

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO

LUIS HENRIQUE FERRACCIU PAGOTTO MENDES
MARIA EDUARDA SOARES ROMANA SILVA
TIAGO FERNANDES SOUCEK

TRABALHO MULTIDISCIPLINAR DE PROGRAMAÇÃO ORIENTADA A
OBJETOS
PLANNER UNIVERSITÁRIO

APUCARANA
2023

LUIS HENRIQUE FERRACCIU PAGOTTO MENDES
MARIA EDUARDA SOARES ROMANA SILVA
TIAGO FERNANDES SOUCEK

**TRABALHO MULTIDISCIPLINAR DE PROGRAMAÇÃO ORIENTADA A
OBJETOS: PLANNER UNIVERSITÁRIO**

Relatório técnico apresentado como requisito parcial à obtenção de nota na disciplina de Programação Orientada a Objetos do curso de graduação em Engenharia de Computação da Universidade Tecnológica Federal do Paraná.
Professor: Lucio Agostinho Rocha.

RESUMO

O trabalho multidisciplinar de Programação Orientada a Objetos propõe um planner universitário, empregando princípios de programação orientada a objetos para otimizar a gestão acadêmica. O programa oferece diversas funcionalidades, incluindo organização de horários de aulas, controle de tarefas e de provas. O sistema é de fácil utilidade, facilitando a adaptação às necessidades específicas de um estudante universitário. O uso de classes e objetos proporciona uma estrutura flexível e eficiente, promovendo a manutenibilidade do código e melhorando a experiência do usuário no planejamento e acompanhamento das atividades acadêmicas.

Palavras-chave: planner universitário; programação orientada a objetos; gestão de tarefas; horários de aulas.

ABSTRACT

The object-oriented guidance project proposes a university planner, employing principles of object-oriented programming to optimize academic management. The program offers various features, including class schedule organization, task and exam tracking. The system is user-friendly, facilitating adaptation to the specific needs of a university student. The use of classes and objects provides a flexible and efficient structure, promoting code maintainability and enhancing the user experience in planning and monitoring academic activities.

Keywords: university planner; object-oriented programming; task management; class schedules.

LISTA DE ILUSTRAÇÕES

Figura 1 – Diagrama de classe.....	14
Figura 2 – Diagrama de Caso de Uso.....	15
Figura 3 – Diagrama de Sequência Cadastrar Universitário	18
Figura 4 - Diagrama de Sequência Realizar Login.....	19
Figura 5 - Diagrama de Sequência Adicionar Prova	19
Figura 6 - Diagrama de Sequência Ler Prova.....	19
Figura 7 - Diagrama de Sequência Adicionar Disciplina	20
Figura 8 - Diagrama de Sequência Ler Disciplina.....	20
Figura 9 - Diagrama de Sequência Adicionar Trabalho	20
Figura 10 - Diagrama de Sequência Ler Trabalho	20
Figura 11 - Diagrama de Estados Cadastrar Universitário	21
Figura 12 - Diagrama de Estados Realizar Login	21
Figura 13 - Diagrama de Estados Adicionar Gasto Mensal.....	21
Figura 14 - Diagrama de Estados Adicionar Prova	22
Figura 15 - Diagrama de Estados Adicionar Tarefa	22
Figura 16 - Diagrama de Estados Concluir Tarefa.....	22
Figura 17 - Diagrama de Estados Adicionar Trabalho	22
Figura 18 - Diagrama de Estados Adicionar Disciplina	22
Figura 19 - Diagrama de Estados Ler Disciplina.....	23
Figura 20 - Diagrama de Estados Inserir Conteúdo	23
Figura 21 - Diagrama de Atividades Cadastrar Universitário	23
Figura 22 - Diagrama de Atividades Realizar Login.....	24
Figura 23 - Diagrama de Atividades Inserir Gasto.....	24
Figura 24 - Diagrama de Atividades Inserir Prova	24
Figura 25 - Diagrama de Atividades Inserir Tarefa	24
Figura 26 - Diagrama de Atividades Concluir Tarefa	25
Figura 27 - Diagrama de Atividades Inserir Trabalho.....	25
Figura 28 - Diagrama de Atividades Inserir Disciplina	25
Figura 29 - Diagrama de Atividades Ver Disciplina	25
Figura 30 - Diagrama de Atividades Inserir Conteúdo no Caderno.....	26

LISTA DE QUADROS

Quadro 1 – Descrição das classes	14
Quadro 2 – Descrição do caso de uso Realizar Cadastro	16
Quadro 3 – Descrição do caso de uso Realizar Login.....	16
Quadro 4 – Descrição do caso de uso Adicionar Gasto Mensal.....	16
Quadro 5 - Descrição do caso de uso Adicionar Prova.....	16
Quadro 6 - Descrição do caso de uso Adicionar Tarefa.....	17
Quadro 7 - Descrição do caso de uso Concluir Tarefa.....	17
Quadro 8 - Descrição do caso de uso Adicionar Trabalho	17
Quadro 9 - Descrição do caso de uso Adicionar Disciplinas	17
Quadro 10 - Descrição do caso de uso Adicionar Conteúdo no Caderno	18
Quadro 11 - Descrição do caso de uso Ver Detalhes da Disciplina	18

SUMÁRIO

1	INTRODUÇÃO	9
2	DESENVOLVIMENTO.....	10
2.1	Funcionalidades disponíveis no Planner Universitário	10
2.1.1	Cadastrar universitário.....	10
2.1.2	Realizar login.....	10
2.1.3	Adicionar gasto mensal	10
2.1.4	Adicionar tarefa	10
2.1.5	Adicionar disciplinas	11
2.1.6	Adicionar prova	11
2.2	Características técnicas do Planner	11
2.2.1	Adaptabilidade.....	11
2.2.2	Proteção	11
2.2.3	Facilidade de Uso	11
2.3	Padrão de Projeto do Planner.....	12
2.3.1	Modelo (Model).....	12
2.3.2	Visão (View).....	12
2.3.3	Controlador (Controller).....	12
2.4	Diagramas UML.....	13
2.4.1	Diagrama de Classe	13
2.4.2	Diagrama de Caso de Uso	15
2.4.3	Diagrama de Sequência	18
2.4.4	Diagrama de Estados	21
2.4.5	Diagrama de Atividades.....	23
2.5	Polimorfismo	26
2.6	Tratamento de Exceção Personalizada	26

2.6.1	CadastroGUI.java - RA já cadastrado e campos vazios	26
2.6.2	LoginGUI.java - RA não cadastrado e credenciais inválidas	27
2.6.3	CadastroUniversitarioGUI.java - Verificar campos vazios	27
2.6.4	ProvasGUI.java - Verificar campos vazios.....	27
3	CONCLUSÃO	28
4	LINK.....	29

1 INTRODUÇÃO

O graduando, seja ele de qualquer curso superior, possui uma rotina diária distinta em função das múltiplas responsabilidades vinculadas à vida acadêmica. Para tanto, a utilização de um programa auxiliar, como um Planner Universitário, é de grande importância, auxiliando a gerir sua rotina.

O software Planner Universitário foi desenvolvido em Java com conceitos de programação orientada a objetos para ser uma solução para este problema, permitindo ao graduando uma variedade de opções para aprimorar sua organização acadêmica, desde a anotação de conteúdo das aulas até o controle de avaliações e a gestão financeira.

Apesar de suas diversas funcionalidades, o Planner Universitário destaca-se por uma interface de fácil utilização, intuitiva e agradável, facilitando o uso. Possui também certa segurança, devido à tela de login, onde o usuário somente tem a permissão de usufruir do sistema se for cadastrado, e ainda se possuir a senha de acesso.

Com tantas funções interessantes, este relatório técnico visa abranger e explicar o software por completo, com os principais recursos e funcionalidades que o Planner pode fornecer ao graduando, além de descrever seus casos de uso e mais com diagramas no padrão UML (Unified Modeling Language).

2 DESENVOLVIMENTO

2.1 Funcionalidades disponíveis no Planner Universitário

Com a intenção de atender às diversas demandas dos discentes durante a graduação, tornou-se necessário incorporar várias funcionalidades ao software. Cada uma delas foi organizada de maneira a garantir uma execução correta, contribuindo assim para o alcance efetivo de sua finalidade proposta.

A seguir, são apresentadas as principais funcionalidades presentes no Planner Universitário:

2.1.1 Cadastrar universitário

Esta funcionalidade tem como finalidade permitir o cadastro de informações de um universitário no sistema, possibilitando a personalização e o uso das funcionalidades do Planner. Ao cadastrar um universitário, o sistema cria um perfil deste usuário, fornecendo uma experiência personalizada e permitindo o acesso às diversas ferramentas disponíveis.

2.1.2 Realizar login

Esta funcionalidade tem como finalidade autenticar o universitário no sistema, permitindo o acesso às funcionalidades personalizadas do Planner. Ao realizar o login, o usuário confirma sua identidade, possibilitando a utilização completa das ferramentas e o acesso aos dados registrados no Planner.

2.1.3 Adicionar gasto mensal

Esta funcionalidade tem como finalidade permitir que o universitário registre de forma organizada seus gastos mensais no sistema Planner. Ao adicionar gastos mensais, o usuário consegue acompanhar suas despesas, contribuindo para uma gestão financeira mais consciente durante o período acadêmico.

2.1.4 Adicionar tarefa

Esta funcionalidade tem como finalidade permitir que o universitário registre tarefas no Planner, contribuindo para uma gestão do tempo e das atividades acadêmicas. Ao adicionar

tarefas, o usuário organiza seu cronograma, facilitando o acompanhamento de prazos e compromissos importantes ao longo do semestre.

2.1.5 Adicionar disciplinas

Esta funcionalidade tem como finalidade permitir que o universitário adicione informações sobre disciplinas no Planner, contribuindo para uma organização mais eficiente de seu cronograma acadêmico. Ao adicionar disciplinas, o usuário pode planejar seu semestre, visualizando os seus horários de aula.

2.1.6 Adicionar prova

Esta funcionalidade tem como finalidade permitir que o universitário registre informações sobre suas provas no Planner, contribuindo para a organização e planejamento dos seus compromissos acadêmicos. Ao adicionar provas, o usuário mantém um controle detalhado das avaliações.

2.2 Características técnicas do Planner

2.2.1 Adaptabilidade

O sistema, por ser desenvolvido em Java, utiliza tecnologias e padrões que conferem adaptações na implantação. Essa abordagem permite fácil adaptação e migração para diferentes plataformas e sistemas operacionais, disponibilizando aos usuários a liberdade de escolher o contexto mais adequado às suas necessidades.

2.2.2 Proteção

O Planner reforça a segurança das informações armazenadas ao implementar medidas de proteção, sendo a autenticação por senha uma das principais proteções.

2.2.3 Facilidade de Uso

O Planner se destaca pela sua usabilidade aprimorada, apresentando uma interface intuitiva e de fácil navegação. Sua interface amigável facilita a usabilidade dos universitários, proporcionando uma experiência sem complicações e direta.

2.3 Padrão de Projeto do Planner

O Planner Universitário adota a abordagem de projeto de software conhecida como MVC, ou Model-View-Controller. Esse padrão arquitetônico é fundamental para a estruturação coerente e modular do sistema, permitindo uma clara separação entre as diversas responsabilidades do software. Abaixo,

2.3.1 Modelo (Model)

O Modelo no Planner Universitário representa a camada de dados e lógica de negócios. Inclui componentes responsáveis por manipular e gerenciar as informações do universitário, como cadastro de disciplinas, gastos mensais, tarefas e provas. Essa camada é independente da interface do usuário, garantindo a integridade e consistência dos dados.

2.3.2 Visão (View)

A Visão é a camada responsável pela apresentação dos dados aos usuários. No Planner, a Visão inclui a interface do usuário, onde os universitários podem interagir com as funcionalidades do sistema. A interface é projetada de maneira intuitiva, refletindo as informações do Modelo de forma clara e organizada.

2.3.3 Controlador (Controller)

O Controlador atua como intermediário entre o Modelo e a Visão, gerenciando as interações do usuário e manipulando as operações relacionadas aos dados. No Planner, o Controlador recebe comandos do usuário por meio da interface e os traduz em operações específicas no Modelo. Ele também atualiza a Visão para refletir as mudanças nos dados.

2.4 Diagramas UML

O Unified Modeling Language (UML) representa uma linguagem gráfica amplamente adotada no desenvolvimento de software para a visualização, especificação, construção e documentação de sistemas complexos. Essa ferramenta versátil oferece uma abordagem padronizada para a representação de modelos conceituais, permitindo uma compreensão visual clara dos aspectos do sistema.

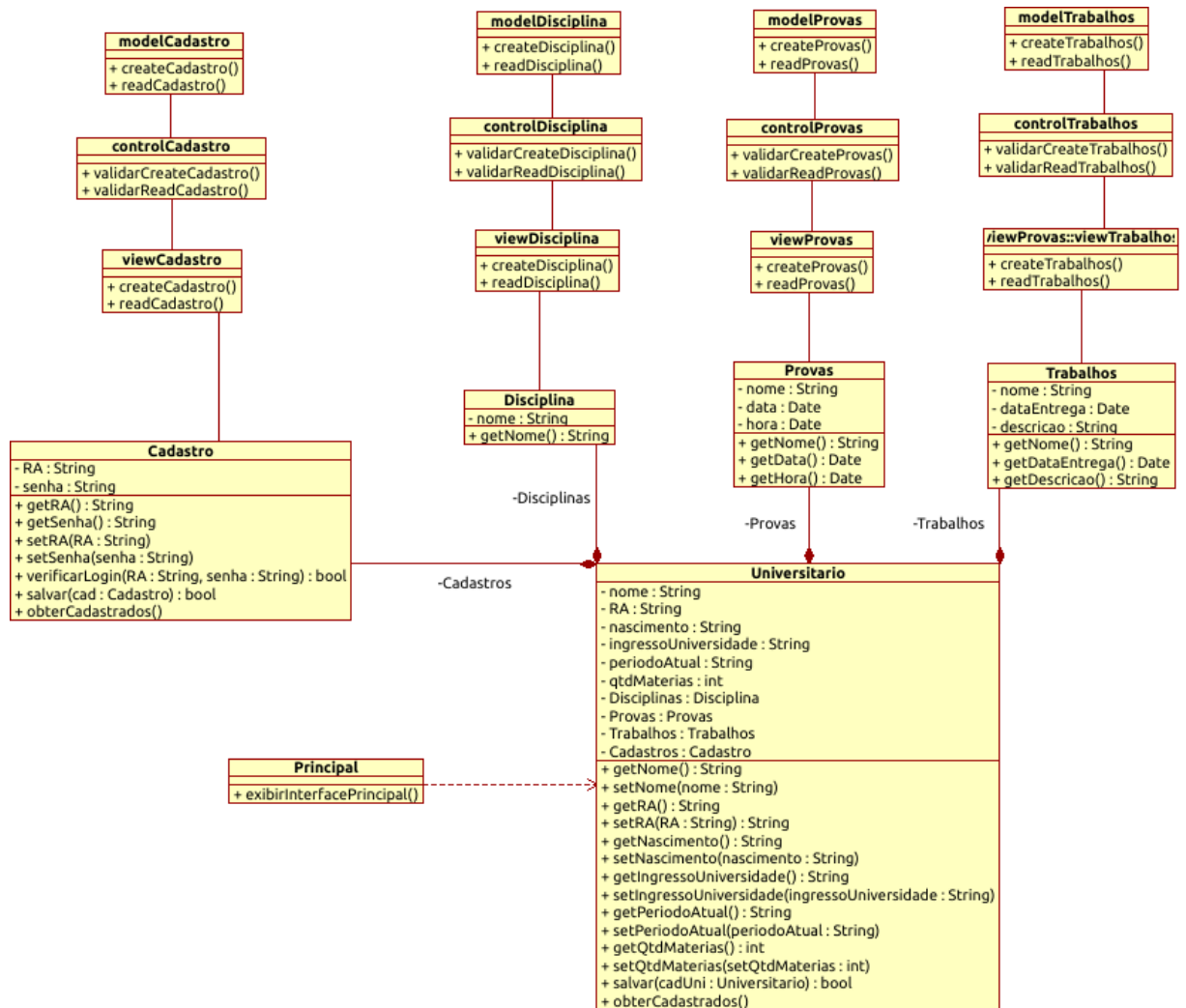
No contexto deste relatório, os diagramas UML desempenham um papel fundamental ao fornecer uma representação visual das diversas interações, estruturas e comportamentos envolvidos no projeto de software, proporcionando uma plataforma comum para a comunicação entre os membros da equipe de desenvolvimento.

A seguir, é mostrado os diagramas de classe, estados, casos de uso, atividades e de sequência do Planner universitário:

2.4.1 Diagrama de Classe

O diagrama de classes do Planner é uma representação visual da estrutura de classes e das relações entre essas classes no sistema do Planner Universitário.

Figura 1 – Diagrama de classe



Fonte: Autoria própria (2023)

Quadro 1 – Descrição das classes

Classe	Descrição
Principal	Esta classe é responsável por gerenciar a interação entre os diferentes componentes gráficos do software
Universitário	Esta classe representa os universitários que utilizam a aplicação para gerenciar suas atividades acadêmicas, junto de seus atributos característicos
Cadastro	Esta classe representa o cadastro propriamente dito de um universitário
Disciplina	Esta classe representa as disciplinas que o universitário tenha durante sua graduação

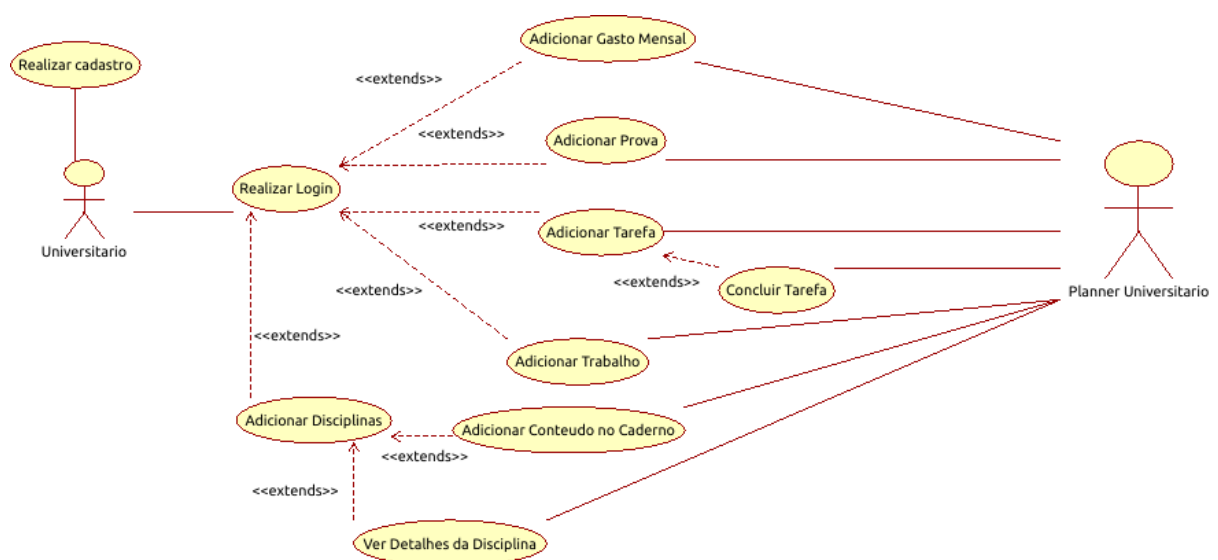
Provas	Esta classe representa as provas que o universitário possa ter e que deseja manter controle sobre
Trabalhos	Esta classe representa os trabalhos que o universitário possa ter e que deseja manter controle sobre
Model	Esta classe gerencia dados acadêmicos, como disciplinas e tarefas
Control	Esta classe coordena interações do universitário, processa entradas e atualiza o modelo
View	Esta classe apresenta informações ao usuário e coleta entradas, facilitando a interação com o Planner Universitário

Fonte: Autoria própria (2023)

2.4.2 Diagrama de Caso de Uso

O diagrama de caso de uso do Planner é uma representação visual que destaca as interações entre os universitários, delineando as principais funcionalidades oferecidas pela aplicação. Este diagrama é essencial para compreender como os universitários interagem com o sistema, identificando suas necessidades e objetivos.

Figura 2 – Diagrama de Caso de Uso



Fonte: Autoria própria (2023)

Quadro 2 – Descrição do caso de uso Realizar Cadastro

Nome Caso de Uso	Realizar Cadastro
Atores	Universitário, Planner Universitário
Finalidade	Realizar o cadastro de um universitário no Planner
Descrição	O próprio usuário (universitário) irá inserir seus dados no Planner
Tabelas Manipuladas	Universitário

Fonte: Autoria própria (2023)

Quadro 3 – Descrição do caso de uso Realizar Login

Nome Caso de Uso	Realizar Login
Atores	Universitário, Planner Universitário
Finalidade	Realizar o login do universitário no sistema
Descrição	O próprio usuário (universitário) irá inserir seus dados na tela de login do Planner para acessar o sistema
Tabelas Manipuladas	Universitário

Fonte: Autoria própria (2023)

Quadro 4 – Descrição do caso de uso Adicionar Gasto Mensal

Nome Caso de Uso	Adicionar Gasto Mensal
Atores	Universitário, Planner Universitário
Finalidade	Inserir o gasto de certo mês no Planner
Descrição	O próprio usuário (universitário) irá inserir seus gastos mensais para ter o somatório dos mesmos
Tabelas Manipuladas	Universitário

Fonte: Autoria própria (2023)

Quadro 5 - Descrição do caso de uso Adicionar Prova

Nome Caso de Uso	Adicionar Prova
Atores	Universitário, Planner Universitário
Finalidade	Inserir provas no Planner
Descrição	O próprio usuário (universitário) irá inserir suas provas, junto com seus respectivos dados
Tabelas Manipuladas	Provas

Fonte: Autoria própria (2023)

Quadro 6 - Descrição do caso de uso Adicionar Tarefa

Nome Caso de Uso	Adicionar Tarefa
Atores	Universitário, Planner Universitário
Finalidade	Inserir tarefas no Planner
Descrição	O próprio usuário (universitário) irá inserir suas tarefas pendentes no Planner
Tabelas Manipuladas	Universitário

Fonte: Autoria própria (2023)

Quadro 7 - Descrição do caso de uso Concluir Tarefa

Nome Caso de Uso	Concluir Tarefa
Atores	Universitário, Planner Universitário
Finalidade	Concluir tarefas pendentes que estão inseridas no Planner
Descrição	O próprio usuário (universitário) irá marcar como concluídas tarefas que estavam pendentes a serem realizadas e que agora já estão concluídas
Tabelas Manipuladas	Universitário

Fonte: Autoria própria (2023)

Quadro 8 - Descrição do caso de uso Adicionar Trabalho

Nome Caso de Uso	Adicionar Trabalho
Atores	Universitário, Planner Universitário
Finalidade	Inserir trabalhos atribuídos ao universitário no Planner
Descrição	O próprio usuário (universitário) irá os dados de seus trabalhos que lhe foram atribuídos
Tabelas Manipuladas	Trabalhos

Fonte: Autoria própria (2023)

Quadro 9 - Descrição do caso de uso Adicionar Disciplinas

Nome Caso de Uso	Adicionar Disciplinas
Atores	Universitário, Planner Universitário
Finalidade	Inserir as disciplinas que o universitário possui no Planner
Descrição	O próprio usuário (universitário) irá inserir as disciplinas em que está matriculado no Planner
Tabelas Manipuladas	Disciplina

Fonte: Autoria própria (2023)

Quadro 10 - Descrição do caso de uso Adicionar Conteúdo no Caderno

Nome Caso de Uso	Adicionar Conteúdo no Caderno
Atores	Universitário, Planner Universitário
Finalidade	Inserir o conteúdo das aulas da disciplina no Planner
Descrição	O próprio usuário (universitário) irá inserir o conteúdo que aprendeu nas aulas da disciplina no caderno
Tabelas Manipuladas	Disciplina

Fonte: Autoria própria (2023)

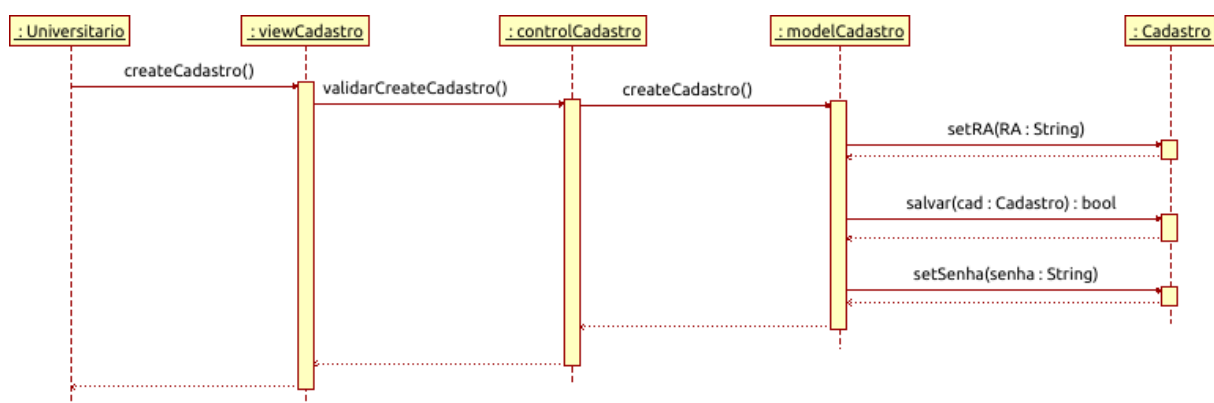
Quadro 11 - Descrição do caso de uso Ver Detalhes da Disciplina

Nome Caso de Uso	Ver Detalhes da Disciplina
Atores	Universitário, Planner Universitário
Finalidade	Ver os detalhes das disciplinas inseridas no Planner
Descrição	O próprio usuário (universitário) irá verificar os detalhes de cada disciplina em que está matriculado
Tabelas Manipuladas	Disciplina

Fonte: Autoria própria (2023)

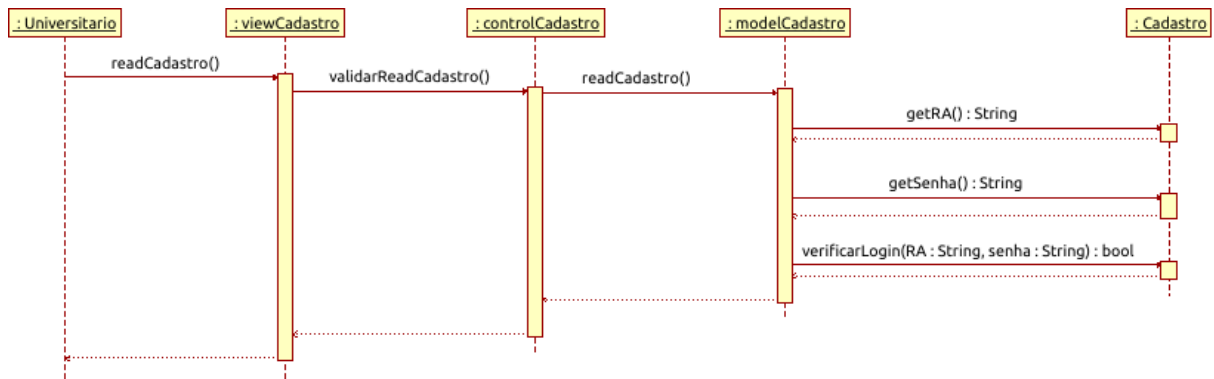
2.4.3 Diagrama de Sequência

O diagrama de sequência do Planner oferece uma visão dinâmica das interações entre os diferentes objetos no sistema durante a execução de cenários específicos. Este tipo de diagrama é de suma importância para entender como as classes e objetos colaboram ao longo do tempo para realizar tarefas específicas.

Figura 3 – Diagrama de Sequência Cadastrar Universitário

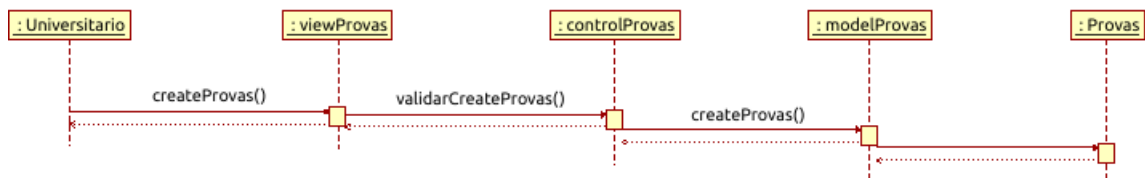
Fonte: Autoria própria (2023)

Figura 4 - Diagrama de Sequência Realizar Login



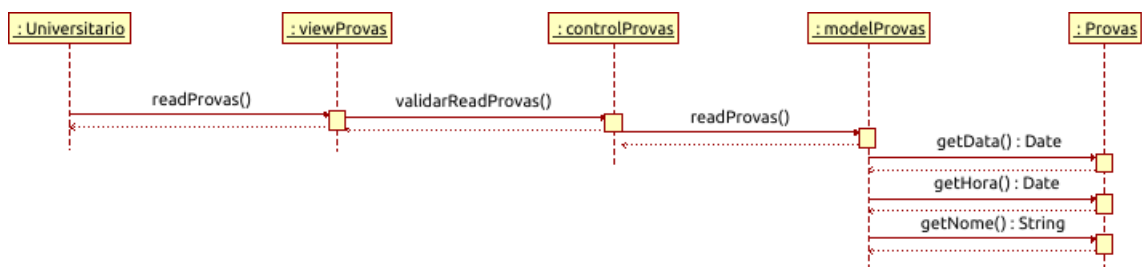
Fonte: Autoria própria (2023)

Figura 5 - Diagrama de Sequência Adicionar Prova



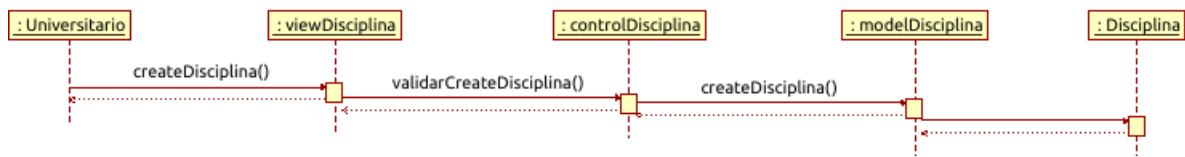
Fonte: Autoria própria (2023)

Figura 6 - Diagrama de Sequência Ler Prova



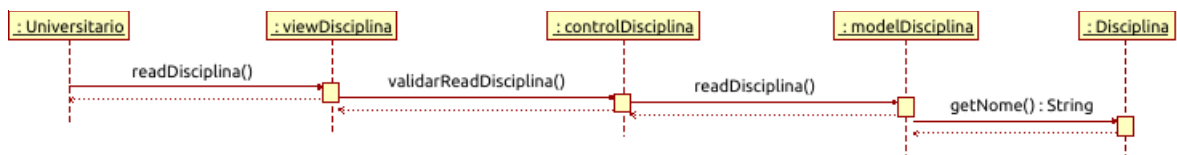
Fonte: Autoria própria (2023)

Figura 7 - Diagrama de Sequência Adicionar Disciplina



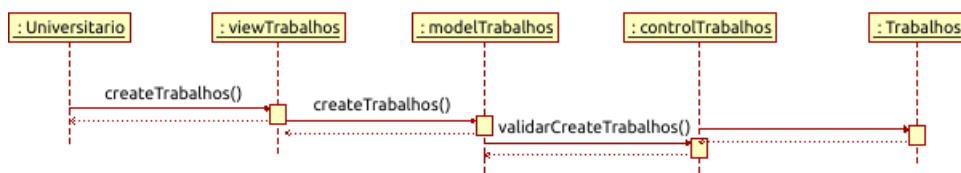
Fonte: Autoria própria (2023)

Figura 8 - Diagrama de Sequência Ler Disciplina



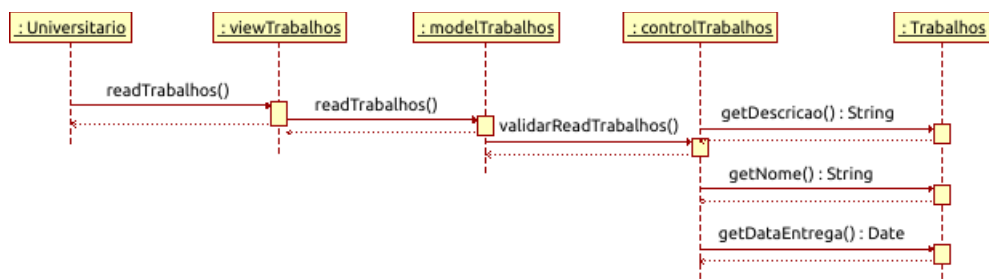
Fonte: Autoria própria (2023)

Figura 9 - Diagrama de Sequência Adicionar Trabalho



Fonte: Autoria própria (2023)

Figura 10 - Diagrama de Sequência Ler Trabalho



Fonte: Autoria própria (2023)

2.4.4 Diagrama de Estados

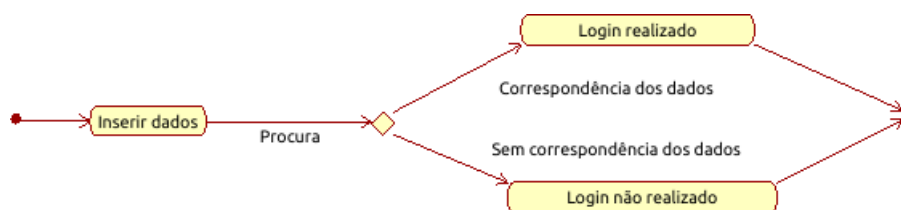
O diagrama de estados do Planner oferece uma representação visual das diferentes condições e transições que os objetos no sistema podem experimentar ao longo do tempo. Este tipo de diagrama é essencial para compreender o comportamento dinâmico de entidades específicas, como disciplinas, tarefas e usuários, à medida que evoluem através de diferentes estados.

Figura 11 - Diagrama de Estados Cadastrar Universitário



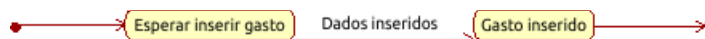
Fonte: Autoria própria (2023)

Figura 12 - Diagrama de Estados Realizar Login



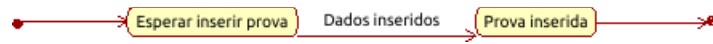
Fonte: Autoria própria (2023)

Figura 13 - Diagrama de Estados Adicionar Gasto Mensal



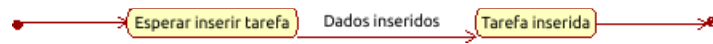
Fonte: Autoria própria (2023)

Figura 14 - Diagrama de Estados Adicionar Prova



Fonte: Autoria própria (2023)

Figura 15 - Diagrama de Estados Adicionar Tarefa



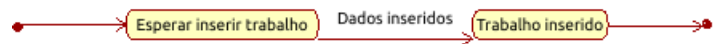
Fonte: Autoria própria (2023)

Figura 16 - Diagrama de Estados Concluir Tarefa



Fonte: Autoria própria (2023)

Figura 17 - Diagrama de Estados Adicionar Trabalho



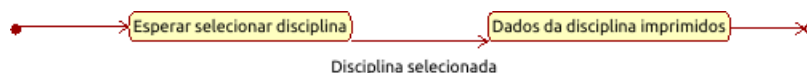
Fonte: Autoria própria (2023)

Figura 18 - Diagrama de Estados Adicionar Disciplina



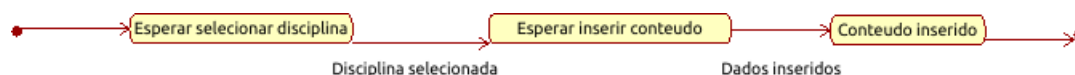
Fonte: Autoria própria (2023)

Figura 19 - Diagrama de Estados Ler Disciplina



Fonte: Autoria própria (2023)

Figura 20 - Diagrama de Estados Inserir Conteúdo

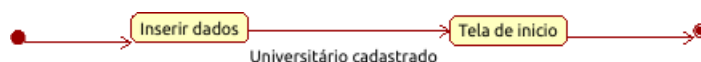


Fonte: Autoria própria (2023)

2.4.5 Diagrama de Atividades

O diagrama de atividade do Planner é uma representação visual que destaca as atividades executadas no sistema, proporcionando uma visão clara dos fluxos de trabalho e processos. Esse tipo de diagrama é essencial para compreender como as ações e tarefas são coordenadas dentro do aplicativo, mostrando a sequência de atividades e as decisões tomadas ao longo do caminho.

Figura 21 - Diagrama de Atividades Cadastrar Universitário



Fonte: Autoria própria (2023)

Figura 22 - Diagrama de Atividades Realizar Login



Fonte: Autoria própria (2023)

Figura 23 - Diagrama de Atividades Inserir Gasto



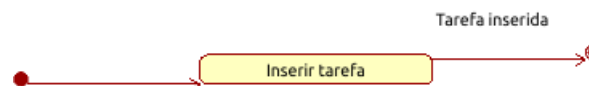
Fonte: Autoria própria (2023)

Figura 24 - Diagrama de Atividades Inserir Prova

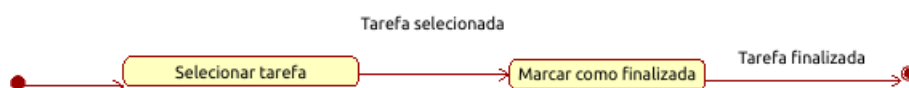


Fonte: Autoria própria (2023)

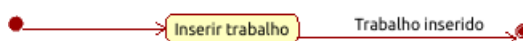
Figura 25 - Diagrama de Atividades Inserir Tarefa



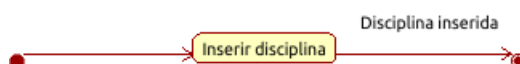
Fonte: Autoria própria (2023)

Figura 26 - Diagrama de Atividades Concluir Tarefa

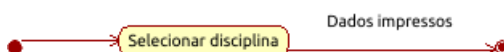
Fonte: Autoria própria (2023)

Figura 27 - Diagrama de Atividades Inserir Trabalho

Fonte: Autoria própria (2023)

Figura 28 - Diagrama de Atividades Inserir Disciplina

Fonte: Autoria própria (2023)

Figura 29 - Diagrama de Atividades Ver Disciplina

Fonte: Autoria própria (2023)

Figura 30 - Diagrama de Atividades Inserir Conteúdo no Caderno



Fonte: Autoria própria (2023)

2.5 Polimorfismo

A interface `Gasto` estabelece um contrato que as classes que a implementam devem seguir. Neste contexto, a única exigência é a implementação do método `calcularTotalGasto()`.

Cada classe que implementa a interface `Gasto`, como `Aluguel`, `Luz`, `Internet`, etc., oferece uma implementação específica para o método `calcularTotalGasto()`. Cada tipo de gasto, portanto, tem conhecimento sobre como calcular seu próprio total.

O método `calcularTotalGasto()` na classe principal (`GastoUniversitarioGUI`) utiliza a interface `Gasto` de maneira polimórfica. Ele cria instâncias de diferentes tipos de gastos (`Aluguel`, `Luz`, etc.) e chama o método `calcularTotalGasto()` em cada uma delas sem se preocupar com a implementação específica.

Essa abordagem é viável porque todas essas classes seguem a mesma interface, e o conceito de polimorfismo possibilita que o método seja invocado de maneira uniforme, proporcionando flexibilidade e reusabilidade no código.

2.6 Tratamento de Exceção Personalizada

O tratamento de exceção personalizada é uma prática fundamental para garantir a robustez e a confiabilidade de um sistema. No contexto das classes Java mencionadas, diversas situações foram contempladas para assegurar uma experiência de usuário consistente e evitar comportamentos inesperados.

2.6.1 CadastroGUI.java - RA já cadastrado e campos vazios

Esta classe aborda a exceção relacionada ao Registro Acadêmico (RA) já existente. Ao tentar cadastrar um universitário com um RA que já está no sistema, uma exceção personalizada

é lançada, informando ao usuário sobre essa duplicidade. Além disso, verificações são realizadas para campos vazios durante o cadastro, garantindo a integridade dos dados.

2.6.2 LoginGUI.java - RA não cadastrado e credenciais inválidas

No contexto do Login, uma exceção é lançada quando o RA fornecido não está cadastrado no sistema. Isso assegura que apenas usuários registrados tenham acesso. Além disso, o tratamento de exceção abrange credenciais inválidas, proporcionando uma resposta adequada ao usuário quando a combinação RA e senha não corresponde a nenhum cadastro válido.

2.6.3 CadastroUniversitarioGUI.java - Verificar campos vazios

O tratamento de exceção personalizada nesta classe concentra-se na prevenção do cadastro de universitários com campos essenciais vazios. Isso evita dados inconsistentes e garante que as informações necessárias sejam fornecidas durante o processo de registro.

2.6.4 ProvasGUI.java - Verificar campos vazios

No contexto das provas, a verificação de campos vazios é essencial para garantir a precisão dos dados inseridos. O tratamento de exceção personalizada nesta classe evita que o usuário prossiga com a operação caso campos obrigatórios não sejam preenchidos, promovendo a integridade dos registros acadêmicos.

3 CONCLUSÃO

Em conclusão, o Planner Universitário foi desenvolvido para facilitar a vida acadêmica dos graduandos, abordando diversas necessidades de organização e gestão. Com funcionalidades como cadastro de universitários, login, controle de gastos mensais, tarefas, disciplinas e provas, o Planner oferece uma solução completa.

A escolha do padrão de design MVC destaca a preocupação com uma estrutura clara e modular, e os diagramas UML proporcionam uma compreensão visual abrangente do sistema, facilitando a comunicação na equipe de desenvolvimento.

O polimorfismo na interface Gasto permite a manipulação flexível de diferentes tipos de gastos, enquanto o tratamento de exceções personalizado contribui para uma experiência do usuário mais estável, evitando problemas inesperados.

Em resumo, o Planner Universitário não apenas simplifica a vida acadêmica, mas também representa uma prática exemplar no desenvolvimento de software, combinando funcionalidades abrangentes, uma arquitetura bem planejada e uma abordagem cuidadosa para exceções.

4 LINK

<https://drive.google.com/drive/folders/1zXEjM5mLDTPCa7-vlhx6vaZdp92ezmOD?usp=sharing>