



# Desenvolvimento de Software 1 - Projeto Flash Pomo

Jala University

David Alex Souza Santos  
Gustavo Henrique Jesus da Silva  
Rhuan Miguel Esteves  
Rinaldo Lira de Albuquerque Lima

# Sumário

Introdução .....	3
BRD .....	4
Desenvolvimento .....	7
Processo de Desenvolvimento .....	16
Daily Meeting Report .....	20
Engenharia de Software .....	42
Tecnologias Utilizadas .....	117
Manual do Usuário .....	119
Mapa do Site .....	132
API Reference .....	133
Schemas .....	134
Purchase Schemas .....	135
PomodoroStory Schemas .....	136
Favorite Schemas .....	137
Box Schemas .....	138
GetByIdResponse .....	139
ResponseCommonDTO .....	140
Flashcard Schemas .....	141
Ensemble Schemas .....	142
User Schemas .....	143
Auth Schemas .....	144
ResponseEntity .....	145
Auth Endpoints .....	146
POST auth/login .....	147
POST auth/register .....	149
User Endpoints .....	151
GET user .....	152
GET user/{userId} .....	153
PUT user/{userId} .....	154
DELETE user/{userId} .....	157
Ensemble Endpoints .....	158
GET ensemble/{ensembleId} .....	159
PUT ensemble/{ensembleId} .....	160

DELETE ensemble/{ensembleId} .....	162
GET ensemble/ .....	163
POST ensemble/ .....	164
Flashcard Endpoints .....	166
GET flashcard/{flashcardId} .....	167
PUT flashcard/{flashcardId} .....	168
DELETE flashcard/{flashcardId} .....	170
GET flashcard/ .....	171
POST flashcard/ .....	172
GET flashcard/ensemble/{ensemble} .....	174
PomodoroStory Endpoints .....	175
GET pomodoro-story/{pomodoroStoryId} .....	176
PUT pomodoro-story/{pomodoroStoryId} .....	177
POST pomodoro-story/{pomodoroStoryId} .....	179
GET pomodoro-story/ .....	181
Favorite Endpoints .....	182
GET favorite/{favoriteld} .....	183
PUT favorite/{favoriteld} .....	184
DELETE favorite/{favoriteld} .....	186
GET favorite/ .....	187
POST favorite/ .....	188
Box Endpoints .....	190
PUT box/{favoriteld} .....	191
DELETE box/{favoriteld} .....	193
GET box/ .....	194
POST box/ .....	195
GET box/{boxId} .....	197
Purchase Endpoints .....	198
GET purchase/{purchaseld} .....	199
PUT purchase/{purchaseld} .....	200
POST purchase/{purchaseld} .....	203
DELETE purchase/{purchaseld} .....	206
GET purchase/ .....	207
Conclusão .....	208
Referencias Bibliográficas .....	209

# Introdução

O aplicativo desenvolvido é voltado para o meio acadêmico, combinando as estratégias de estudo Pomodoro e Flashcards em um único hub. Ele oferece funcionalidades que permitem a criação e edição de flashcards, além de gerenciar o tempo de estudo de forma eficiente através da técnica Pomodoro.

Além disso, o aplicativo conta com funcionalidades de mercado, onde os usuários podem negociar seus flashcards usando uma moeda própria da aplicação. Essa integração de ferramentas proporciona um ambiente de aprendizado dinâmico e interativo, favorecendo o desenvolvimento acadêmico e a troca de conhecimento entre os usuários.

## Objetivo do Relatório

O objetivo deste relatório é documentar a criação, o desenvolvimento e o funcionamento de um software voltado para o contexto acadêmico, abrangendo suas etapas de concepção, implementação, modelagem e sua aplicabilidade no ambiente de estudos.

# BRD

A business requirements document (BRD) é um relatório detalhado que descreve tudo o que um novo projeto requer para obter sucesso. Este documento delineia os objetivos do projeto, o que é esperado ao longo do ciclo de vida do projeto e o que é necessário para alcançar o projeto.

## Resumo Executivo

Este relatório descreve o desenvolvimento de um software acadêmico criado como parte do curso de Desenvolvimento de Software 1 da Jala University. O sistema desenvolvido combina a técnica Pomodoro de gerenciamento de tempo com o uso de flashcards, criando um ambiente de aprendizado dinâmico.

## Objetivo do projeto

O aplicativo permite aos usuários gerenciar suas atividades de estudo com as seguintes funcionalidades:

- Gerenciamento de Usuários: Cadastro, edição e exclusão de usuários.
- Pomodoro: Ativação, pausa e desativação do timer de estudo.
- Relatórios do Pomodoro: Geração e consulta de relatórios baseados na técnica Pomodoro.
- Flashcards: Criação, edição, consulta e exclusão de flashcards.
- Histórico de Flashcards: Adição e remoção de flashcards do histórico.
- Favoritos: Gerenciamento de flashcards favoritos.
- Categorias de Flashcards: Criação e organização de flashcards por categorias.
- Marketplace de Flashcards: Compra e venda de flashcards utilizando uma moeda interna.

## Escopo do projeto

A equipe será composta por um Product Owner, dois desenvolvedores e um Scrum Master, trabalhando sob a metodologia ágil Scrum.

## Requerimento de negócios

Requisito de Negócio	Prioridade	Nível Crítico
Integração de Funcionalidades Pomodoro e Flashcards	1	Crítico
Facilidade de Uso	2	Médio
Escalabilidade	3	Médio
Acessibilidade e Segurança	4	Médio
Mercado de Flashcards	5	Baixo

## Key Stakeholders

Nome	Ocupação
Rhuan Miguel Esteves	Product Owner
David Alex Souza Santos	Desenvolvedor Front
Gustavo Henrique de Jesus da Silva	Desenvolvedor Back
Rinaldo Gabriel Lira de Albuquerque Lima	Scrum Master

## Restrições de Projeto

<b>Restrição</b>	<b>Descrição</b>
Prazo	Completo em 8 semanas, devido ao tempo de conclusão do curso
Recursos Humanos	A equipe tem apenas 4 membros para o desenvolvimento do projeto
Orçamento	O projeto deve ser desenvolvido com ferramentas gratuitas ou de baixo custo, sem despesas adicionais
Tecnologia	A aplicação deve ser desenvolvida com tecnologias já conhecidas pelos integrantes da equipe
Escopo	As funcionalidades, como sign up, log in, flashcards e pomodoro devem ser entregues como mínimo viável

## Custo Benefício

<b>Benefício</b>	<b>Descrição</b>
Desenvolvimento	Experiência prática em desenvolvimento de software e metodologias ágeis, alinhando-se ao mercado de trabalho.
Treinamento e Aprendizado	Facilita a organização, produtividade e retenção de conhecimento, beneficiando usuários finais.

# Desenvolvimento

## Projeto

Este projeto visa desenvolver um aplicativo voltado para o meio acadêmico, combinando as estratégias de estudo Pomodoro e Flashcards em uma única plataforma. O objetivo é criar uma solução completa que permita aos usuários gerenciar seu tempo de estudo de forma eficiente e organizar seus materiais de aprendizado de maneira efetiva.

Uma das principais funcionalidades do aplicativo será a integração da técnica Pomodoro. Essa técnica de gerenciamento de tempo ajudará os usuários a manterem sua concentração e produtividade durante os momentos de estudo. Além disso, o aplicativo contará com ferramentas de criação e edição de flashcards, uma estratégia amplamente utilizada por estudantes para memorização e revisão de conteúdo.

Outra funcionalidade chave será o mercado interno, onde os usuários poderão negociar seus flashcards usando uma moeda própria da plataforma. Essa integração de um ambiente de compra e venda de material de estudo criará um ecossistema dinâmico de aprendizado, incentivando a troca de conhecimento e a colaboração entre os usuários.

Ao combinar estratégias de estudo comprovadas, gerenciamento de tempo e um mercado interno, o aplicativo se posicionará como uma solução inovadora e completa para as necessidades do meio acadêmico. Espera-se que essa integração de ferramentas melhore o desempenho e o engajamento dos estudantes, além de fomentar a colaboração e o compartilhamento de conhecimento entre os usuários.

O desenvolvimento deste aplicativo visa atender às demandas do público acadêmico, oferecendo uma plataforma que integre eficientemente as principais estratégias de estudo e aprendizado.

## Objetivo Geral

O objetivo geral deste projeto é desenvolver um aplicativo que combine as estratégias de estudo Pomodoro e Flashcards, oferecendo aos usuários do meio acadêmico uma plataforma integrada e eficiente para gerenciar seu tempo de estudo e organizar seus materiais de aprendizado.

## Objetivos Específicos

- 1. Implementar a funcionalidade de Pomodoro:** integrar a técnica Pomodoro no aplicativo, permitindo que os usuários gerenciem seu tempo de estudo de forma eficiente, alternando períodos de trabalho intenso e breves intervalos.
- 2. Desenvolver ferramentas de criação e edição de Flashcards:** criar uma interface intuitiva para que os usuários possam criar, armazenar, editar e organizar seus flashcards de forma eficaz.
- 3. Estabelecer um Mercado Interno de Flashcards:** implementar um sistema de compra e venda de flashcards entre os usuários, utilizando uma moeda própria da aplicação, a fim de fomentar a troca de conhecimento e a colaboração na comunidade acadêmica.
- 4. Integrar as funcionalidades de Pomodoro e Flashcards:** garantir uma experiência fluida e integrada entre as estratégias de estudo Pomodoro e o uso de Flashcards, otimizando o fluxo de aprendizado dos usuários.
- 5. Desenvolver uma Interface Intuitiva e Acessível:** criar uma interface de usuário intuitiva, responsiva e acessível, de modo a facilitar a adoção e a usabilidade do aplicativo por parte do público-alvo.
- 6. Promover a Colaboração e Compartilhamento de Conhecimento:** Através do Mercado Interno de Flashcards, incentivar a colaboração e o compartilhamento de conhecimento entre os usuários do aplicativo, fomentando o aprendizado mútuo.

## Mercado

### Análise de Mercado

#### Público Alvo

O aplicativo tem como público-alvo estudantes de ensino médio ao nível universitário, abrangendo:

Perfil	Descrição
Estudantes do Ensino Médio	Alunos que buscam ferramentas para melhorar sua organização e produtividade nos estudos, preparando-se para o ambiente universitário.
Estudantes Universitários	Alunos de graduação e pós-graduação que precisam gerenciar seu tempo de estudo e organizar seu material de aprendizado de forma eficiente.
Estudantes Autônomos	Indivíduos que estudam de forma independente, como preparatórios para concursos ou cursos livres, e necessitam de estratégias de estudo eficazes.

Esse público-alvo foi selecionado com base na necessidade de ferramentas que combinem eficientemente a gestão do tempo de estudo (técnica Pomodoro) e a organização de conteúdo (flashcards), além da oportunidade de compartilhar e trocar conhecimento por meio do mercado interno de flashcards, atendendo às demandas dos estudantes desde o ensino médio até o nível universitário.

### Análise da Concorrência

Ao analisar o mercado, foram identificados alguns aplicativos e plataformas que oferecem funcionalidades semelhantes ao aplicativo proposto, embora de forma fragmentada. Alguns dos principais concorrentes são:

Concorrente	Funcionalidades	Pontos Fortes	Pontos Fracos
Anki	- Criação e revisão de flashcards - Gerenciamento de deck de flash cards	- Ampla base de usuários - Algoritmo de repetição espaçada eficiente	- Não possui integração com a técnica Pomodoro - Interface pouco intuitiva
Forest	- Técnica Pomodoro - Gerenciamento de tempo de estudo	- Interface amigável - Gamificação da técnica Pomodoro	- Não possui funcionalidades de flashcards - Apenas gerenciamento de tempo, sem organização de conteúdo
Quizlet	- Criação e revisão de flashcards - Compartilhamento de flashcards	- Ampla biblioteca de flashcards compartilhados - Ferramentas de aprendizado interativas	- Não possui integração com a técnica Pomodoro - Foco apenas na funcionalidade de flashcards, sem gerenciamento de tempo

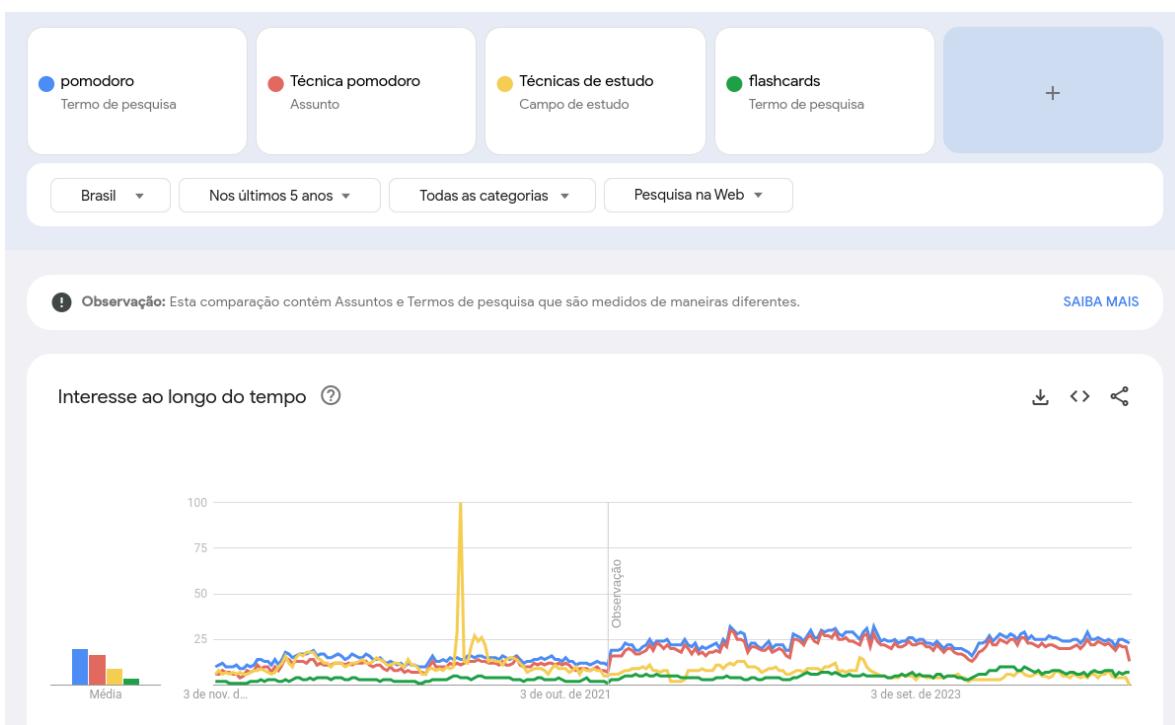
Essa análise da concorrência permite identificar as principais lacunas no mercado, as quais o aplicativo proposto pode preencher ao combinar as funcionalidades de gerenciamento de tempo (Pomodoro) e organização de conteúdo (flashcards), além de oferecer um mercado interno de flashcards para a troca de conhecimento entre os usuários.

#### Tendências:

- **Crescente demanda por ferramentas de gerenciamento de tempo e organização de estudo:** Estudantes e profissionais do meio acadêmico buscam cada vez mais soluções integradas que os ajudem a gerenciar seu tempo de estudo e aprendizado de forma eficiente.

- **Adoção da técnica Pomodoro no ambiente acadêmico:** A técnica Pomodoro vem ganhando popularidade entre estudantes e pesquisadores, que a utilizam para aumentar sua produtividade durante os momentos de estudo e trabalho.
- **Necessidade de compartilhamento e colaboração no aprendizado:** Existe uma demanda crescente por plataformas que permitam a troca de conhecimento e a colaboração entre estudantes e profissionais do meio acadêmico.

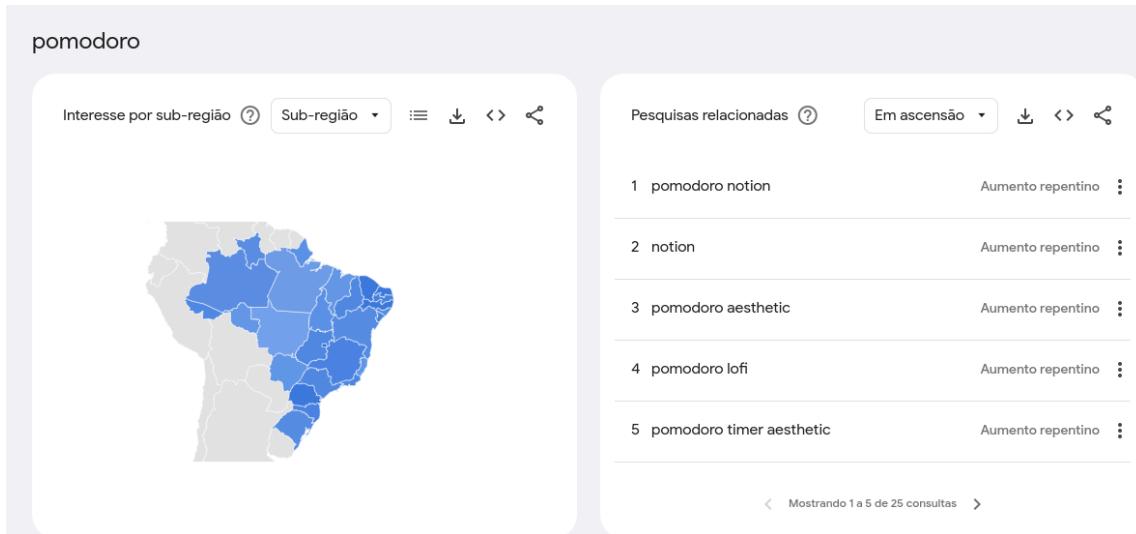
Usando alguns termos para pesquisa no Google Trends vemos que há uma busca ativa sobre os termos chaves como: pomodoro, flashcards, etc



grafico\_trends.png

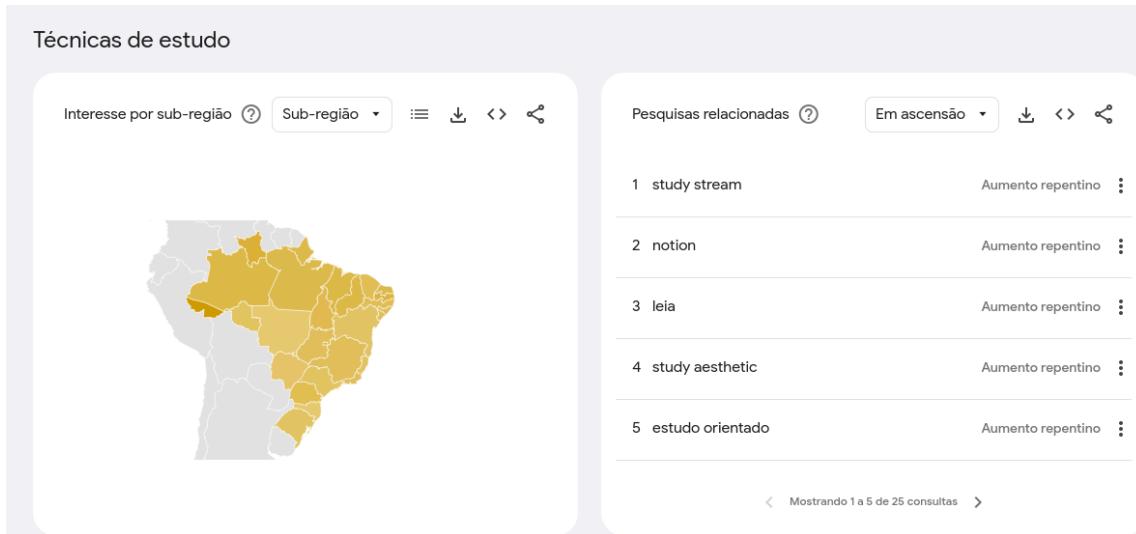
### Gráficos por termos:

- Pomodoro



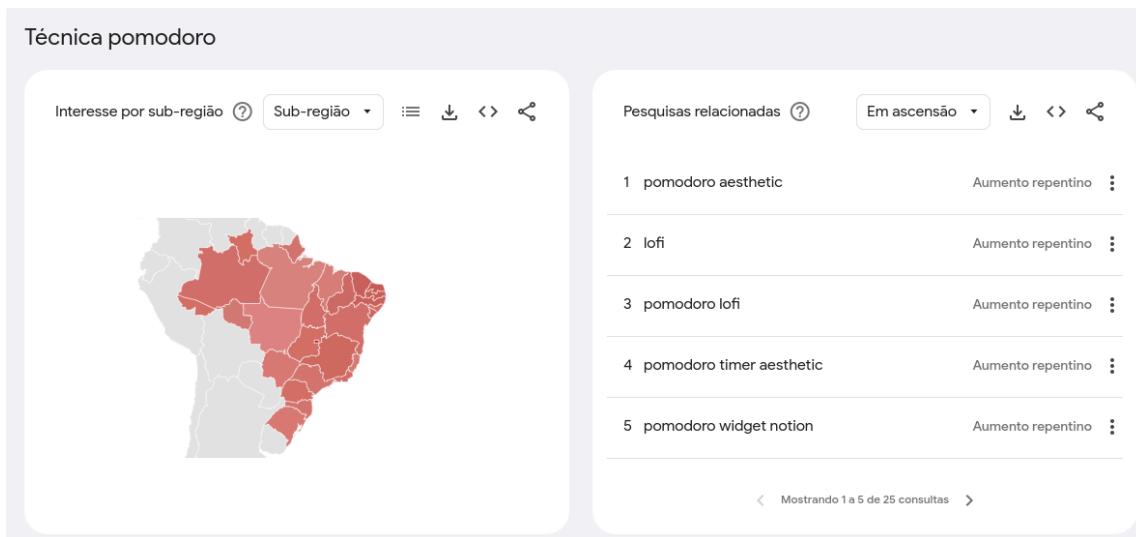
pomodoro.png

- Técnica Estudo



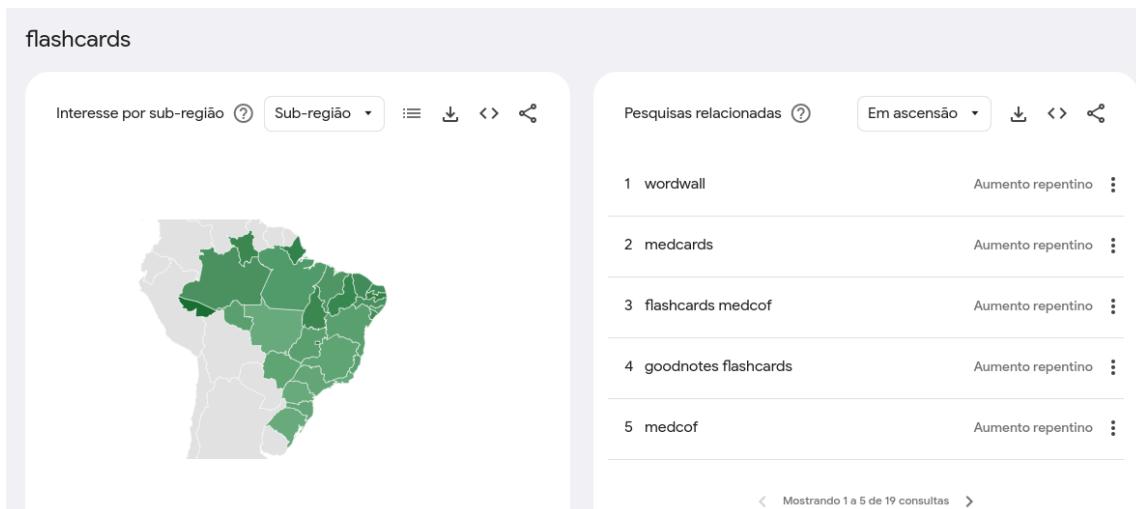
tecnica\_estudo.png

- Técnica Pomodoro



tecnica\_pomodoro.png

- Flashcards



flashcards.png

## Oportunidades:

- **Integração de ferramentas de gerenciamento de tempo e organização de conteúdo:** O mercado carece de soluções que combinem de forma eficiente e fluida a técnica Pomodoro e a utilização de flashcards, atendendo às necessidades dos estudantes.
- **Criação de um ecossistema de compartilhamento de conhecimento:** O desenvolvimento de um mercado interno de flashcards, onde os usuários possam comprar, vender e trocar materiais de estudo, representa uma oportunidade

significativa para fomentar a colaboração e a troca de conhecimento no meio acadêmico.

- **Diferenciação por meio da integração de funcionalidades:** A maioria das soluções atuais se concentra em apenas uma das necessidades (gerenciamento de tempo ou organização de flashcards), deixando uma lacuna a ser preenchida por um aplicativo que integre essas funcionalidades de forma completa.

Essas tendências e oportunidades de mercado indicam que o aplicativo proposto pode preencher uma demanda relevante do público-alvo, oferecendo uma solução integrada e inovadora para o gerenciamento do tempo de estudo e a organização do conteúdo de aprendizado, além de promover a colaboração e a troca de conhecimento entre os usuários.

## **Posicionamento e Diferencial Competitivo**

Com base na análise do mercado e das necessidades do público-alvo, o aplicativo proposto se posicionará como uma solução integrada e inovadora, com os seguintes diferenciais competitivos:

1. **Combinação de Pomodoro e Flashcards:** O aplicativo será a única solução do mercado que combina de forma fluida e eficiente a técnica Pomodoro para gerenciamento do tempo de estudo e as funcionalidades de criação, organização e revisão de flashcards. Essa integração permitirá que os usuários otimizem seu processo de aprendizado.
2. **Mercado Interno de Flashcards:** A implementação de um mercado interno, onde os usuários poderão comprar, vender e trocar flashcards usando uma moeda própria da plataforma, será um diferencial competitivo. Essa funcionalidade fomentará a colaboração e a troca de conhecimento entre os estudantes e profissionais do meio acadêmico.
3. **Foco no Público Acadêmico:** Ao concentrar-se especificamente nas necessidades dos estudantes do ensino médio, universitários e profissionais do meio acadêmico, o aplicativo se diferenciará de soluções genéricas de gerenciamento de tempo e flashcards, oferecendo uma experiência personalizada e adaptada às demandas desse público-alvo.
4. **Interface Intuitiva e Acessível:** O aplicativo priorizará o desenvolvimento de uma interface de usuário intuitiva, responsiva e acessível, facilitando a adoção e a

usabilidade por parte dos estudantes e profissionais do meio acadêmico, independentemente de seu nível de familiaridade com esse tipo de ferramenta.

Esse posicionamento estratégico, aliado aos diferenciais competitivos do aplicativo, permitirá que o produto se destaque no mercado e se torne uma solução de referência para o gerenciamento eficiente do tempo de estudo e a organização do conteúdo de aprendizado no meio acadêmico.

# Processo de Desenvolvimento

## User Stories

ID	Descrição	Prioridade
US 001	Como usuário, eu quero criar uma conta, para que eu possa fazer login e utilizar o software.	Alta
US 002	Como usuário, eu quero fazer login no sistema, para que eu possa acessar minhas informações e funcionalidades.	Alta
US 003	Como usuário, eu quero realizar o logout na minha conta, para que eu possa encerrar minha sessão de forma segura e proteger minhas informações pessoais.	Média
US 004	Como usuário eu quero editar meus dados na minha conta de usuário.	Média
US 005	Como usuário, eu quero poder excluir minha conta, para que eu possa remover meu perfil do sistema e minhas informações pessoais.	Baixa
US 006	Como usuário, eu quero poder ativar o método Pomodoro no sistema, para que eu possa gerenciar melhor meu tempo e produtividade durante minhas tarefas.	Alta
US 007	Como usuário, eu quero poder pausar o método Pomodoro no sistema, caso eu não queira mais utilizá-lo.	Média
US 008	Como usuário eu quero criar meus flashcards.	Alta

US 00 9	Como usuário eu quero editar meus flashcards.	Médi a
US 010	Como usuário eu quero excluir meus flashcards.	Médi a
US 011	Como usuário eu quero criar ensembles.	Médi a
US 012	Como usuário eu quero editar minhas ensembles.	Médi a
US 013	Como usuário eu quero excluir minhas ensembles.	Baixa

## Organizaçao da equipe

Ocupação	Descrição
Product Owner	Responsável por gerenciar o produto e garantir que as funcionalidades atendam às necessidades dos usuários.
Desenvolvedor Front	Responsável pelo desenvolvimento da interface do usuário, garantindo uma experiência intuitiva e agradável.
Desenvolvedor Back	Responsável pelo desenvolvimento da lógica e funcionalidades do aplicativo, implementando as regras de negócio.
Scrum Master	Responsável por facilitar o processo de desenvolvimento, removendo impedimentos e aplicando as práticas ágeis da equipe.
Customer	Representa os usuários finais do aplicativo, cujas necessidades e feedback são fundamentais para o desenvolvimento do produto.

## Organização Inicial

Nome	Ocupação
David Alex Souza Santos	Desenvolvedor Front
Gustavo Henrique de Jesus da Silva	Desenvolvedor Back
Rhuan Miguel Esteves	Desenvolvedor Front
Rinaldo Gabriel Lira de Albuquerque Lima	Desenvolvedor Back

# Daily Meeting Report

Tabela 1: Daily Meeting Report - 12/11/2024

Group Name	Module Name	Members	Role performed	o que fez hoje?	O que você fará amanhã?	existem impedimentos em seu caminho?	Outro impedimento?
GRUPO 4 - FLASHPOMO	DS 1	Rinaldo	Scrum Master	Terminei a modelagem de dados 1/2	Vou estudar sobre react	não	não
		David	Developer	Terminei a modelagem de dados 2/2	Pesquisarei ferramentas p/ projeto e estudar react	não	não
		Gustavo	Developer	Finalizei a parte 1/3 da prototipagem	Revisão do Trrello e estudar sobre Spring Boot	não	não
		Rhuan	Developer	Finalizei a parte 2/3 e 3/3 da prototipagem	Vou estudar sobre react	não	não
Sprint #	1.0						

Tabela 2: Daily Meeting Report - 13/11/2024

Group Name	Module Name	Members	Role performed	o que fez hoje?	O que você fará amanhã?	existem impedimentos em seu caminho?	Outro impedimento?
GRUPO 4 - FLASHPOMO	DS 1	Rinaldo	Scrum Master	Comecei o estudo sobre react e ajustei parte do backlog	Estudarei sobre html e css	não	=VLOOKUP(B19, 'Hoja 1'! B3: G9, 5, FALSE)
		David	Developer	Pesquisei ferramentas p/ projeto e estudei react	Estudarei sobre react	não	=VLOOKUP(B20, 'Hoja 1'! B3: G9, 5, FALSE)
		Gustavo	Developer	Revisão do Trelo e estudar sobre Spring Boot	Desenvolvedor o frontend de login e sign up	não	=VLOOKUP(B21, 'Hoja 1'! B3: G9, 5, FALSE)
		Rhuan	Developer	Arrumei o backlog	programar a tela de login	não	=VLOOKUP(B22, 'Hoja 1'! B3: G9, 5, FALSE)
Sprint #	1.0						

Tabela 3: Daily Meeting Report - 14/11/2024

Group Name	Module Name	Members	Role performed	o que fez hoje?	O que você fará amanhã?	existem impedimentos em seu caminho?	Outro impedimento?
GRUPO 4 - FLA SHPO MO	DS 1	Rinaldo	Scrum Master	Estudei react html e css, aprendi o básico	Irei estudar melhor e criar o banco de dados	não	não
		David	Developer	Estudei React e spring	estudar spring e criar tabelas	não	não
		Gustavo	Developer	Desenvolvedor o frontend de login e signup	Implementação do cadastramento	não	Não
		Rhuan	Developer	Estudei React e criei o frontend	começarei a desenvolver a api	não	não
Sprint #	1.0						

Tabela 4: Daily Meeting Report - 15/11/24

Group Name	Module Name	Members	Role performed	o que fez hoje?	O que você fará amanhã?	existem impedimentos em seu caminho?	Outro impedimento?
GRUPO 4 - FLA SHPOMO	DS 1	Rinaldo	Scrum Master	terminei o banco de dados JPA	irei estudar mais sobre a linguagem	não	não
		David	Developer	Continuei a criação das tabelas com spring	estudar mais sobre spring boot	-	-
		Gustavo	Developer	Continuei a implementação do cadastro	Implementação do login	não	não
		Rhu an	Developer	continuei a programação	irei estudar as linguagens	não	não
Sprint #	1.0						

Tabela 5: Daily Meeting Report - 18/11/2024

Group Name	Module Name	Members	Role performed	o que fez hoje?	O que você fará amanhã?	existem impedimentos em seu caminho?	Outro impedimento?
GRUPO 4 - FLA SHPOMO	DS 1	Rinaldo	Developer	Estudei Java	Ajudar na criação das rotas	não	não
		David	Developer	Estudei Java	Criar rotas	não	não
		Gustavo	Developer	Finalizei a criação da API	Ajudar na criação das rotas	não	não
		Rhuan	Scrum master	Estudei Java	Ajudar na criação das rotas	não	não
Sprint #	1.0		Revisão do projeto para dar continuidade				

Tabela 6: Daily Meeting Report - 19/11/2024

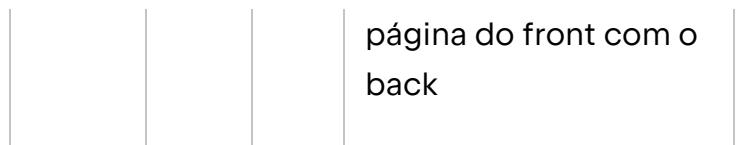
Group Name	Module Name	Members	Role performed	o que fez hoje?	O que você fará amanhã?	existem impedimentos em seu caminho?	Outro impedimento?
GRUPO 4 - FLASHPOMO	DS 1	Rinaldo	Developer	Documentação	Melhorarei a documentação	não	não
		David	Developer	Ajudei na documentação	Continuar o front end	não	não
		Gustavo	Developer	Ajudei na documentação	Fazer o UC01	não	não
		Rhuan	Scrum Master	Ajudei na documentação	Continuar o front end	não	não
Sprint #	1.0		Definir as funções no projeto para seguir implementando as funcionalidades, front, back e doc				

Tabela 7: Daily Meeting Report - 20/11/2024

Group Name	Module Name	Members	Role performed	o que fez hoje?	O que você fará amanhã?	existem impedimentos em seu caminho?	Outro impedimento?
GRUPO 4 - FLASHPOMO	DS 1	Rinaldo	Developer	Documentação	Melhorar ei a documentação	sim, intern et	não
		David	Developer	Ajudei na documentação	Continuar o front end	não	não
		Gustavo	Developer	Endpoint de exclusão do usuário	Endpoint de exclusão do usuário	não	não
		Rhuan	Scrum Master	Continuei no front end	Continuar o front end	não	não
Sprint #	1.0		Colocar em ponto a continuação do projeto e verificar andamento				

Tabela 8: Daily Meeting Report - 21/11/2024

Grou p Na me	Mo dul e N am e	Me mb ers	Role performed	o que fez ho je?	O que v ocê far á aman hã?	existem i mpedim entos em seu cami nho?	Outro impe dime nto?
GRUP O 4 - FLAS HPO MO	DS 1	Rin ald o	Developer	Docu menta ção	Auxiliar ei na in tegraçã o do pe rfil	sim, inter net	não
		Da vid	Developer	Ajudei na do cume ntaçã o	Termin ar o fro nt end	não	não
		Gu sta vo	Developer	End p oint d e ediç ão	Integra ção do perfil d e usuár o	não	não
		Rh ua n	Scrum Master	Conti nuei n o fron t end	Termin ar o fro nt end	não	não
Sprint #	1.0		Colocar em ponto a co ntinuação do projeto e verificar o andamento da integração da nova				



página do front com o  
back

Tabela 9: Daily Meeting Report - 21/11/2024

Grou p Na me	Mo dul e N am e	Me mb ers	Role performed	o que fe z hoje?	O que v ocê far á aman hã?	existem i mpedim entos e m seu ca minho?	Outr o imp edim ent o?
GRU PO 4 - FLA SHPO MO	DS 1	Rin ald o	Developer	Auxiliei na integração	Revisar o seguimento do projeto	não	não
		Da vid	Developer	Terminei a parte dooo front necessária	Revisar o seguimento do projeto	não	não
		Gu sta vo	Developer	Integrei o projeto	Revisar o seguimento do projeto	não	não
		Rh ua n	Scrum Master	Terminei a parte dooo front necessária	Revisar o seguimento do projeto	não	não
Sprint #	2.0		Colocar em ponto a continuação do projeto e verificar o andamento da integração da no				

			va página do front co m o back
--	--	--	-----------------------------------

Tabela 10: Daily Meeting Report - 25/11/2024

Grou p Na me	Mo dul e N am e	M em be rs	Role performed	o que fez hoje?	O que v ocê fará amanhã?	existem impen di mentos em seu caminh o?	Outr o im pedi ment o?
GRU PO 4 - FLA SHP OMO	DS 1	Rin ald o	Developer	Planejei o que vamos implementar durante a semana	Ajudar o Gustavo com o Back-End	não	não
		Da vid	Scrum Master	Planejei o que vamos implementar durante a semana	Começ ar a faze r o box sell	não	não
		Gu sta vo	Developer	Planejei o que vamos implementar durante a semana	Continuar imple mentaç ão do Back-End	não	não
		Rh ua n	Developer	Planejei o que vamos implementar durante a semana	Começ ar a faze r o box buy	não	não

Sprint #	2.0	Colocar em ponto a continuação do projeto e verificar o andamento da integração da nova página do front com o back	
----------	-----	--------------------------------------------------------------------------------------------------------------------	--

Tabela 11: Daily Meeting Report - 26/11/2024

Grou p Na me	Mo dul e N am e	Me mbers	Role performed	o que fez ho je?	O que você fará a manhã?	existem i mpedime ntos em seu cami nho?	Outro impe dimento?
GRUP O 4 - FLAS HPO MO	DS 1	Rin ald o	Developer	Endp oints do Fla shcar d	Conti nuarei no Ba ck-En d	não	não
		Da vid	Scrum Master	A tela box s ell	Conti nuarei no Fro nt-En d	não	não
		Gu sta vo	Developer	Endp oints do Es emble	Conti nuarei no Ba ck-En d	não	não
		Rh ua n	Developer	Esem bles d o Das hboar d	Conti nuarei no Fro nt-En d	não	não
Sprint #	2.0		Colocar em ponto a co ntinuação do projeto e verificar o andamento d a integração da nova p				

		áginas do front com o b ack
--	--	--------------------------------

Tabela 12: Daily Meeting Report - 27/11/2024

Grou p Na me	Mo dul e N am e	Me mb ers	Role performed	o que fe z hoje?	O que você f ará a manh ã?	existem i mpedim entos e m seu ca minho?	Outr o imp ediment o?
GRUP O 4 - FLAS HPO MO	DS 1	Rin ald o	Developer	Contin uei no En dpoints do Flash card	Conti nuarei no Ba ck-En d	não	não
		Da vid	Scrum Master	Contin uei trabal hando n a tela bo x sell	Conti nuarei no Fro nt-En d	não	não
		Gu sta vo	Developer	Contin uei no En dpoints do Ese mble	Conti nuarei no Ba ck-En d	não	não
		Rh ua n	Developer	Contin uei no Es embles do Dash board	Conti nuarei no Fro nt-En d	não	não
Sprint #	2.0		Colocar em ponto a c ontinuação do projeto e verificar o andament o da integração da no				

		va página do front co m o back	
--	--	-----------------------------------	--

Tabela 13: Daily Meeting Report - 28/11/2024

Group Name	Module Name	Members	Role performed	o que fez hoje?	O que você fará amanhã?	existem impedimentos em seu caminho?	Outro impedimento?
GRUPO 4 - FLA~~ SHPO MO	DS 1	Rinaldo	Developer	Mexi nas rotas do flash card	Continuarei no Back-End	não	não
		David	Scrum Master	Implementei modal na tela box sell	Continuarei no Front-End	não	não
		Gustavo	Developer	Mexi nas rotas do flash card	Continuarei no Back-End	não	não
	~~	Rhuan	Developer	Terminei o dashboard e o flashcard	Continuarei no Front-End	não	não
Sprint #	2.0		Colocar em ponto a continuação do projeto e verificar o andamento da integração da nov				

		a página do front com o back
--	--	---------------------------------

Tabela 14: Daily Meeting Report - 29/11/2024

Grou p Na me	Mo dul e N am e	Me mb ers	Role performed	o que fe z hoje?	O que você f ará a manh ã?	existem i mpedim entos e m seu ca minho?	Outr o imp ediment o?
GRUP O 4 - FLAS HPO MO	DS 1	Rin ald o	Developer	Ajudei c om os te stes dos endpoin ts	Conti nuarei no Ba ck-En d	não	não
		Da vid	Scrum Master	Terminei o docum ento do lab6	Conti nuarei no Fro nt-En d	não	não
		Gu sta vo	Developer	Testei os endpoin ts	Conti nuarei no Ba ck-En d	não	não
		Rh ua n	Developer	Finalizei as telas principai s do fron t end	Conti nuarei no Fro nt-En d	não	não
Sprint #	2.0		Colocar em ponto a c ontinuação do projeto e verificar o andament o da integração da no				

			va página do front co m o back	
--	--	--	-----------------------------------	--

Tabela 15: Daily Meeting Report - 02/12/2024

Group Name	Module Name	Members	Role performed	o que fez hoje?	O que você fará amanhã?	existem impedimentos em seu caminho?	Outro impedimento?
GRUPO 4 - FLAS HPO MO	DS 1	Rinaldo	Developer	MODEL AGEM DE DA DOS	DOC: DIAGRAMAS - SEQUENCIA	não	não
		David	Developer	CASOS DE USO: UCO 1 a UCO 5	DOC: CASOS DE USO	não	não
		Gustavo	Developer	ENDPOINT API	DOC: ENDPOINTS	não	não
		Rhuan	Developer	INTEGRAÇÃO FRONT-END	DOC: FRONT-END	não	não
Sprint #	3.0		Colocar em ponto a continuação do projeto e verificar o andamento da documentação				

# Engenharia de Software

A engenharia de software é a abordagem sistemática para o desenvolvimento, operação e manutenção de sistemas de software. Nesse contexto, a organização da documentação técnica do nosso projeto é essencial para garantir a qualidade, a manutenibilidade e a evolução do sistema.

Como afirma Pressman (2010), "a documentação técnica é a espinha dorsal da engenharia de software, pois captura os requisitos, a arquitetura, a implementação e os procedimentos operacionais de um sistema".

## Requisitos

### Requisitos Funcionais

Requisito	Descrição	Prioridade	Caso de Uso
RF01	O sistema deverá permitir o cadastro, edição e exclusão de usuários.	1	UC01
RF02	O sistema deverá permitir ao usuário logar e deslogar sua conta.	2	UC02, UC03
RF03	O sistema deverá permitir ao usuário iniciar, desativar e pausar o temporizador.	1	UC04
RF04	O sistema deverá permitir realizar ao usuário as funções de criar e consultar o relatório dos pomodoros.	3	UC05
RF05	O sistema deverá permitir ao usuário realizar as funções de criar, editar, consultar e excluir flashcards.	1	UC06
RF06	O sistema deverá permitir ao usuário realizar as funções de adicionar e remover flashcards do histórico.	2	UC07
RF07	O sistema deverá permitir ao usuário realizar as funções de adicionar e remover flashcards dos favoritos.	2	UC08
RF08	O sistema deverá permitir ao usuário realizar as funções de criar, consultar, editar e excluir conjuntos dos flashcards.	1	UC09
RF09	O sistema deverá permitir ao usuário a compra de flashcards usando a moeda do sistema.	3	UC10
RF10	O sistema deverá permitir a criação de box sells para venda dos flashcards.	1	UC11

RF11	O sistema deverá permitir ao usuário a venda de flashcard s.	1	UC12
------	--------------------------------------------------------------	---	------

## Requisitos Não Funcionais

Requisito	Descrição	Prioridade
RNF01	O sistema deve ter uma interface responsiva e adaptável a diferentes tamanhos de tela.	1
RNF02	O sistema deve ter um design intuitivo e de fácil usabilidade.	1
RNF03	O sistema deve ter um tempo de resposta rápido, mesmo com um grande número de usuários.	2
RNF04	O sistema deve garantir a segurança e privacidade dos dados dos usuários.	1
RNF05	O sistema deve ser acessível, seguindo as diretrizes de acessibilidade web.	2
RNF06	O sistema deve ser multiplataforma, funcionando em diferentes sistemas operacionais e dispositivos.	2
RNF07	O sistema deve fornecer documentação clara e detalhada para os desenvolvedores.	3
RNF08	O sistema deve realizar backups automáticos diários para prevenir perda de dados.	3
RNF09	O sistema deve permitir integrações com APIs externas para funcionalidades adicionais.	3

# Arquitetura de Informação

A arquitetura de informação (AI) para a documentação de software do nosso projeto deve ser orientada a fornecer uma estrutura clara e intuitiva para organizar e disseminar o conhecimento técnico. Segundo Rosenfeld e Morville (2015), a AI "envolve a concepção da estrutura e da navegação de um espaço de informação para facilitar a execução de tarefas e o acesso ao conteúdo".

Nesse contexto, a documentação técnica do nosso projeto pode ser estruturada da seguinte forma:

## 1. Visão Geral do Sistema:

- Descrição de alto nível do sistema, suas funcionalidades e objetivos.
- Diagrama de Casos de Uso para representar as interações entre atores e o sistema.
- Citação: "A documentação de visão geral fornece uma compreensão comum do sistema, estabelecendo uma base para discussões e decisões posteriores" (IEEE, 2011).

## 2. Documentação Arquitetural:

- Diagrama de Componentes para ilustrar a estrutura modular do sistema.
- Diagrama de Implantação para descrever a infraestrutura e a distribuição dos componentes.
- Citação: "A documentação arquitetural é essencial para entender o design e as decisões de alto nível que regem o sistema" (Rozanski e Woods, 2011).

## 3. Documentação de Subsistemas e Módulos:

- Diagramas de Classe para detalhar as estruturas de dados e os relacionamentos.
- Diagramas de Sequência para explicar os principais fluxos de interação.
- Citação: "A documentação de subsistemas e módulos ajuda os desenvolvedores a compreender a implementação e as responsabilidades de cada componente" (Kruchten, 2004).

#### 4. Guias de Desenvolvimento e Implantação:

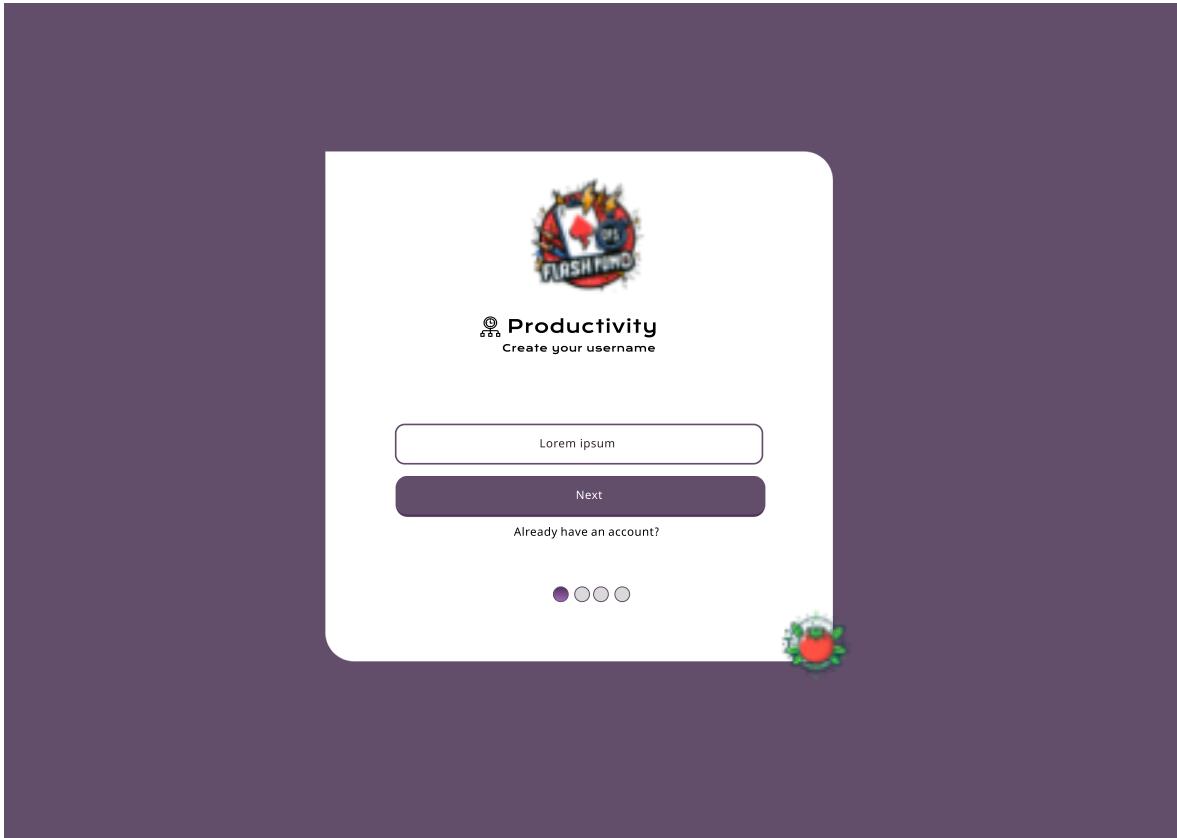
- Instruções passo a passo para configurar o ambiente de desenvolvimento.
- Procedimentos de construção, empacotamento e implantação do sistema.
- Citação: "A documentação de desenvolvimento e implantação é crucial para permitir que novos membros da equipe comecem a contribuir rapidamente" (McConnell, 1996).

#### 5. Referência de API e Bibliotecas:

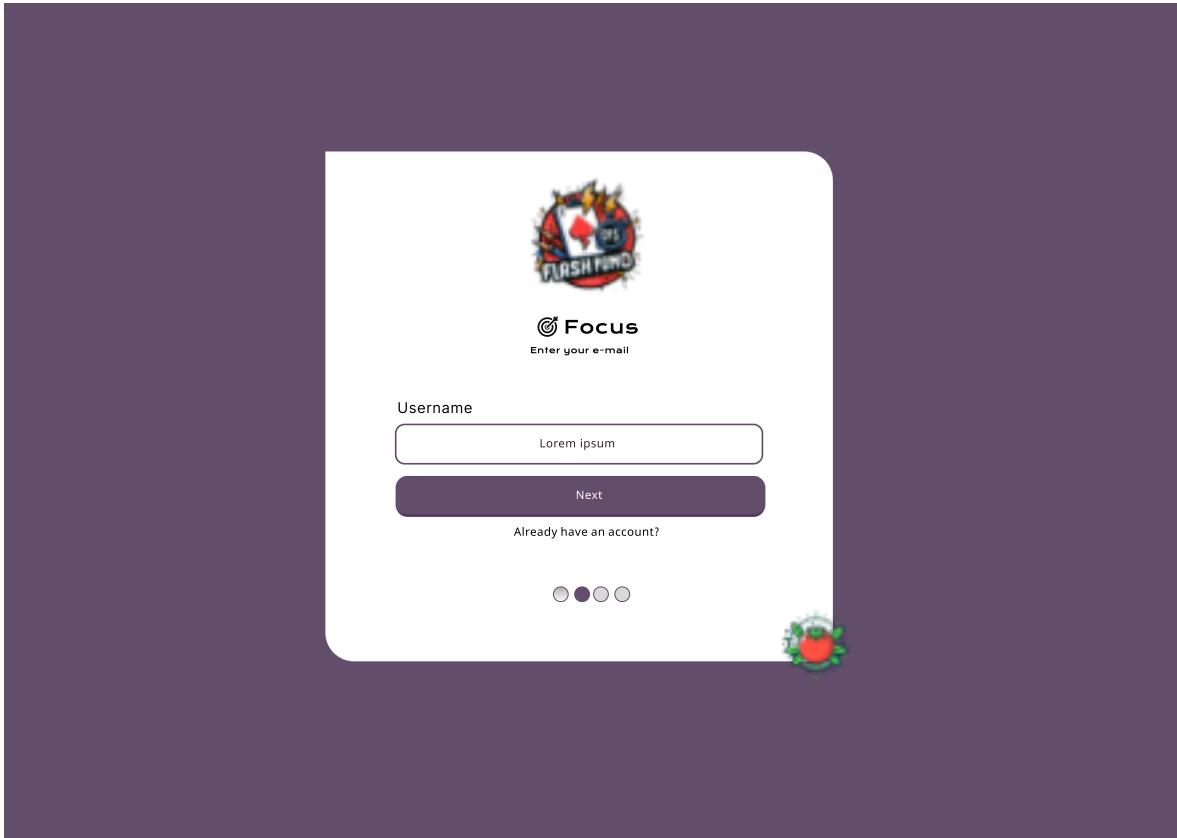
- Documentação detalhada das interfaces de programação (APIs) e bibliotecas utilizadas.
- Exemplos de uso e padrões de interação.
- Citação: "A documentação de referência de API e bibliotecas facilita a integração e a reutilização de componentes em todo o sistema" (Krug, 2014).

Essa estrutura de arquitetura de informação ajudará a garantir que a documentação do nosso projeto de software seja compreensível, acessível e eficaz para os desenvolvedores e outras partes interessadas.

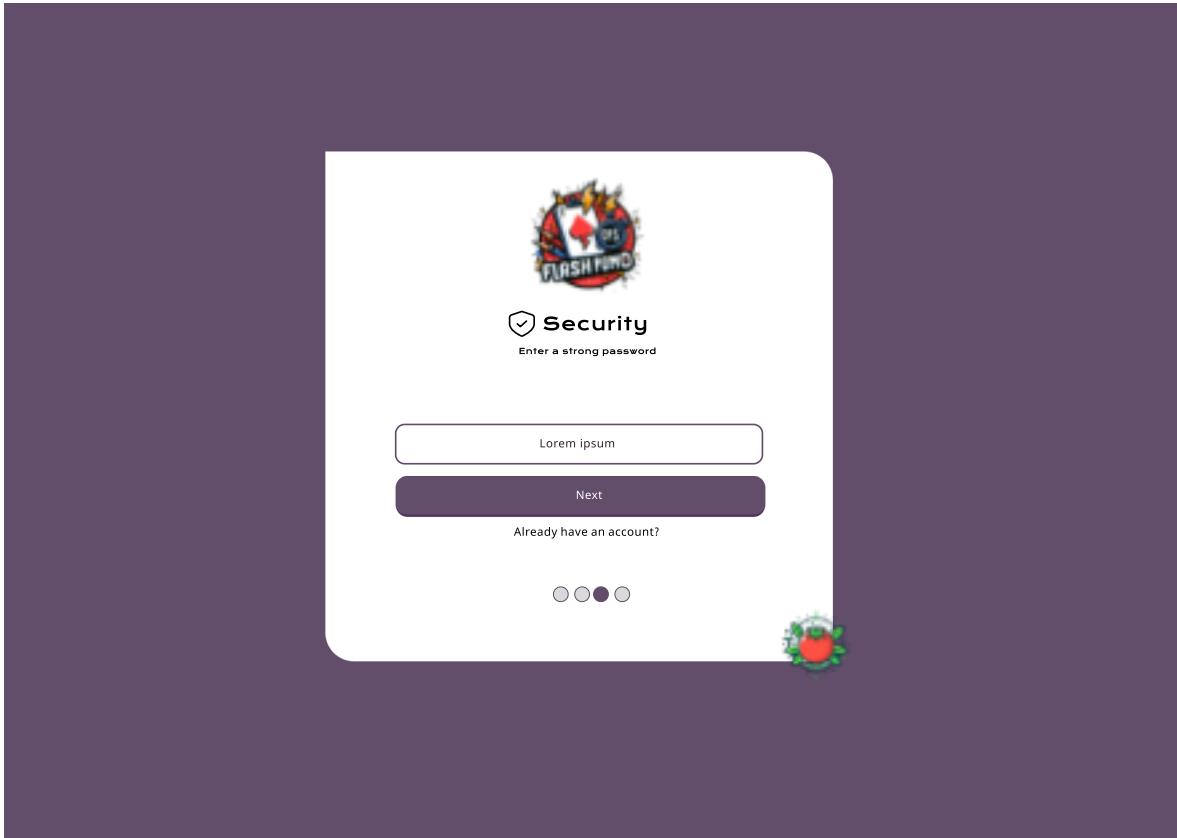
### Wireframe



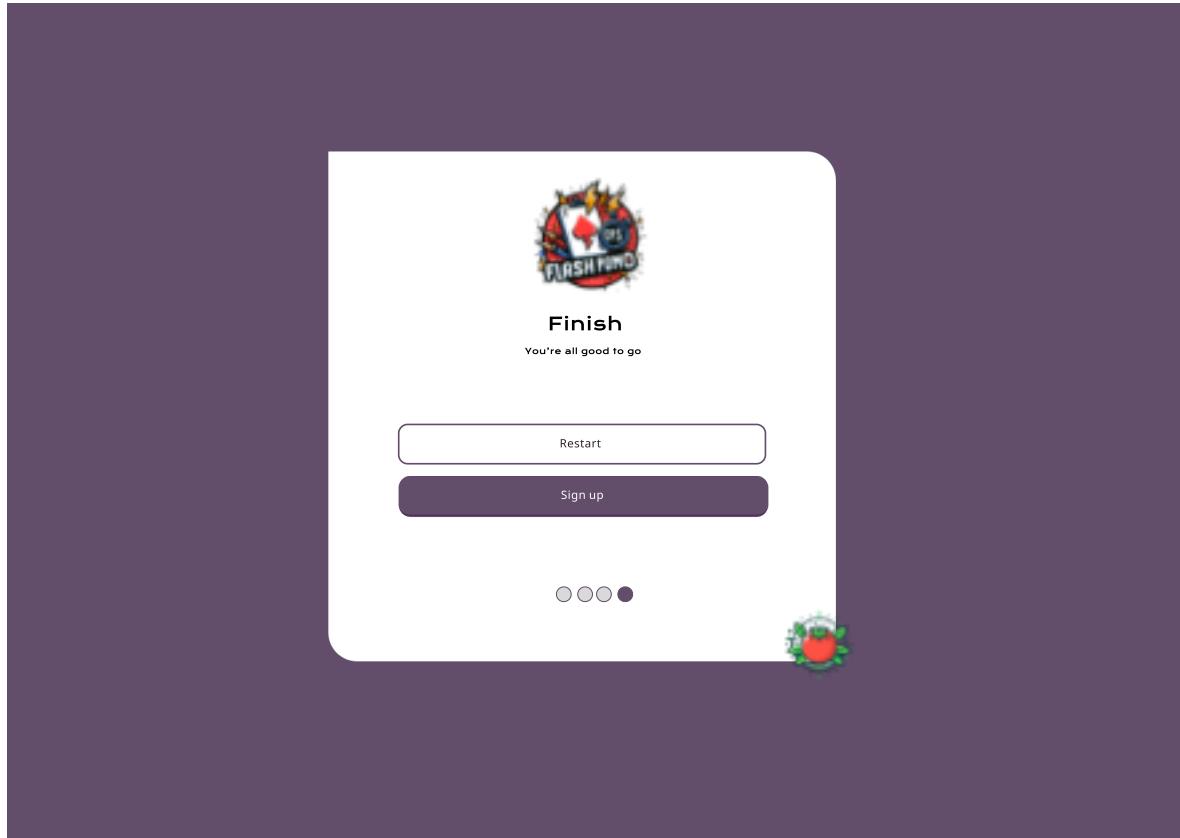
CADASTRO - 1 - USERNAME.svg



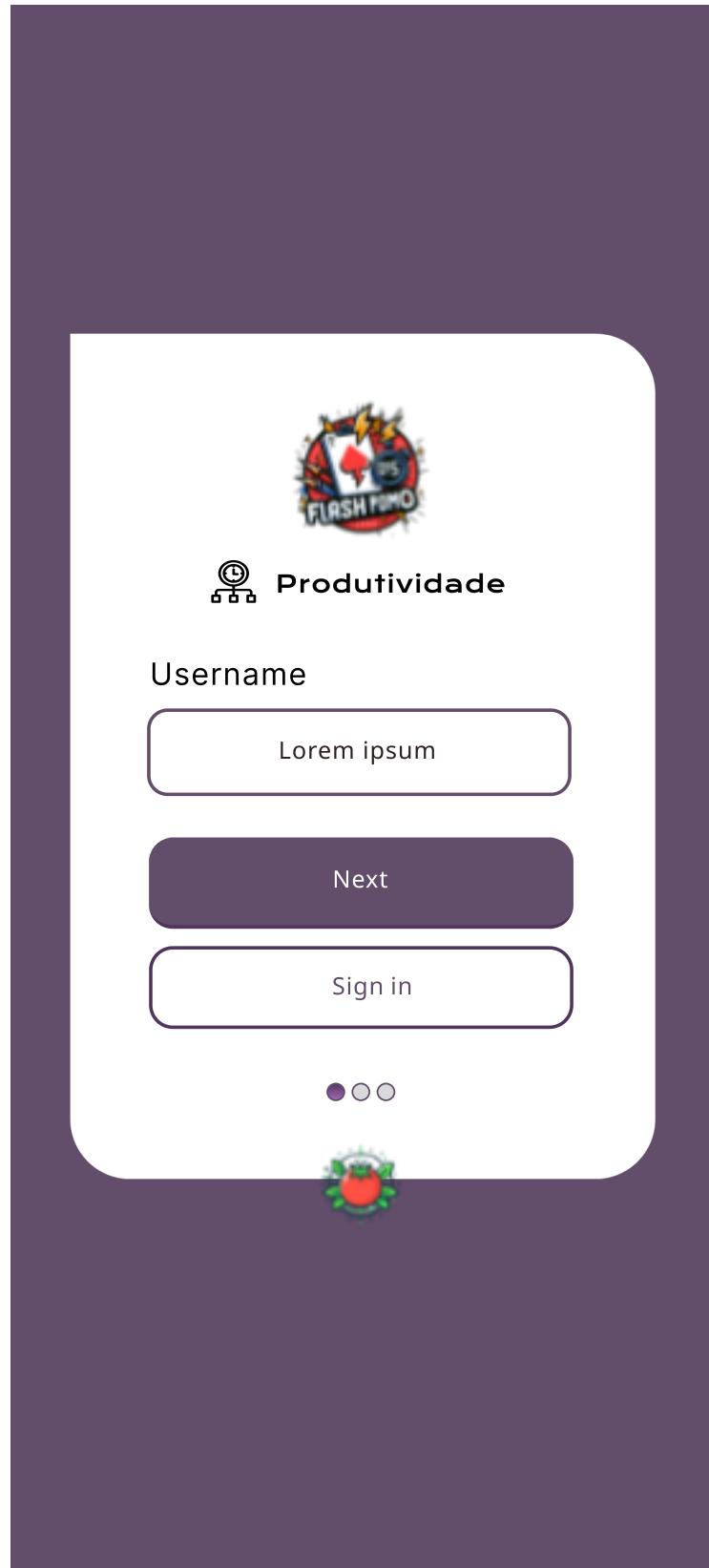
CADASTRO - 1 - EMAIL.svg



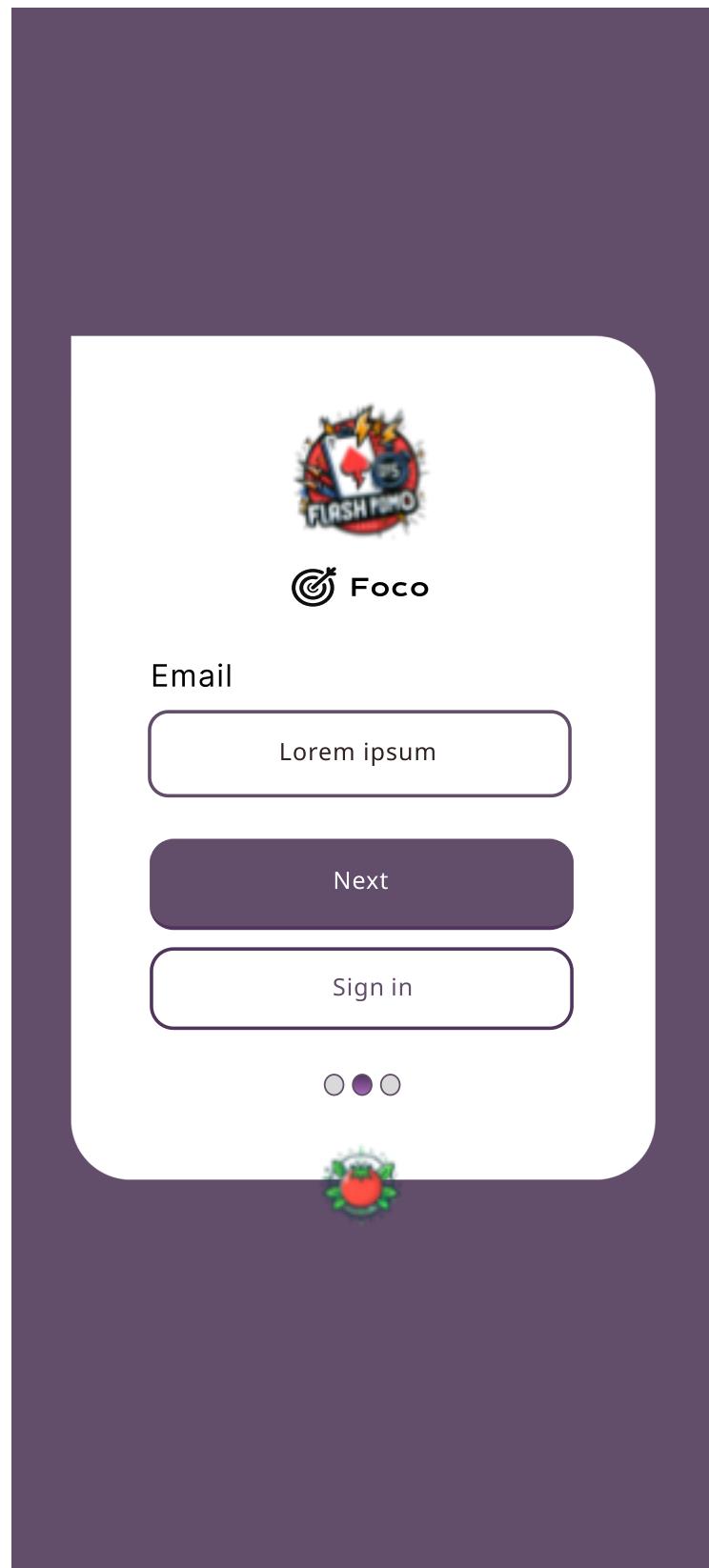
CADASTRO - 1 - PASSWORD.svg

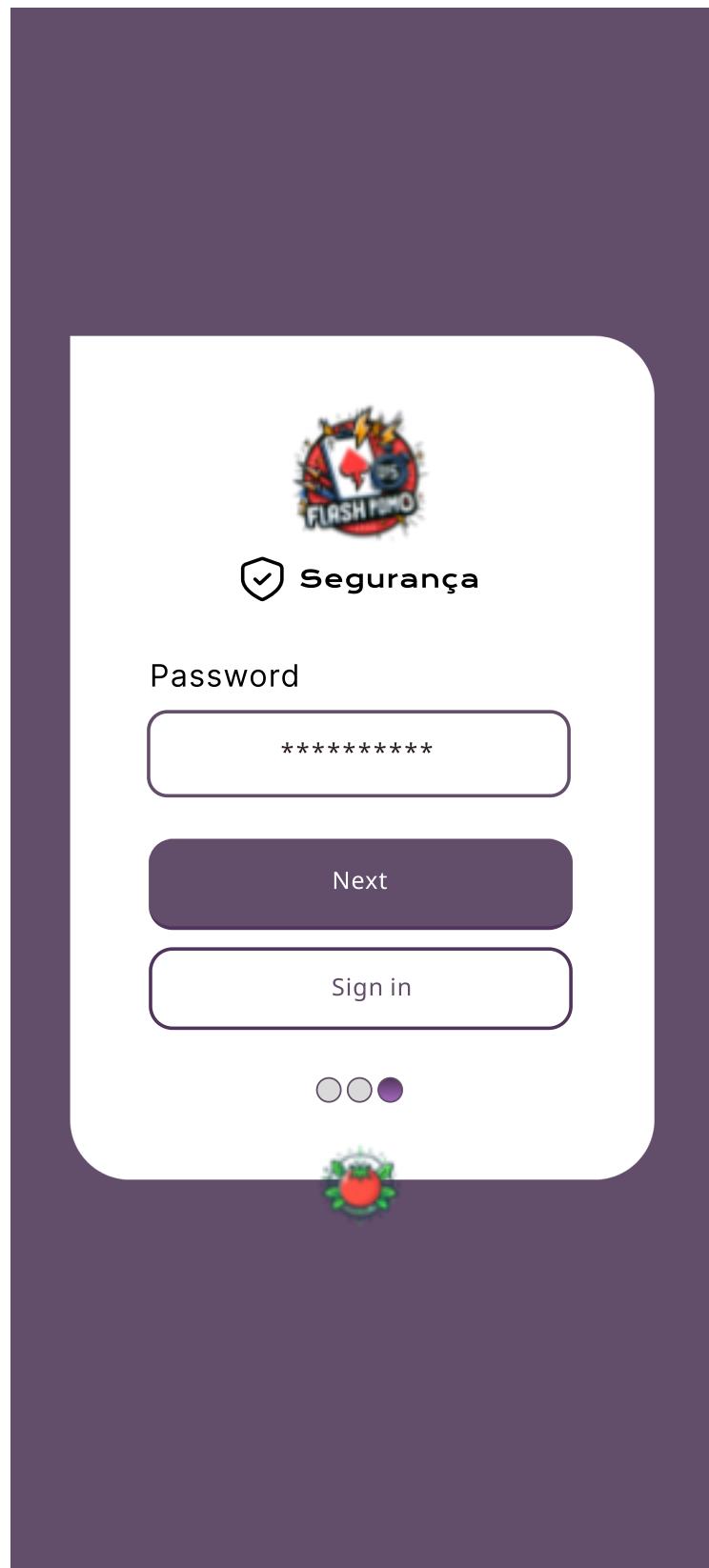


CADASTRO - 1 - SIGN UP.svg

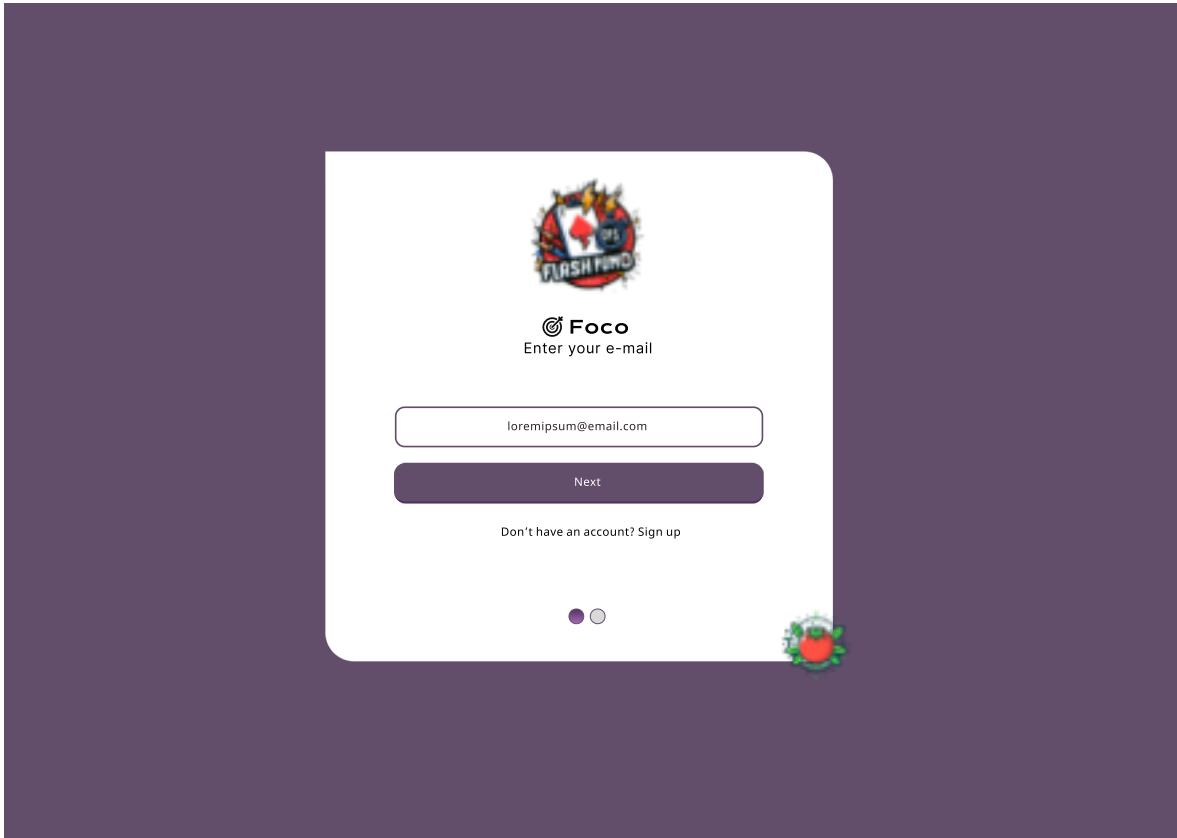


CADASTRO - 1 - MOBILE.svg

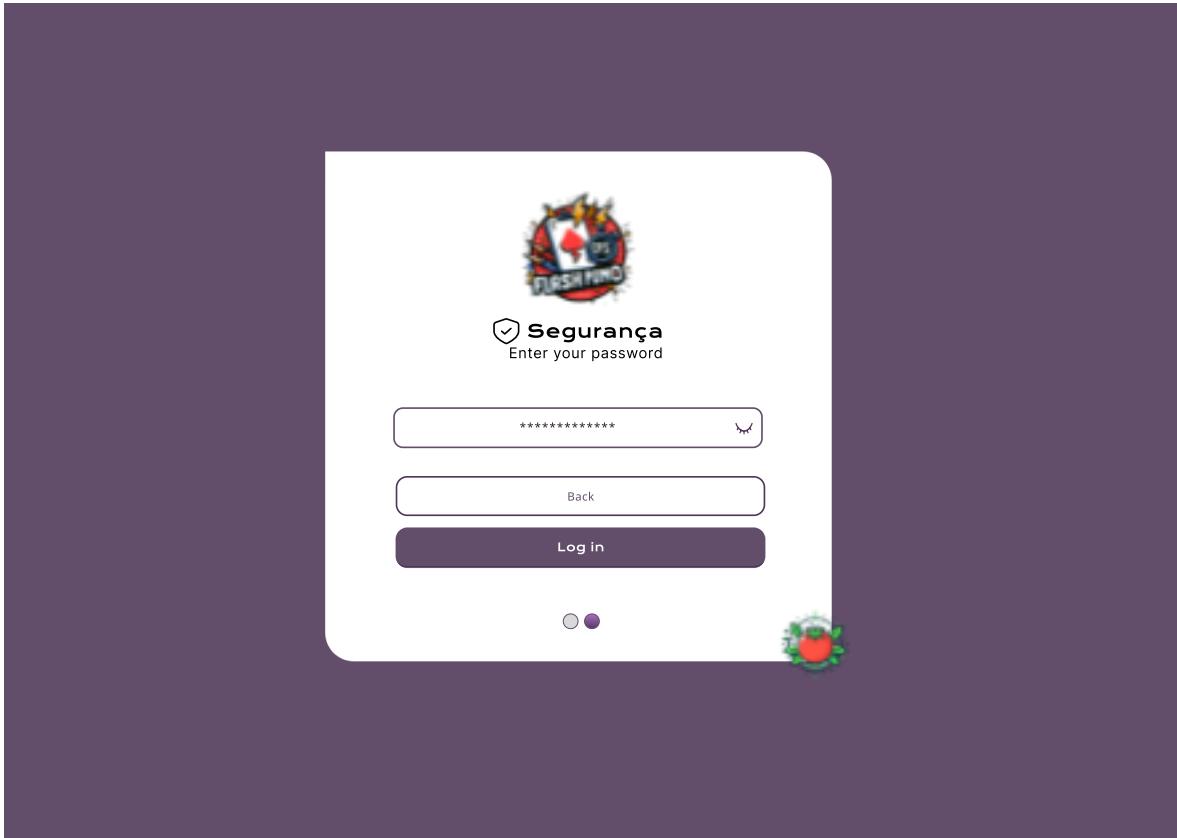




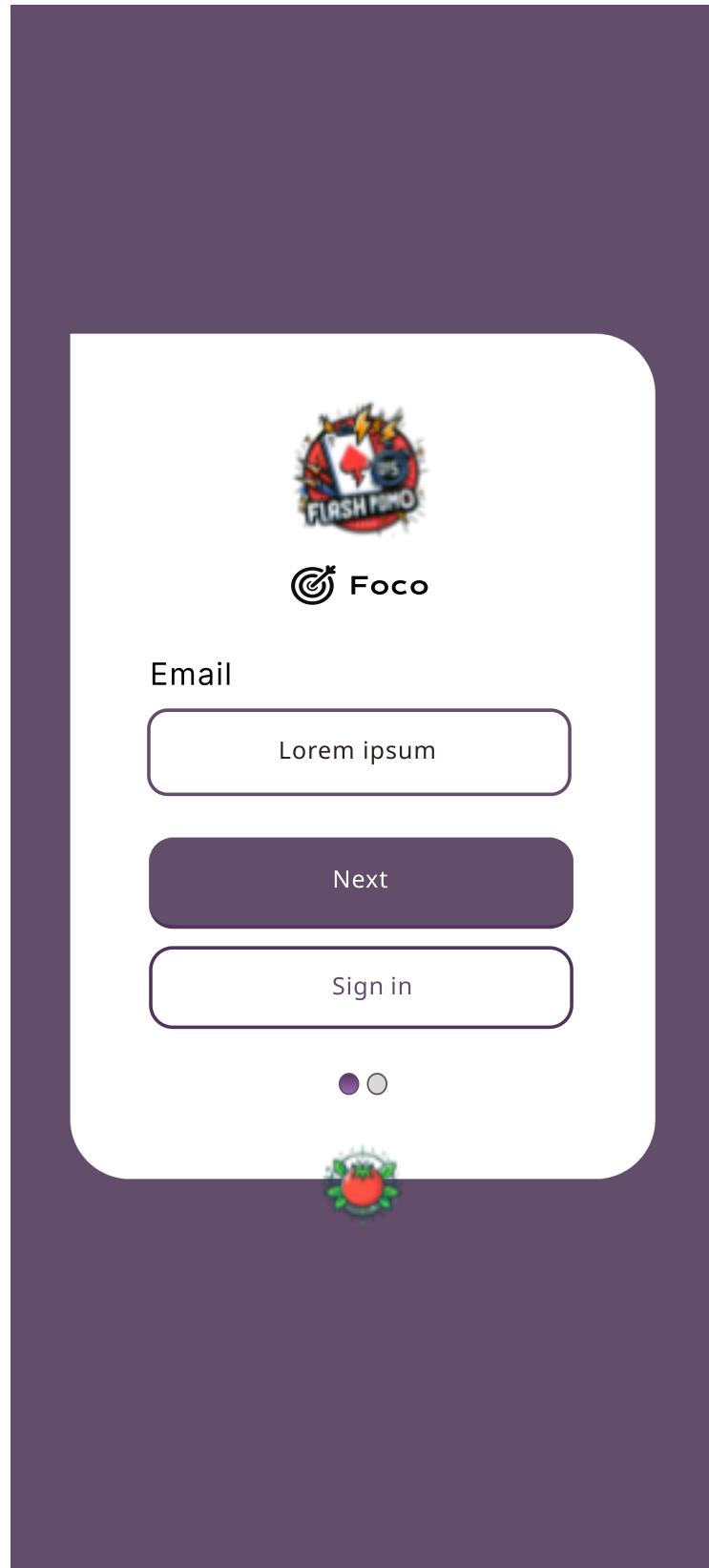
CADASTRO - 3 - MOBILE.svg



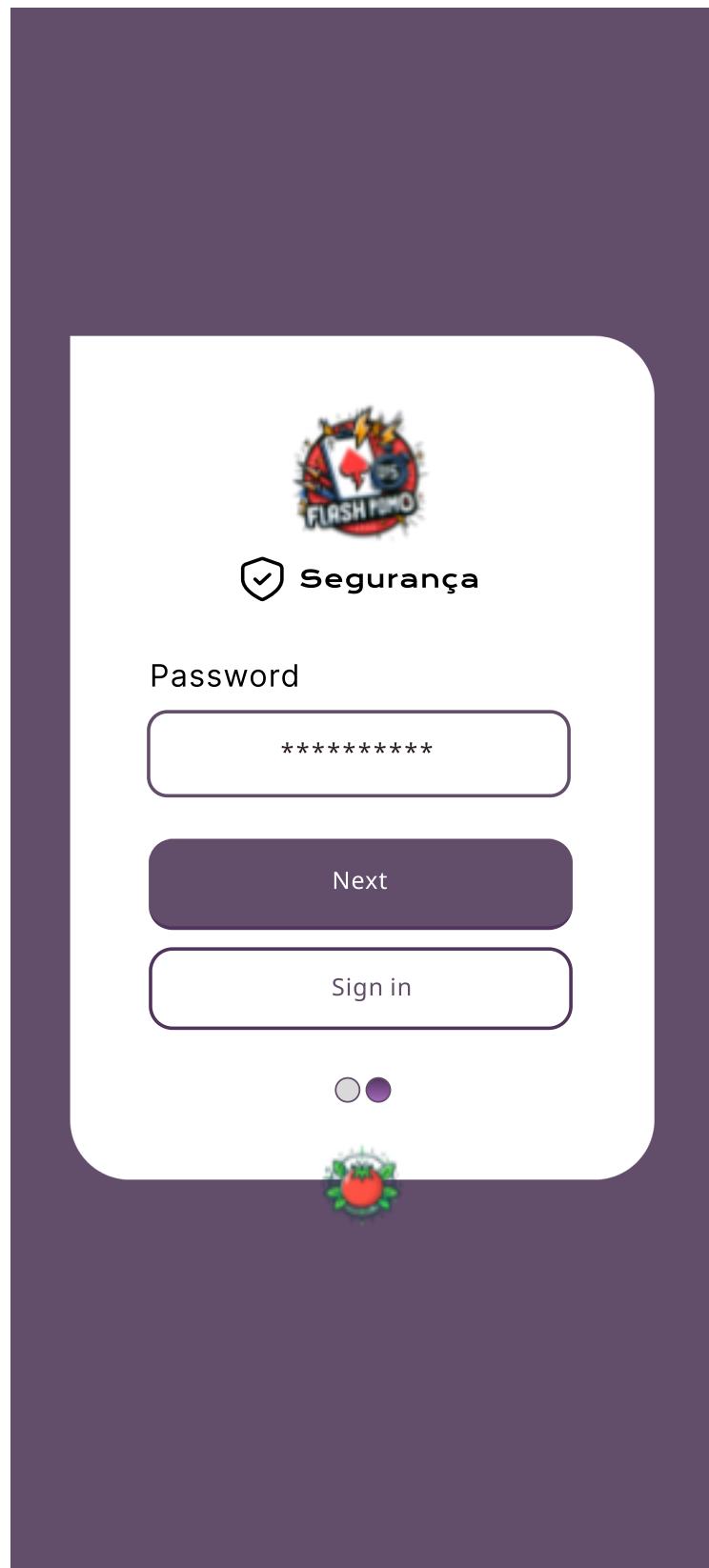
LOGIN - 1 - DESKTOP.svg



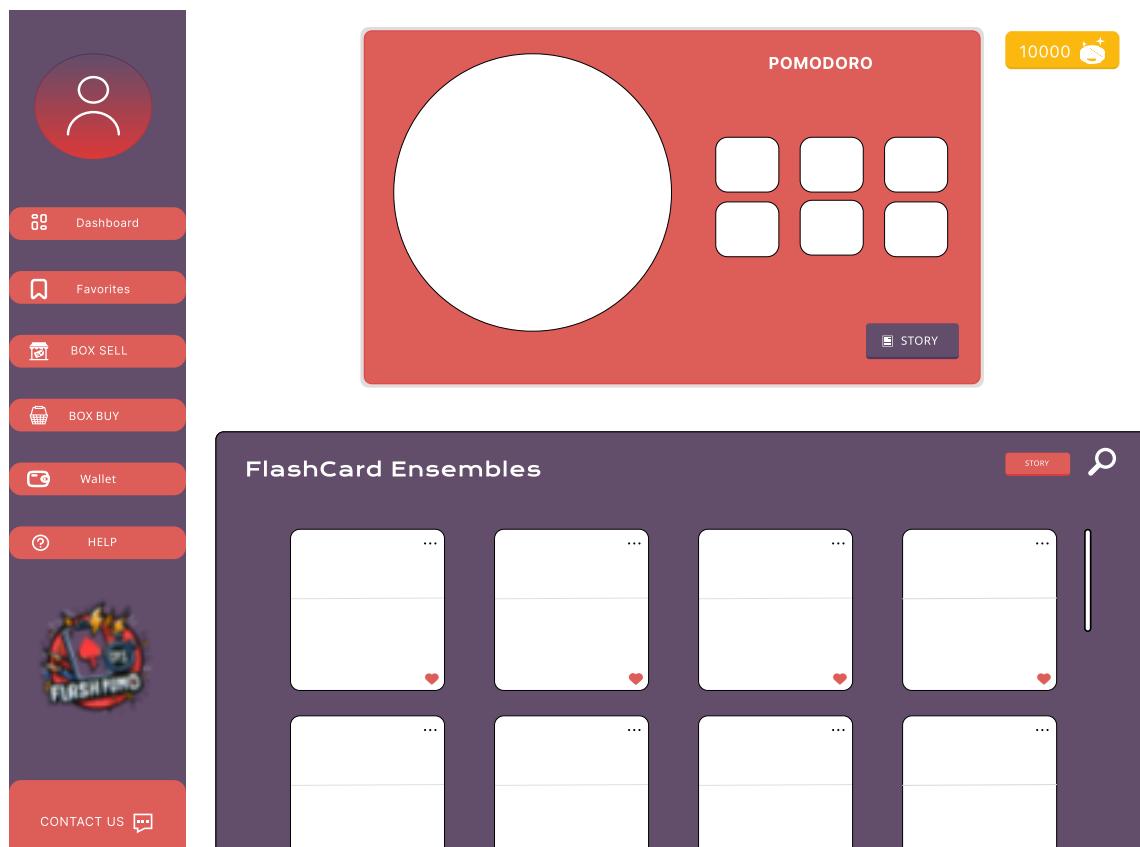
LOGIN - 1 - SECURITY.svg



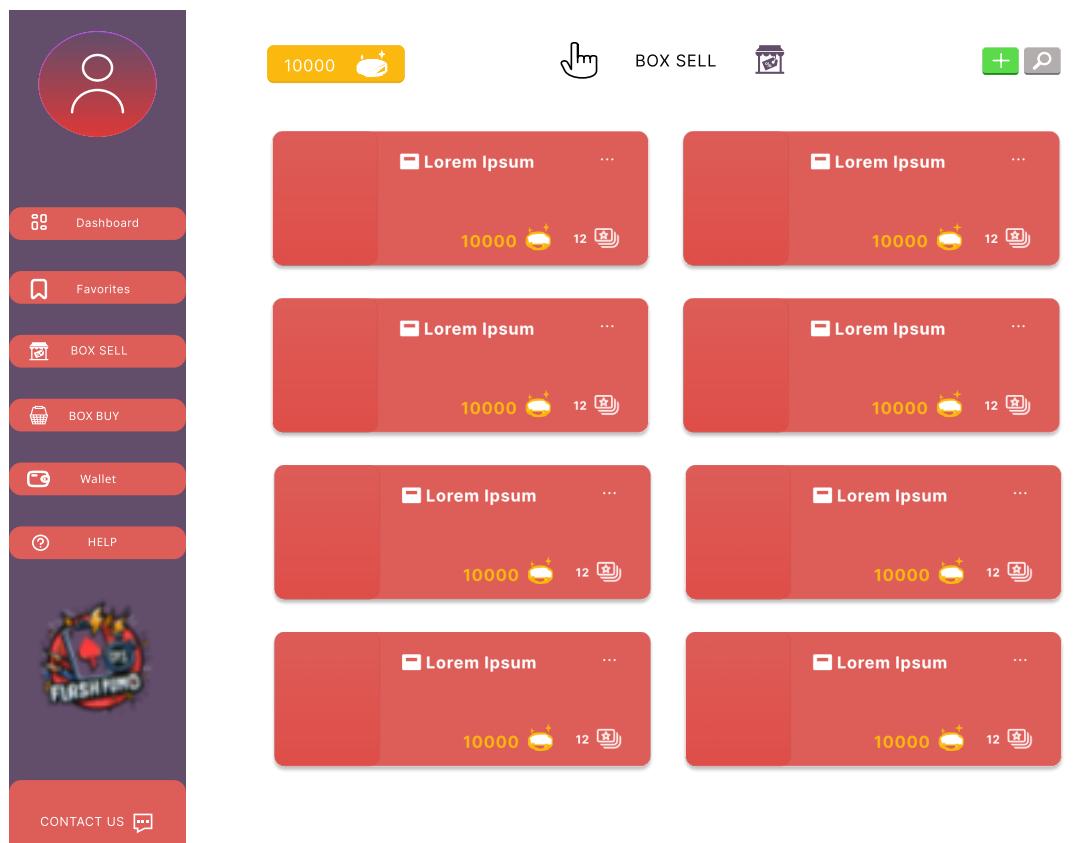
LOGIN - 2 - MOBILE.svg



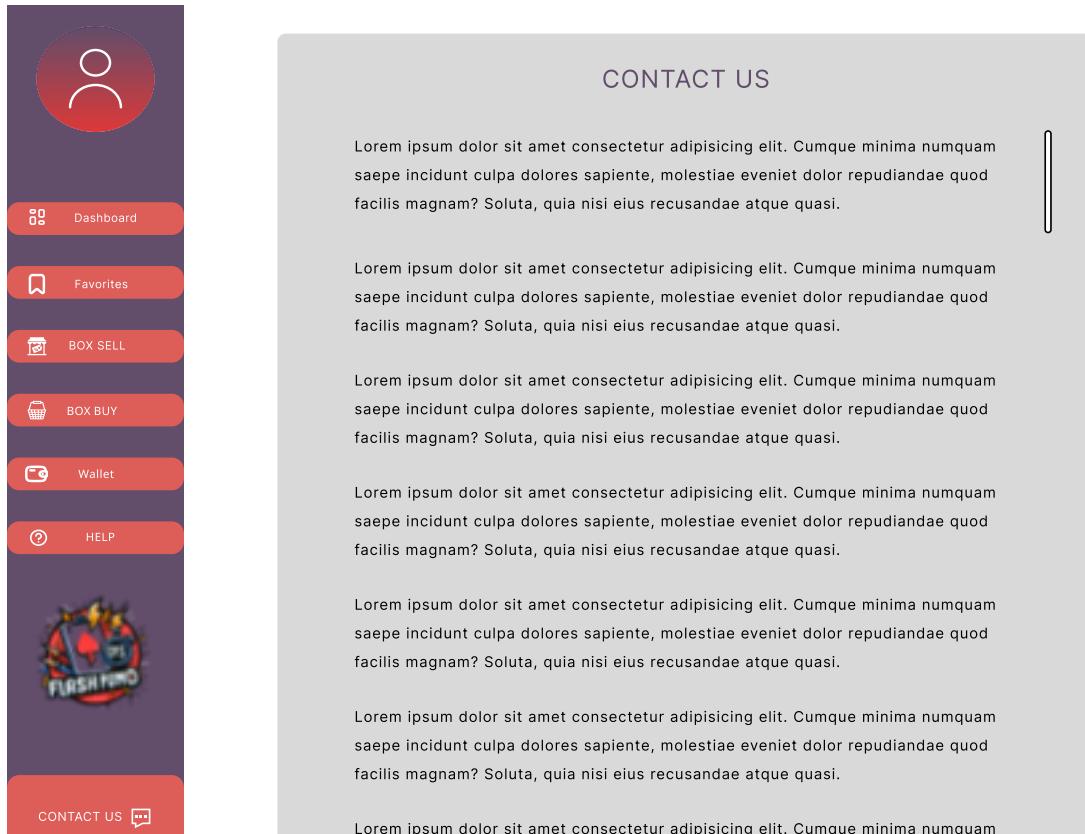
LOGIN - 3 - MOBILE.svg



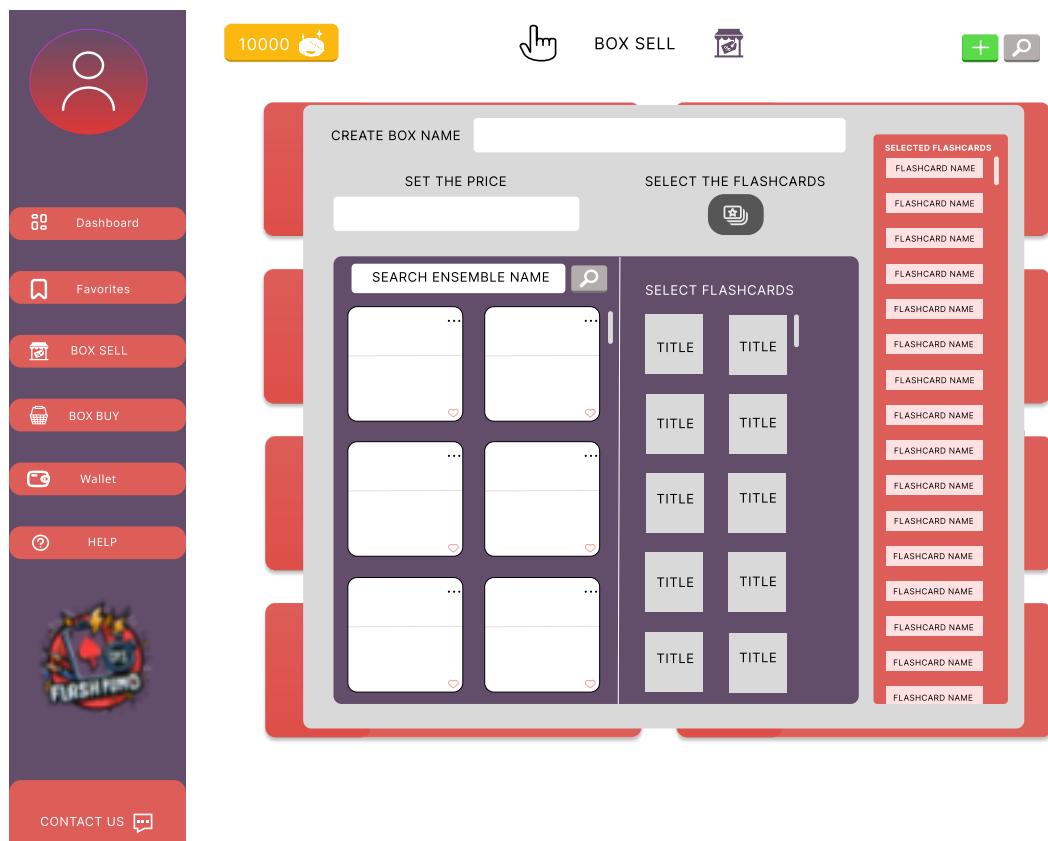
INTERFACE - USER - DASHBOARD.svg



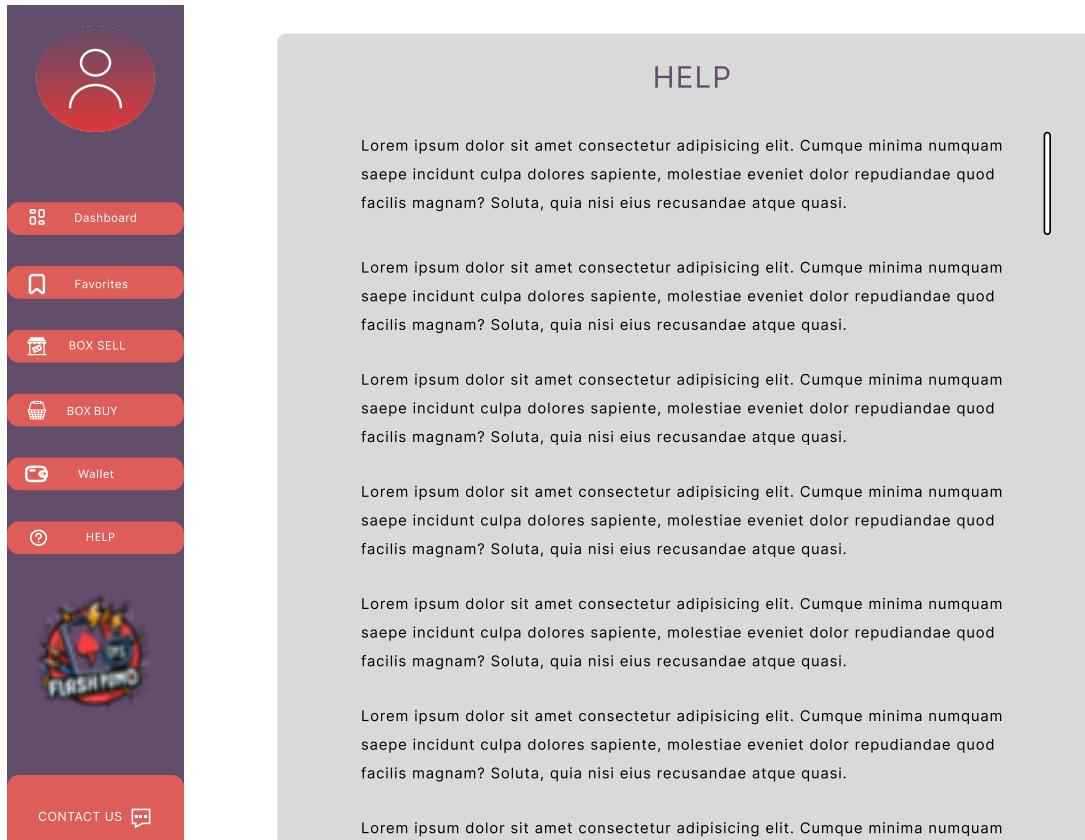
DESKTOP - USER - BOX SHELL.svg



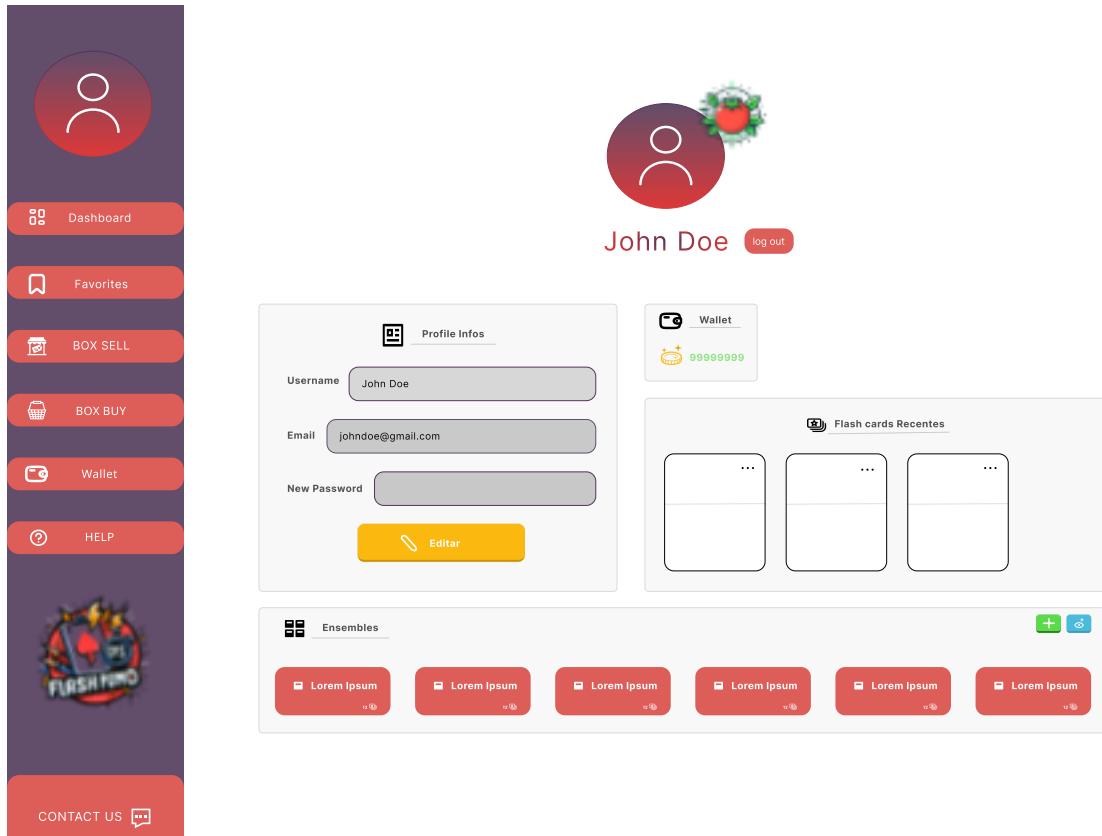
DESKTOP - USER - CONTACT US.svg



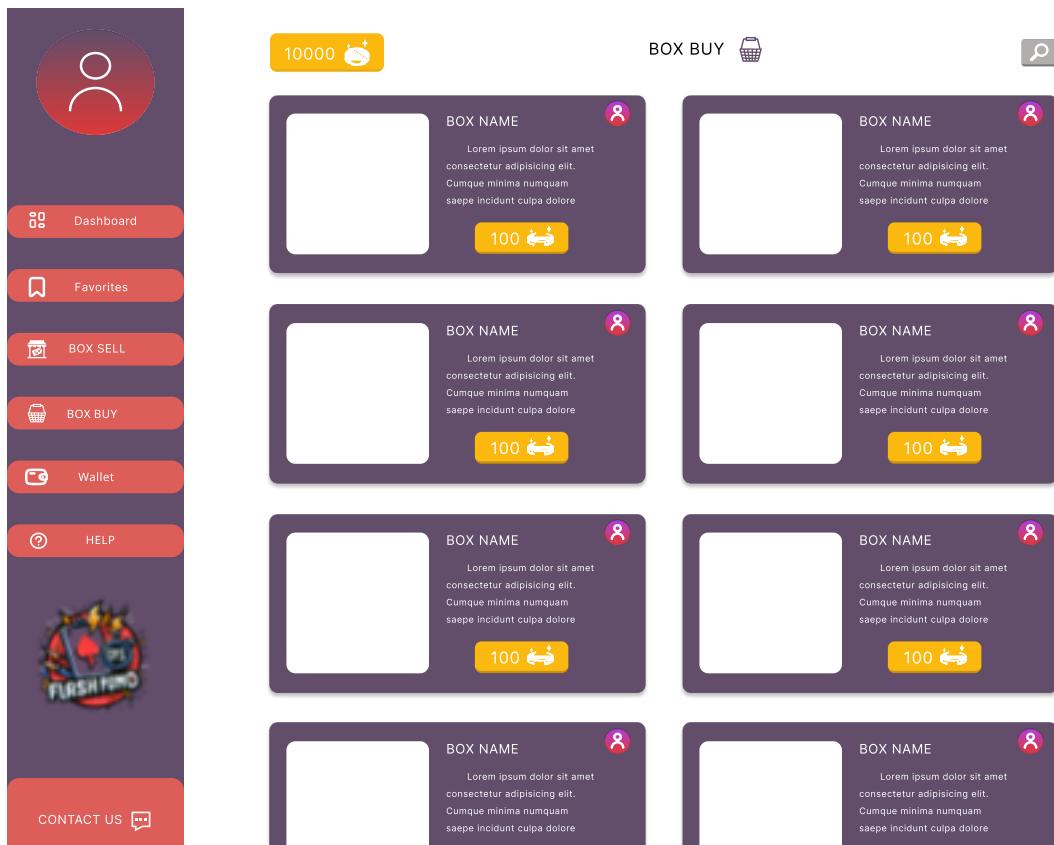
DESKTOP - USER - CREATING BOX SHELL.svg



DESKTOP - USER - HELP.svg



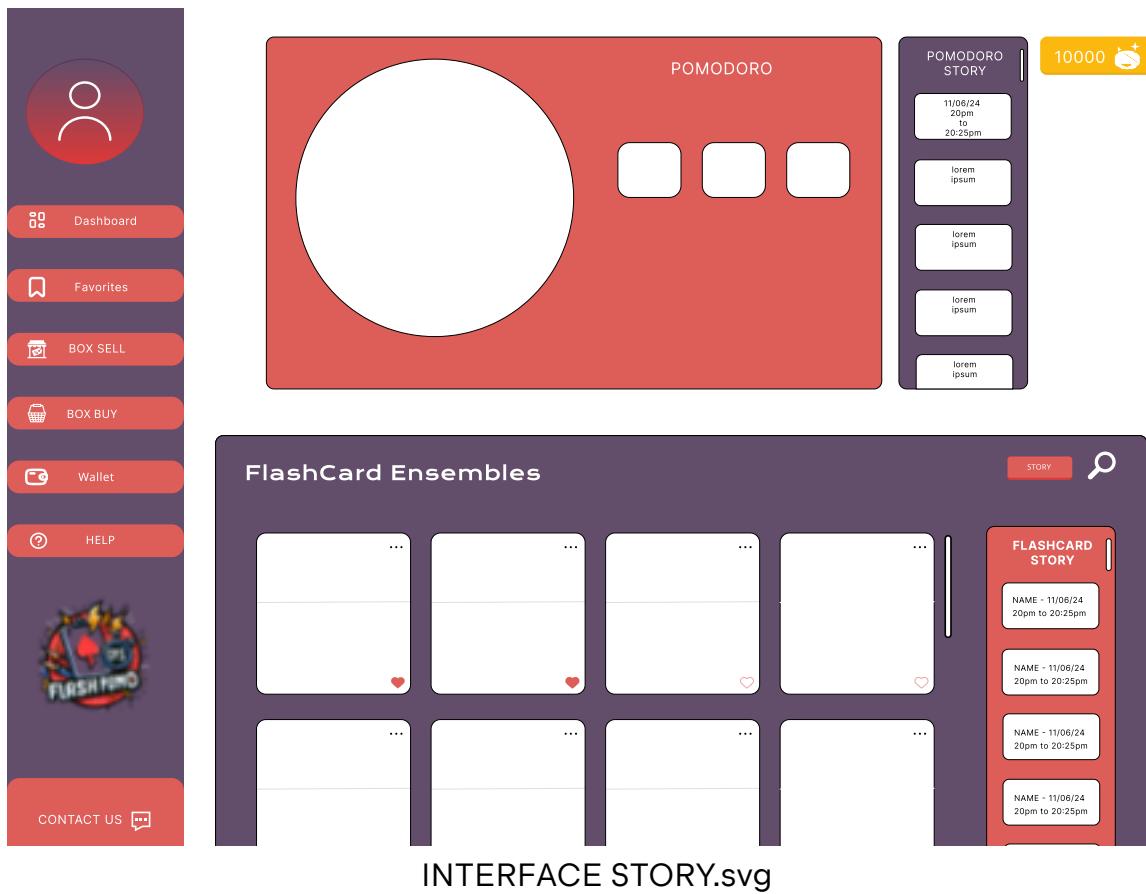
DESKTOP - USER - USER PROFILE.svg



INTERFACE BOX BUY.svg



INTERFACE FAVORITE ENSEMBLES.svg



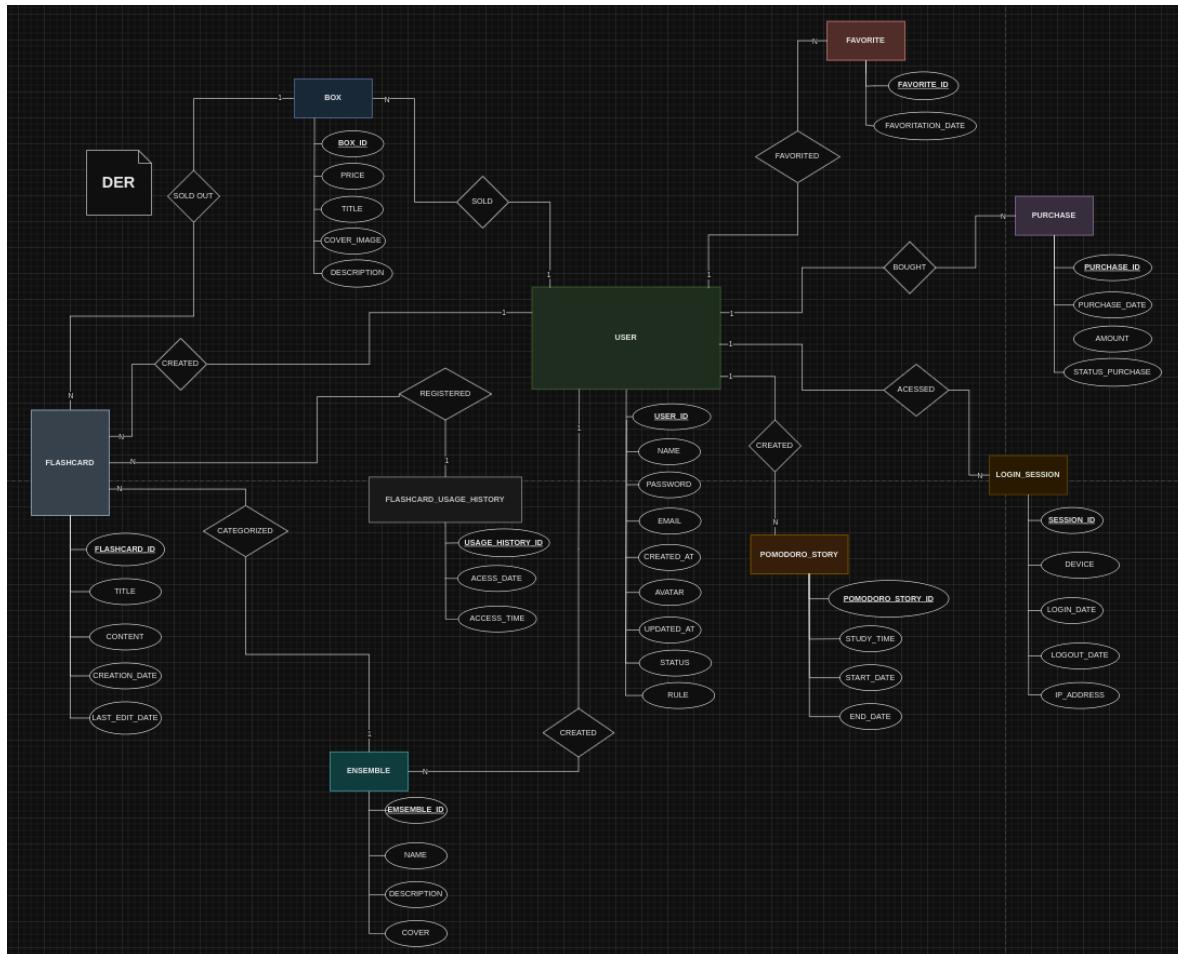
INTERFACE STORY.svg

## Banco de Dados

**⚠ Resource:**

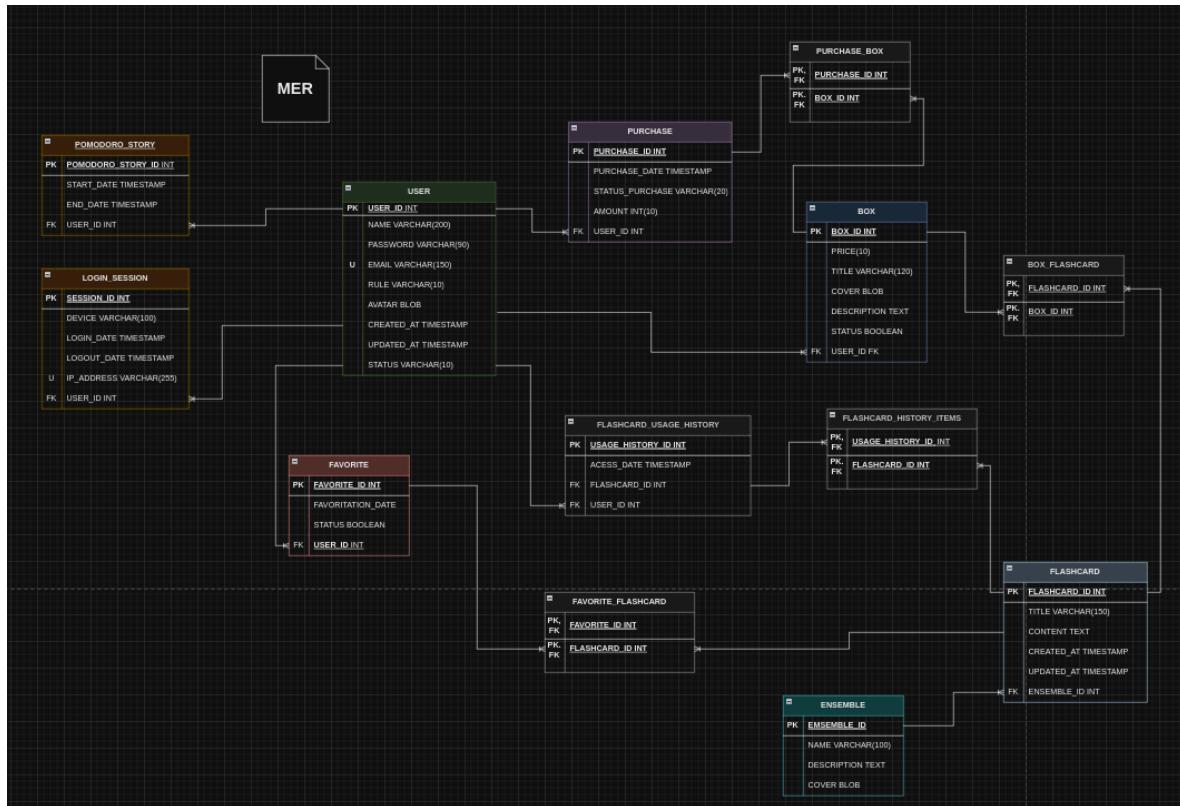
<https://drive.google.com/file/d/1IfJADPBk8vBuoPQ7H7mxsLTCevSHhe98/view?usp=sharing>  
`(https://drive.google.com/file/d/1IfJADPBk8vBuoPQ7H7mxsLTCevSHhe98/vie  
w?usp=sharing)`

## Diagrama Entidade Relacionamento



DER.png

## Modelo Entidade Relacionamento



MER.png

## Modelo Físico

### Tabela USER

-- Tabela para armazenar informações dos usuários

```

CREATE TABLE USER (
    USER_ID INT PRIMARY KEY,           -- Identificador único do
    usuário                                -- Nome do usuário
    NAME VARCHAR(200),                  -- Senha do usuário para
    PASSWORD VARCHAR(90),                -- Email único do usuário
    EMAIL VARCHAR(150) UNIQUE,          -- Função ou tipo de usuário
    RULE VARCHAR(10),                   -- Imagem ou avatar do
    AVATAR BLOB,                      -- Data e hora em que o
    usuário foi criado
    CREATED_AT TIMESTAMP,              -- Data e hora da última
    UPDATED_AT TIMESTAMP,
)

```

```

atualização
    STATUS VARCHAR(10)          -- Status do usuário
);

```

**Tabela ENSEMBLE**

- Tabela para armazenar informações sobre os conjuntos de flashcards

```

CREATE TABLE ENSEMBLE (
    ENSEMBLE_ID INT PRIMARY KEY,           -- Identificador único do
    conjunto
    NAME VARCHAR(100),                   -- Nome do conjunto
    DESCRIPTION TEXT,                    -- Descrição do conjunto
    COVER BLOB                          -- Capa ou imagem
representando o conjunto
);

```

**Tabela FLASHCARD**

- Tabela para armazenar informações dos flashcards

```

CREATE TABLE FLASHCARD (
    FLASHCARD_ID INT PRIMARY KEY,        -- Identificador único do
    flashcard
    TITLE VARCHAR(150),                 -- Título do flashcard
    CONTENT TEXT,                      -- Conteúdo ou descrição do
    flashcard
    CREATED_AT TIMESTAMP,              -- Data de criação do
    flashcard
    UPDATED_AT TIMESTAMP,              -- Data da última
    atualização do flashcard
    ENSEMBLE_ID INT,                  -- Identificador do conjunto
    de flashcards a que este flashcard pertence
    FOREIGN KEY (ENSEMBLE_ID) REFERENCES ENSEMBLE(ENSEMBLE_ID)  --
);

```

```
Relacionamento com o conjunto de flashcards
);
```

### Tabela FAVORITE

- Tabela para armazenar informações sobre os favoritos dos usuários

```
CREATE TABLE FAVORITE (
    FAVORITE_ID INT PRIMARY KEY,           -- Identificador único do
    favorito
    FAVORITATION_DATE_TIMESTAMP TIMESTAMP, -- Data e hora em que o
    flashcard foi marcado como favorito
    STATUS BOOLEAN,                      -- Status do favorito (se
    está ativo ou não)
    USER_ID INT,                         -- Identificador do usuário
    que marcou o favorito
    FOREIGN KEY (USER_ID) REFERENCES USER(USER_ID) --
    Relacionamento com o usuário
);
```

### Tabela Pivo FAVORITE\_FLASHCARD

- Tabela para associar flashcards a favoritos

```
CREATE TABLE FAVORITE_FLASHCARD (
    FAVORITE_ID INT,                     -- Identificador do favorito
    FLASHCARD_ID INT,                   -- Identificador do
    flashcard
    PRIMARY KEY (FAVORITE_ID, FLASHCARD_ID), -- Chave composta para
    garantir que um flashcard não seja duplicado nos favoritos
    FOREIGN KEY (FLASHCARD_ID) REFERENCES FLASHCARD(FLASHCARD_ID),
    -- Relacionamento com o flashcard
    FOREIGN KEY (FAVORITE_ID) REFERENCES FAVORITE(FAVORITE_ID) --
```

```
Relacionamento com o favorito
);
```

#### Tabela PivoFLASHCARD\_USAGE\_HISTORY

- Tabela para armazenar o histórico de uso dos flashcards pelos usuários

```
CREATE TABLE FLASHCARD_USAGE_HISTORY (
    USAGE_HISTORY_ID INT PRIMARY KEY,      -- Identificador único do
                                            -- histórico de uso
    ACCESS_DATE TIMESTAMP,                  -- Data e hora de acesso ao
                                            -- flashcard
    FLASHCARD_ID INT,                      -- Identificador do
                                            -- flashcard
    USER_ID INT,                          -- Identificador do usuário
                                            -- que acessou o flashcard
    FOREIGN KEY (FLASHCARD_ID) REFERENCES FLASHCARD(FLASHCARD_ID), -- Relacionamento com o flashcard
    FOREIGN KEY (USER_ID) REFERENCES USER(USER_ID) -- Relacionamento com o usuário
);
```

#### Tabela PivoFLASHCARD\_HISTORY\_ITEMS

- Tabela para armazenar os itens do histórico de uso dos flashcards

```
CREATE TABLE FLASHCARD_HISTORY_ITEMS (
    USAGE_HISTORY_ID INT,                  -- Identificador do
                                            -- histórico de uso
    FLASHCARD_ID INT,                      -- Identificador do
                                            -- flashcard
    PRIMARY KEY (USAGE_HISTORY_ID, FLASHCARD_ID), -- Chave composta
                                            -- para garantir a associação
    FOREIGN KEY (USAGE_HISTORY_ID) REFERENCES
    FLASHCARD_USAGE_HISTORY(USAGE_HISTORY_ID), -- Relacionamento com o
```

```

histórico de uso
  FOREIGN KEY (FLASHCARD_ID) REFERENCES FLASHCARD(FLASHCARD_ID) -
- Relacionamento com o flashcard
);

```

**Tabela TABLE\_BOX**

- Tabela para armazenar as caixas de flashcards compradas pelos usuários

```

CREATE TABLE BOX (
  BOX_ID INT PRIMARY KEY,                               -- Identificador único da
  caixa
  PRICE INT,                                         -- Preço da caixa
  TITLE VARCHAR(120),                                -- Título da caixa
  COVER BLOB,                                         -- Capa da caixa
  DESCRIPTION TEXT,                                   -- Descrição da caixa
  STATUS BOOLEAN,                                     -- Status da caixa (ativa,
  inativa, etc.)
  USER_ID INT,                                         -- Identificador do usuário
  que comprou ou criou a caixa
  FOREIGN KEY (USER_ID) REFERENCES USER(USER_ID) -- Relacionamento com o usuário
);

```

**Tabela BOX\_FLASHCARD**

- Tabela para associar flashcards a caixas

```

CREATE TABLE BOX_FLASHCARD (
  BOX_ID INT,                                         -- Identificador da caixa
  FLASHCARD_ID INT,                                    -- Identificador do
  flashcard
  PRIMARY KEY (BOX_ID, FLASHCARD_ID), -- Chave composta para
  garantir que um flashcard só apareça uma vez na caixa
  FOREIGN KEY (BOX_ID) REFERENCES BOX(BOX_ID), -- Relacionamento

```

```

com a caixa
    FOREIGN KEY (FLASHCARD_ID) REFERENCES FLASHCARD(FLASHCARD_ID) -
- Relacionamento com o flashcard
);

```

**Tabela PURCHASE**

- Tabela para registrar as compras realizadas pelos usuários

```

CREATE TABLE PURCHASE (
    EMAIL VARCHAR(150) UNIQUE,           -- Email único do usuário
    que fez a compra
    RULE VARCHAR(10),                   -- Função do usuário que
    fez a compra
    AVATAR BLOB,                      -- Avatar do usuário que
    fez a compra
    PURCHASE_ID INT PRIMARY KEY,       -- Identificador único da
    compra
    PURCHASE_DATE TIMESTAMP,           -- Data e hora da compra
    STATUS VARCHAR(20),                -- Status da compra
    (pendente, concluída, etc.)
    AMOUNT INT,                       -- Quantidade de itens na
    compra
    USER_ID INT,                      -- Identificador do usuário
    que fez a compra
    FOREIGN KEY (USER_ID) REFERENCES USER(USER_ID) --
    Relacionamento com o usuário
);

```

**Tabela PivoPURCHASE\_BOX**

- Tabela para associar caixas a compras

```

CREATE TABLE PURCHASE_BOX (
    PURCHASE_ID INT,                  -- Identificador da compra

```

```

    BOX_ID INT,                      -- Identificador da caixa
comprada
    PRIMARY KEY (PURCHASE_ID, BOX_ID), -- Chave composta para
garantir que a mesma caixa não apareça mais de uma vez na mesma
compra
    FOREIGN KEY (PURCHASE_ID) REFERENCES PURCHASE(PURCHASE_ID), -- Relacionamento com a compra
    FOREIGN KEY (BOX_ID) REFERENCES BOX(BOX_ID) -- Relacionamento
com a caixa
);

```

**Tabela POMODORO\_STORY**

- Tabela para registrar as sessões de pomodoro dos usuários:

```

CREATE TABLE POMODORO_STORY (
    POMODORO_STORY_ID INT PRIMARY KEY,      -- Identificador único da
sessão de pomodoro
    START_DATE_TIMESTAMP TIMESTAMP,          -- Data e hora de início
da sessão
    END_DATE_TIMESTAMP TIMESTAMP,           -- Data e hora de término
da sessão
    USER_ID INT,                          -- Identificador do
usuário que realizou a sessão
    FOREIGN KEY (USER_ID) REFERENCES USER(USER_ID) -- Relacionamento com o usuário
);

```

**Tabela LOGIN\_SESSION**

- Tabela para registrar as sessões de login dos usuários:

```

CREATE TABLE LOGIN_SESSION (
    SESSION_ID INT PRIMARY KEY,            -- Identificador único da
sessão de login

```

```

    DEVICE VARCHAR(100) ,           -- Dispositivo usado para
login
    LOGIN_DATE_TIMESTAMP TIMESTAMP , -- Data e hora do login
    LOGOUT_DATE_TIMESTAMP TIMESTAMP , -- Data e hora do logout
    IP_ADDRESS VARCHAR(255) ,        -- Endereço IP de onde o
usuário fez login
    USER_ID INT ,                  -- Identificador do
usuário que fez login
    FOREIGN KEY (USER_ID) REFERENCES USER(USER_ID) --
Relacionamento com o usuário
);

```

## UML

A Unified Modeling Language (UML) é uma linguagem de modelagem visual padronizada amplamente utilizada na indústria de desenvolvimento de software. Conforme definido pela Object Management Group (OMG), "a UML fornece uma maneira padronizada de visualizar a estrutura e o comportamento de um sistema de software" (OMG, 2017).

Os principais diagramas UML incluem:

1. Diagrama de Casos de Uso (Use Case Diagram): Representa as principais funcionalidades do sistema e as interações entre os atores (usuários, sistemas externos, etc.) e o sistema.
2. Diagrama de Classes (Class Diagram): Descreve a estrutura estática do sistema, mostrando as classes, seus atributos, métodos e os relacionamentos (herança, associação, composição, etc.) entre elas.
3. Diagrama de Sequência (Sequence Diagram): Ilustra a sequência de interações entre objetos ao longo do tempo, capturando a dinâmica do sistema e a troca de mensagens entre eles.
4. Diagrama de Atividades (Activity Diagram): Modela o fluxo de trabalho e as atividades envolvidas em um processo ou em um caso de uso, destacando a ordem e as condições de execução.

Outros diagramas UML, como Diagrama de Pacotes, Diagrama de Componentes e Diagrama de Implantação, são utilizados para modelar diferentes aspectos do sistema, como sua organização modular, elementos de implementação e implantação.

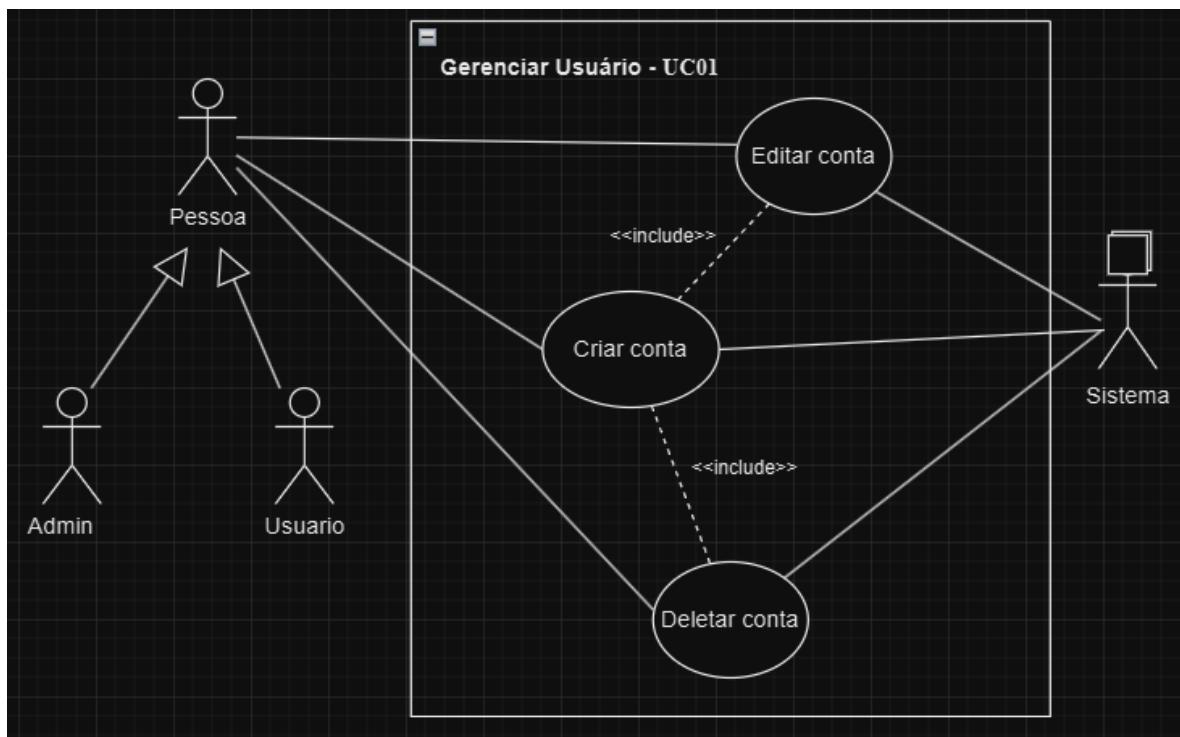
De acordo com a especificação da OMG, "a UML é uma linguagem de modelagem visual unificada que permite aos desenvolvedores especificar, visualizar, construir e documentar os artefatos de um sistema de software" (OMG, 2017).

## **Diagrama de Casos de Uso**

### **UC01 - Gerenciar Usuário**

<b>Nome</b>	<b>Gerenciar usuário.</b>
<b>Descrição</b>	Este caso de uso é responsável pela criação e edição de contas de usuários.
<b>Fluxo de criação</b>	<ol style="list-style-type: none"> <li>1. O usuário seleciona, na área de login, a opção "criar conta".</li> <li>2. O usuário preenche as informações requisitadas pelo sistema, como nome da conta, e-mail e senha.</li> <li>3. O usuário confirma a senha.</li> <li>4. O usuário aceita os termos de uso.</li> <li>5. O sistema valida os dados inseridos.</li> <li>6. O sistema cria a conta do usuário, exibindo uma mensagem de sucesso.</li> </ol>
<b>Fluxo de edição</b>	<ol style="list-style-type: none"> <li>1. O usuário seleciona o menu.</li> <li>2. O sistema exibe as opções de menu.</li> <li>3. O usuário seleciona "conta".</li> <li>4. O sistema mostra as informações da conta.</li> <li>5. O usuário seleciona a opção "editar conta".</li> <li>6. O usuário altera as informações permitidas, como nome ou imagem de perfil.</li> <li>7. O sistema valida as alterações.</li> <li>8. O sistema atualiza as informações da conta e exibe uma mensagem de sucesso.</li> </ol>
<b>Autor</b>	Usuário e Sistema
<b>Pré-condições</b>	O usuário deve estar logado no sistema para editar a conta.

Pós-condições	Dados do usuário atualizados ou nova conta criada com sucesso.
Fluxo de Exceção	<ol style="list-style-type: none"> <li>1. Se o e-mail já estiver em uso, o sistema exibe uma mensagem indicando erro e solicita a correção.</li> <li>2. Se informações obrigatórias não forem preenchidas, o sistema exibe uma mensagem indicando os campos pendentes.</li> <li>3. O sistema não executa a ação até que todos os erros sejam corrigidos.</li> </ol>

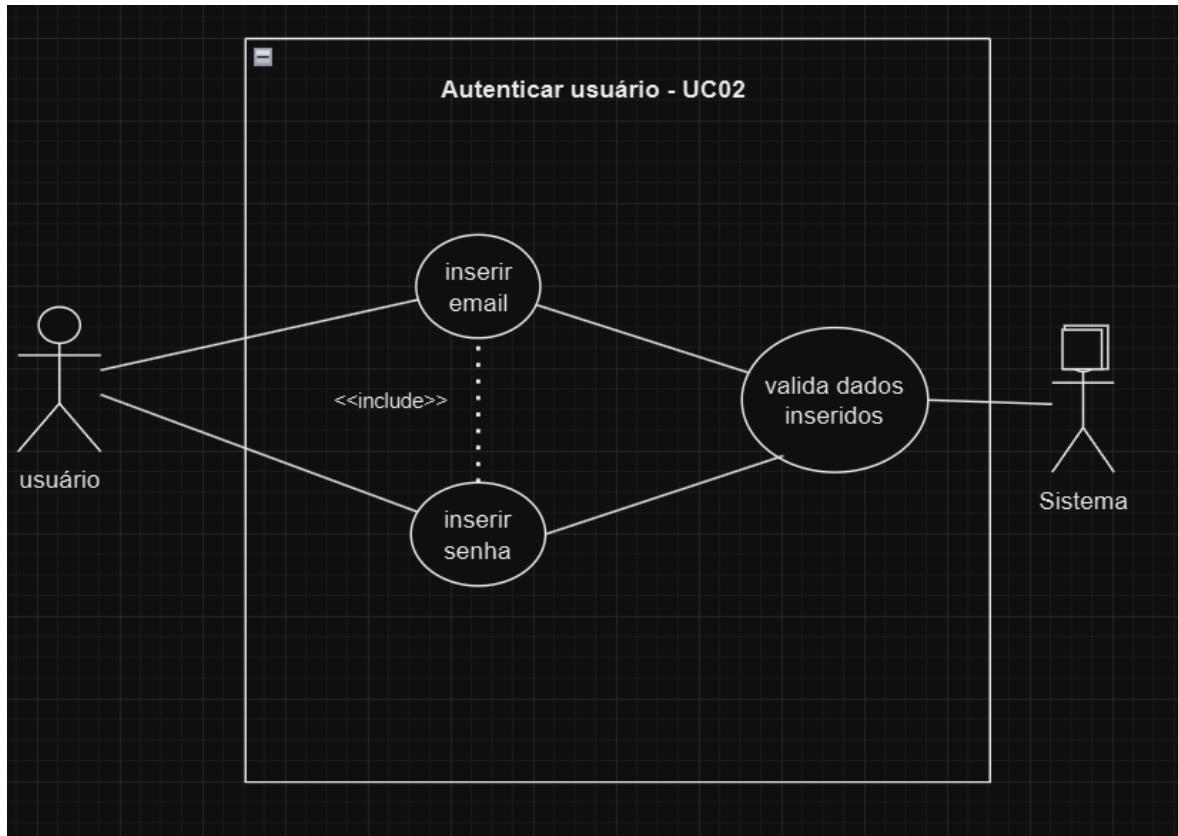


Gerenciar\_Usuário

Imagen Caso de Uso UC01

**UC02 - Autenticar Usuário**

<b>Nome</b>	<b>Autenticar usuário.</b>
<b>Descrição</b>	Este caso de uso é responsável pelo log in da conta de usuário.
<b>Fluxo Entrar</b>	<ol style="list-style-type: none"> <li>1. O usuário insere seu e-mail na página de login.</li> <li>2. O usuário insere sua senha na página de login.</li> <li>3. O sistema valida os dados inseridos.</li> <li>4. Caso os dados sejam válidos, o usuário é autenticado e redirecionado à página inicial.</li> </ol>
<b>Autor</b>	Usuário
<b>Pré-condições</b>	O usuário deve estar registrado no sistema.
<b>Pós-condições</b>	Entrada no sistema.
<b>Fluxo de Exceção</b>	<ol style="list-style-type: none"> <li>1. Caso o e-mail não esteja registrado, o sistema exibe uma mensagem de erro.</li> <li>2. Caso a senha esteja incorreta, o sistema exibe uma mensagem indicando erro e solicita nova tentativa.</li> </ol>

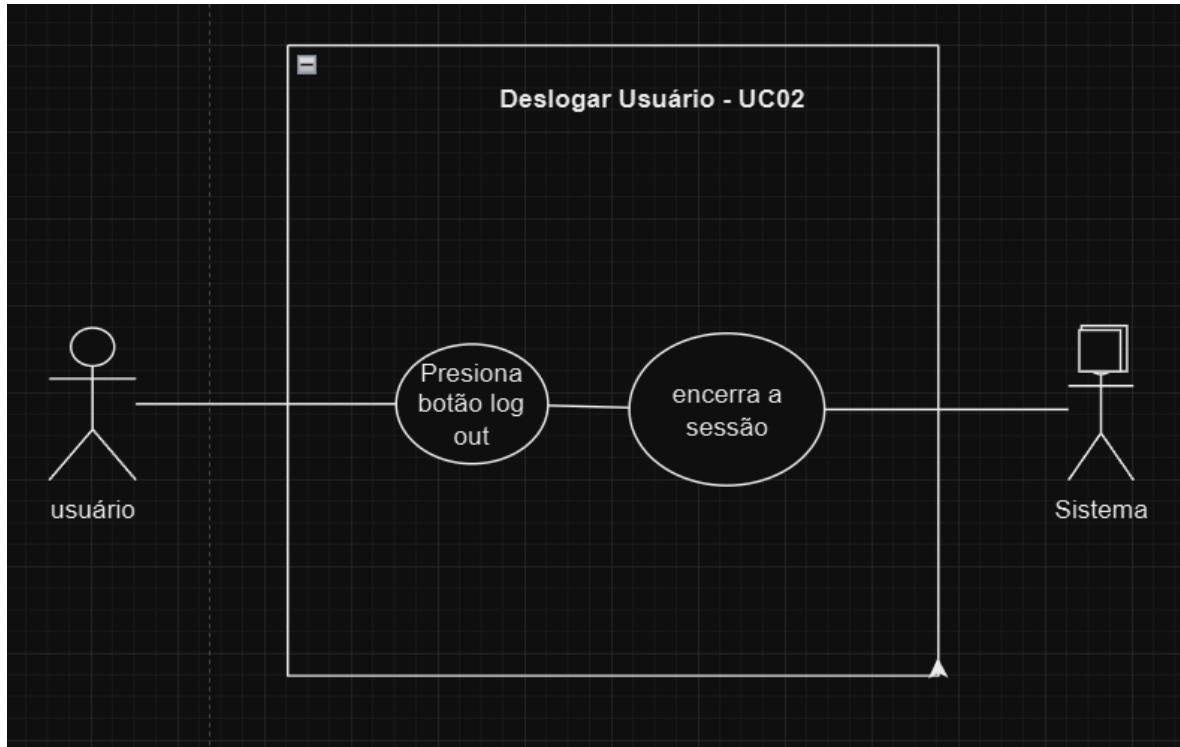


Uc02

Imagen Caso de Uso UC02

### UC03 - Deslogar Usuário

<b>Nome</b>	<b>Deslogar usuário.</b>
<b>Descrição</b>	Este caso de uso é responsável pelo log out da conta de usuário.
<b>Fluxo Sair</b>	<ol style="list-style-type: none"> <li>1. O usuário pressiona no perfil do menu.</li> <li>2. O usuário pressiona o botão "log out".</li> <li>3. O sistema exibe um pop-up para confirmação.</li> <li>4. O usuário confirma o log out.</li> <li>5. O sistema encerra a sessão e redireciona o usuário para a página de login.</li> </ol>
<b>Autor</b>	Usuário
<b>Pré-condições</b>	O usuário deve estar logado no sistema.
<b>Pós-condições</b>	Saída do sistema.



Uc03

Imagen Caso de Uso UC03

#### UC04 - Gerenciamento de Pomodoro

<b>Nome</b>	<b>Gerenciamento de Pomodoro</b>
<b>Descrição</b>	Este caso de uso descreve as funcionalidades de ativação, desativação, pausa e retomada do temporizador de Pomodoro no sistema.
<b>Fluxo de ativação</b>	<ol style="list-style-type: none"> <li>1. O usuário acessa a funcionalidade de Pomodoro no sistema.</li> <li>2. O sistema verifica se o usuário está autenticado e tem acesso à funcionalidade.</li> <li>3. O sistema exibe a interface do temporizador de Pomodoro.</li> <li>4. O usuário clica no botão "Iniciar Pomodoro".</li> <li>5. O sistema ativa o temporizador, definindo a duração padrão (ex.: 25 minutos).</li> <li>6. O sistema inicia a contagem regressiva e exibe o tempo restante.</li> <li>7. Ao atingir o final do ciclo, o sistema emite um aviso sonoro.</li> <li>8. O sistema exibe uma mensagem informando a conclusão.</li> </ol>
<b>Fluxo de pausa</b>	<ol style="list-style-type: none"> <li>1. Durante um ciclo ativo, o usuário clica no botão "Pausar Pomodoro".</li> <li>2. O sistema interrompe a contagem regressiva.</li> <li>3. O sistema altera o estado para "Pausado".</li> <li>4. O usuário pode retomar ou parar o ciclo.</li> </ol>
<b>Fluxo de retomada</b>	<ol style="list-style-type: none"> <li>1. Durante um ciclo pausado, o usuário clica no botão "Retomar Pomodoro".</li> <li>2. O sistema verifica se há um ciclo em estado "Pausado".</li> <li>3. O sistema retoma a contagem regressiva.</li> <li>4. O sistema atualiza a interface.</li> <li>5. Ao atingir o final do ciclo, o sistema emite um aviso sonoro e exibe uma mensagem informando a conclusão.</li> </ol>

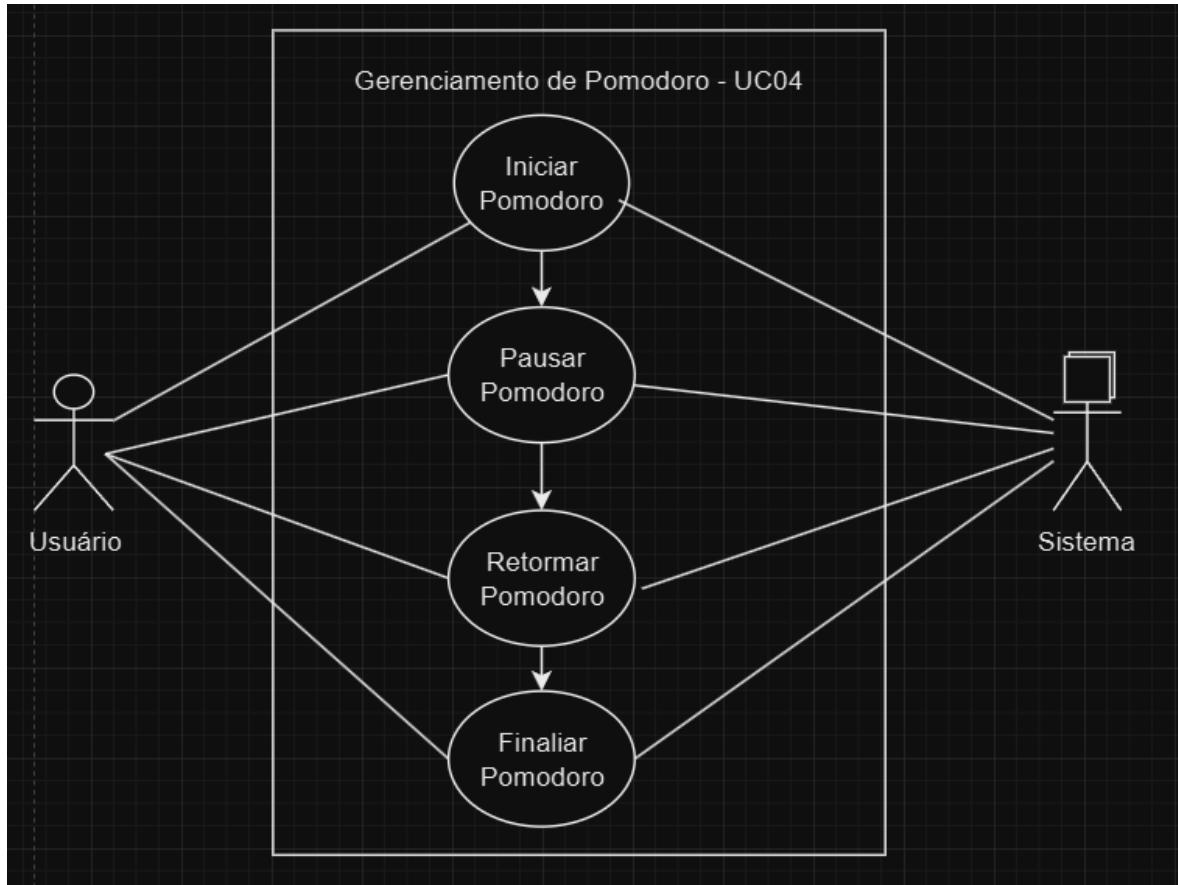
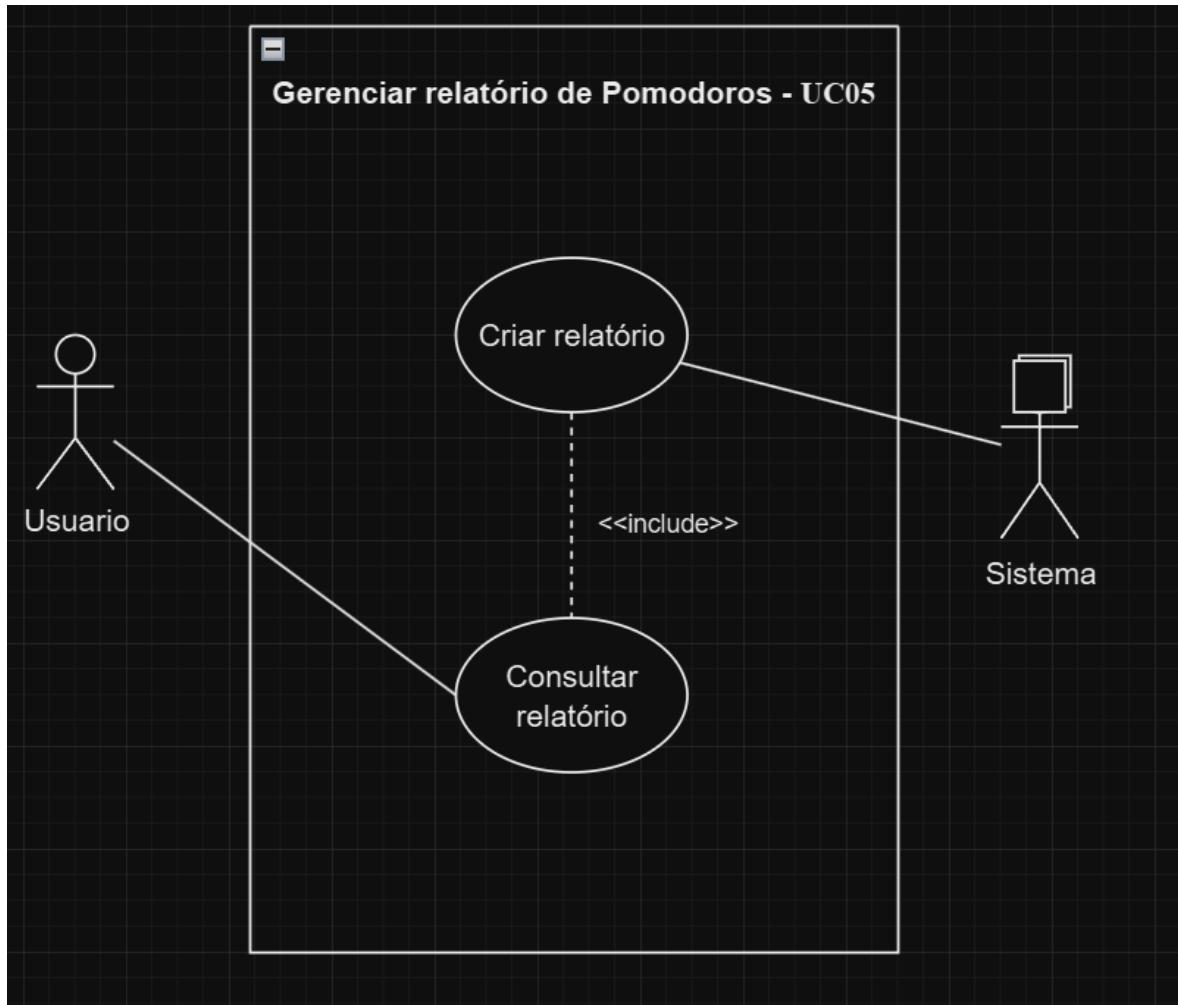


Imagen Caso de Uso UC04

### UC05 - Gerenciar Relatório de Pomodoros

<b>Nome</b>	<b>Gerenciar Relatório de Pomodoros</b>
<b>Descrição</b>	Este caso de uso é responsável pela criação e consulta de relatórios dos Pomodoros.
<b>Fluxo de criação</b>	<ol style="list-style-type: none"> <li>1. O usuário seleciona a opção “criar relatório de Pomodoro”.</li> <li>2. O usuário informa o período desejado (datas de início e fim).</li> <li>3. O sistema gera o relatório com base nos dados.</li> <li>4. O sistema valida os dados e exibe o relatório gerado.</li> <li>5. O usuário confirma o relatório.</li> </ol>
<b>Fluxo de consulta</b>	<ol style="list-style-type: none"> <li>1. O usuário seleciona a opção “consultar relatório de Pomodoro”.</li> <li>2. O usuário informa o período desejado ou escolhe relatórios existentes.</li> <li>3. O sistema exibe o relatório solicitado.</li> <li>4. O usuário visualiza o relatório.</li> </ol>
<b>Autor</b>	Usuário e Sistema
<b>Pré-condições</b>	O usuário deve estar logado no sistema.
<b>Pós-condições</b>	Relatório de Pomodoros criado ou consultado.



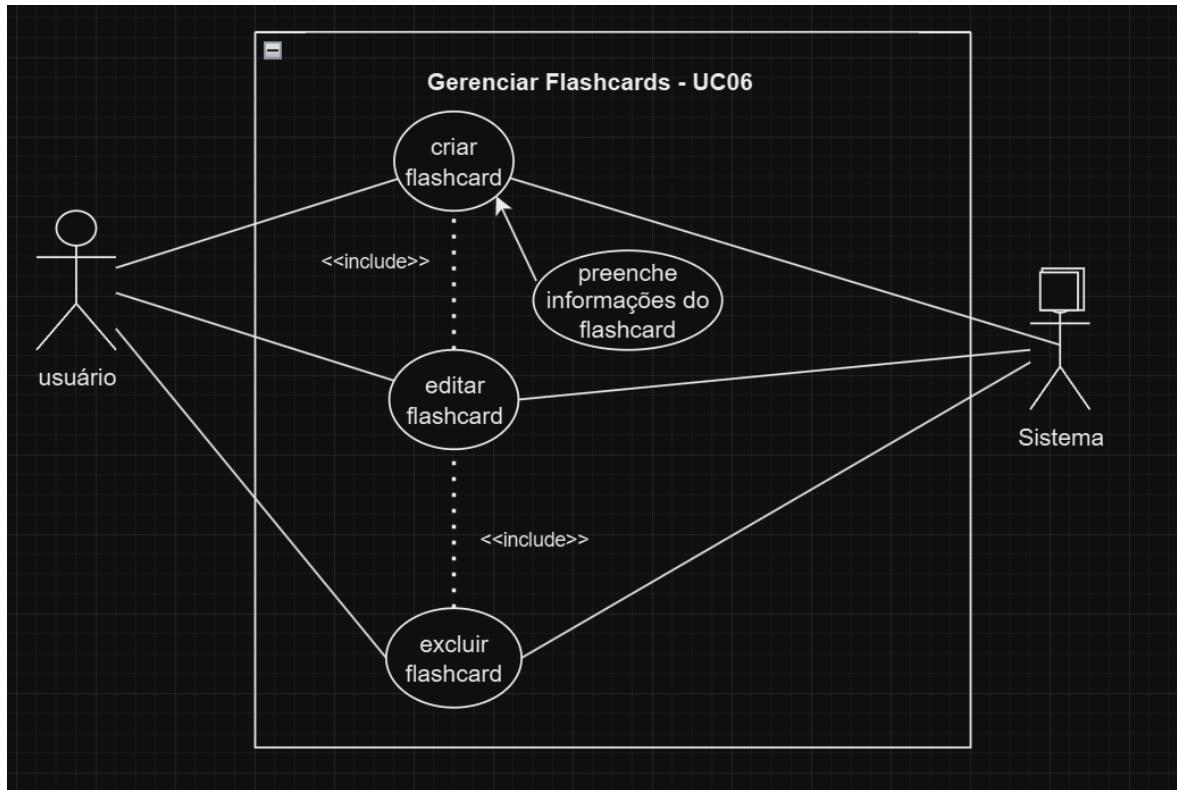
Uc05

Imagen Caso de Uso UC05

#### UC06 - Gerenciar Flashcards

<b>Nome</b>	<b>Gerenciar Flashcards</b>
<b>Descrição</b>	Este caso de uso é responsável pela criação, edição e consulta de flash cards do usuário.
<b>Fluxo de criação</b>	<ol style="list-style-type: none"> <li>1. O usuário seleciona a opção de adicionar flashcards “+” logo abaixo do Pomodoro.</li> <li>2. O sistema exibe um formulário para que o usuário preencha as informações do novo flashcard.</li> <li>3. O usuário preenche as informações solicitadas.</li> <li>4. O usuário clica no botão "Salvar".</li> <li>5. O sistema valida as informações.</li> <li>6. Caso válidas, o sistema salva o novo flashcard.</li> <li>7. O sistema exibe uma mensagem de sucesso.</li> <li>8. O sistema atualiza a lista de flashcards do usuário.</li> </ol>
<b>Fluxo de edição</b>	<ol style="list-style-type: none"> <li>1. O usuário seleciona os três pontos do flashcard.</li> <li>2. O sistema exibe as opções de favoritar, editar ou excluir.</li> <li>3. O usuário seleciona a opção “editar”.</li> <li>4. O usuário preenche os novos dados do flashcard.</li> <li>5. O sistema valida as informações.</li> <li>6. Caso válidas, o flashcard é atualizado no banco de dados.</li> <li>7. O sistema exibe uma mensagem de sucesso.</li> <li>8. O sistema atualiza a lista de flashcards do usuário.</li> </ol>
<b>Autor</b>	Usuário e Sistema
<b>Pré-condições</b>	O usuário deve estar logado no sistema.

Pós-condições	Alteração da lista de flashcards do usuário.
---------------	----------------------------------------------

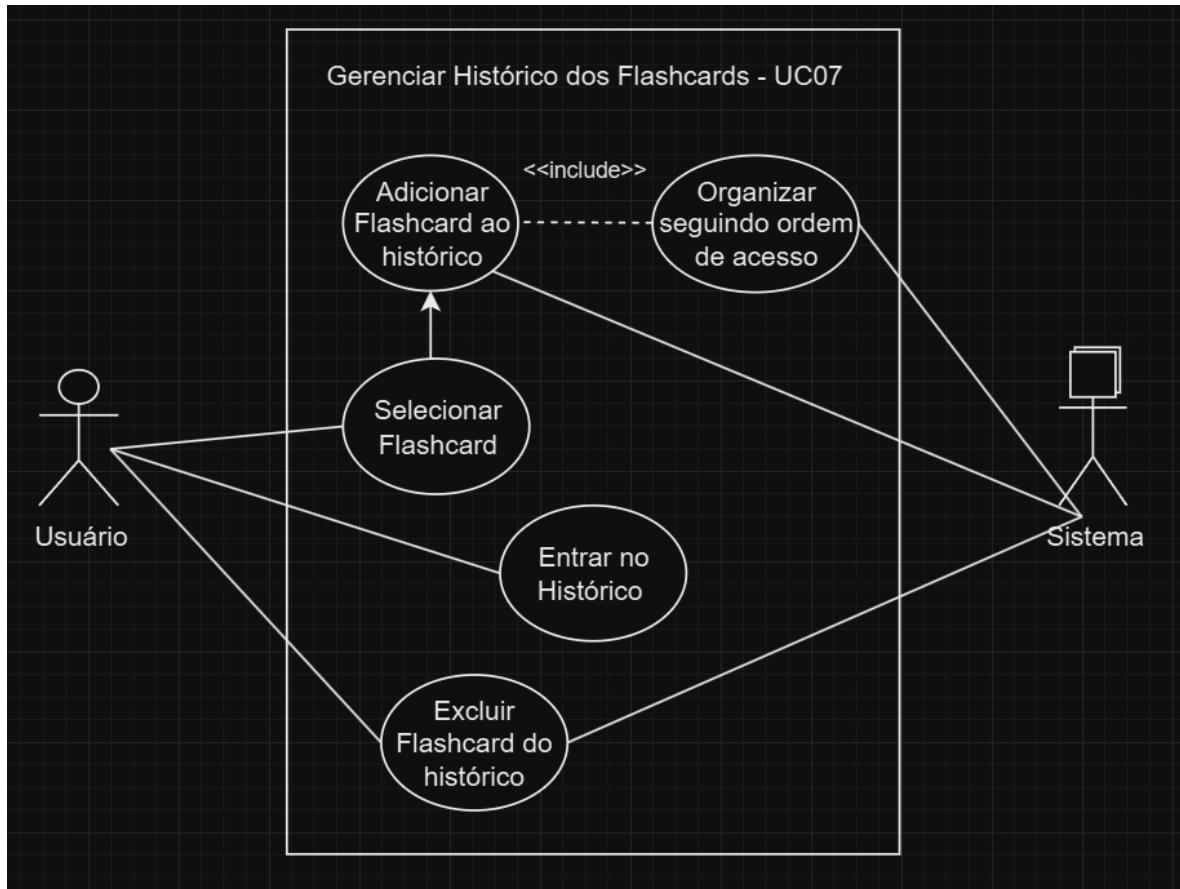


Uc06

Imagen Caso de Uso UC06

### UC07 - Gerenciar Histórico dos Flashcards

<b>Nome</b>	<b>Gerenciar Histórico dos Flashcards</b>
<b>Descrição</b>	Este caso de uso é responsável pela adição e remoção de flashcards do histórico.
<b>Fluxo principal</b>	<ol style="list-style-type: none"> <li>1. O usuário seleciona um flashcard.</li> <li>2. O sistema adiciona o flashcard no histórico.</li> <li>3. O sistema organiza os flashcards conforme a ordem de acesso.</li> </ol>
<b>Fluxo de reação</b>	<ol style="list-style-type: none"> <li>1. O usuário seleciona a opção “menu”.</li> <li>2. O sistema exibe as opções de menu.</li> <li>3. O usuário seleciona a opção “flashcards”.</li> <li>4. O usuário acessa a interface de gerenciamento de flashcards.</li> <li>5. O usuário seleciona a opção “Histórico”.</li> <li>6. O usuário seleciona o flashcard correspondente.</li> <li>7. O usuário seleciona a opção “Remover do Histórico”.</li> <li>8. O sistema exibe uma mensagem de confirmação.</li> <li>9. Caso o usuário confirme, o sistema remove o flashcard do histórico. Caso contrário, volta à tela anterior.</li> </ol>
<b>Autor</b>	Usuário e Sistema
<b>Pré-condições</b>	O usuário deve estar logado no sistema.
<b>Pós-condições</b>	Alteração no histórico de flashcards do usuário.



Uc07

Imagen Caso de Uso UC07

### UC08 - Gerenciar Favoritos

Nome	<b>Gerenciar Favoritos</b>
<b>Descrição</b>	Este caso de uso é responsável pela adição e remoção de flashcards dos favoritos.
<b>Fluxo de favoritar</b>	<ol style="list-style-type: none"> <li>1. O usuário seleciona os três pontos no quadrado do flashcard específico.</li> <li>2. O sistema exibe as opções de favoritar, editar ou excluir.</li> <li>3. O usuário seleciona a opção “favoritar”.</li> </ol>
<b>Fluxo de desfavoritar</b>	<ol style="list-style-type: none"> <li>1. O usuário seleciona os três pontos no quadrado do flashcard específico.</li> <li>2. O sistema exibe as opções de favoritar, editar ou excluir.</li> <li>3. O usuário seleciona a opção “desfavoritar”.</li> </ol>
<b>Autor</b>	Usuário e Sistema
<b>Pré-condições</b>	O usuário deve estar logado no sistema.
<b>Pós-condições</b>	Alteração na lista de favoritos do usuário.

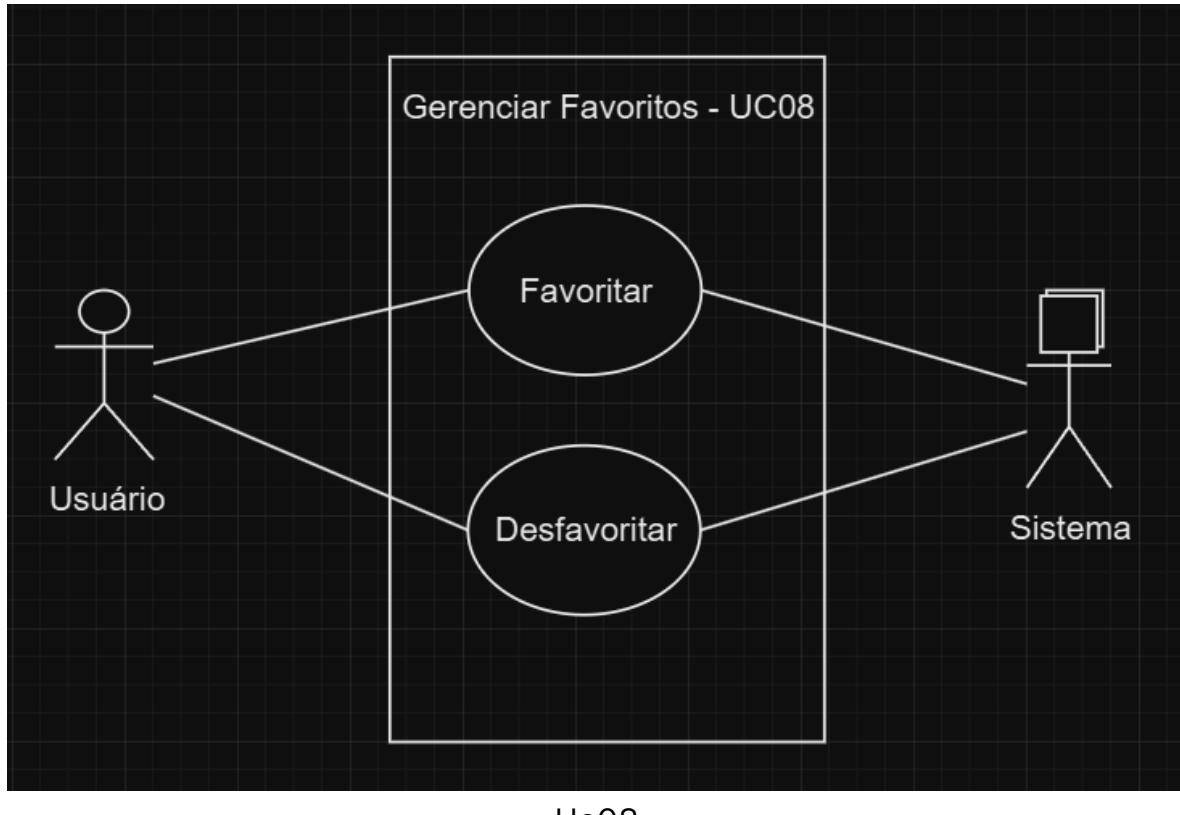
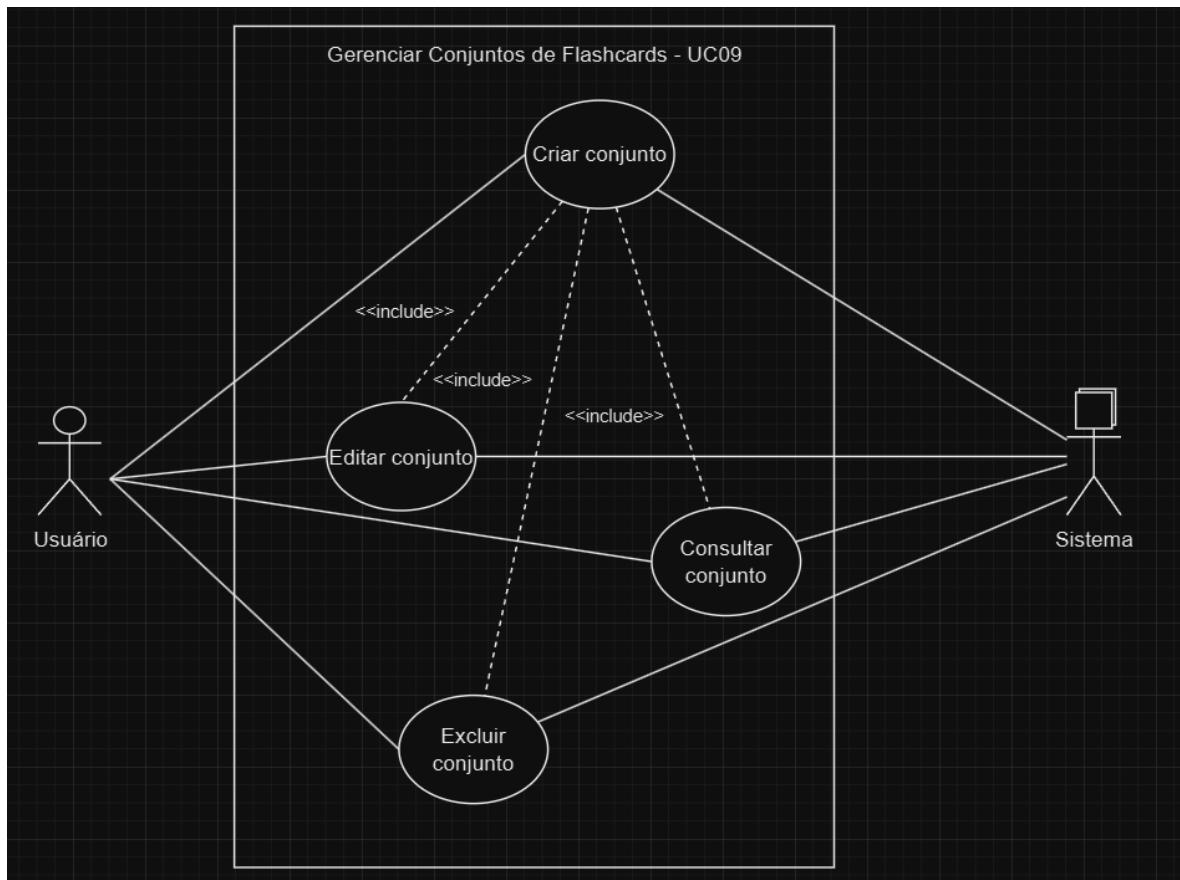


Imagen Caso de Uso UC08

### UC09 - Gerenciar Conjuntos de Flashcards

<b>Nome</b>	<b>Gerenciar Conjuntos</b>
<b>Descrição</b>	Este caso de uso é responsável pela criação, consulta, edição e exclusão das categorias de flashcards.
<b>Fluxo de criação</b>	<ol style="list-style-type: none"> <li>1. O usuário seleciona a opção de criar conjunto “+ com uma pasta” logo abaixo do Pomodoro.</li> <li>2. O sistema exibe um formulário para que o usuário preencha o nome, descrição e cor do novo conjunto.</li> <li>3. O usuário preenche os requisitos.</li> <li>4. O usuário confirma a criação do conjunto.</li> <li>5. O sistema valida as informações.</li> <li>6. O sistema atualiza as informações e adiciona o novo conjunto.</li> </ol>
<b>Fluxo de consulta</b>	<ol style="list-style-type: none"> <li>1. O usuário seleciona a lupa na parte dos conjuntos.</li> <li>2. O sistema abre um campo de texto para escrita.</li> <li>3. O usuário digita o nome do conjunto desejado.</li> <li>4. O usuário confirma a pesquisa.</li> <li>5. O sistema mostra os conjuntos que correspondem à pesquisa.</li> </ol>
<b>Fluxo de edição</b>	<ol style="list-style-type: none"> <li>1. O usuário seleciona a opção “ver mais” logo abaixo dos conjuntos.</li> <li>2. O sistema exibe um modal com todos os conjuntos.</li> <li>3. O usuário seleciona os três pontos de um conjunto.</li> <li>4. O sistema exibe as opções de editar ou excluir.</li> <li>5. O usuário escolhe “editar”.</li> <li>6. O sistema exibe o conjunto com opções para alterar flashcards, nome, descrição ou cor.</li> <li>7. O usuário confirma as alterações.</li> <li>8. O sistema valida e atualiza o conjunto no banco de dados.</li> </ol>

<b>Autor</b>	Usuário e Sistema
<b>Pré-condições</b>	O usuário deve estar logado no sistema.
<b>Pós-condições</b>	Criar, consultar, editar ou excluir as categorias dos flashcards.

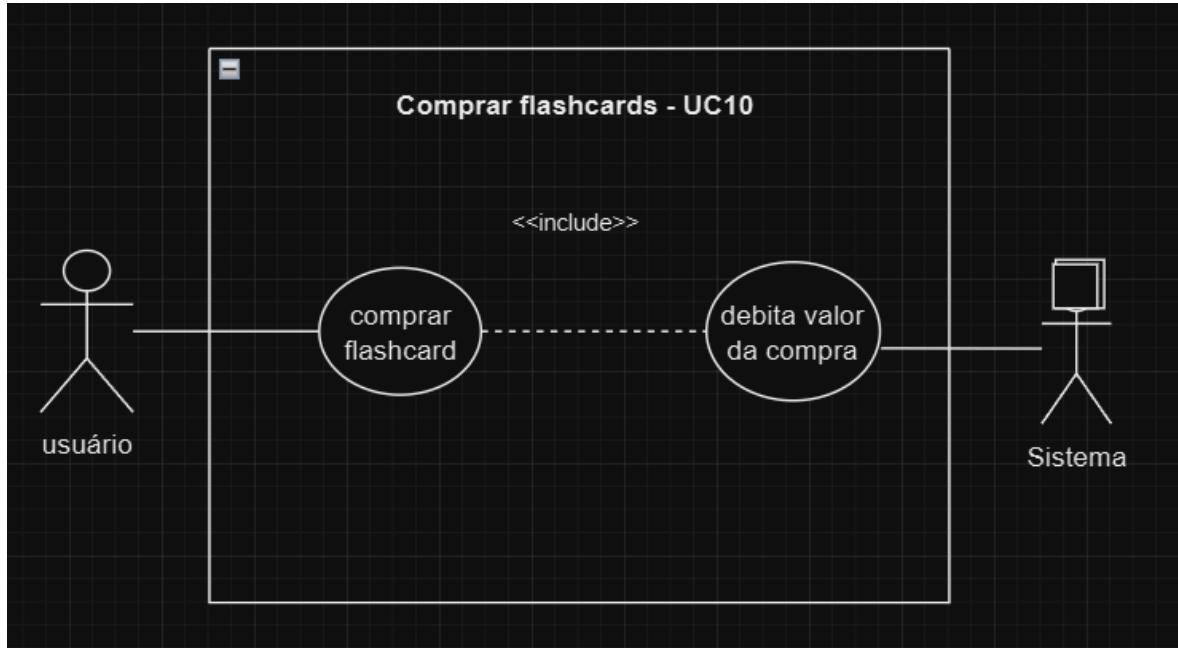


Uc09

Imagen Caso de Uso UC09

### UC10 - Comprar Flashcards

<b>Nome</b>	<b>Comprar Flashcards</b>
<b>Descrição</b>	Este caso de uso é responsável pela compra de flashcards usando a moeda do sistema.
<b>Fluxo de Compra</b>	<ol style="list-style-type: none"> <li>1. O usuário acessa a interface da loja.</li> <li>2. O sistema exibe uma lista de flashcards disponíveis para compra, com filtros por categoria e preço.</li> <li>3. O usuário seleciona o flashcard desejado.</li> <li>4. O sistema verifica se o usuário possui saldo suficiente.</li> <li>5. Caso tenha saldo, o sistema exibe uma mensagem de confirmação.</li> <li>6. O usuário confirma a compra.</li> <li>7. O sistema atualiza a lista de flashcards do usuário e débita o saldo da conta.</li> <li>8. O sistema exibe o flashcard comprado ao usuário.</li> </ol>
<b>Autor</b>	Usuário e Sistema
<b>Pré-condições</b>	O usuário deve estar logado no sistema e possuir saldo suficiente.
<b>Pós-condições</b>	Alteração no saldo e atualização da lista de flashcards do usuário.

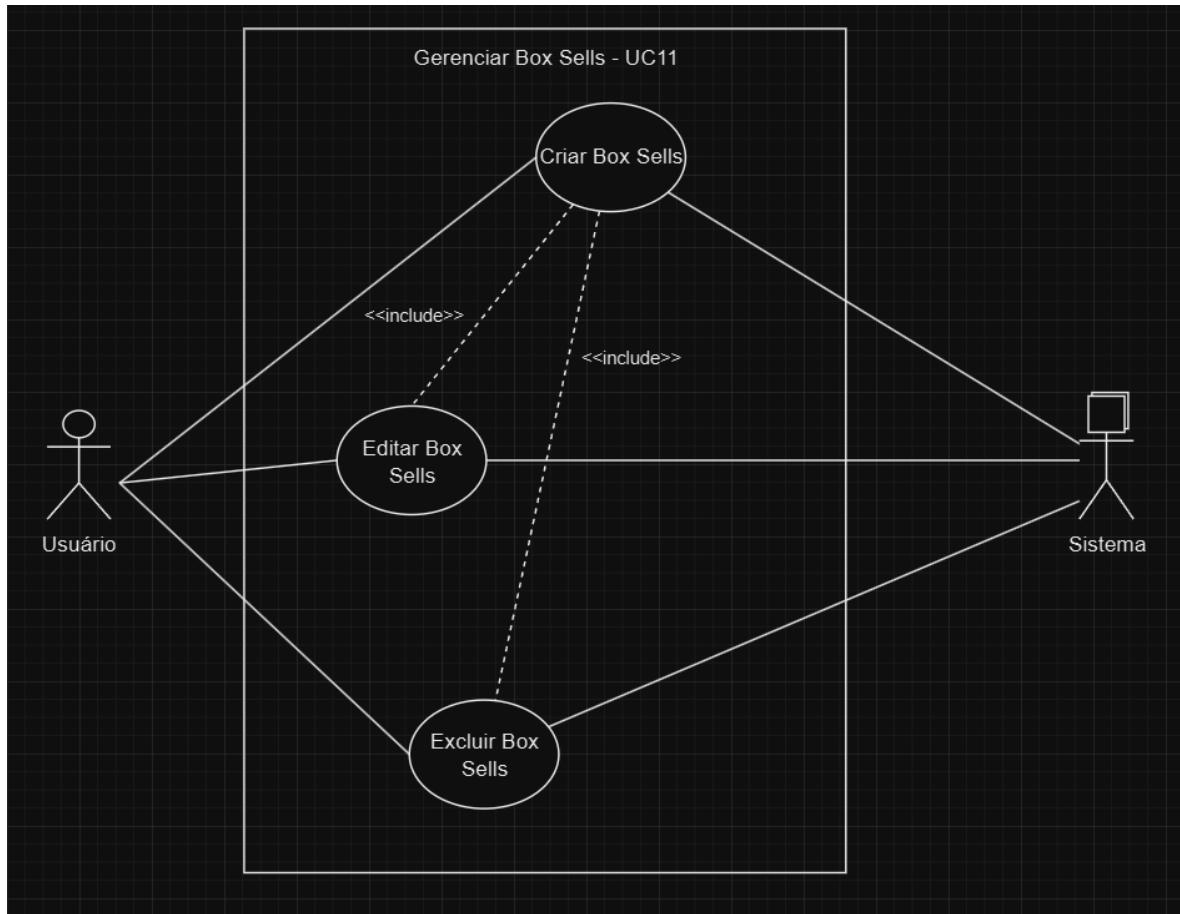


Uc10

Imagen Caso de Uso UC10

### UC11 - Gerenciar Box Sells

<b>Nome</b>	<b>Gerenciamento de Box Sells</b>
<b>Descrição</b>	Este caso de uso é responsável pela criação, edição e exclusão de Box Sells.
<b>Fluxo de criação</b>	<ol style="list-style-type: none"> <li>1. O usuário acessa a interface de gerenciamento de Box Sells.</li> <li>2. O usuário seleciona a opção “Criar Box Sell”.</li> <li>3. O sistema solicita ao usuário que defina um nome e uma descrição.</li> <li>4. O usuário escolhe os flashcards a serem incluídos.</li> <li>5. O sistema exibe um resumo da Box Sell com os flashcards selecionados.</li> <li>6. O usuário define o preço de venda.</li> <li>7. O usuário confirma a criação.</li> <li>8. O sistema adiciona a Box Sell à loja e à lista de vendas disponíveis.</li> </ol>
<b>Fluxo de edição</b>	<ol style="list-style-type: none"> <li>1. O usuário acessa a interface de gerenciamento de Box Sells.</li> <li>2. O usuário seleciona a Box Sell que deseja editar.</li> <li>3. O sistema exibe os detalhes da Box Sell.</li> <li>4. O usuário altera nome, descrição ou flashcards incluídos.</li> <li>5. O usuário confirma as alterações.</li> <li>6. O sistema valida e atualiza a Box Sell no banco de dados.</li> </ol>
<b>Autor</b>	Usuário e Sistema
<b>Pré-condições</b>	O usuário deve estar logado no sistema e ter permissões para gerenciar Box Sells.
<b>Pós-condições</b>	Alteração na lista de Box Sells disponíveis e no banco de dados.

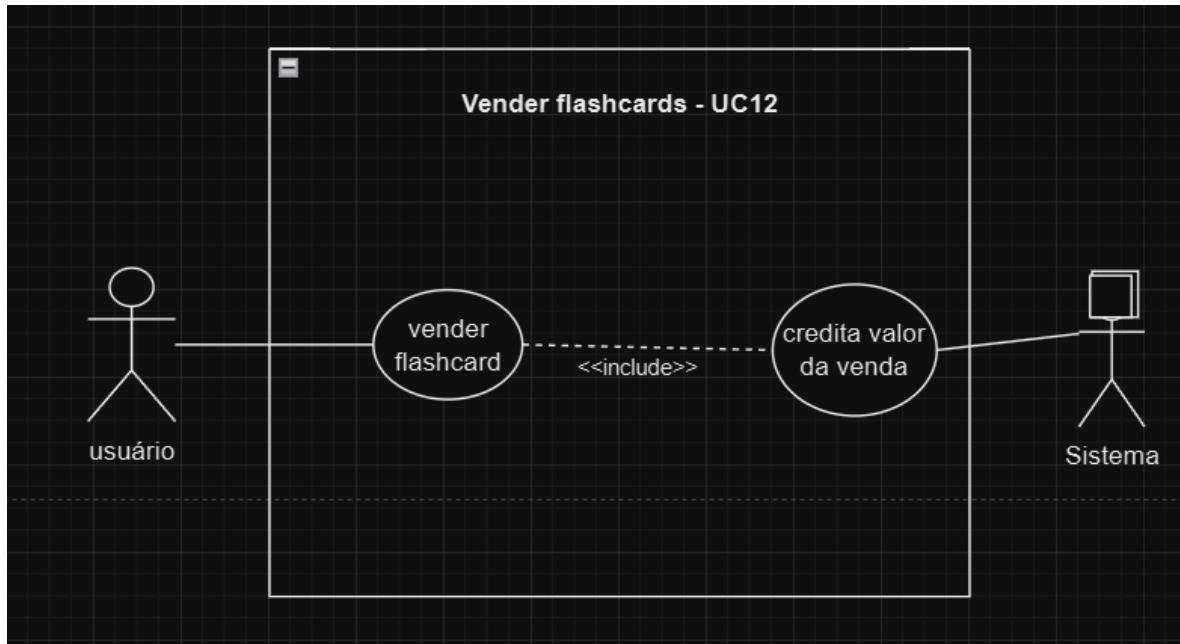


Uc11

Imagen Caso de Uso UC11

### UC12 - Vender Flashcards

<b>Nome</b>	<b>Vender Flashcards</b>
<b>Descrição</b>	Este caso de uso é responsável pela venda de flashcards usando a moeda do sistema.
<b>Fluxo de Venda</b>	<ol style="list-style-type: none"> <li>1. O usuário acessa a interface da loja.</li> <li>2. O usuário seleciona a opção “Venda de Flashcards”.</li> <li>3. O usuário escolhe o flashcard a ser vendido.</li> <li>4. O usuário define o valor da venda em moedas do sistema.</li> <li>5. O usuário confirma a venda.</li> <li>6. O sistema lista o flashcard para outros usuários.</li> <li>7. Quando vendido, o sistema credita o valor na conta do usuário e remove o flashcard.</li> </ol>
<b>Autor</b>	Usuário e Sistema
<b>Pré-condições</b>	O usuário deve estar logado no sistema.
<b>Pós-condições</b>	Alteração no saldo e na lista de flashcards do usuário.



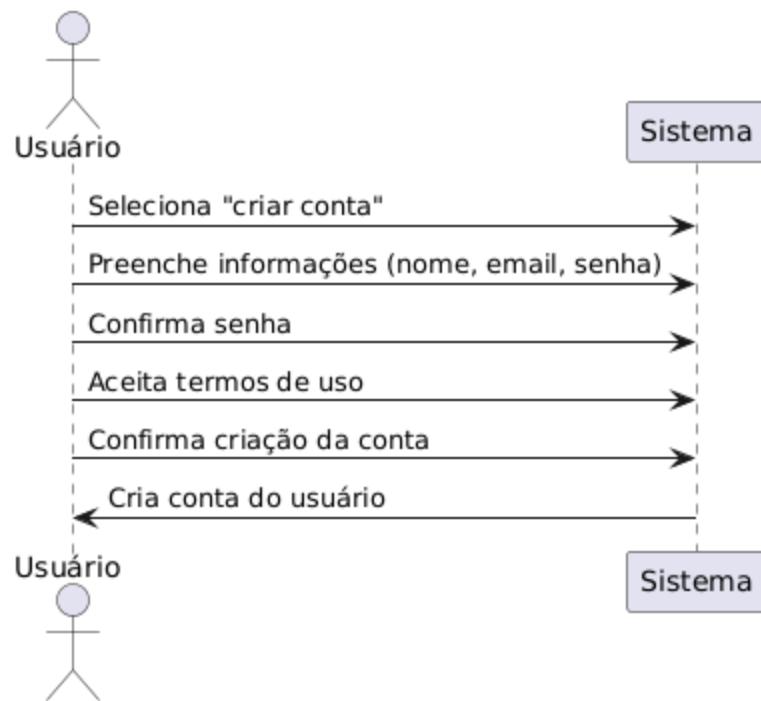
Uc12

Imagen Caso de Uso UC12

## Diagramas de Sequência

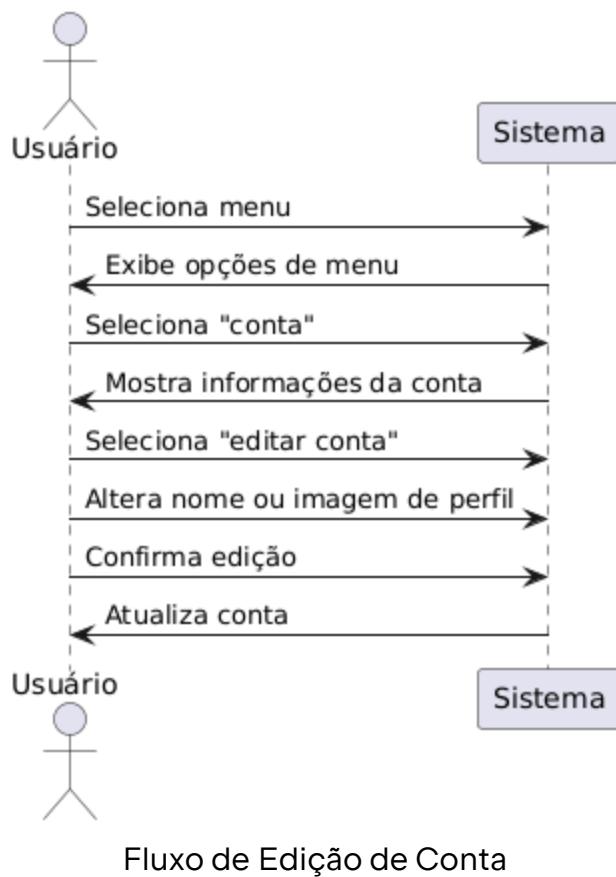
### Diagrama de Sequencia

#### Criação da conta

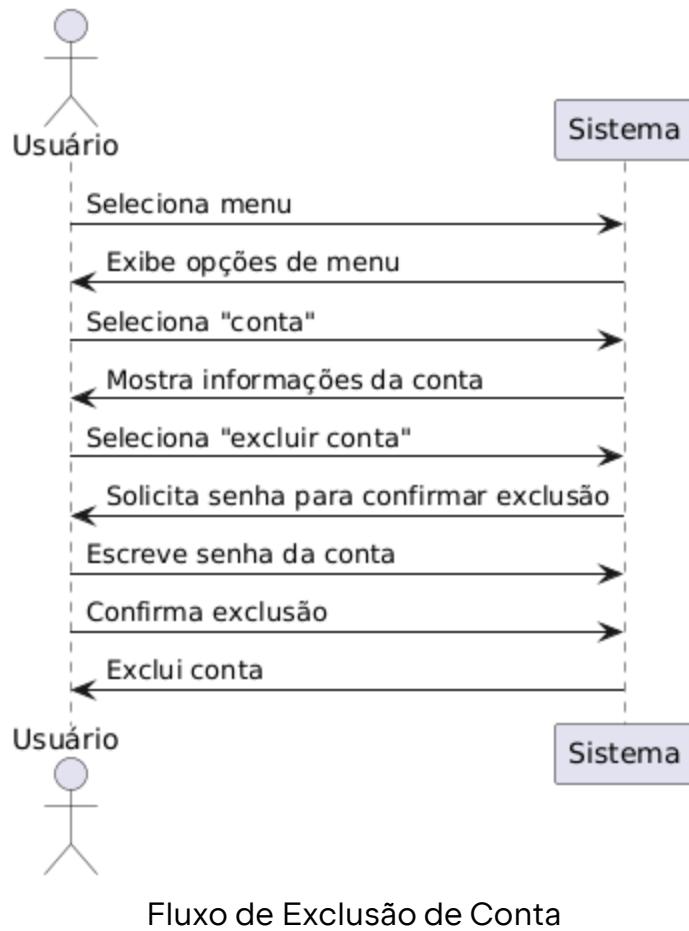


Fluxo de Criação de Conta

## Edição da conta

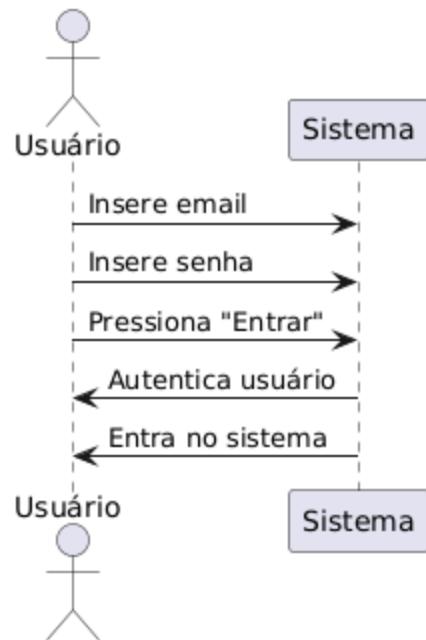


## Exclusão da conta



## Diagrama de Sequencia : UC-02

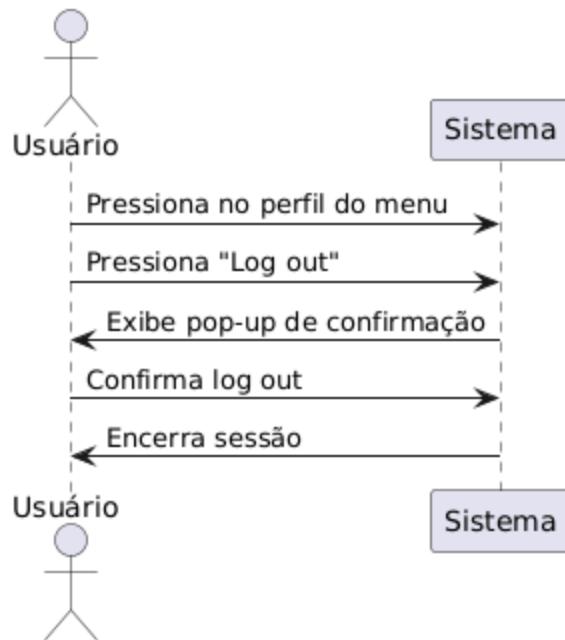
Entrada no Sistema



Fluxo de Entrar do Sistema

## Diagrama de Sequencia : UC-03

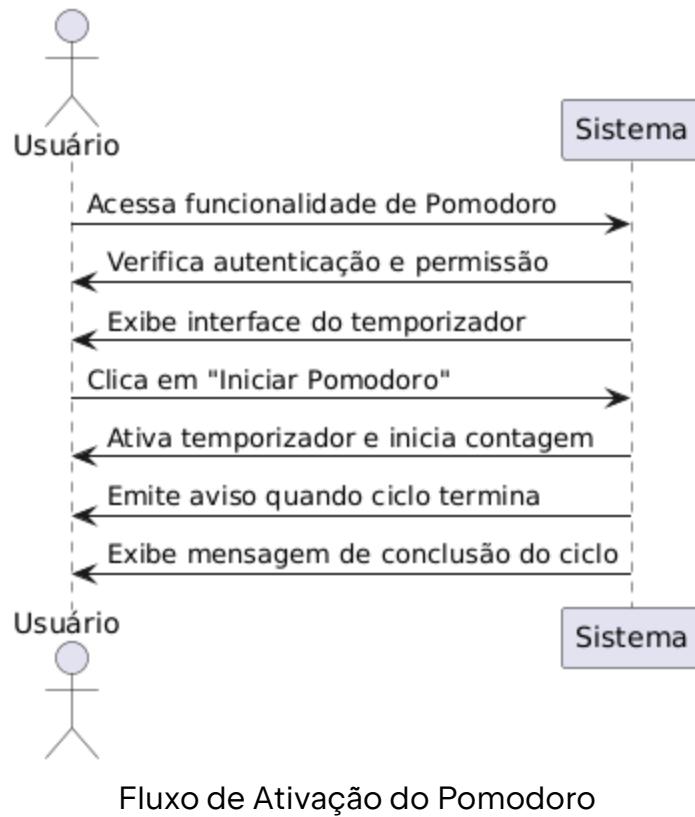
Saida do Sistema



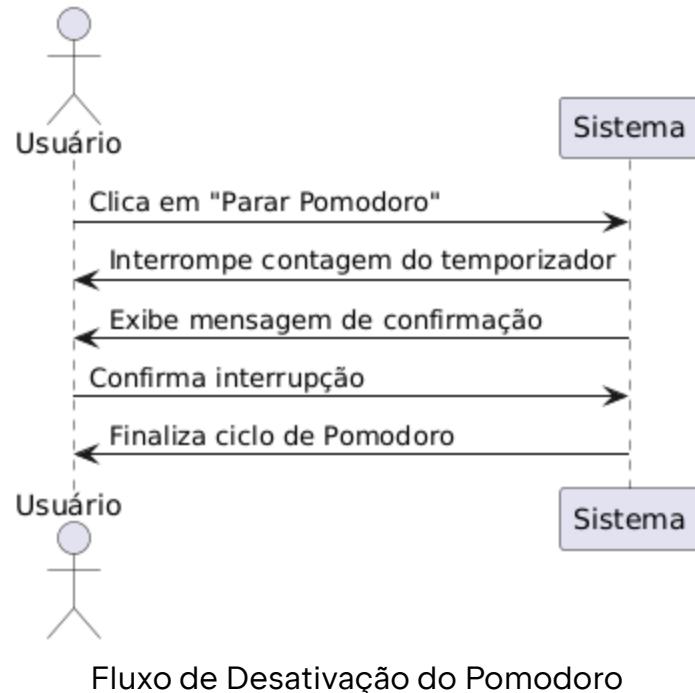
Fluxo de Saida do Sistema

## Diagrama de Sequencia : UC-04

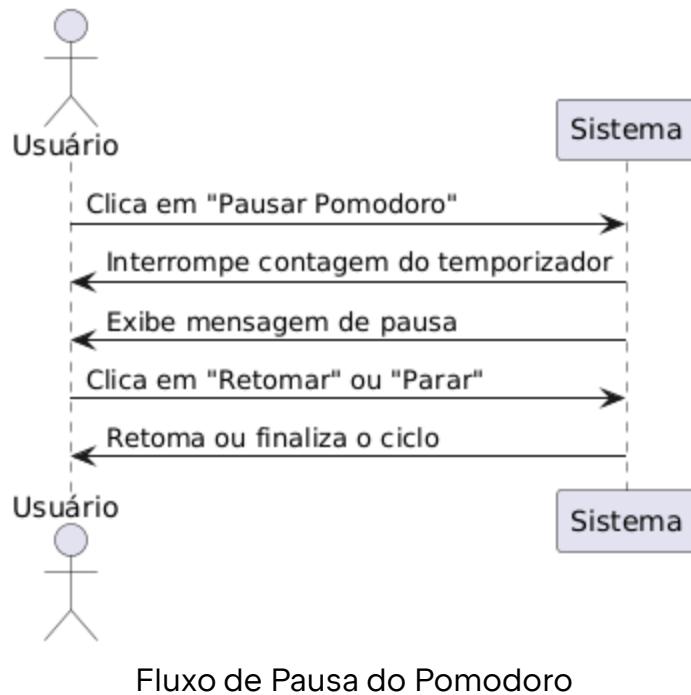
Ativação do Pomodoro



### Desativação do Pomodoro



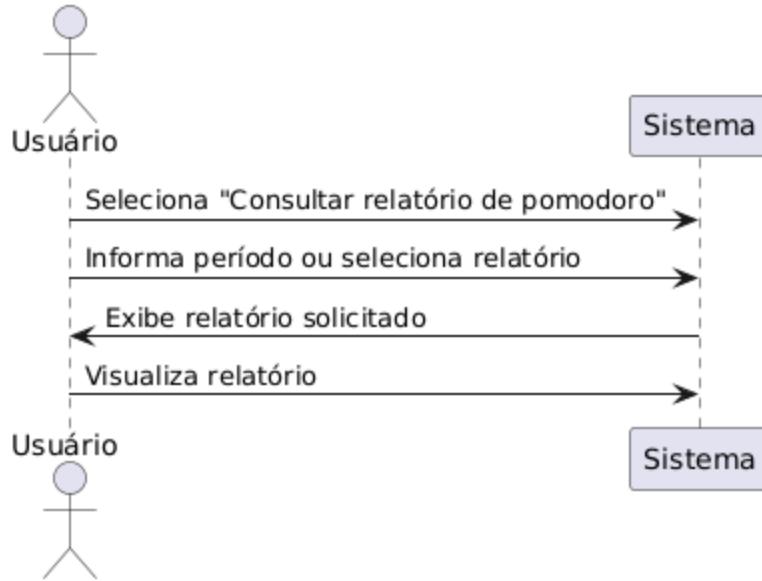
### Pausa do Pomodoro



Fluxo de Pausa do Pomodoro

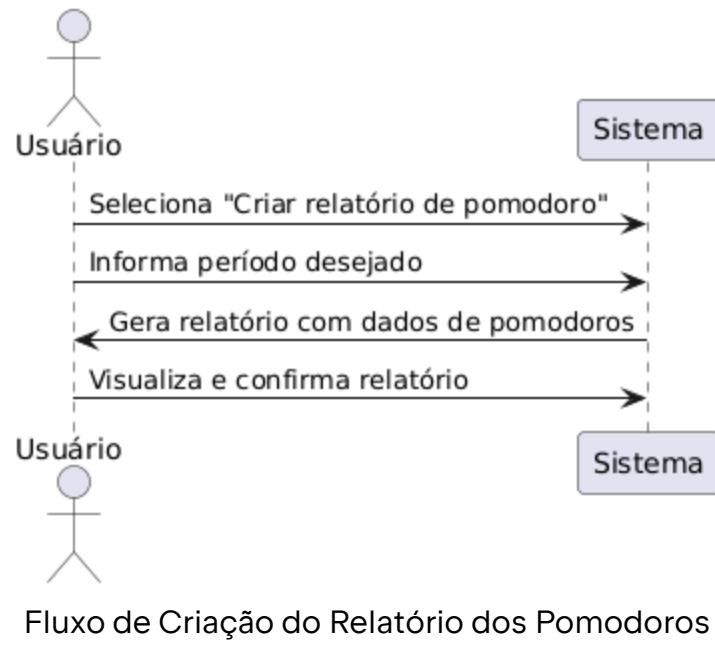
## Diagrama de Sequencia : UC-05

### Consulta do Relatórios



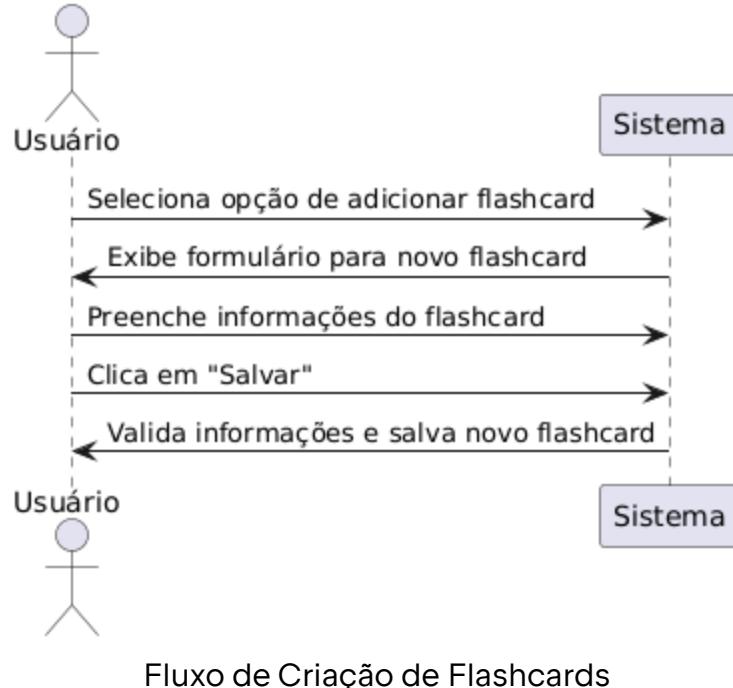
Fluxo de Consulta dos Relatório dos Pomodoros

### Criação dos Relatórios

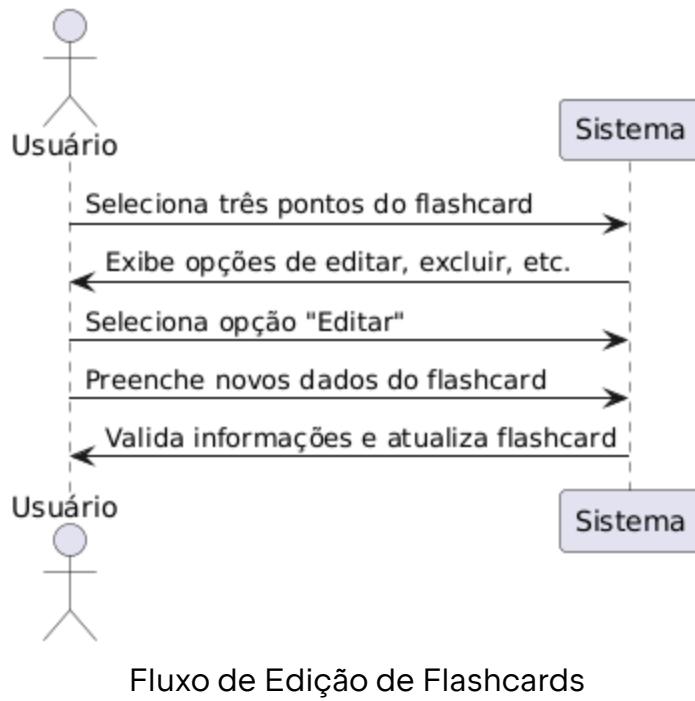


## Diagrama de Sequencia : UC-06

### Criação dos Flashcards

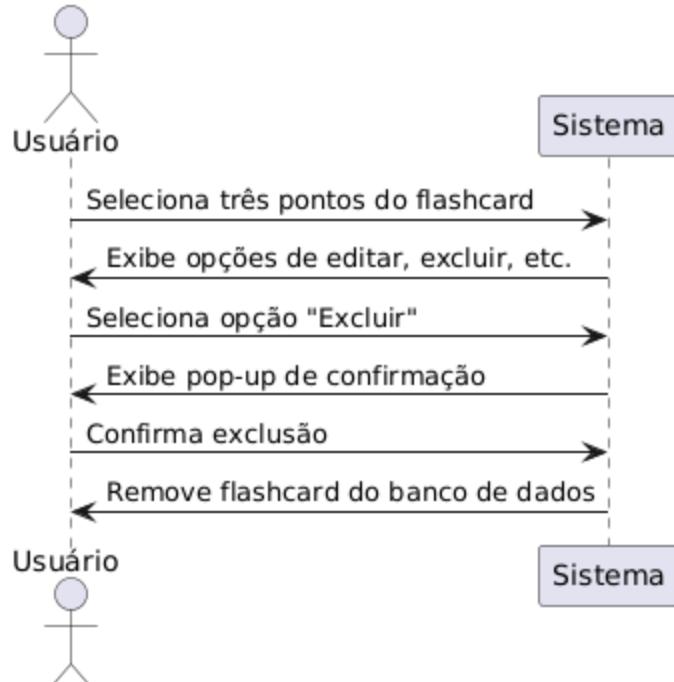


### Edição dos Flashcards



Fluxo de Edição de Flashcards

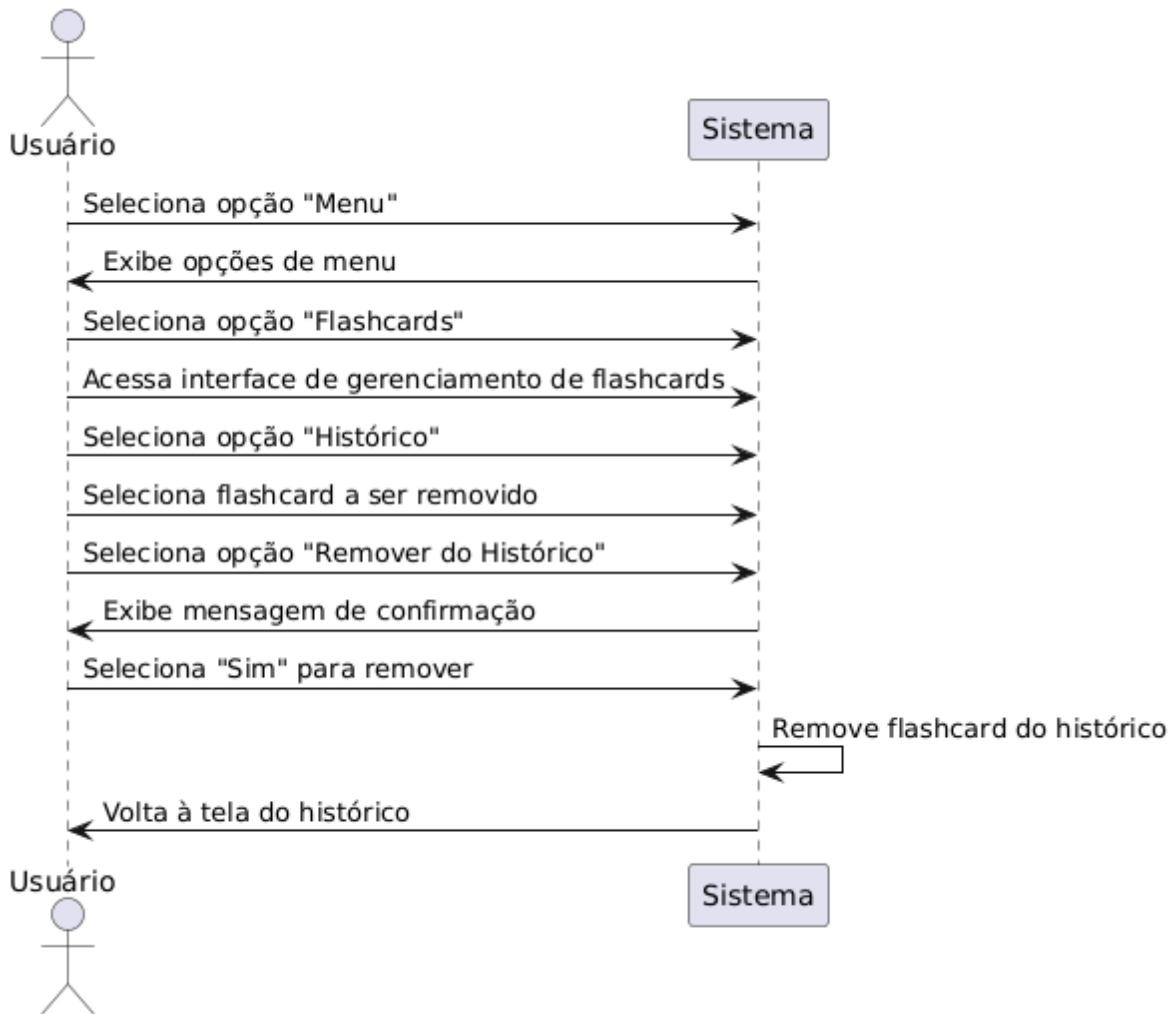
### Exclusão dos Flashcards



Fluxo de Exclusão de Flashcards

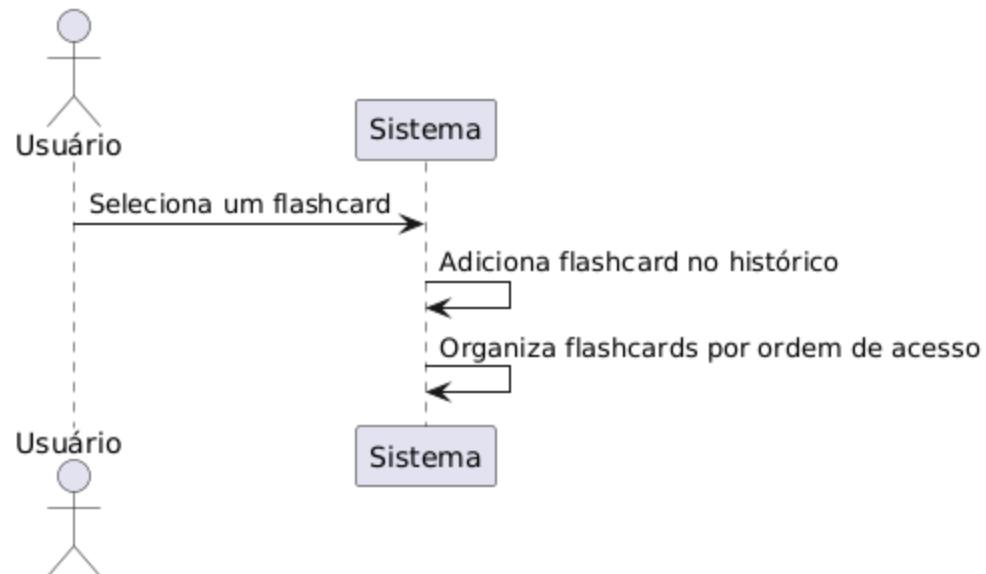
### Diagrama de Sequencia : UC-07

#### Remoção do Histórico dos Flashcards



Fluxo de Remoção de um Log do Histórico

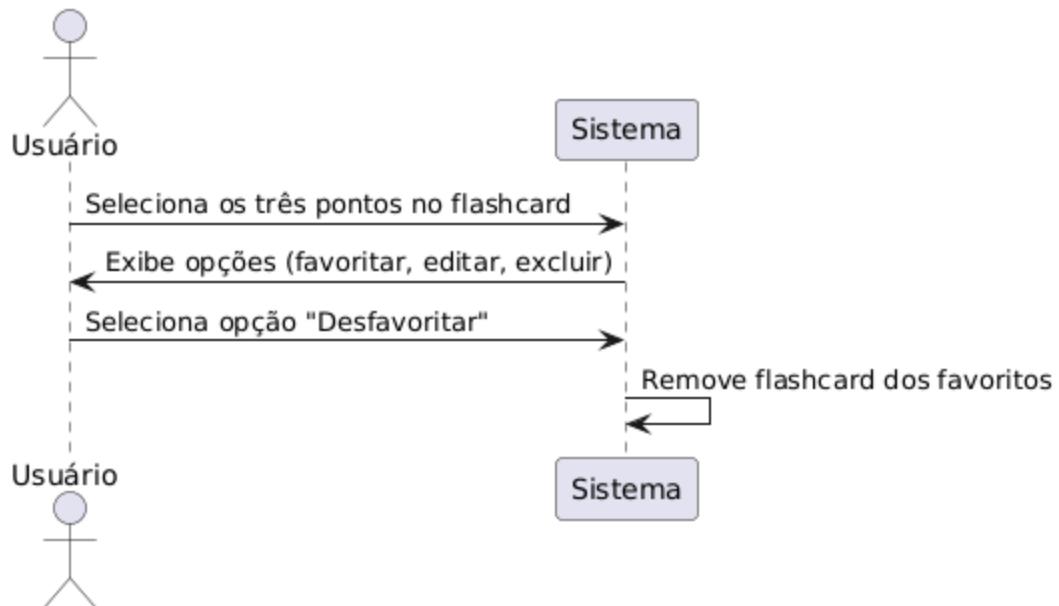
**Seleção de um Flashcard do Histórico**



Fluxo de Seleção de um Log do Histórico

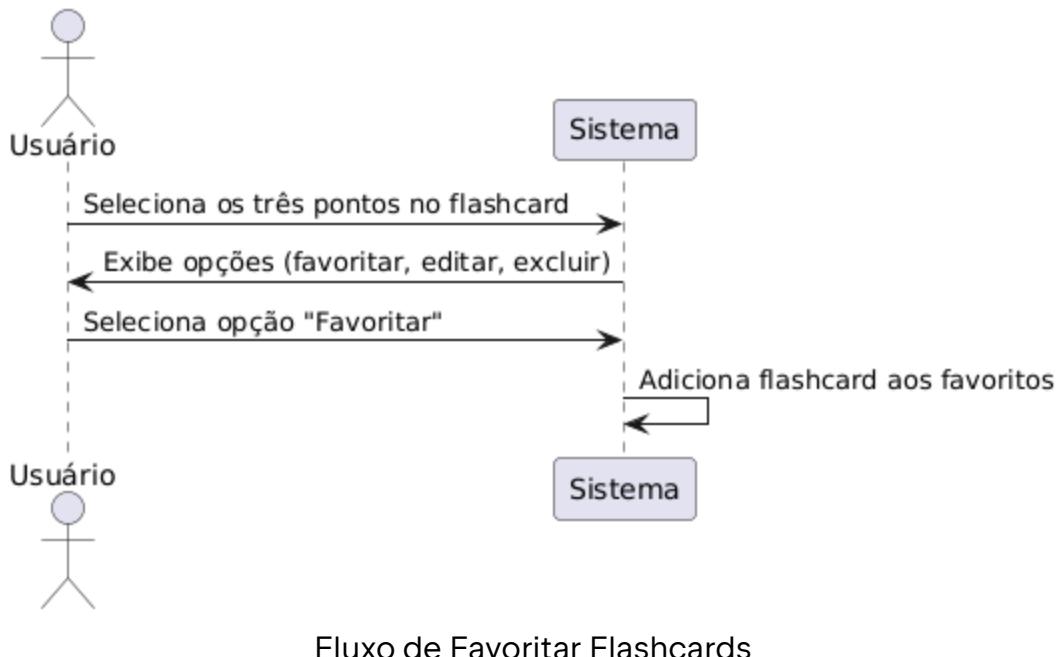
## Diagrama de Sequencia : UC-08

### Desfavoritar Flashcard dos Favoritos



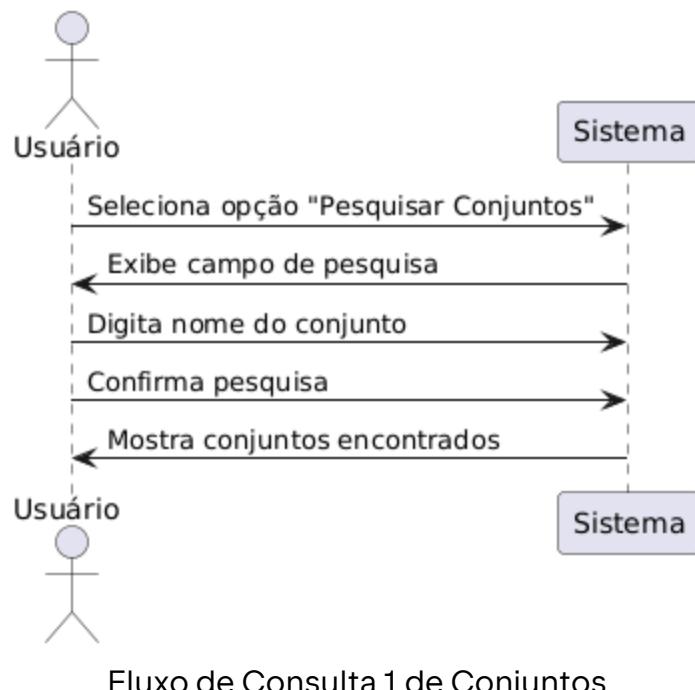
Fluxo de Desfavoritar Flashcards

### Favoritar Flashcard dos Favoritos

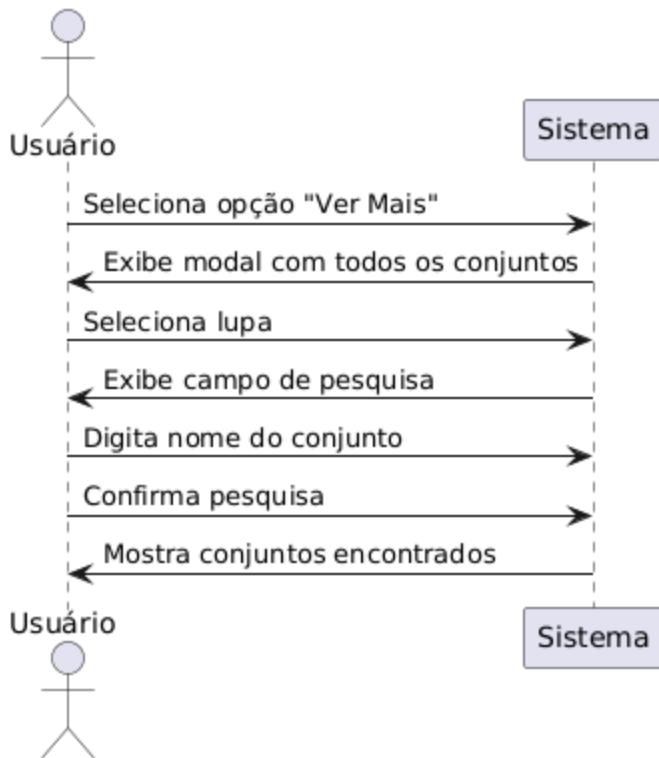


## Diagrama de Sequencia : UC-09

### 1 Tipo de Consulta dos Conjuntos

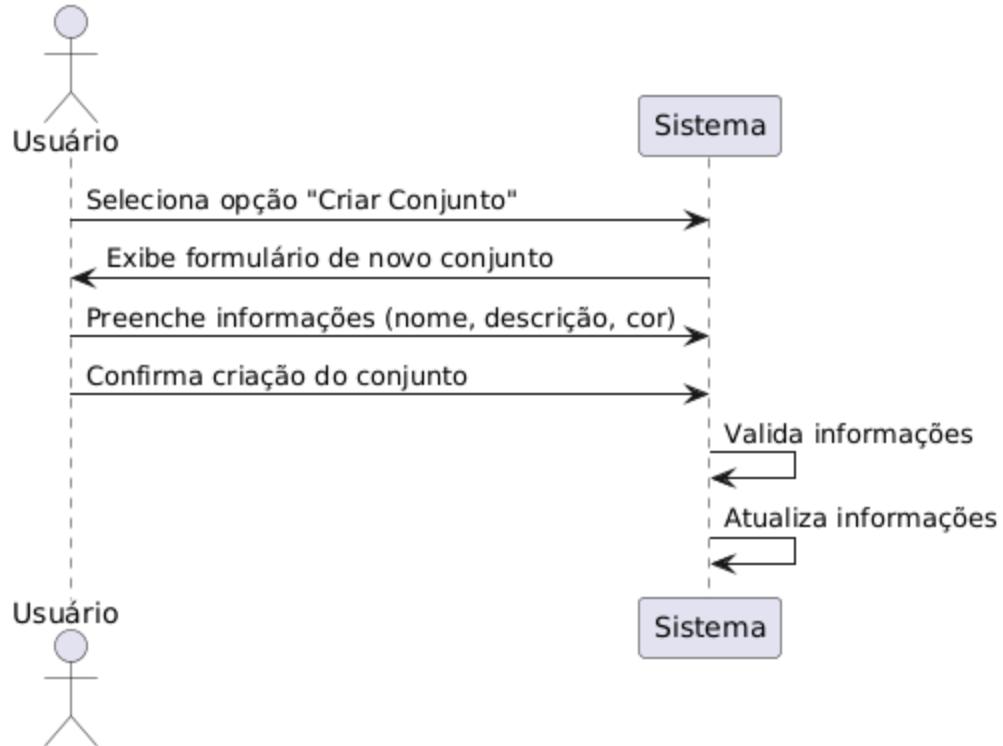


### 2 Tipo de Consulta dos Conjuntos



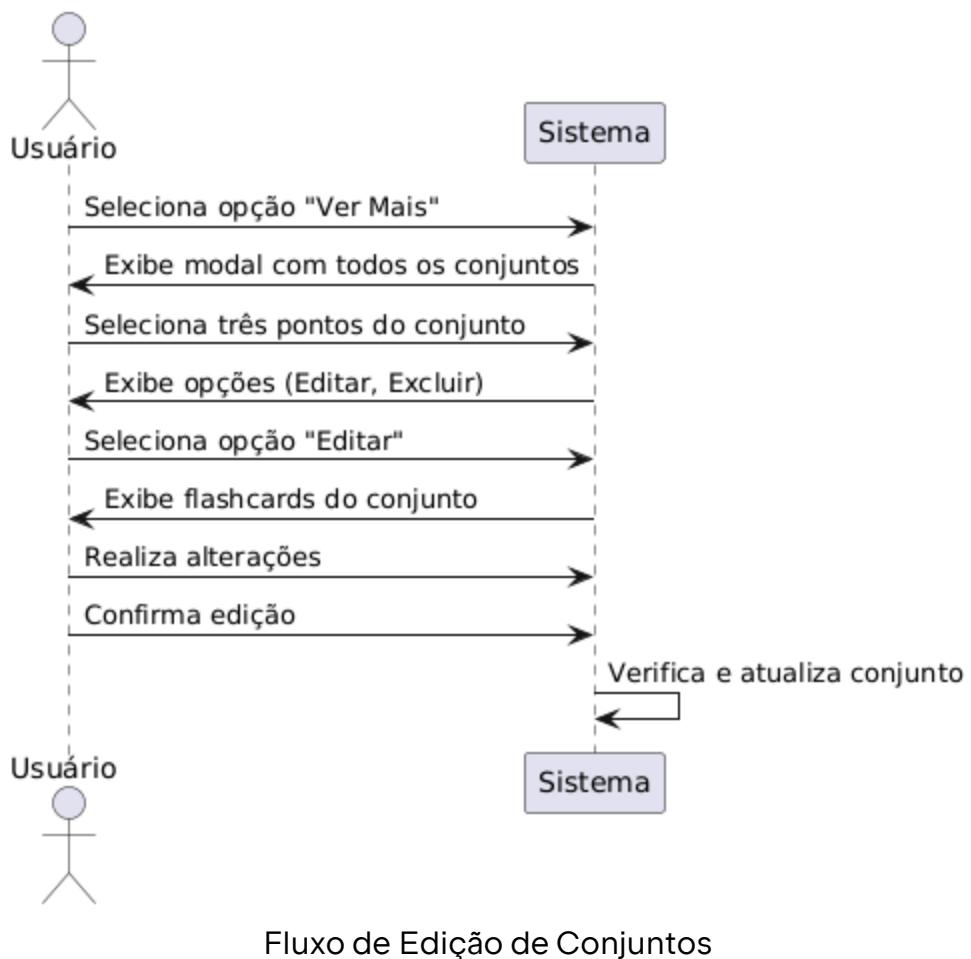
Fluxo de Consulta 2 de Conjuntos

## Criação dos Conjuntos

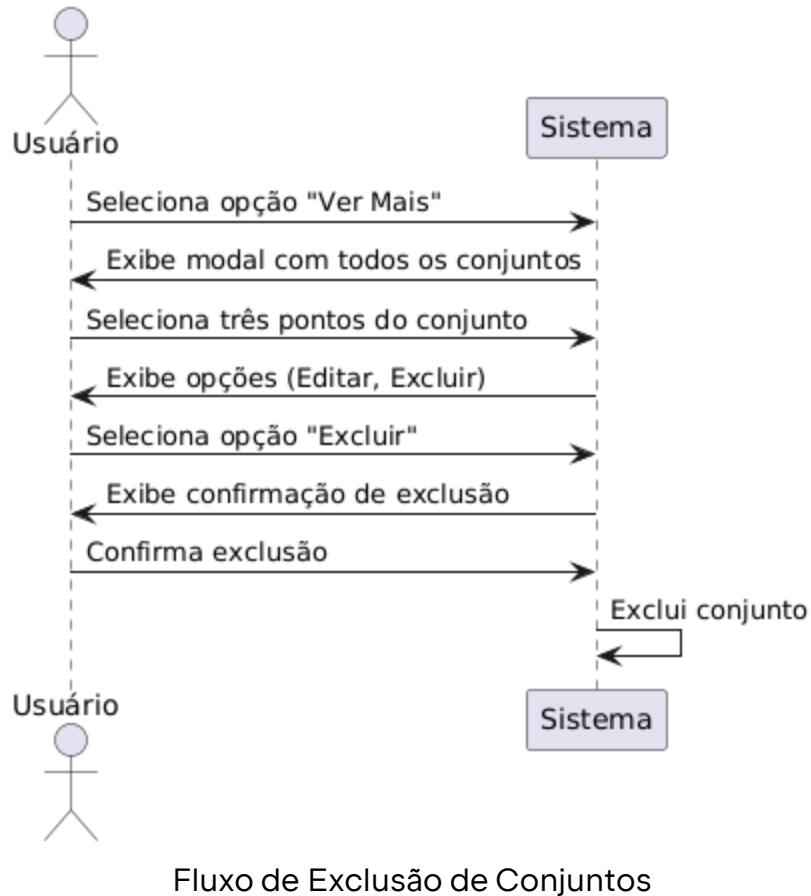


Fluxo de Criação de Conjuntos

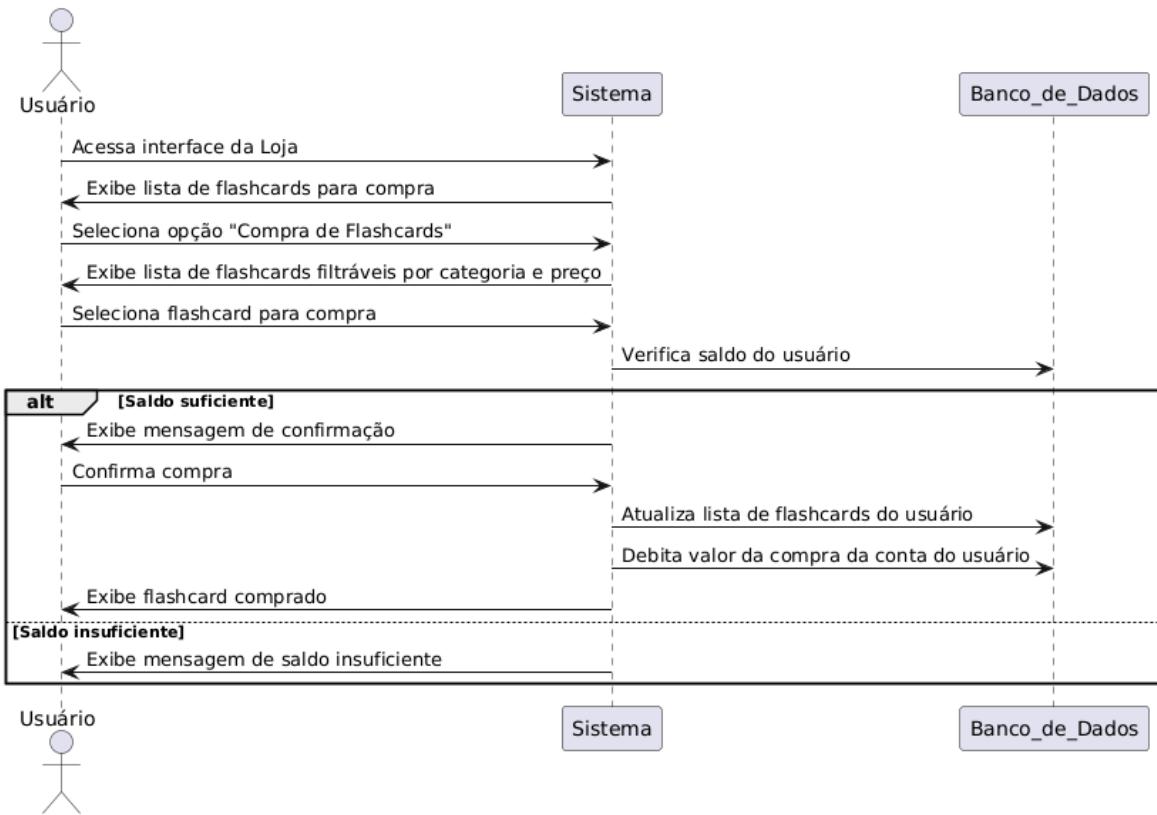
## Edição dos Conjuntos



## Exclusão dos Conjuntos



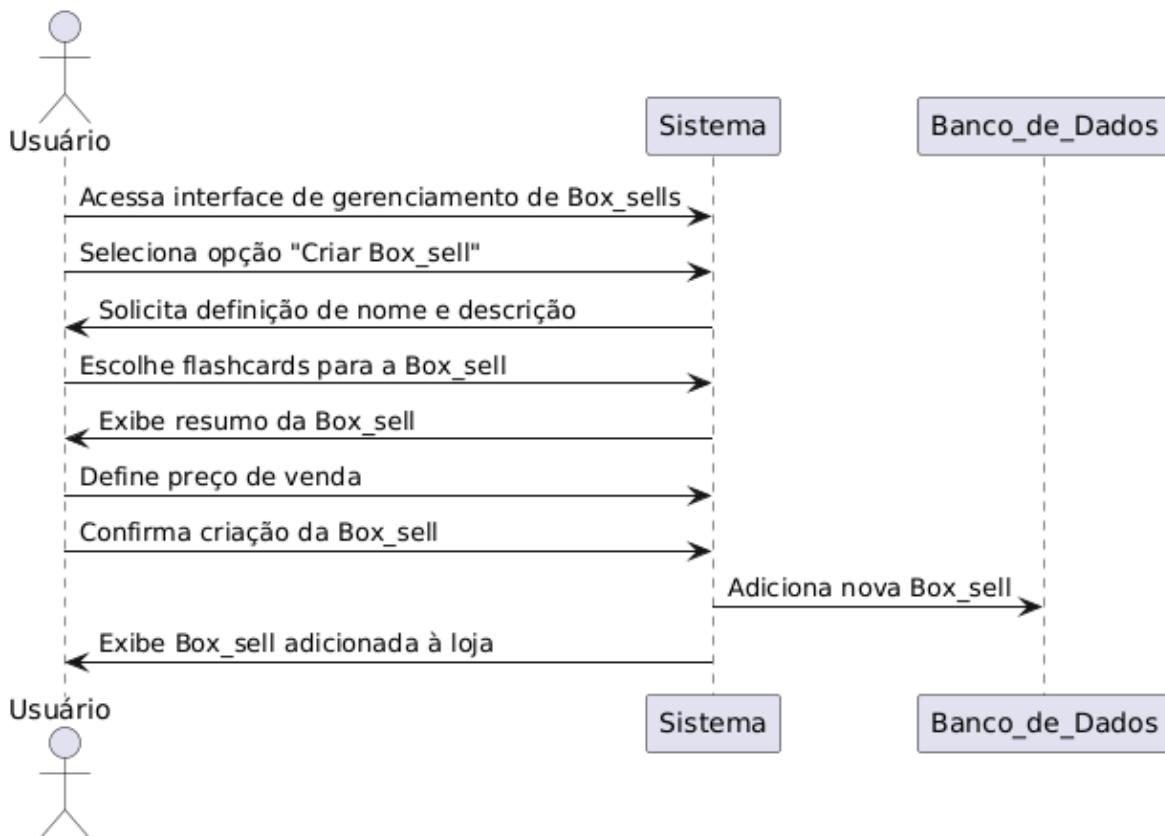
## Diagrama de Sequencia : UC-10



Fluxo de Compra dos Flashcards

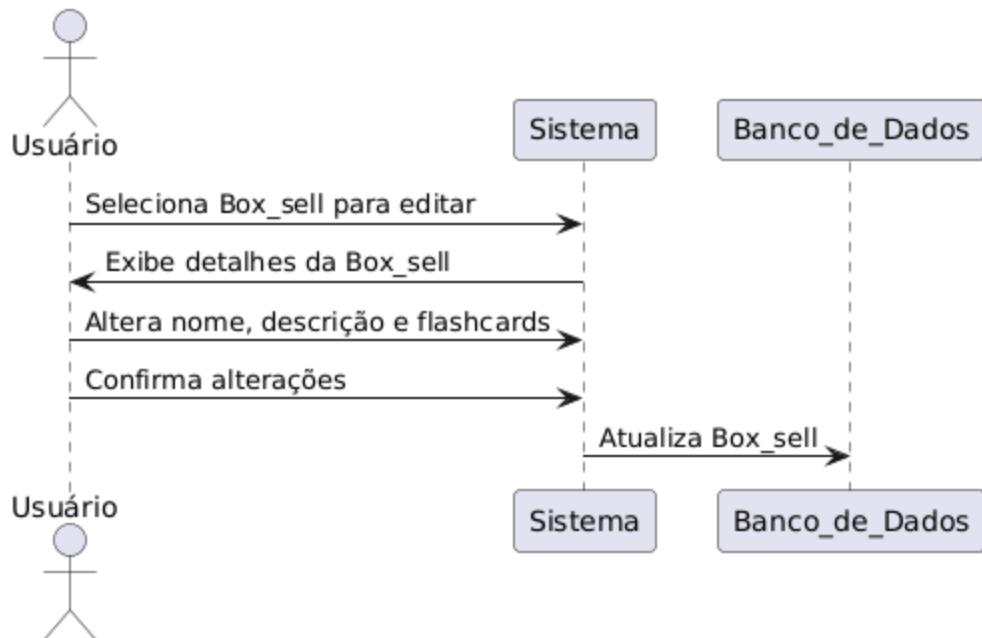
## Diagrama de Sequencia : UC-11

### Criação das Box Sells



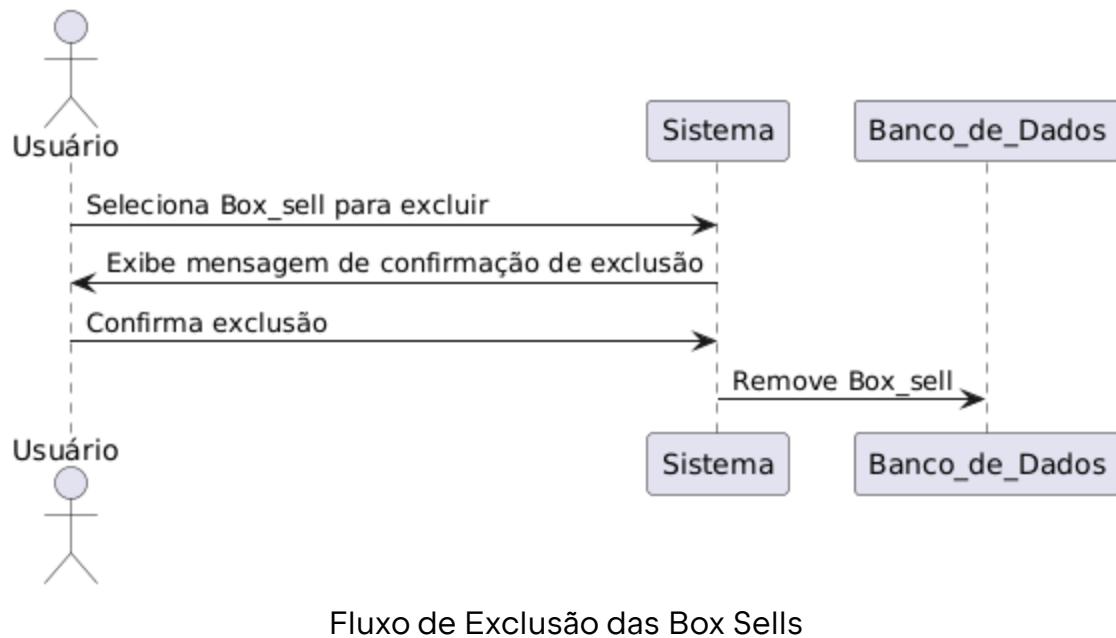
Fluxo de Criação das Box Sells

## Edição das Box Sells



Fluxo de Edição das Box Sells

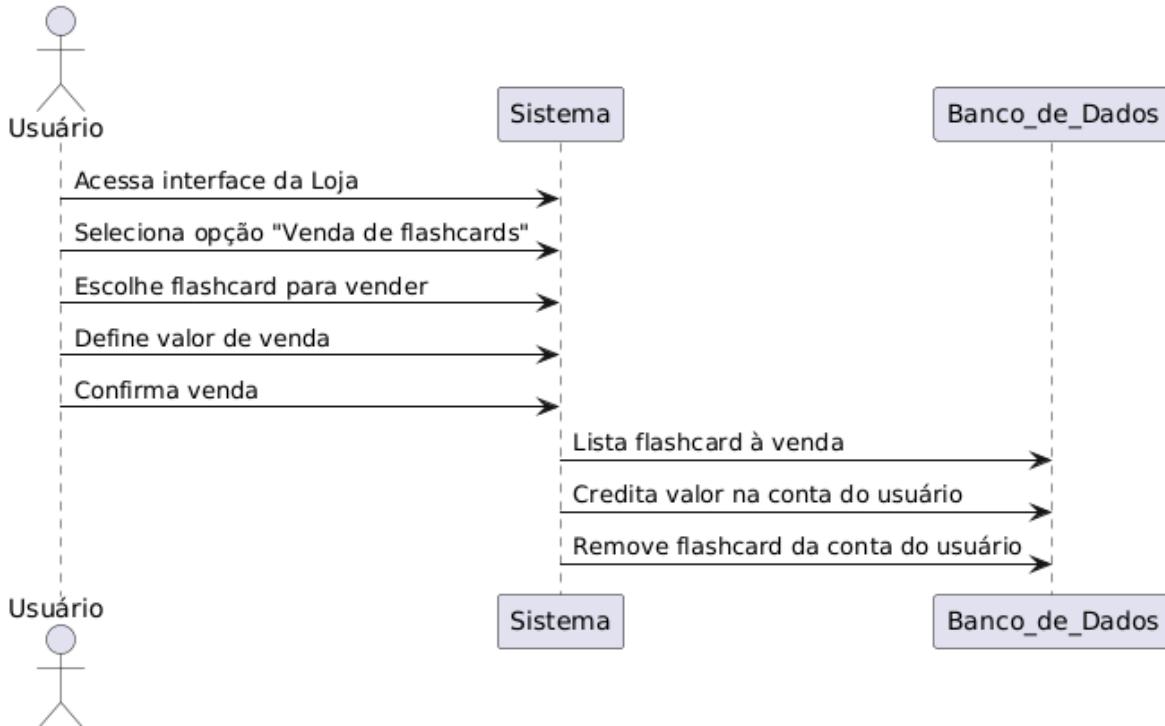
## Exclusão das Box Sells



Fluxo de Exclusão das Box Sells

## Diagrama de Sequencia : UC-12

### Venda das Box Sells



Fluxo de Venda dos Flashcards

# Tecnologias Utilizadas

## Linguagens e Tecnologias

**JavaScript:** Utilizado para a construção da interface web interativa do sistema, incluindo funcionalidades do lado do cliente.

**Java:** Empregado no desenvolvimento do backend da aplicação, utilizando o framework Spring Boot para a construção dos serviços e APIs.

**CSS:** Responsável pela estilização visual dos componentes e layout da interface do usuário.

**HTML:** Usado para a estruturação semântica das páginas web que compõem a aplicação.

**Spring Boot:** Framework Java utilizado para a construção rápida e eficiente do backend, fornecendo recursos de injeção de dependência, segurança, persistência de dados, entre outros.

**JWT (JSON Web Tokens):** Utilizado para a implementação de autenticação e autorização no sistema, garantindo a segurança das interações entre cliente e servidor.

## Banco de Dados

**MySQL:** Banco de dados relacional escolhido para armazenar e persistir os dados da aplicação de forma estruturada.

## Controle de Versão

**Git:** Sistema de controle de versão adotado para o gerenciamento do código-fonte do projeto.

**GitLab:** Plataforma de hospedagem e colaboração de projetos Git utilizada para o versionamento do código.

## IDEs

**IntelliJ IDEA:** Ambiente de desenvolvimento integrado (IDE) escolhido para a codificação e debug do backend em Java.

**Visual Studio Code:** IDE utilizada no desenvolvimento do frontend, provendo

ferramentas e extensões para trabalhar com JavaScript, HTML e CSS.

## Prototipagem

**Figma:** Ferramenta de design e prototipagem utilizada para criar os mockups e designs da interface do usuário.

## Gerenciamento de Projeto

**Trello:** Plataforma de gerenciamento de projetos e tarefas empregada para organizar e acompanhar o andamento do desenvolvimento.

## Documentação

**WriterSide:** Ferramenta selecionada para a criação e estruturação da documentação técnica do projeto, visando a facilitar a compreensão e manutenção do sistema.

Essa combinação de tecnologias, ferramentas e práticas de engenharia de software nos permitirá construir uma aplicação web robusta, escalável e fácil de manter ao longo do tempo. w

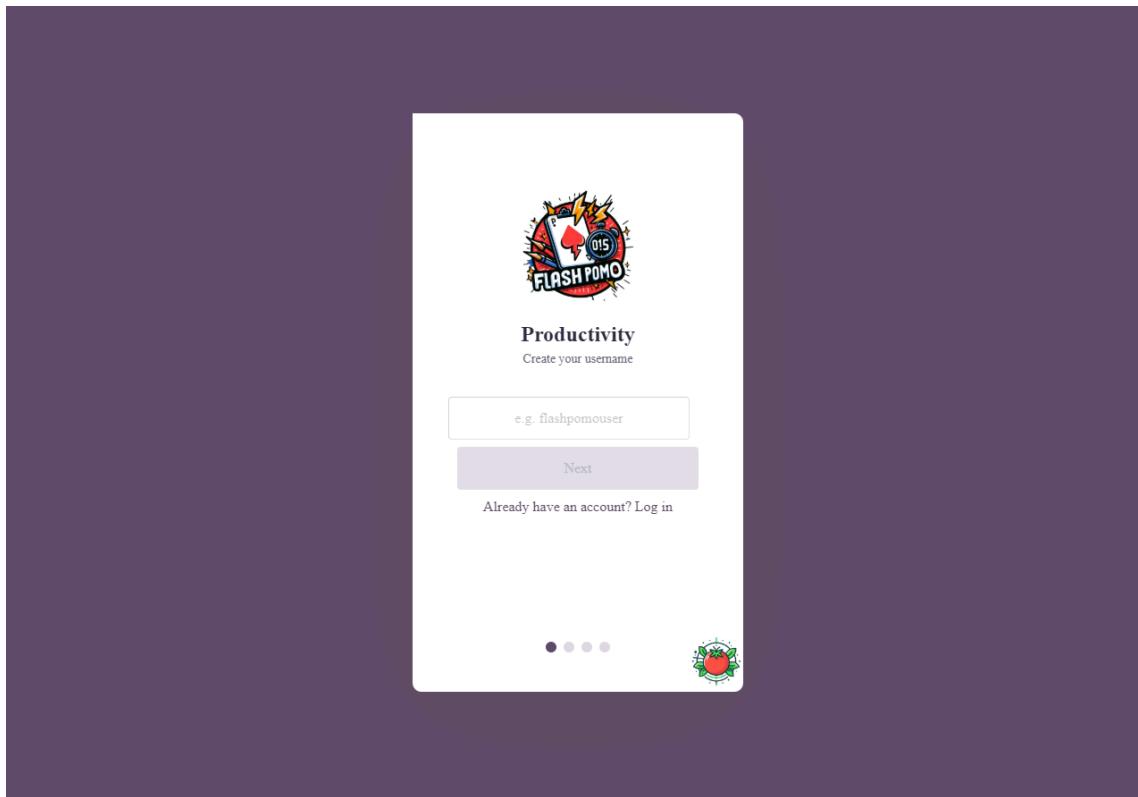
# Manual do Usuário

O manual do usuário tem como objetivo orientar o usuário final do sistema a utilizá-lo de forma adequada e eficiente, contendo informações sobre as funcionalidades, recursos, instruções de uso, solução de problemas e demais informações relevantes ao usuário.

## Cadastro de Usuários e Login

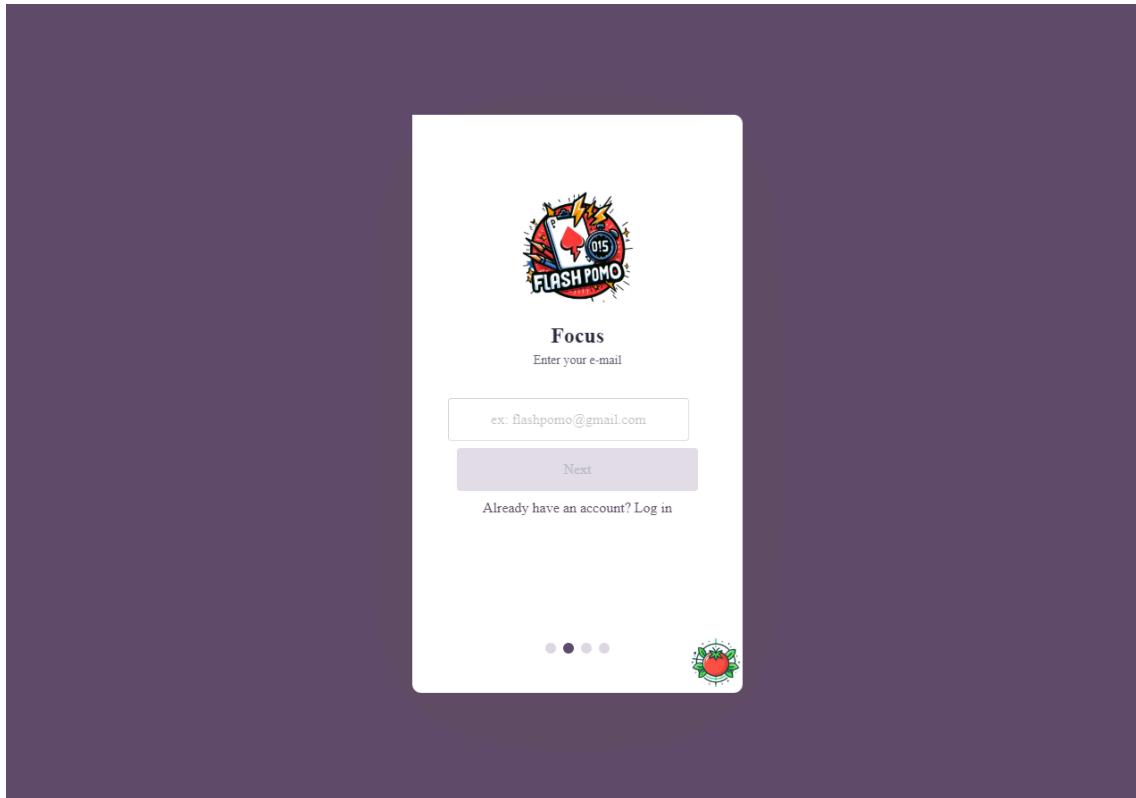
Para realizar o cadastro, o usuário deve informar os dados corretamente, preenchendo os campos necessários, cada campo devidamente preenchido deve ser confirmado pressionando o botão de continuar.

1. Primeiro o usuário criará o nome de usuário.



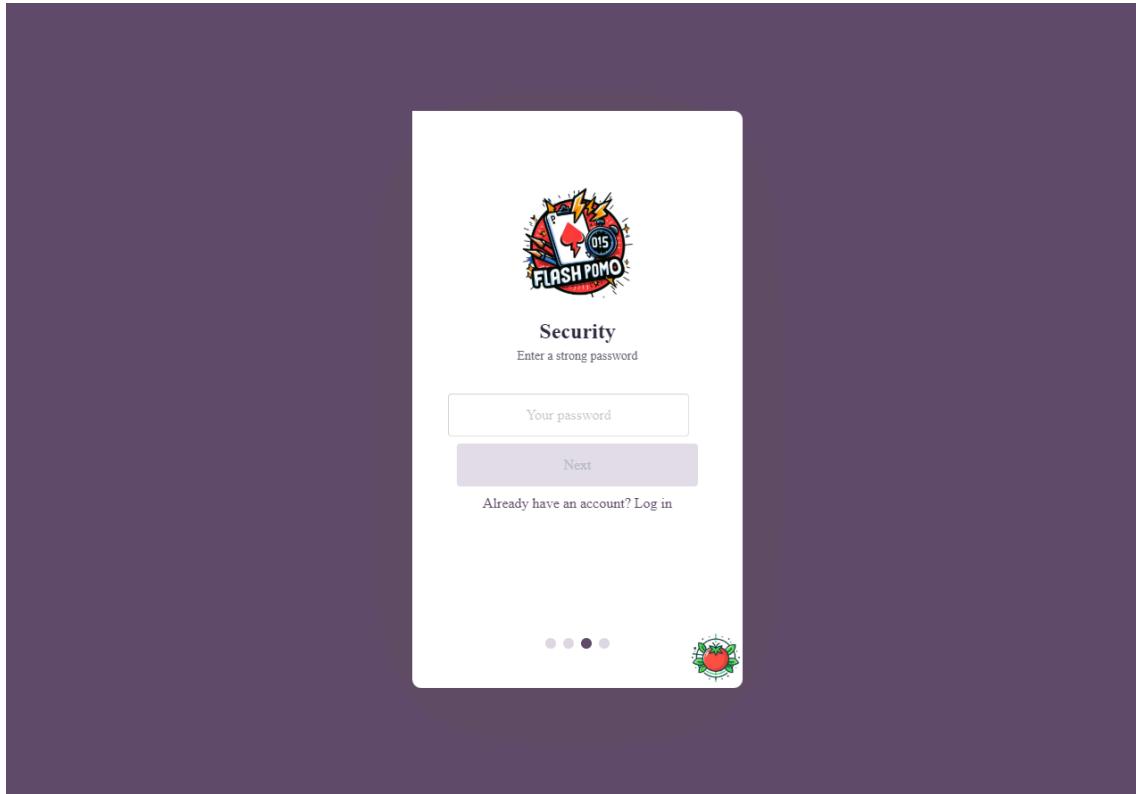
Cadastro nome de usuário

2. Em seguida o usuário informará o email.



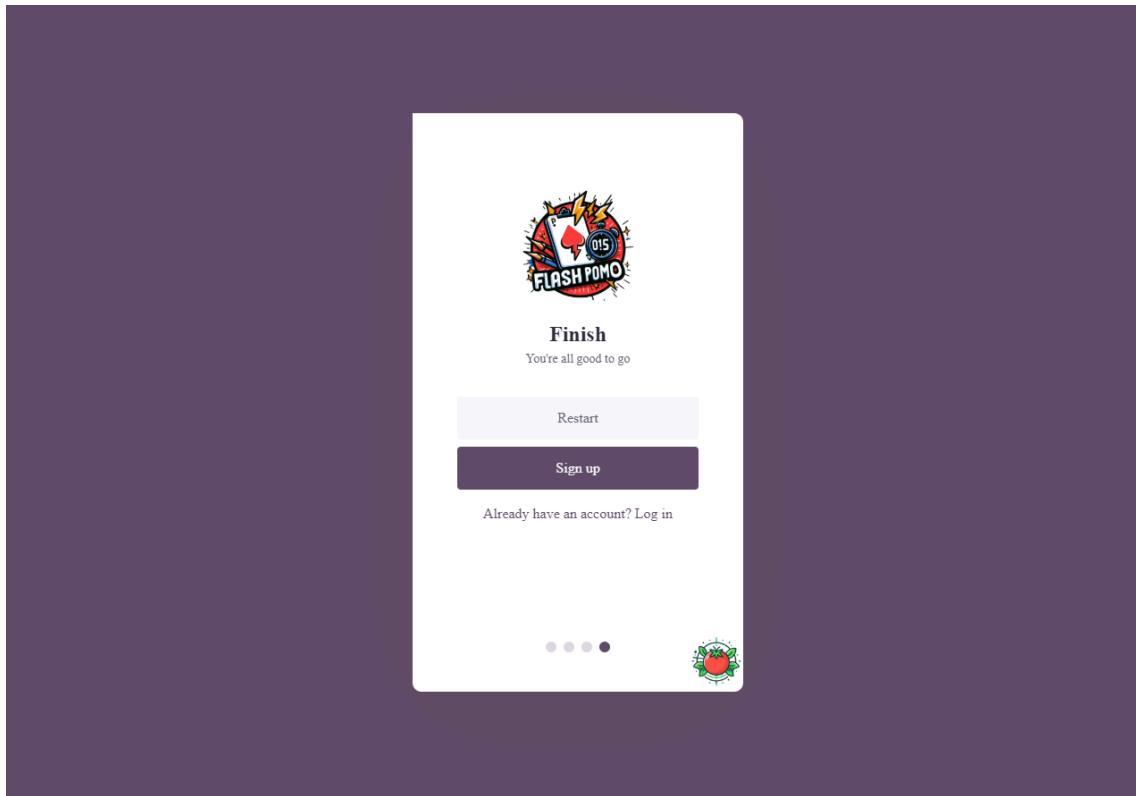
Cadastro email

3. O usuário criará sua senha.



Cadastro senha de usuário

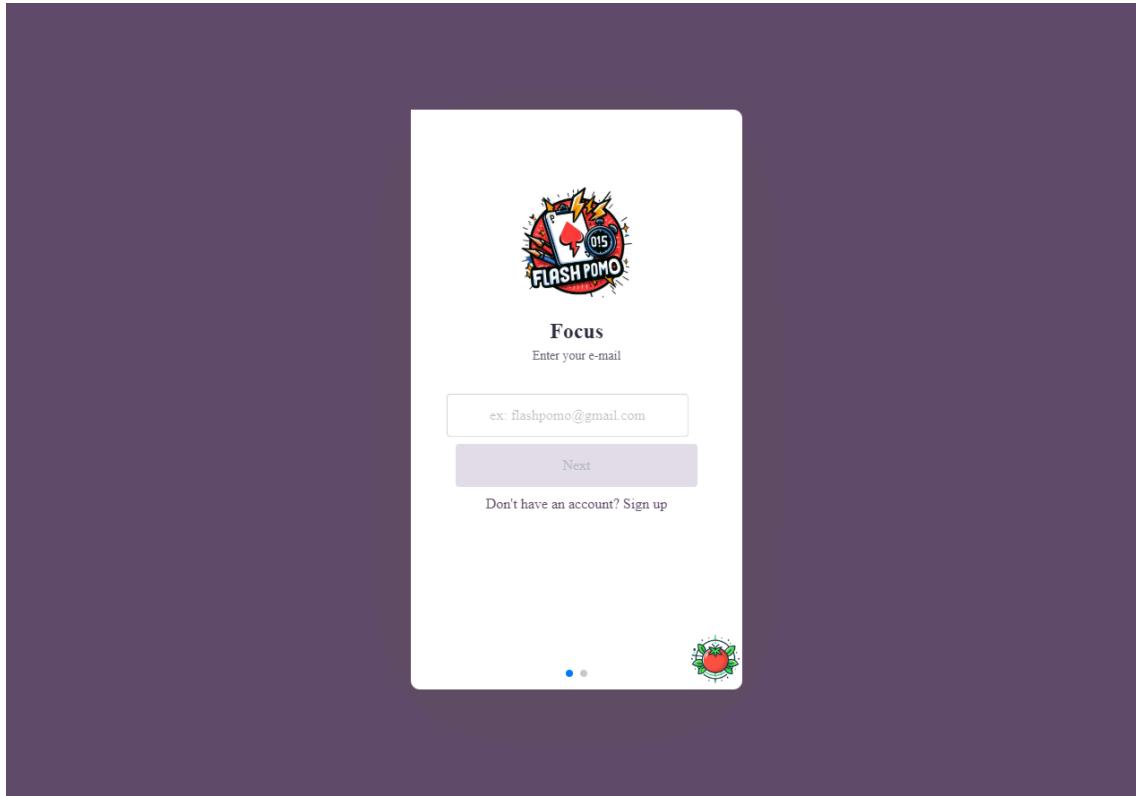
4. Após informar os devidos dados, o usuário confirma a criação de conta.



Signup4

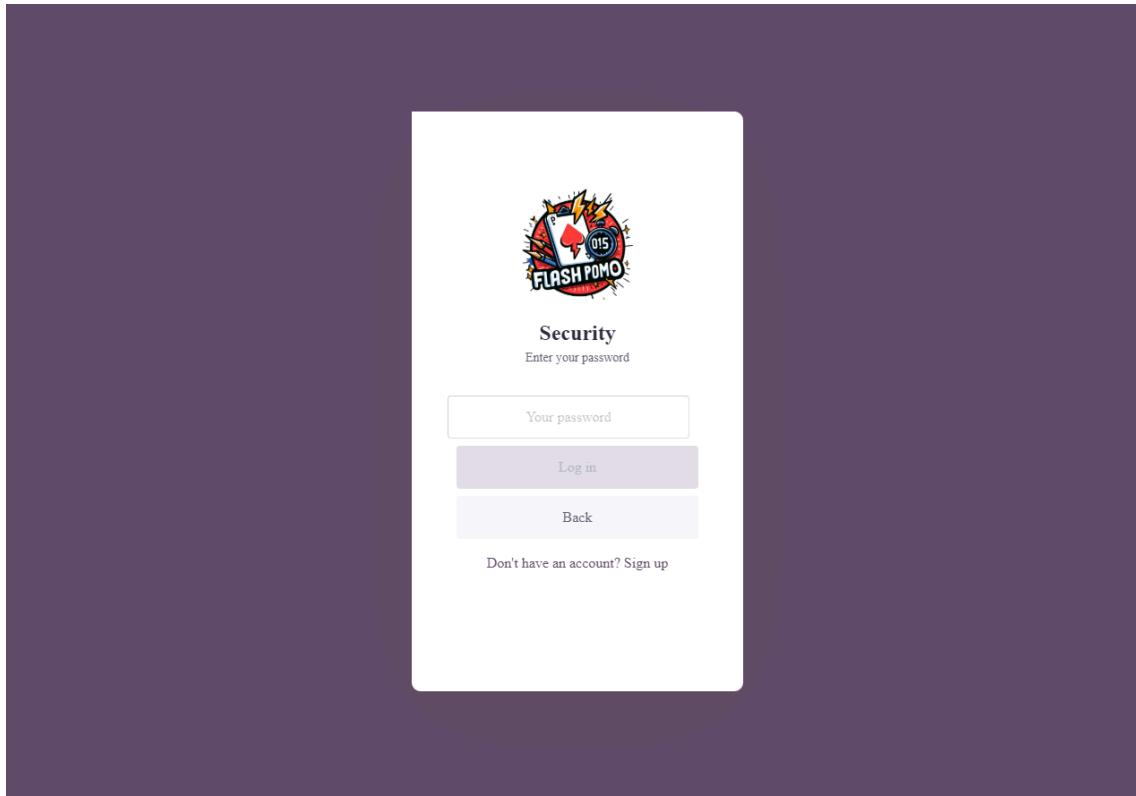
**Pronto, a conta do usuário foi criada! Agora o usuário pode fazer log in.**

5. O usuário na página de log in, informa o email.



Login1

6. O usuário digita sua senha e confirma o log in

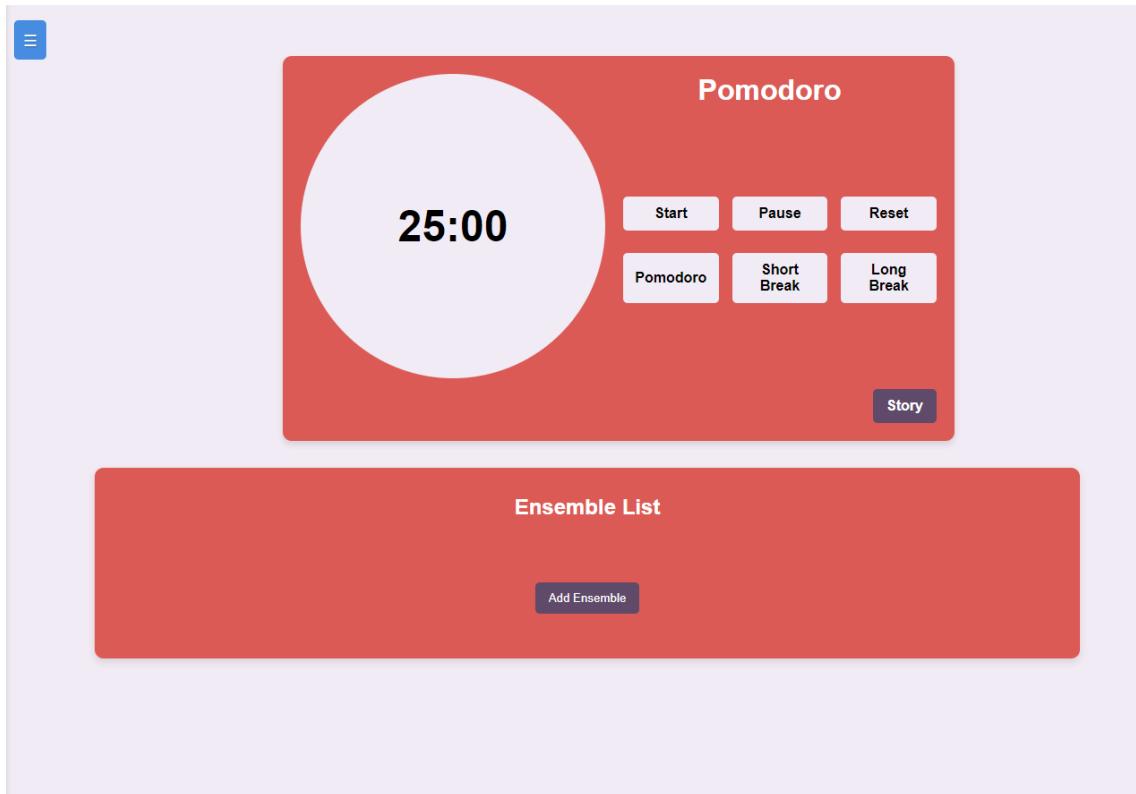


Login2

## Criar FlashCard

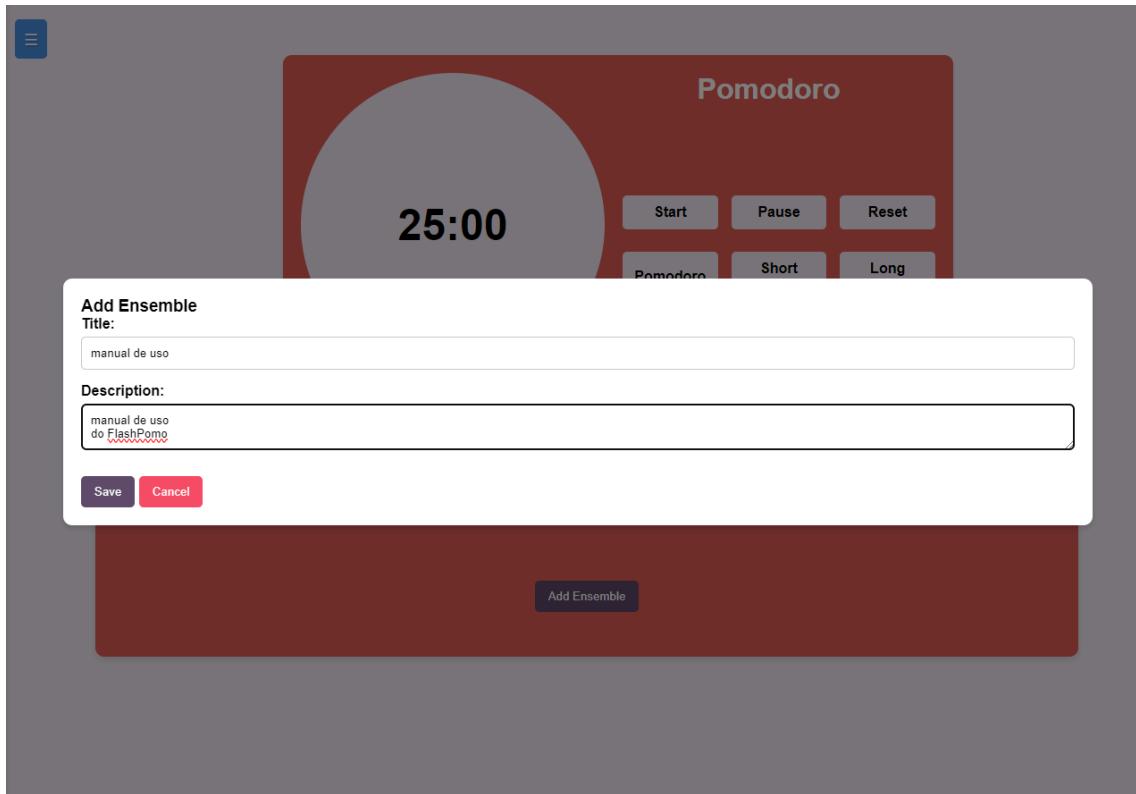
Na página de Dashboard o usuário precisará criar primeiramente o ensemble que será o conjunto presente dos flashcards, para tal selecionará o botão add ensemble, o sistema pedirá o título e a descrição do ensemble.

1. Na seção Ensemble List.



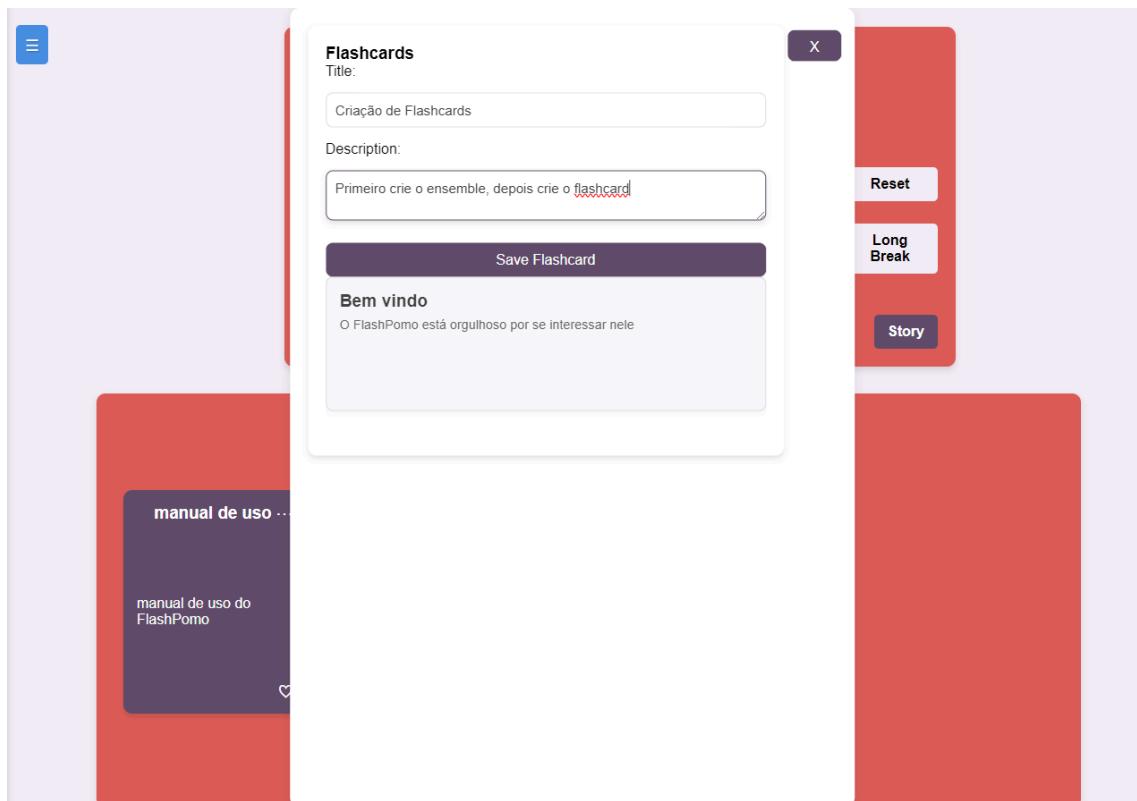
Dashboard1

2. Crie o ensemble, adicione o título e descrição, e salve.



Add ensemble

3. Pressione no ensemble e crie os flashcards, o título e descrição, depois é só salvar

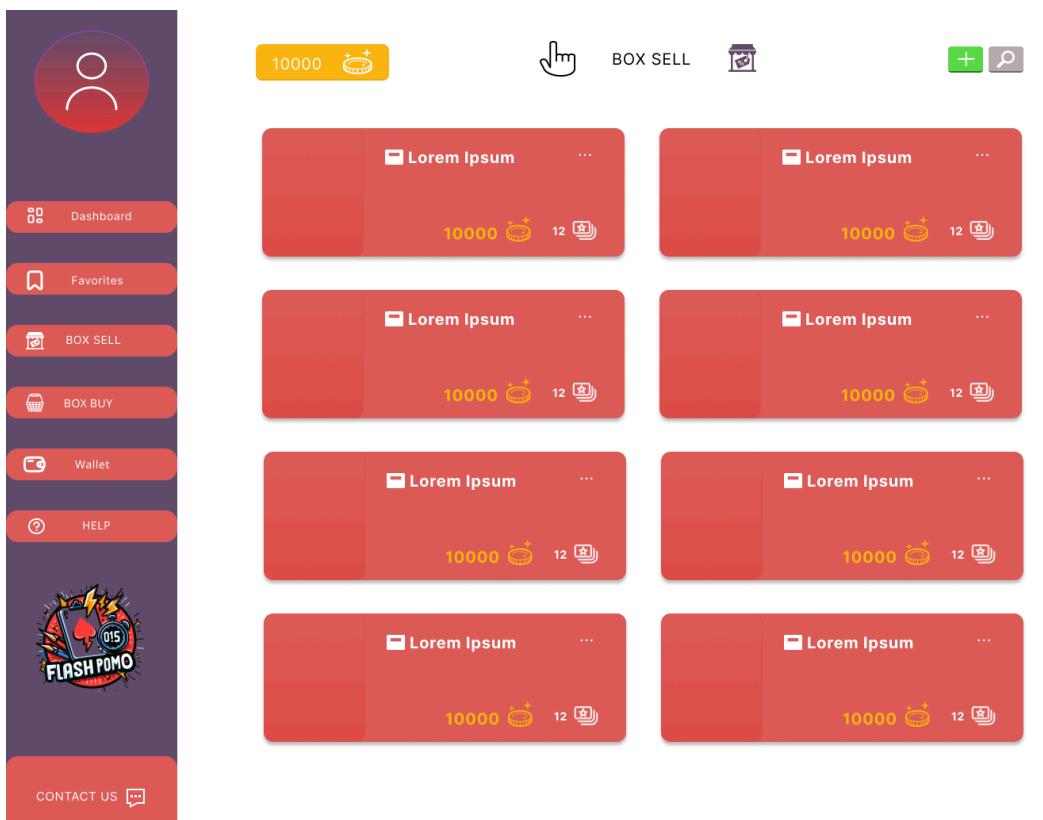


Add flashcard

## Criar Box e Vender

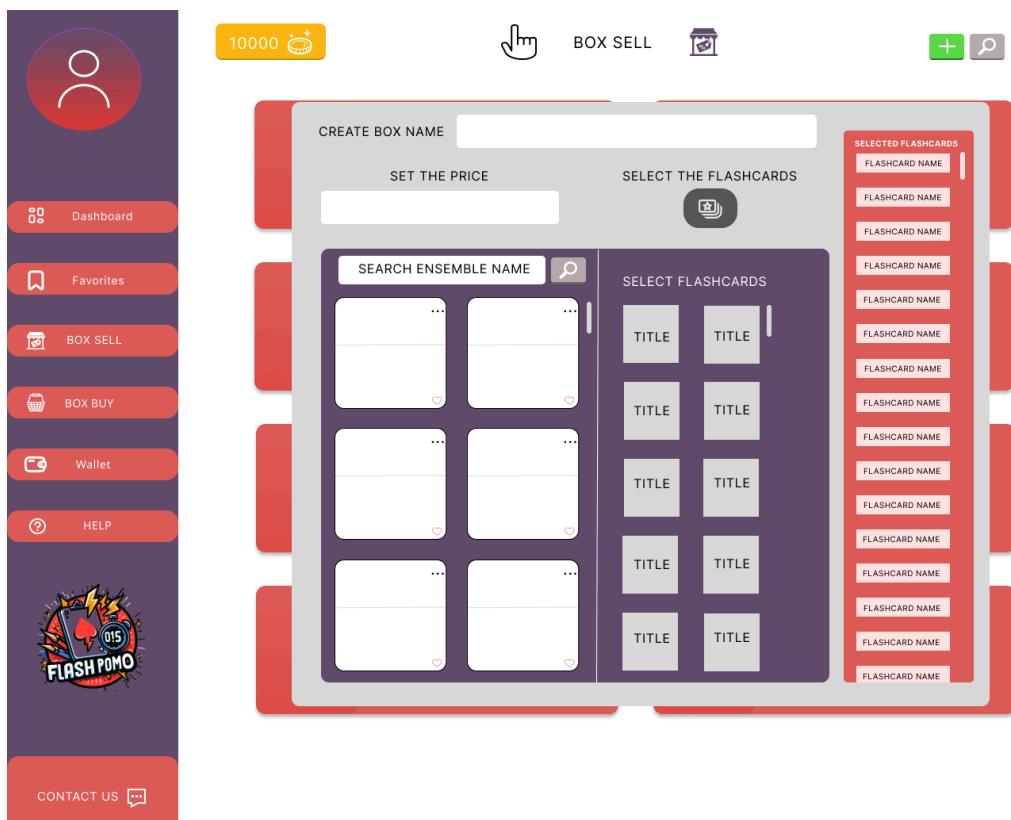
Para vender uma box, primeiro o usuário precisa, criá-la.

1. No menu, o usuário entrará na página Box Sell.



Desktop user box shell

2. Nela precisará selecionar o botão de adicionar um "+" ao lado do botão de pesquisa. Em seguida deverá criar a box com os flashcards já existentes e, os quais podem pertencer a vários ensembles, e logo após selecionar o preço.

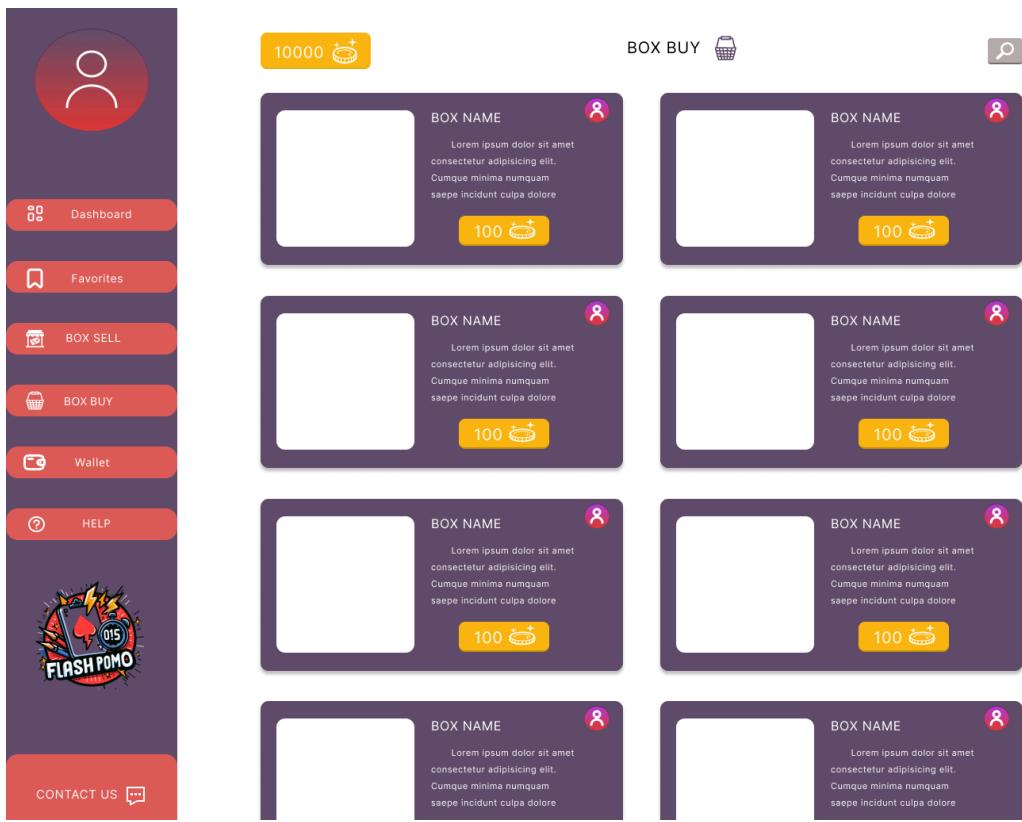


Desktop user creating box shell

## Comprar Box

Para comprar uma box, o usuário deverá entrar na página Box Buy, e pesquisar pela box que ele deseja.

1. O usuário selecionará o botão presente do valor da box, e em seguida confirmará a compra.

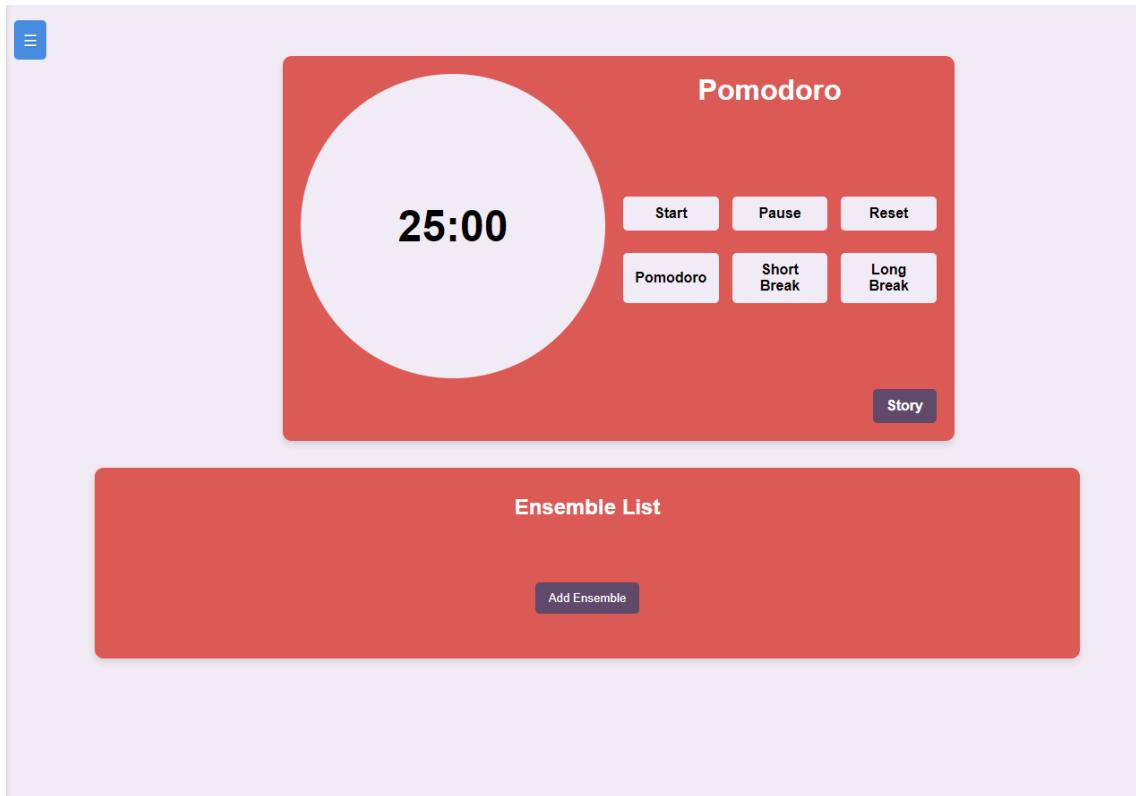


Interface box buy

## Usar Pomodoro

Para o usuário usar o pomodoro, ele deverá primeiro estar no Dashboard.

1. O usuário poderá escolher no pomodoro as funções Pomodoro, de 25min, short break de 5min, e long break de 15min. também poderá usar as funções de start para começar, pause para pausar e restart para recomeçar.



Dashboard1

# Mapa do Site

Start typing here...

# API Reference

Esta seção fornece uma visão geral detalhada dos endpoints da API disponíveis em nosso aplicativo de software. Ela inclui informações sobre os formatos de solicitação e resposta, bem como exemplos e diretrizes de uso.

# Schemas

A seção de Schemas da API é uma parte essencial da documentação, pois ela define os formatos de dados esperados pelos diferentes endpoints da nossa API.

Os schemas descrevem, em detalhes, a estrutura e as regras de validação dos objetos JSON que são utilizados nas solicitações e respostas da API. Essa padronização é fundamental para garantir a consistência e a interoperabilidade entre o cliente (aplicação que consome a API) e o servidor (nossa aplicação que fornece a API).

# Purchase Schemas

## PurchaseRequestDTO

# PomodoroStory Schemas

## PomodoroStoryRequestDTO

# Favorite Schemas

**FavoriteRequestDTO**

# Box Schemas

## BoxRequestDTO

# **GetByIdResponse**

**.GetByIdResponse**

# **ResponseCommonDTO**

**ResponseCommonDTO**

# Flashcard Schemas

**Flashcard**

**FlashcardRequestEditDTO**

**FlashcardRequestDTO**

# Ensemble Schemas

**Ensemble**

**EnsembleRequestDTO**

# User Schemas

**User**

**UserEditRequestDTO**

**ResponseUserEditDTO**

# Auth Schemas

**RegisterRequestDTO**

**LoginRequestDTO**

# **ResponseEntity**

**ResponseType**

# Auth Endpoints

A autenticação é um aspecto crucial de aplicações web, permitindo que os usuários acessem suas contas de forma segura e executem diversas ações. Os endpoints de autenticação mais comuns incluem registro, login, logout e gerenciamento de senha.

O endpoint de registro permite que os usuários criem uma nova conta, fornecendo informações como nome de usuário, email e senha. Após um registro bem-sucedido, o usuário normalmente recebe um token de autenticação, que pode ser usado em solicitações autenticadas posteriores.

O endpoint de login é usado para autenticar os usuários e obter um token de autenticação, que é então usado para autorizar as ações do usuário. Esse token é tipicamente um JSON Web Token (JWT) que contém as informações do usuário e é assinado pelo servidor.

# POST auth/login

POST /auth/login

## Request parameters

### Body

**application/json** LoginRequestDTO **required**

Child attributes

**email** object

Child attributes

**password** object

Child attributes

### JSON example

```
{  
  "email": {},  
  "password": {}  
}
```

## Responses

**200 OK**

Content type **\*/\***

OK

**\*/\* ResponseEntity**

Child attributes

**headers** `HttpHeaders` **required**

Child attributes

# POST auth/register

POST /auth/register

## Request parameters

### Body

**application/json** RegisterRequestDTO required

Child attributes

**name** object

Child attributes

**email** object

Child attributes

**password** object

Child attributes

**avatar** object

Child attributes

**createdAt** object

Child attributes

**status** object

Child attributes

**role** object

Child attributes

## JSON example

```
{  
  "name": {},  
  "email": {},  
  "password": {},  
  "avatar": {},  
  "createdAt": {},  
  "status": {},  
  "role": {}  
}
```

## Responses

200 OK

Content type    \*/\*

OK

  \*/ ResponseEntity

Child attributes

**headers** HttpHeaders **required**

Child attributes

# User Endpoints

Endpoints para User

# GET user

GET /user

## Responses

200 OK

Content type \*/\*

OK

\*/\* object

Child attributes

# GET user/{userId}

GET /user/{userId}

## Request parameters

### Path

**userId** object **required**

Child attributes

## Responses

200 OK

Content type \*/\*

OK

\*/\* GetByIdResponse

Child attributes

**name** object

Child attributes

**email** object

Child attributes

# PUT user/{userId}

**PUT** /user/{userId}

## Request parameters

### Path

**userId** object **required**

Child attributes

### Body

**application/json** UserEditRequestDTO **required**

Child attributes

**name** object

Child attributes

**email** object

Child attributes

**avatar** object

Child attributes

**password** object

Child attributes

**role** object

Child attributes

**updatedAt** object

Child attributes

## JSON example

```
{  
  "name": {},  
  "email": {},  
  "avatar": {},  
  "password": {},  
  "role": {},  
  "updatedAt": {}  
}
```

## Responses

200 OK

Content type    \*/\*

OK

`/* ResponseUserEditDTO`

    Child attributes

**name** object

        Child attributes

**email** object

        Child attributes

**password** object

        Child attributes

**avatar** object

Child attributes

**role** object

Child attributes

# DELETE user/{userId}

**DELETE** /user/{userId}

## Request parameters

### Path

**userId** object **required**

Child attributes

## Responses

**200 OK**

Content type `*/*`

OK

`*/* ResponseCommonDTO`

Child attributes

**message** object

Child attributes

# Ensemble Endpoints

Endpoints para Ensemble.

# GET ensemble/{ensembleId}

**GET** /ensemble/{ensembleId}

## Request parameters

### Path

**ensembleId** object **required**

Child attributes

## Responses

**200 OK**

**Content type** \*/\*

OK

\*/\* **ResponseEntity**

Child attributes

**headers** HttpHeaders **required**

Child attributes

# PUT ensemble/{ensembleId}

**PUT** /ensemble/{ensembleId}

## Request parameters

### Path

**ensembleId** object **required**

Child attributes

### Body

**application/json** EnsembleRequestDTO **required**

Child attributes

**name** object

Child attributes

**description** object

Child attributes

**userId** object

Child attributes

### JSON example

```
{  
  "name": {},  
  "description": {},  
  "userId": {}  
}
```

```
"userId": {}  
}
```

## Responses

200 OK

Content type \*/\*

OK

\*/\* ResponseEntity

Child attributes

**headers** HttpHeaders required

Child attributes

# DELETE ensemble/{ensembleId}

**DELETE** /ensemble/{ensembleId}

## Request parameters

### Path

**ensembleId** object **required**

Child attributes

## Responses

**200 OK**

Content type `*/*`

OK

`*/*` `ResponseEntity`

Child attributes

**headers** `HttpHeaders` **required**

Child attributes

# GET ensemble/

GET

/ensemble/

## Responses

200 OK

Content type    \*/\*

OK

\*/\* object

Child attributes

# POST ensemble/

POST /ensemble/

## Request parameters

### Body

**application/json** EnsembleRequestDTO required

Child attributes

**name** object

Child attributes

**description** object

Child attributes

**userId** object

Child attributes

### JSON example

```
{  
  "name": {},  
  "description": {},  
  "userId": {}  
}
```

## Responses

200 OK

**Content type** \*/\*

OK

\*/\* **ResponseType**

Child attributes

**headers** `HttpHeaders` **required**

Child attributes

# Flashcard Endpoints

Endpoints para Flashcard

# GET flashcard/{flashcardId}

**GET** /flashcard/{flashcardId}

## Request parameters

### Path

**flashcardId** object **required**

Child attributes

## Responses

**200 OK**

Content type `*/*`

OK

`*/*` **ResponseType**

Child attributes

**headers** HttpHeaders **required**

Child attributes

# PUT flashcard/{flashcardId}

**PUT** /flashcard/{flashcardId}

## Request parameters

### Path

**flashcardId** object **required**

Child attributes

### Body

**application/json** FlashcardRequestEditDTO **required**

Child attributes

**title** object

Child attributes

**content** object

Child attributes

**createdAt** object

Child attributes

**updatedAt** object

Child attributes

**ensembleId** object

Child attributes

**status** object

Child attributes

## JSON example

```
{  
  "title": {},  
  "content": {},  
  "createdAt": {},  
  "updatedAt": {},  
  "ensembleId": {},  
  "status": {}  
}
```

## Responses

200 OK

Content type    \*//\*

OK

\*//\* ResponseEntity

Child attributes

**headers** HttpHeaders [required](#)

Child attributes

# DELETE flashcard/{flashcardId}

**DELETE** /flashcard/{flashcardId}

## Request parameters

### Path

**flashcardId** object **required**

Child attributes

## Responses

**200 OK**

Content type `*/*`

OK

`*/*` **ResponseEntity**

Child attributes

**headers** `HttpHeaders` **required**

Child attributes

# GET flashcard/

GET /flashcard/

## Responses

200 OK

Content type \*/\*

OK

\*/\* object

Child attributes

# POST flashcard/

POST /flashcard/

## Request parameters

### Body

**application/json** FlashcardRequestDTO required

Child attributes

**title** object

Child attributes

**content** object

Child attributes

**createdAt** object

Child attributes

**updatedAt** object

Child attributes

**ensembleId** object

Child attributes

### JSON example

```
{  
  "title": {},  
  "content": {},  
  "createdAt": {},  
  "updatedAt": {},  
  "ensembleId": {}  
}
```

```
"createdAt": {},  
"updatedAt": {},  
"ensembleId": {}  
}
```

## Responses

200 OK

Content type    \*/\*

OK

  \*/ ResponseEntity

  Child attributes

**headers** HttpHeaders **required**

    Child attributes

# GET flashcard/ensemble/{ensemble}

GET /flashcard/ensemble/{ensemble}

## Request parameters

### Path

**ensemble** object required

Child attributes

## Responses

200 OK

Content type \*/\*

OK

\*//\* object

Child attributes

# PomodoroStory Endpoints

Endpoints para PomodoroStory

# GET pomodoro-story/{pomodoroStoryId}

GET /pomodoro-story/{pomodoroStoryId}

## Request parameters

### Path

**pomodoroStoryId** object required

Child attributes

## Responses

200 OK

Content type \*/\*

OK

\*/ ResponseEntity

Child attributes

**headers** HttpHeaders required

Child attributes

# PUT pomodoro-story/{pomodoroStoryId}

**PUT** /pomodoro-story/{pomodoroStoryId}

## Request parameters

### Path

**pomodoroStoryId** object **required**

Child attributes

### Body

**application/json** PomodoroStoryRequestDTO **required**

Child attributes

**pomodoroStoryId** object

Child attributes

**startDateTimestamp** object

Child attributes

**endDateTimestamp** object

Child attributes

**userId** object

Child attributes

## JSON example

```
{  
  "pomodoroStoryId": {},  
  "startDateTimestamp": {},  
  "endDateTimestamp": {},  
  "userId": {}  
}
```

## Responses

200 OK

Content type    \*/\*

OK

  \*/\* ResponseEntity

  Child attributes

**headers** HttpHeaders **required**

    Child attributes

# POST pomodoro-story/{pomodoroStoryId}

POST /pomodoro-story/{pomodoroStoryId}

## Request parameters

### Path

**pomodoroStoryId** object **required**

Child attributes

### Body

**application/json** PomodoroStoryRequestDTO **required**

Child attributes

**pomodoroStoryId** object

Child attributes

**startDateTimestamp** object

Child attributes

**endDateTimestamp** object

Child attributes

**userId** object

Child attributes

## JSON example

```
{  
  "pomodoroStoryId": {},  
  "startDateTimestamp": {},  
  "endDateTimestamp": {},  
  "userId": {}  
}
```

## Responses

200 OK

Content type    \*/\*

OK

  \*/\* ResponseEntity

  Child attributes

**headers** HttpHeaders **required**

    Child attributes

# GET pomodoro-story/

GET /pomodoro-story/

## Responses

200 OK

Content type \*/\*

OK

\*/\* ResponseEntity

Child attributes

**headers** HttpHeaders **required**

Child attributes

# Favorite Endpoints

Endpoints para Favorite

# GET favorite/{favoriteld}

**GET** /favorite/{favoriteld}

## Request parameters

### Path

**favoriteld** object **required**

Child attributes

## Responses

**200 OK**

**Content type** \*/\*

OK

\*/\* **ResponseEntity**

Child attributes

**headers** HttpHeaders **required**

Child attributes

# PUT favorite/{favoriteld}

**PUT** /favorite/{favoriteld}

## Request parameters

### Path

**favoriteld** object **required**

Child attributes

### Body

**application/json** FavoriteRequestDTO **required**

Child attributes

**farvoriteld** object

Child attributes

**favoritationDateTimestamp** object

Child attributes

**status** object

Child attributes

**userId** object

Child attributes

## JSON example

```
{  
    "farvoriteId": {},  
    "favoritatemDateTimestamp": {},  
    "status": {},  
    "userId": {}  
}
```

## Responses

200 OK

Content type    \*/\*

OK

\*/\* ResponseEntity

Child attributes

**headers** HttpHeaders [required](#)

Child attributes

# DELETE favorite/{favoriteId}

**DELETE** /favorite/{favoriteId}

## Request parameters

### Path

**favoriteId** object **required**

Child attributes

## Responses

**200 OK**

**Content type** \*/\*

OK

\*/\* **ResponseEntity**

Child attributes

**headers** HttpHeaders **required**

Child attributes

# GET favorite/

**GET** /favorite/

## Responses

**200 OK**

**Content type** \*/\*

OK

\*/\* **ResponseType**

Child attributes

**headers** HttpHeaders **required**

Child attributes

# POST favorite/

POST /favorite/

## Request parameters

### Body

**application/json** FavoriteRequestDTO **required**

Child attributes

**farvoriteId** object

Child attributes

**favoritactionDateTimestamp** object

Child attributes

**status** object

Child attributes

**userId** object

Child attributes

### JSON example

```
{  
  "farvoriteId": {},  
  "favoritactionDateTimestamp": {},  
  "status": {},  
  "userId": {}  
}
```

```
"userId": {}  
}
```

## Responses

200 OK

Content type \*/\*

OK

\*/\* ResponseEntity

Child attributes

**headers** HttpHeaders required

Child attributes

# Box Endpoints

Endpoints para Box

# PUT box/{favoriteld}

PUT /box/{favoriteld}

## Request parameters

### Path

**favoriteld** null

### Body

**application/json** BoxRequestDTO required

Child attributes

**boxId** object

Child attributes

**price** object

Child attributes

**title** object

Child attributes

**cover** object

Child attributes

**description** object

Child attributes

**status** object

Child attributes

**userId** object

Child attributes

## JSON example

```
{  
  "boxId": {},  
  "price": {},  
  "title": {},  
  "cover": {},  
  "description": {},  
  "status": {},  
  "userId": {}  
}
```

## Responses

**200 OK**

**Content type** \*/\*

OK

/\* ResponseEntity

Child attributes

**headers** HttpHeaders **required**

Child attributes

# DELETE box/{favoritId}

**DELETE** /box/{favoritId}

## Request parameters

### Path

**favoritId** null

## Responses

**200 OK**

**Content type** \*/\*

OK

\*/\* ResponseEntity

Child attributes

**headers** HttpHeaders **required**

Child attributes

# GET box/

GET /box/

## Responses

200 OK

Content type \*/\*

OK

\*/\* ResponseEntity

Child attributes

**headers** HttpHeaders required

Child attributes

# POST box/

POST /box/

## Request parameters

### Body

**application/json** BoxRequestDTO **required**

Child attributes

**boxId** object

Child attributes

**price** object

Child attributes

**title** object

Child attributes

**cover** object

Child attributes

**description** object

Child attributes

**status** object

Child attributes

**userId** object

Child attributes

## JSON example

```
{  
    "boxId": {},  
    "price": {},  
    "title": {},  
    "cover": {},  
    "description": {},  
    "status": {},  
    "userId": {}  
}
```

## Responses

200 OK

Content type    \*/\*

OK

  \*/ ResponseEntity

Child attributes

**headers** HttpHeaders **required**

    Child attributes

# GET box/{boxId}

GET

/box/{boxId}

## Request parameters

### Path

**boxId** object **required**

Child attributes

## Responses

**200 OK**

Content type `*/*`

OK

`*/*` `ResponseEntity`

Child attributes

**headers** `HttpHeaders` **required**

Child attributes

# Purchase Endpoints

Endpoints para Purchase

# GET purchase/{purchaseId}

**GET** /purchase/{purchaseId}

## Request parameters

### Path

**purchaseId** object **required**

Child attributes

## Responses

**200 OK**

Content type `*/*`

OK

`*/*` **ResponseType**

Child attributes

**headers** `HttpHeaders` **required**

Child attributes

# PUT purchase/{purchaseId}

**PUT** /purchase/{purchaseId}

## Request parameters

### Path

**purchaseId** object **required**

Child attributes

### Body

**application/json** PurchaseRequestDTO **required**

Child attributes

**purchaseId** object

Child attributes

**email** object

Child attributes

**rule** object

Child attributes

**avatar** object

Child attributes

**purchaseDate** object

Child attributes

**status** object

Child attributes

**amount** object

Child attributes

**userId** object

Child attributes

## JSON example

```
{  
  "purchaseId": {},  
  "email": {},  
  "rule": {},  
  "avatar": {},  
  "purschaseDate": {},  
  "status": {},  
  "amount": {},  
  "userId": {}  
}
```

## Responses

**200 OK**

**Content type** \*/\*

OK

/\* ResponseEntity

Child attributes

**headers** HttpHeaders **required**

Child attributes

# POST purchase/{purchaseId}

**POST** /purchase/{purchaseId}

## Request parameters

### Path

**purchaseId** object **required**

Child attributes

### Body

**application/json** PurchaseRequestDTO **required**

Child attributes

**purchaseId** object

Child attributes

**email** object

Child attributes

**rule** object

Child attributes

**avatar** object

Child attributes

**purchaseDate** object

Child attributes

**status** object

Child attributes

**amount** object

Child attributes

**userId** object

Child attributes

## JSON example

```
{  
  "purchaseId": {},  
  "email": {},  
  "rule": {},  
  "avatar": {},  
  "purschaseDate": {},  
  "status": {},  
  "amount": {},  
  "userId": {}  
}
```

## Responses

**200 OK**

**Content type** \*/\*

OK

/\* ResponseEntity

Child attributes

**headers** HttpHeaders **required**

Child attributes

# DELETE purchase/{purchaseId}

**DELETE** /purchase/{purchaseId}

## Request parameters

### Path

**purchaseId** object **required**

Child attributes

## Responses

**200 OK**

Content type `*/*`

OK

`*/*` **ResponseEntity**

Child attributes

**headers** `HttpHeaders` **required**

Child attributes

# GET purchase/

**GET** /purchase/

## Responses

**200 OK**

**Content type** \*/\*

OK

\*/\* **ResponseType**

Child attributes

**headers** HttpHeaders **required**

Child attributes

# Conclusão

O projeto atingiu uma parte significativa dos objetivos definidos inicialmente, com a implementação completa de 5 requisitos funcionais (RF01, RF02, RF03, RF05 e RF08) e 7 requisitos não funcionais (RNF01, RNF02, RNF03, RNF04, RNF05, RNF07 e RNF09). Além disso, conseguimos desenvolver a interface front-end e integrar a funcionalidade de criação de conjuntos (ensembles) com o back-end, o que representa um importante avanço na entrega do produto.

Apesar das limitações de tempo e das dificuldades técnicas enfrentadas, como a curva de aprendizado em ferramentas e tecnologias, a equipe demonstrou resiliência e colaboração ao superar esses desafios. O uso do modelo ágil Scrum foi fundamental para o sucesso do projeto, permitindo uma boa divisão de tarefas, revisões constantes, e melhorias contínuas ao longo de cada sprint. A metodologia também nos ajudou a lidar com prazos e promover um ambiente de trabalho estruturado e produtivo.

Durante a execução do projeto, adquirimos habilidades valiosas, como o trabalho em equipe, o hábito de realizar reuniões diárias (dailys) e a capacidade de lidar com prazos e adversidades. Esses aprendizados serão essenciais para futuros projetos e carreiras profissionais.

Em resumo, embora algumas funcionalidades ainda estejam pendentes, o projeto demonstrou um grande progresso, entregando um produto funcional e alinhado aos requisitos priorizados. Este trabalho serviu não apenas como um exercício técnico, mas também como uma experiência de crescimento e aprendizado para todos os envolvidos.

# Referencias Bibliográficas

- OMG. (2017). OMG Unified Modeling Language (OMG UML), Version 2.5.1. Object Management Group. <https://www.omg.org/spec/UML/2.5.1/> (<https://www.omg.org/spec/UML/2.5.1/>)
- Asana. (2024). Business Requirements Document Template: 7 Components. Recuperado de <https://asana.com/resources/business-requirements-document-template> (<https://asana.com/resources/business-requirements-document-template>)
- Mozilla Developer Network. MDN Web Docs - HTML (<https://developer.mozilla.org/en-US/docs/Web/HTML>) e CSS (<https://developer.mozilla.org/en-US/docs/Web/CSS>).
- Spring Team. Spring Boot Documentation (<https://docs.spring.io/spring-boot/docs/current/reference/html/>).
- W3schools Java. Java Tutorial (<https://www.w3schools.com/java/>)
- W3Schools. Git Tutorial (<https://www.w3schools.com/git/>).
- Pressman, R. S. (2009). *Ingeniería de Software* 9. Pearson.
- Cris Rúa. Ágil Es - Introdução às Metodologias Ágeis ([https://www.youtube.com/watch?v=16loziutCZs&ab\\_channel=%C3%81gilEs-PorCrisR%C3%BAa](https://www.youtube.com/watch?v=16loziutCZs&ab_channel=%C3%81gilEs-PorCrisR%C3%BAa)).
- Sommerville, I. (2011). *Engenharia de Software*. 10<sup>a</sup> edição. OCR.
- AEVO. Metodologia Ágil (<https://blog.aevo.com.br/metodologia-agil/>).
- Pontotel. Metodologia Scrum (<https://www.pontotel.com.br/metodologia-scrum/>).
- Lucidchart. Diagrama de Caso de Uso UML (<https://www.lucidchart.com/pages/pt/diagrama-de-caso-de-uso-uml#discoveryTop>).
- Lucidchart. O Que é Diagrama de Sequência UML (<https://www.lucidchart.com/pages/pt/o-que-e-diagrama-de-sequencia-uml>).

- IBM. Sequence Diagrams - UML (<https://www.ibm.com/docs/pt-br/rsm/7.5.0?topic=uml-sequence-diagrams>).
- MySQL. MySQL Tutorial Excerpt (<https://dev.mysql.com/doc/mysql-tutorial-excerpt/8.0/en/>).
- W3Schools. JavaScript Tutorial (<https://www.w3schools.com/js/>).