lista com as tarefas registradas na agenda, permitindo vinculá-las diretamente às sessões Pomodoro.

A seção de Configurações do Pomodoro permite ajustar os parâmetros da técnica. Essas configurações incluem:

- Tempo de Trabalho (min): Duração de cada ciclo de estudo.
- Pausa Curta (min): Duração dos intervalos rápidos entre os ciclos.
- Pausa Longa (min): Tempo dos intervalos mais prolongados após completar um número específico de ciclos.
- Pomodoros antes da Pausa Longa: Quantos ciclos são necessários para que uma pausa longa seja iniciada.

Após ajustar, o botão "Salvar Configurações" assegura que as mudanças sejam aplicadas nas futuras sessões.

Figura 31 - Captura das telas da função Pomodoro.



Fonte: Elaborado pelos autores (2024).

Na tela de criação da rotina Pomodoro, o usuário insere os dados relacionados à tarefa a ser realizada: título, descrição opcional e tempo total estimado em minutos. Após preencher, ao clicar em "Gerar Rotina", o aplicativo cria uma sequência de ciclos Pomodoro baseada nas configurações definidas.

Em seguida, uma janela de resumo é exibida, apresentando detalhes importantes da rotina configurada:

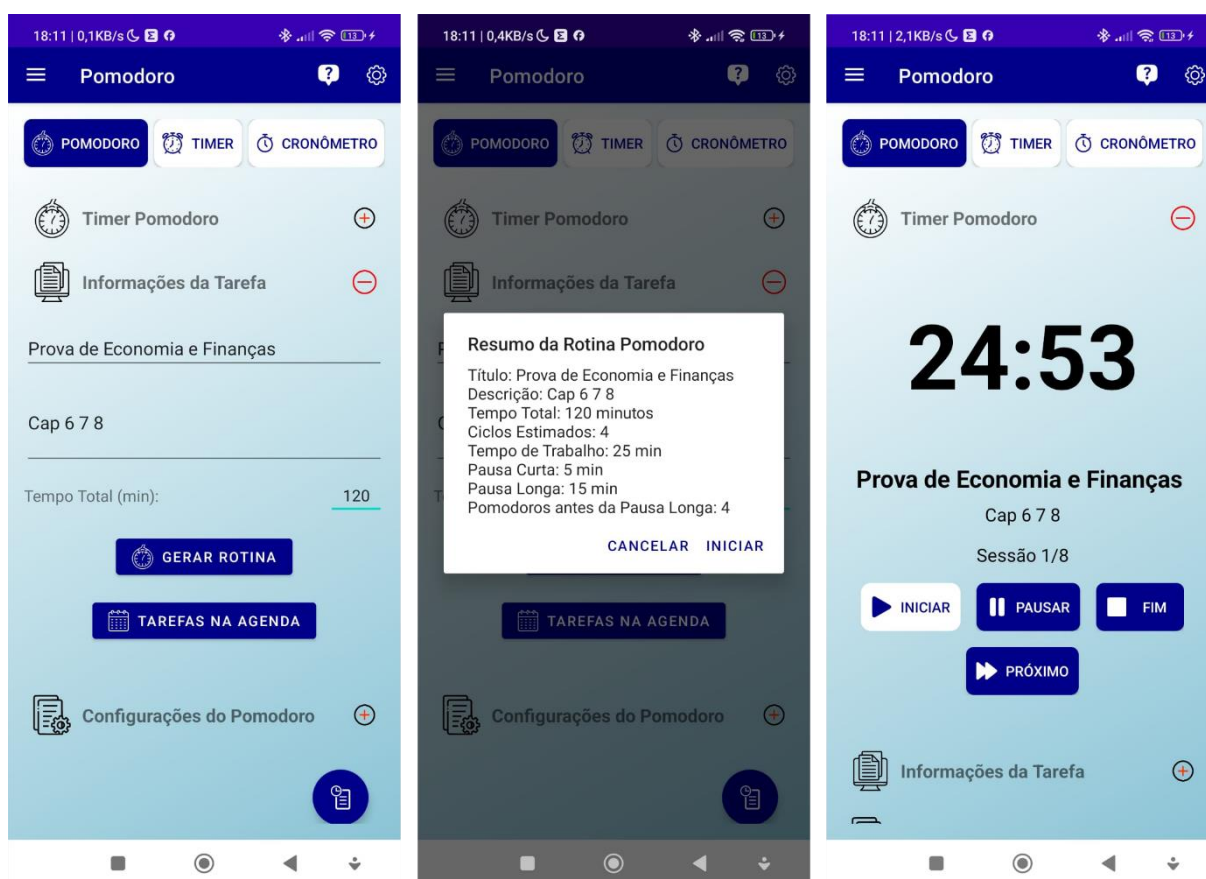
- Título e descrição da tarefa.
- Tempo total estimado.
- Número de ciclos estimados, calculados com base no tempo total e na duração dos intervalos e períodos de trabalho.
- Duração de cada tempo de trabalho, pausa curta e pausa longa.
- Quantidade de Pomodoros antes de uma pausa longa.

O usuário pode revisar os detalhes e optar por "Iniciar" ou "Cancelar" a execução. Ao iniciar, a tela do temporizador exibe a contagem regressiva do ciclo atual. O título e a descrição da tarefa permanecem visíveis para manter o foco no objetivo.

Durante a execução, o aplicativo oferece controles simples:

- Iniciar ou Pausar a contagem regressiva.
- Encerrar a rotina com o botão "Fim".
- Avançar para o próximo ciclo usando o botão "Próximo".

Figura 32 - Captura das telas da função de Início do Pomodoro.



Fonte: Elaborado pelos autores (2024).

Durante cada etapa da rotina Pomodoro, o usuário recebe mensagens que reforçam o propósito da técnica e incentivam a continuidade.

Mensagem ao Final do Ciclo de Trabalho, ao concluir um ciclo de 25 minutos de estudo, o aplicativo exibe: "Muito bom! Você terminou a sessão de estudo de 25 minutos. Faça uma pausa."

Essa mensagem reconhece o esforço e lembra da importância do descanso antes do próximo ciclo.

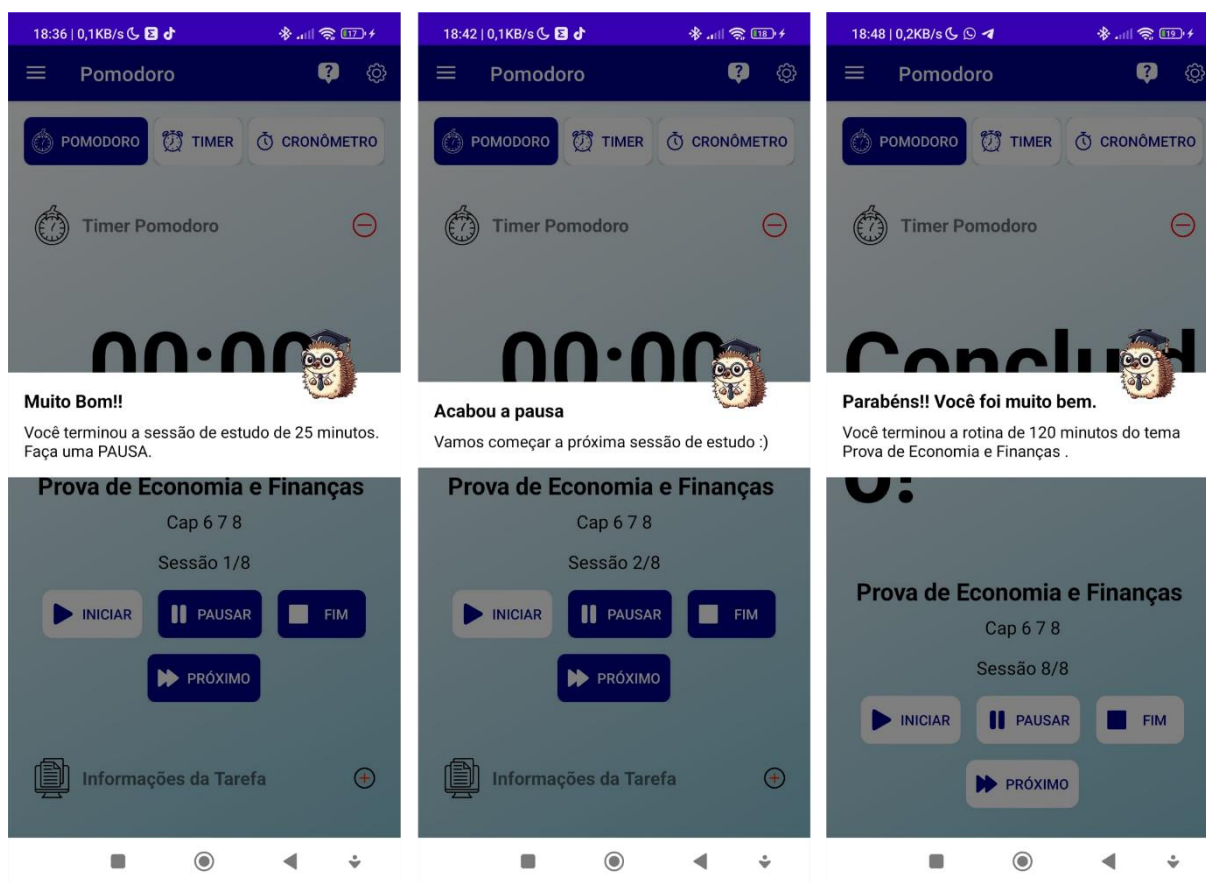
Mensagem ao final da pausa curta, quando o intervalo termina, o aplicativo notifica: "Acabou a pausa. Vamos começar a próxima sessão de estudo :)". Isso garante que o usuário retome o foco no momento certo, mantendo o ritmo planejado.

Mensagem ao final da rotina Pomodoro, ao finalizar todos os ciclos programados, o aplicativo celebra: "Parabéns! Você foi muito bem. Você terminou a rotina de 120 minutos do tema Prova de Economia e Finanças."

Essa mensagem cria um senso de realização, reforçando o empenho do usuário e incentivando o uso contínuo da funcionalidade.

As mensagens são acompanhadas por um mascote amigável, adicionando um toque interativo e acolhedor, tornando o processo de estudo mais leve e motivador.

Figura 33 - Captura das telas das Mensagens exibidas no Pomodoro.



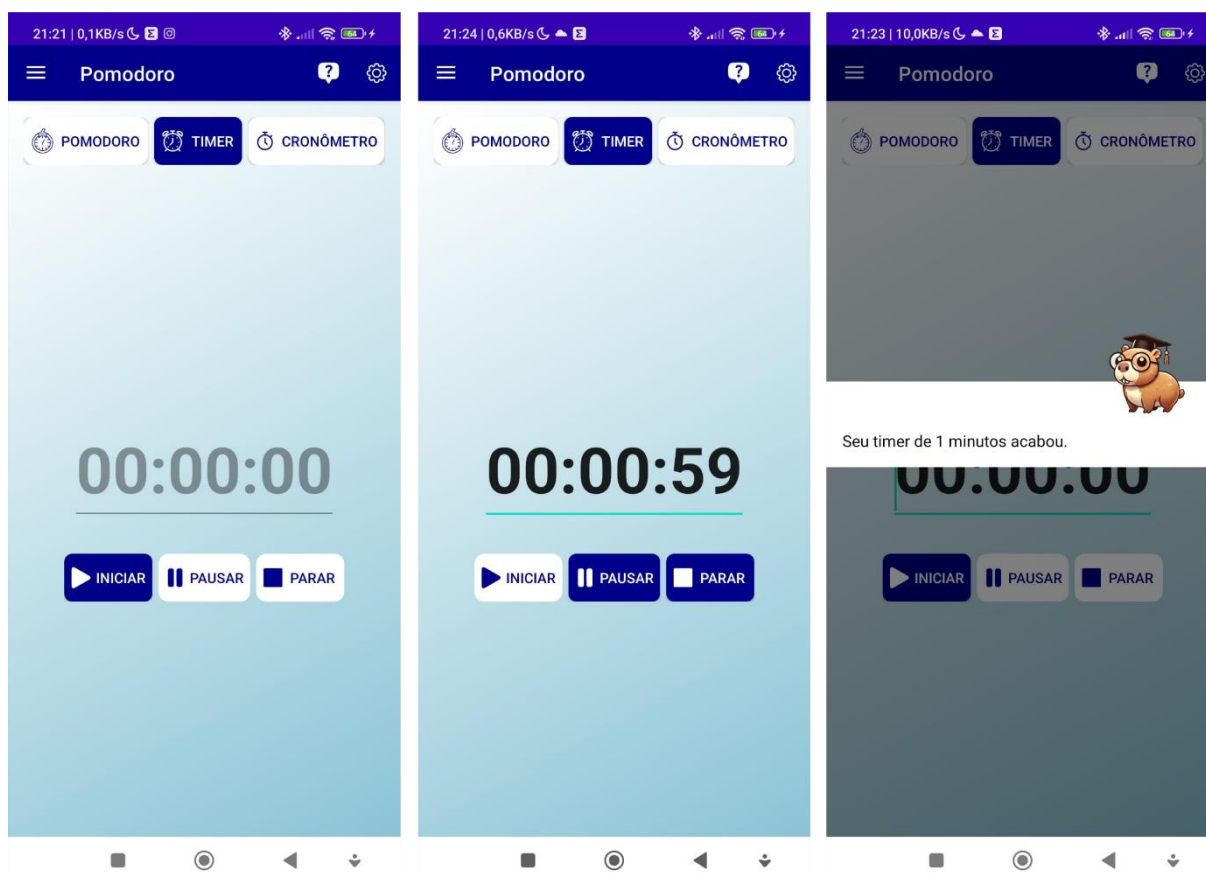
Fonte: Elaborado pelos autores (2024).

O Timer é uma funcionalidade que permite ao usuário configurar e acompanhar a contagem regressiva de intervalos de tempo específicos, ajudando no controle e organização de tarefas diárias. A interface é simples e intuitiva, com botões que facilitam o uso.

Na tela inicial do Timer, o cronômetro aparece zerado, com as opções principais: Iniciar, Pausar e Parar. O usuário define o tempo desejado para a contagem regressiva e, ao pressionar "Iniciar", o cronômetro começa a contar. Durante a contagem, "Pausar" permite interromper temporariamente o tempo, retomando-o depois do ponto em que foi pausado. "Parar" encerra a contagem antes do término do tempo definido.

Conforme o Timer progride, o aplicativo mostra o tempo restante de forma destacada, facilitando o acompanhamento em tempo real. Quando o tempo chega ao fim, uma notificação aparece com uma mensagem motivadora, como: **"Seu timer de 1 minuto acabou."** Essa mensagem vem acompanhada pelo mascote do aplicativo, reforçando a experiência positiva.

Figura 34 - Captura das telas do Timer e da mensagem exibida.

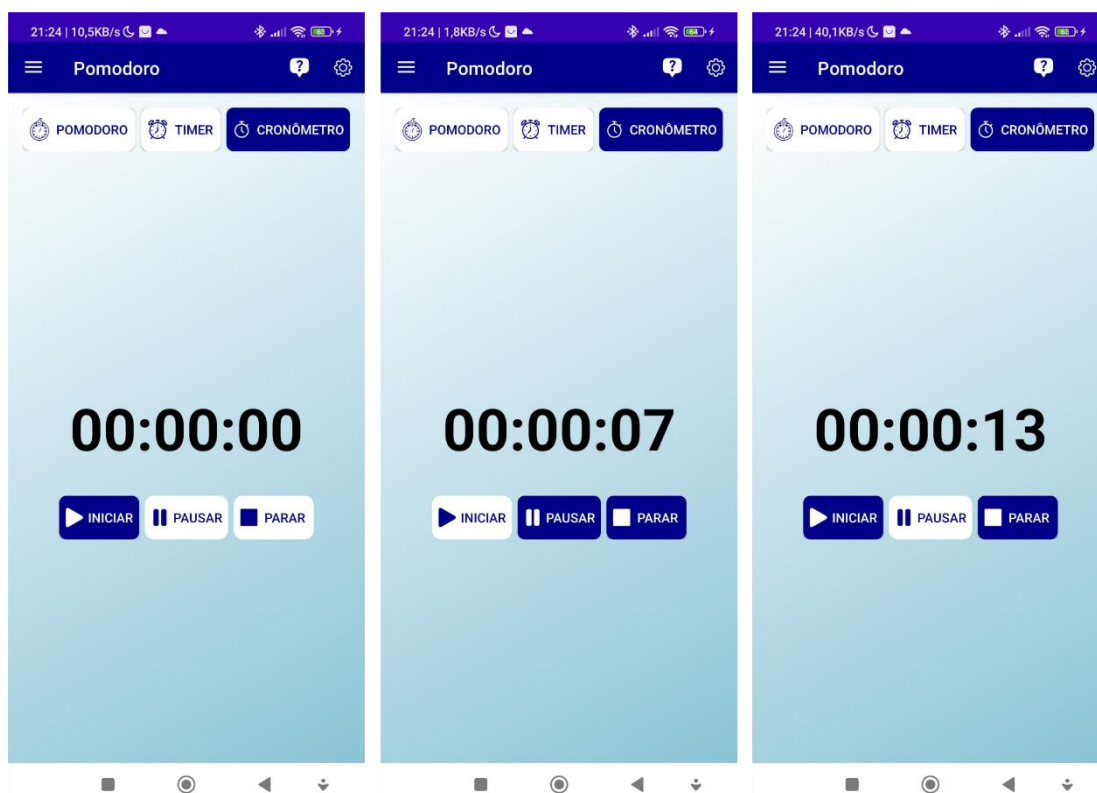


Fonte: Elaborado pelos autores (2024).

Na tela inicial do Cronômetro, o tempo é exibido zerado, e o usuário tem à disposição os botões principais: Iniciar, Pausar e Parar. Ao clicar em "Iniciar", o cronômetro começa a contar o tempo, mostrando segundos e minutos em tempo real. Se for necessário interromper a contagem sem zerar o tempo decorrido, o botão "Pausar" permite essa ação. O botão "Parar" redefine o cronômetro para zero, encerrando a contagem.

Durante o uso, o tempo decorrido é mostrado em números grandes e claros, garantindo fácil leitura, mesmo à distância. Essa funcionalidade é ideal para atividades que exigem monitoramento contínuo, como exercícios físicos, estudos ou qualquer tarefa que requer controle de tempo.

Figura 35 - Captura das telas do Cronômetro.



Fonte: Elaborado pelos autores (2024).

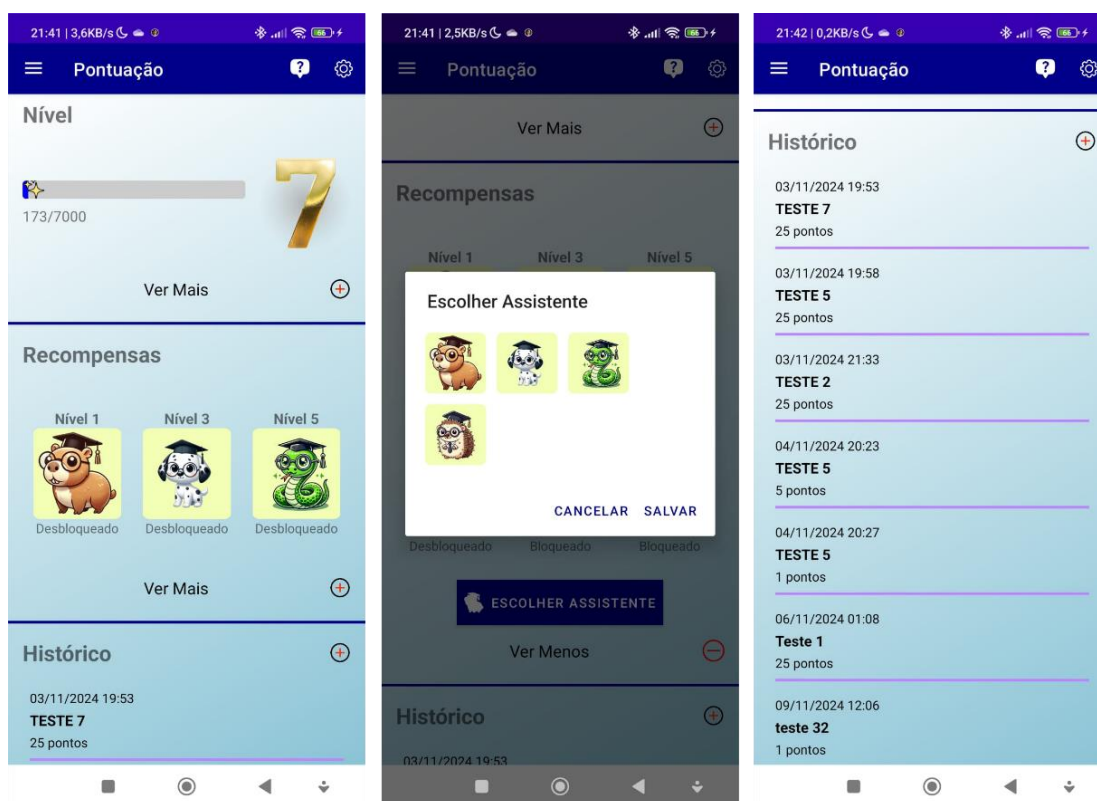
O sistema de Pontuação é baseado em um mecanismo de gamificação que atribui pontos ao usuário por atividades realizadas, como uso da técnica Pomodoro, conclusão de tarefas e outras funcionalidades. Esses pontos são acumulados para que o usuário avance de nível e desbloqueie recompensas exclusivas.

Na tela principal de Pontuação, o usuário vê seu nível atual e a barra de progresso indicando quantos pontos faltam para o próximo nível. As recompensas são exibidas como assistentes virtuais, cada um desbloqueado ao atingir níveis específicos. O botão "Escolher Assistente" permite selecionar um assistente já desbloqueado para acompanhar o usuário no aplicativo, reforçando a personalização.

O histórico de pontos conquistados é apresentado em uma lista cronológica na parte inferior da tela, permitindo que o usuário acompanhe detalhadamente suas atividades e os pontos ganhos em cada uma. Esse histórico ajuda a manter a transparência e o controle sobre o sistema de pontuação.

Por meio dessa funcionalidade, o Estudo360 motiva os usuários a se envolverem mais ativamente com as ferramentas do aplicativo, proporcionando um ambiente divertido e interativo.

Figura 36 - Captura das telas da função Pontuação e Recompensas.



Fonte: Elaborado pelos autores (2024).

8 CONSIDERAÇÕES FINAIS

O desenvolvimento do aplicativo Estudo360 representa uma contribuição significativa para o apoio ao gerenciamento acadêmico de estudantes universitários. O projeto foi concebido com o objetivo de integrar, em uma única plataforma móvel, diversas funcionalidades que facilitam a organização dos compromissos acadêmicos, otimização do tempo de estudo e acompanhamento do desempenho escolar.

Ao longo deste trabalho, foram abordados os principais aspectos teóricos e práticos relacionados ao desenvolvimento de aplicativos móveis, desde a fase de planejamento até a implementação e testes. A utilização de metodologias ágeis, em especial o Scrum, permitiu uma gestão eficiente do projeto, possibilitando a adaptação a mudanças e o cumprimento dos prazos estabelecidos.

A escolha das ferramentas e tecnologias, como o Android Studio, Kotlin e Firebase, mostrou-se acertada, proporcionando um ambiente de desenvolvimento robusto e adequado às necessidades do aplicativo. A integração com o Firebase foi fundamental para a implementação de funcionalidades em tempo real, como a sincronização de dados e o sistema de autenticação de usuários.

O Estudo360 destaca-se por oferecer uma interface amigável e intuitiva, resultado de um cuidadoso processo de design centrado no usuário. A aplicação dos princípios de usabilidade e design visual, fundamentados em autores como Norman (2006) e Nielsen (1993), contribuiu para a criação de uma experiência agradável e eficiente para o usuário.

Entre as principais funcionalidades implementadas, destacam-se:

- Agenda e Calendário Acadêmico: Permitem o gerenciamento de compromissos e eventos acadêmicos, com integração ao calendário institucional.
- Horário das Aulas: Facilita o acesso rápido aos horários e locais das disciplinas, incluindo a visualização de mapas do campus.
- Notas e Faltas: Proporciona o acompanhamento do desempenho acadêmico, com cálculo automático de médias e controle de frequência.
- Técnica Pomodoro: Auxilia na gestão do tempo de estudo, aumentando a produtividade e o foco dos estudantes.

- Gamificação: Incentiva o engajamento do usuário por meio de um sistema de pontos, níveis e recompensas.

O desenvolvimento do Estudo360 enfrentou desafios significativos, como a integração de múltiplas funcionalidades em um único aplicativo e a garantia de uma performance adequada em dispositivos móveis com diferentes especificações. Esses desafios foram superados por meio de um planejamento detalhado, testes rigorosos e ajustes contínuos durante o processo de desenvolvimento.

A validação do aplicativo com usuários reais, através de testes beta, forneceu feedback valioso que permitiu melhorias na usabilidade e na estabilidade do sistema. A resposta positiva dos usuários evidencia o potencial do Estudo360 em contribuir para a organização e sucesso acadêmico dos estudantes.

Como perspectivas futuras, vislumbra-se a expansão do aplicativo com novas funcionalidades, como:

- Integração com Plataformas Educacionais: Sincronização com sistemas acadêmicos para atualização automática de notas e materiais didáticos.
- Colaboração entre Usuários: Criação de grupos de estudo e compartilhamento de recursos entre estudantes.
- Versão iOS: Desenvolvimento de uma versão compatível com dispositivos da Apple, ampliando o alcance do aplicativo.
- Análises Avançadas: Implementação de dashboards com insights sobre o desempenho acadêmico e hábitos de estudo do usuário.

Em conclusão, o Estudo360 atingiu os objetivos propostos, oferecendo uma ferramenta completa e eficaz para auxiliar estudantes no gerenciamento de suas atividades acadêmicas. O projeto contribuiu para o aprimoramento das habilidades técnicas e gerenciais dos desenvolvedores, além de representar um potencial impacto positivo na comunidade acadêmica. Espera-se que o aplicativo continue evoluindo e se consolidando como um aliado indispensável na jornada acadêmica dos estudantes.

9 REFERÊNCIAS BIBLIOGRÁFICAS

ADOBE. Adobe Photoshop. Disponível em: <https://www.adobe.com/br/products/photoshop.html>. Acesso em: 03 nov. 2024.

ANDROID DEVELOPERS. Android Studio. Disponível em: <https://developer.android.com/studio>. Acesso em: 03 nov. 2024.

BAXTER, Kathy; COURAGE, Catherine; CAINE, Kelly. Entendendo Seus Usuários: Um Guia Prático de Métodos de Pesquisa de UX. Porto Alegre: Bookman, 2015.

BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. UML: Guia do Usuário. 2. ed. Rio de Janeiro: Elsevier, 2006.

CHANGE VISION. Astah UML. Disponível em: <https://astah.net/products/astah-uml/>. Acesso em: 03 nov. 2024.

COCKBURN, Alistair. Escrevendo Casos de Uso Eficazes. Porto Alegre: Bookman, 2001.

COREL. CorelDRAW Graphics Suite. Disponível em: <https://www.coreldraw.com/br/>. Acesso em: 03 nov. 2024.

FIGMA. Figma: The Collaborative Interface Design Tool. Disponível em: <https://www.figma.com/>. Acesso em: 03 nov. 2024.

FIREBASE. Firebase Documentation. Disponível em: <https://firebase.google.com/docs>. Acesso em: 03 nov. 2024.

JETBRAINS. Kotlin Programming Language. Disponível em: <https://kotlinlang.org/>. Acesso em: 03 nov. 2024.

JACOBSON, Ivar. Engenharia de Software Orientada a Objetos com UML. Rio de Janeiro: Elsevier, 2000.

JOHNSON, Jeff. Design de Interface do Usuário: Um Guia para Designers de Interface Gráfica. Rio de Janeiro: Elsevier, 2014.

KANDINSKY, Wassily. Ponto e Linha sobre o Plano. 5. ed. São Paulo: Martins Fontes, 2011.

KRUG, Steve. Não Me Faça Pensar: Uma Abordagem de Bom Senso à Usabilidade na Web. 3. ed. Rio de Janeiro: Alta Books, 2014.

LIDWELL, William; HOLDEN, Kritina; BUTLER, Jill. Princípios Universais do Design. 2. ed. Porto Alegre: Bookman, 2015.

LUPTON, Ellen. Pensar com Tipos: Um Guia Crítico para Designers, Escritores, Editores e Estudantes. São Paulo: Gustavo Gili, 2015.

NIELSEN, Jakob. Engenharia de Usabilidade. Rio de Janeiro: Campus, 1993.

NORMAN, Donald A. O Design do Dia a Dia. Rio de Janeiro: Rocco, 2006.

REDMOND, Eric; WILSON, Jim R. Seven Databases in Seven Weeks: A Guide to Modern Databases and the NoSQL Movement. Dallas: Pragmatic Bookshelf, 2012.

SADALAGE, Pramod J.; FOWLER, Martin. NoSQL Essencial: Um Guia Conciso para o Mundo Emergente da Persistência Poliglota. São Paulo: Novatec Editora, 2013.

SCHWABER, Ken; SUTHERLAND, Jeff. Guia do Scrum: O Guia Definitivo do Scrum: As Regras do Jogo. 2017. Disponível em: <https://scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-Portuguese-BR.pdf>. Acesso em: 20 nov. 2023.

SOMMERVILLE, Ian. Engenharia de Software. 9. ed. São Paulo: Pearson Prentice Hall, 2011.

VON HALLE, Barbara. The Handbook of Data Management. 2nd ed. New York: Auerbach Publications, 2002.