

Centro Universitário SENAC



Projeto POO - Equipe 6 Gestão de Moradores

Orientador: Carlos Henrique Veríssimo Pereira

São Paulo

2023

Identificação da Equipe

Id da Equipe: POOK483-Equipe 6		
ID de aluno	Nome	Papel
1141267353	Gabriel de Melo Silva	Gerente de Projeto
1142168549	Kevin de Sousa dos Santos	Quality Assurance e Desenvolvedor Front-End
1142445416	João Vitor Matulis	Arquiteto de software
1142473154	Bernardo Galvão	Desenvolvedor Back-End e DBA

Link do Github:

<https://github.com/JoaoVictorMatulis/SENAC-POOK483-Equipe6-GestaoMoradores/>

Índice

1	DESCRIÇÃO GERAL DO CLIENTE	5
1.1	DESCRIÇÃO DA NECESSIDADE	5
1.2	OBJETIVO	5
1.3	ESCOPO	5
2	REQUISITOS DO CLIENTE	5
2.1	REQUISITOS FUNCIONAIS	5
2.2	REQUISITOS NÃO FUNCIONAIS	6
3	CASOS DE USO(UC)	6
3.1	DIAGRAMA DE CASO DE USO - VISÃO GERAL	6
3.2	CASO DE USO - CASO DE USO #1	7
3.2.1	<i>Diagrama do Caso de uso #1</i>	7
3.2.2	<i>Detalhamento do Caso de uso #1</i>	7
3.3	CASO DE USO - CASO DE USO #2	8
3.3.1	<i>Diagrama do Caso de uso #2</i>	8
3.3.2	<i>Detalhamento do Caso de uso #2.1 - Criar acesso</i>	9
3.3.3	<i>Detalhamento do Caso de uso #2.2 - Remover acesso</i>	11
3.3.4	<i>Detalhamento do Caso de uso #2.3 - Atualizar acesso</i>	12
3.3.5	<i>Detalhamento do Caso de uso #2.4 - Ver moradores registrados</i>	14
3.3.6	<i>Detalhamento do Caso de uso #2.5 - Ver dados de um morador em específico</i>	14
3.4	CASO DE USO - CASO DE USO #3 - VERIFICAR REGISTROS DE ENTRADA E SAÍDA	15
3.4.1	<i>Diagrama do Caso de uso #3</i>	15
3.4.2	<i>Detalhamento do Caso de uso #3.1 - Verificar registros de entrada e de saída</i>	16
4	CLASSES	18
4.1	DIAGRAMA DE CLASSE - GERAL	18
4.2	DIAGRAMA DE CLASSE - ATUALIZAR CADASTRO	19
4.3	DIAGRAMA DE CLASSE - CRIAR ACESSO	20
4.4	DIAGRAMA DE CLASSE - EXIBIR MORADORES	21
4.5	DIAGRAMA DE CLASSE - EXIBIR REGISTROS	22

4.6	DIAGRAMA DE CLASSE - REMOVER CADASTRO	23
4.7	DIAGRAMA DE CLASSE - LOGIN	24
5	BANCO DE DADOS	25
6	DESENHO DA ARQUITETURA DO SISTEMA	25
7	CRONOGRAMA SEGUIDO	26
8	CONCLUSÃO	27

1 Descrição Geral do Cliente

1.1 Descrição da Necessidade

A necessidade	<i>Implementar uma solução de controle de clientes e vigilância de uma loja autônoma em um condomínio fechado.</i>
Afeta	<i>Moradores, funcionários e administração.</i>
O seu impacto é	<i>Aumento de segurança, prevenção de furtos e monitoramento da atividade da loja.</i>
Benefícios com a solução	<i>Os donos da loja evitam furtos, maximizando seus lucros, e podendo baratear os produtos aos moradores.</i>

1.2 Objetivo

O objetivo do sistema é prevenir furtos na loja, e aumento da segurança.

1.3 Escopo

- CRUD de moradores no sistema por um Admin
- Registro de entrada e saída de moradores na loja

2 Requisitos do Cliente

2.1 Requisitos Funcionais

RF1: Manutenção dos moradores no sistema.

RF2: Registro de entrada e saída dos moradores na loja.

RF3: Gerar relatório com todos os registros do Mês

2.2 Requisitos Não Funcionais

RNF1: O programa terá 2 modos: O modo gerencial, e o modo de registro

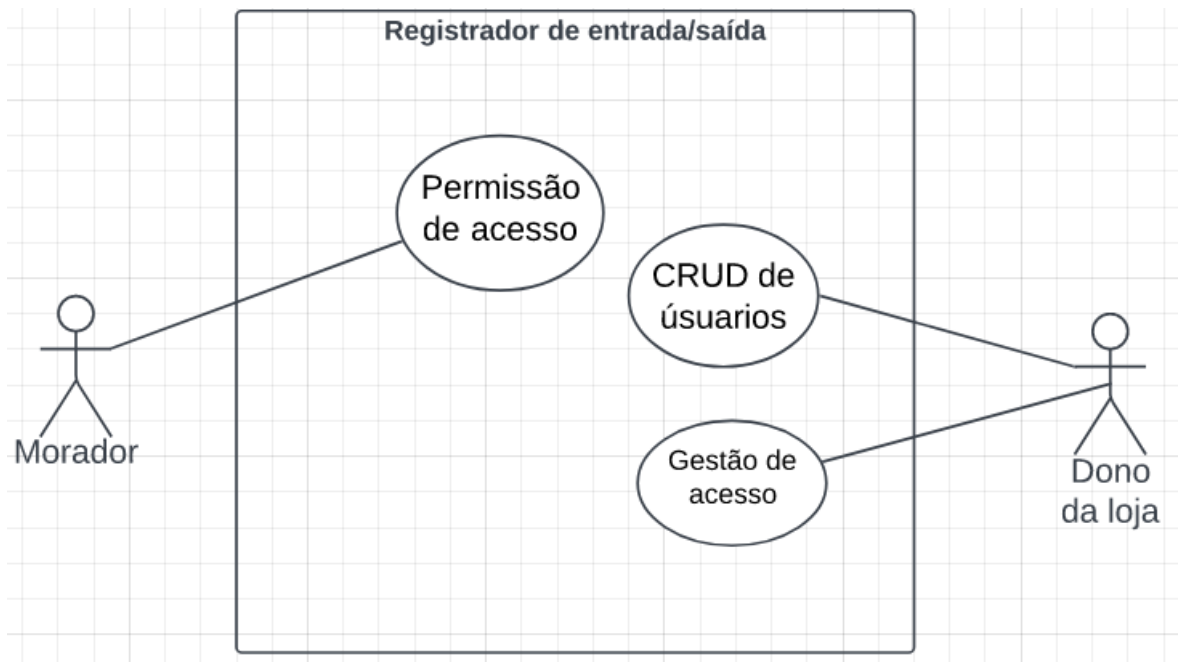
RNF2: O modo gerencial só pode ser acessado por um perfil de Admin

RNF3: O modo de registro pode ser inicializado rapidamente

RNF4: Somente o Admin poderá fazer o CRUD dos moradores

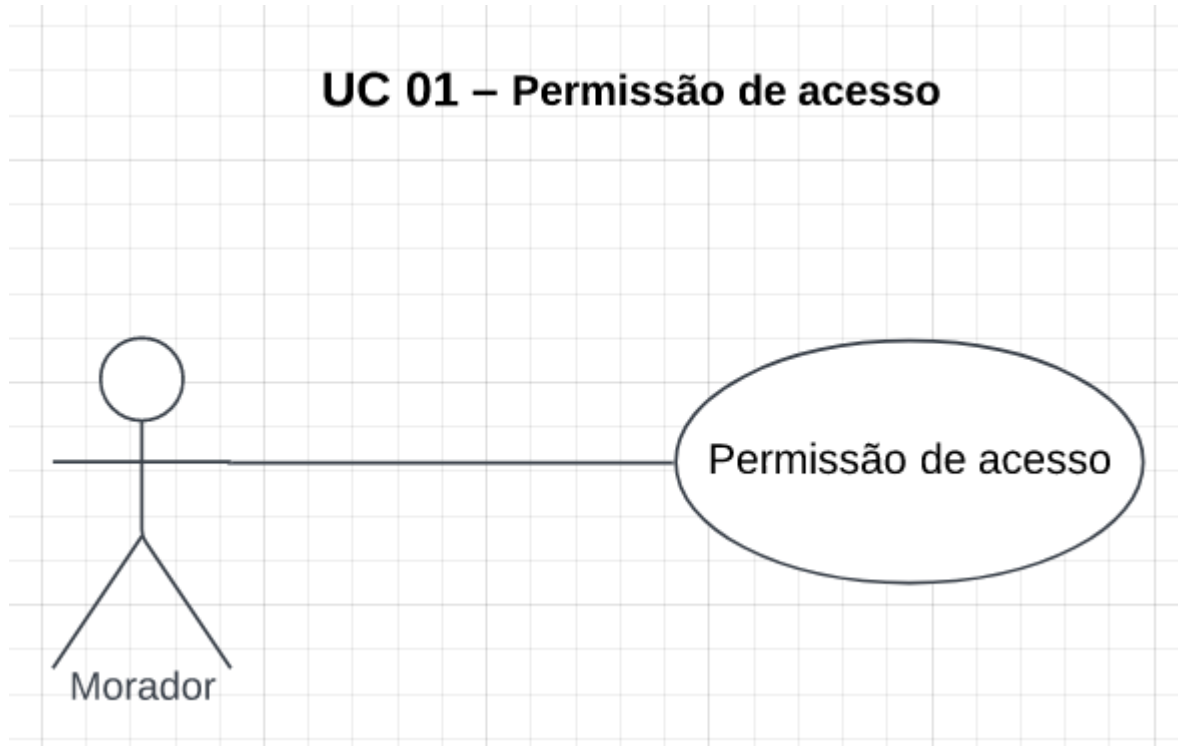
3 Casos de Uso(UC)

3.1 Diagrama de Caso de Uso - Visão Geral



3.2 Caso de Uso - Caso de Uso #1 – Inserir Senha

3.2.1 Diagrama do Caso de uso #1



3.2.2 Detalhamento do Caso de uso #1 - Permissão de acesso

Nome do Sistema que contém o Caso de Uso: Registrador de entrada/saída

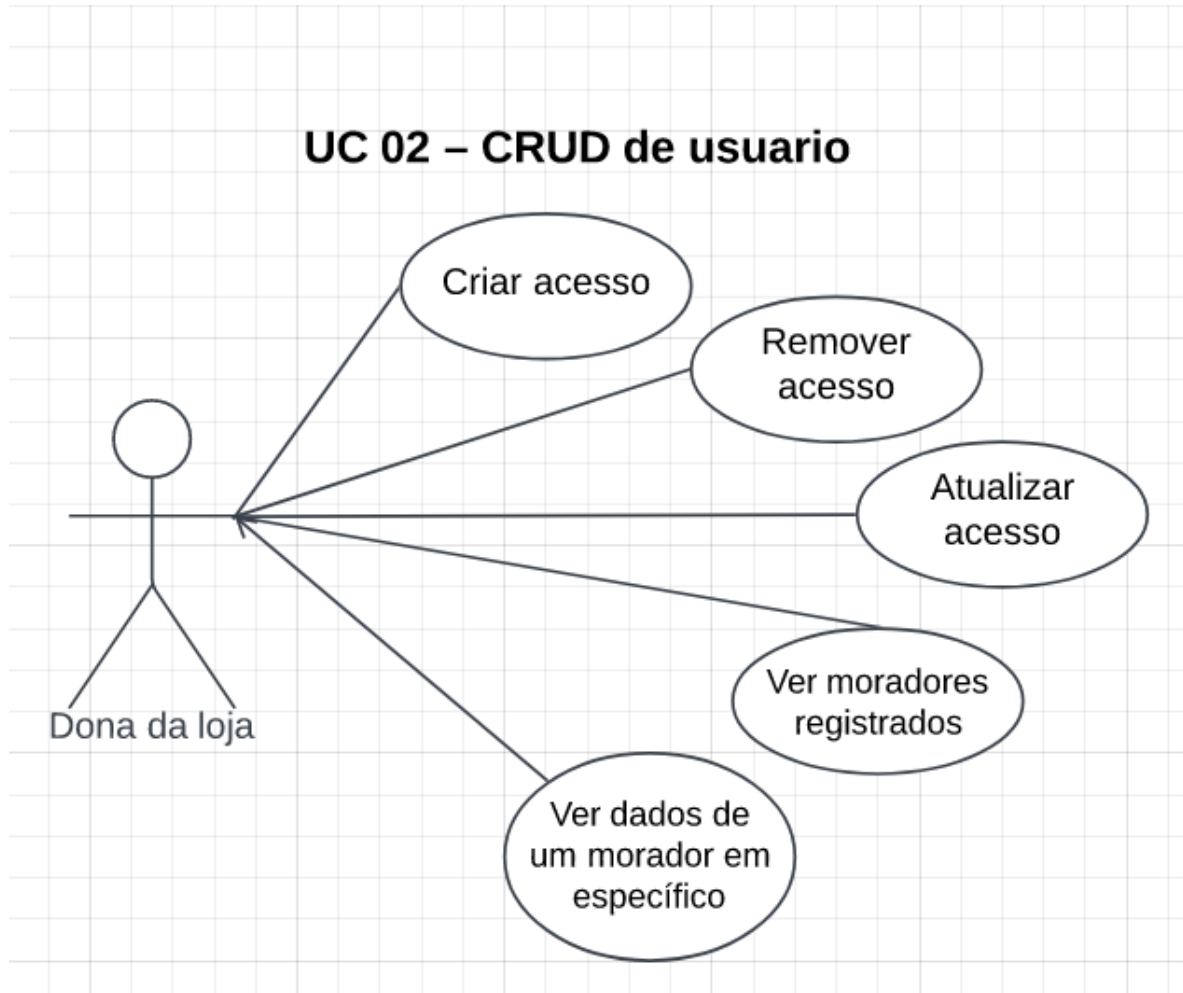
Especificação do caso de uso: Permissão de acesso

Nome do caso de uso	Permissão de acesso
Sumário	Caso de uso em que o morador insere sua senha pessoal para acessar a loja
Atores	Morador

Pré-Condição	Morador já cadastrado no banco de dados
Fluxo Principal	<ol style="list-style-type: none"> 1- O morador insere sua senha. 2- O sistema verifica a senha. [A1] 3- O sistema destranca a porta da loja. 4- O sistema registra o horário de entrada/saída do morador. 5- O caso de uso se encerra.
Fluxos Alternativos	<p>A1- Senha inválida</p> <ol style="list-style-type: none"> 1.1- O sistema retorna uma mensagem informando que a senha está inválida. 1.2- O sistema pede para informar uma senha válida. 1.3-Retornar para o passo 1 do fluxo principal.
Pós-Condição	Registrar horário de entrada/saída do morador

3.3 Caso de Uso - Caso de Uso #2 - CRUD

3.3.1 Diagrama do Caso de uso #2



3.3.2 Detalhamento do Caso de uso #2.1 - Criar acesso

Nome do Sistema que contém o Caso de Uso: Registrador de entrada/saída

Especificação do caso de uso: 2.1 - Criar acesso

Nome do caso de uso	Criar acesso
---------------------	--------------

Sumário	Caso de uso em que o dono da loja quer criar um novo acesso para um morador
Atores	Dono da loja
Pré-Condição	Dono da loja identificado no sistema e informar uma senha e um morador não existentes no banco de dados
Fluxo Principal	<ol style="list-style-type: none"> 1- O dono da loja informa um morador 2- O sistema verifica se o morador já existe no banco de dados. [A1] 3- O dono da loja informa uma senha 3- O sistema verifica a senha. [A2] 4- O sistema registra o morador e sua senha no banco de dados 5- O sistema retorna uma mensagem confirmando a criação do acesso. 6- O sistema envia um email para o morador informando sua senha 7- O caso de uso se encerra.
Fluxos Alternativos	<p>A1- Morador já existente no banco de dados</p> <ol style="list-style-type: none"> 1.1- O sistema retorna uma mensagem informando que o morador já existe no banco de dados 1.2- O sistema pergunta se o dono quer ver os dados de acesso do morador ou atualizar os dados para esse morador. <ol style="list-style-type: none"> 1.2.1- Caso escolha ver os dados do morador, ir para o caso de uso 2.5 1.2.2- Caso escolha atualizar os dados do morador ir para o caso de uso 2.3

	<p>1.3- O caso de uso se encerra.</p> <p>A2- Senha repetida no banco de dados</p> <p>1.1- O sistema retorna uma mensagem informando que a senha está já existe no banco de dados e a qual morador ela pertence.</p> <p>1.2- O sistema pede para informar outra senha.</p> <p>1.3-Retornar para o passo 3 do fluxo principal.</p>
Pós-Condição	Registramento do morador e sua senha no banco de dados

3.3.3 Detalhamento do Caso de uso #2.2 - Remover acesso

Nome do Sistema que contém o Caso de Uso: Registrador de entrada/saída

Especificação do caso de uso: 2.2 Remover acesso

Nome do caso de uso	Remover acesso
Sumário	Caso de uso em que o dono da loja quer remover o acesso de um morador de sua loja.
Atores	Dono da loja
Pré-Condição	Dono da loja identificado no sistema, e um morador registrado no banco de dados

Fluxo Principal	<p>1- O dono da loja informa um morador</p> <p>2- O sistema verifica se o morador existe no banco de dados. [A1]</p> <p>3- O sistema mostra os dados do morador</p> <p>4- O sistema confirma se o dono quer a remoção do morador.[A2]</p> <p>5- O sistema remove o morador do banco de dados</p> <p>6- O sistema retorna uma mensagem confirmando a remoção do morador.</p> <p>8- O caso de uso se encerra</p>
Fluxos Alternativos	<p>A1- Morador não existe no banco de dados</p> <p>1.1- O sistema retorna uma mensagem informando que não foi possível achar o morador no banco de dados</p> <p>1.2- O caso de uso se encerra.</p> <p>A2- O dono cancela a remoção do morador</p> <p>1.1- O caso de uso se encerra.</p>
Pós-Condição	Remoção do morador do banco de dados

3.3.4 Detalhamento do Caso de uso #2.3 - Atualizar acesso

Nome do Sistema que contém o Caso de Uso: Registrador de entrada/saída

Especificação do caso de uso: 2.3 Atualizar acesso

Nome do caso de uso	Atualizar acesso
Sumário	Caso de uso em que o dono da loja quer atualizar o acesso de um morador
Atores	Dono da loja
Pré-Condição	Dono da loja identificado no sistema, e um morador registrado no banco de dados
Fluxo Principal	<ol style="list-style-type: none">1- O dono da loja informa um morador2- O sistema verifica se o morador existe no banco de dados. [A1]3- O dono da loja informa uma nova senha para esse morador3- O sistema verifica a senha. [A2]4- O sistema atualiza a senha do morador no banco de dados5- O sistema retorna uma mensagem confirmando a atualização da senha.6- O sistema envia em email para esse morador informando sua nova senha7- O caso de uso se encerra.

Fluxos Alternativos	<p>A1- Morador não existe no banco de dados</p> <p>1.1- O sistema retorna uma mensagem informando que não foi possível achar o morador no banco de dados</p> <p>1.2- O caso de uso se encerra.</p> <p>A2- Senha repetida no banco de dados</p> <p>1.1- O sistema retorna uma mensagem informando que a senha está já existe no banco de dados e a qual morador ela pertence.</p> <p>1.2- O sistema pede para informar outra senha.</p> <p>1.3-Retornar para o passo 3 do fluxo principal.</p>
Pós-Condição	Atualização dos dados do morador no banco de dados

3.3.5 Detalhamento do Caso de uso #2.4 - Ver moradores registrados

Nome do Sistema que contém o Caso de Uso: Registrador de entrada/saída

Especificação do caso de uso: 2.4 Ver moradores registrados

Nome do caso de uso	Ver moradores registrados
Sumário	Caso de uso em que o dono da loja quer ver todos os moradores registrados no banco de dados
Atores	Dono da loja
Pré-Condição	Dono da loja identificado no sistema

Fluxo Principal	<ol style="list-style-type: none"> 1- O dono da loja pede todos os moradores registrados 2- O sistema retorna uma lista dos moradores registrados 3- O caso de uso se encerra.
Pós-Condição	Verificação dos moradores registrados

3.3.6 Detalhamento do Caso de uso #2.5 - Ver dados de um morador em específico

Nome do Sistema que contém o Caso de Uso: Registrador de entrada/saída

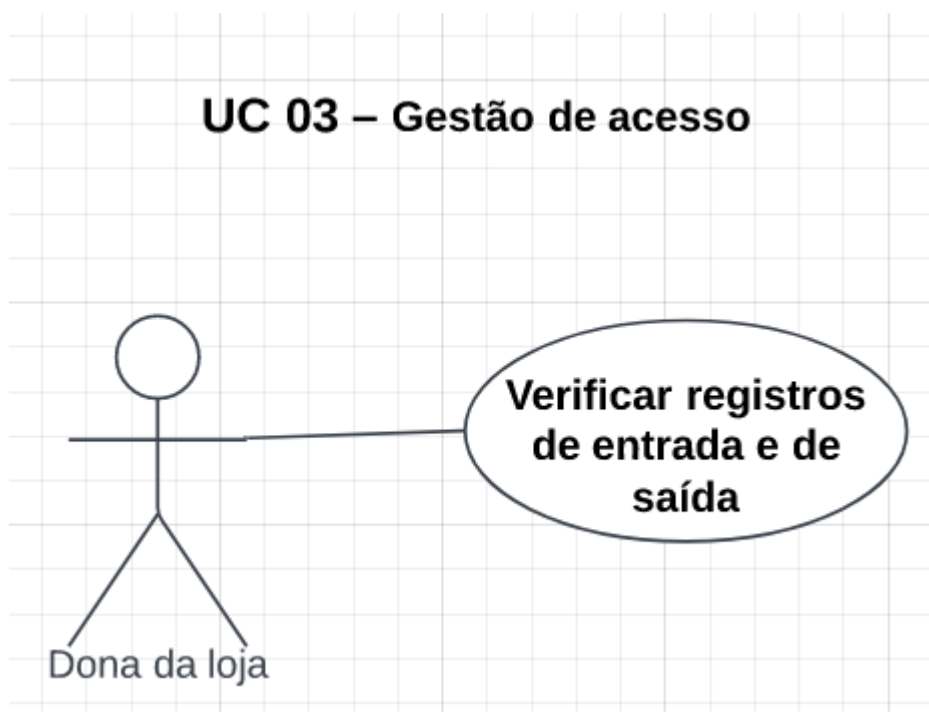
Especificação do caso de uso: 2.5 Ver dados de um morador em específico

Nome do caso de uso	Ver dados de um morador em específico
Sumário	Caso de uso em que o dono da loja quer ver os dados de um morador em específico
Atores	Dono da loja
Pré-Condição	Dono da loja identificado no sistema, e um morador registrado no banco de dados
Fluxo Principal	<ol style="list-style-type: none"> 1- O dono da loja informa um morador 2- O sistema verifica se o morador existe no banco de dados. [A1] 3- O sistema retorna os dados desse morador 4- O caso de uso se encerra.

Fluxos Alternativos	<p>A1- Morador não existe no banco de dados</p> <p>1.1- O sistema retorna uma mensagem informando que não foi possível achar o morador no banco de dados</p> <p>1.2- O caso de uso se encerra.</p>
Pós-Condição	Verificação de dados de um morador em específico

3.4 Caso de uso - Caso de Uso #3 - Verificar registros de entrada e saída

3.4.1 Diagrama do Caso de uso #3



3.4.2 Detalhamento do Caso de uso #3.1 - Verificar registros de entrada e de saída

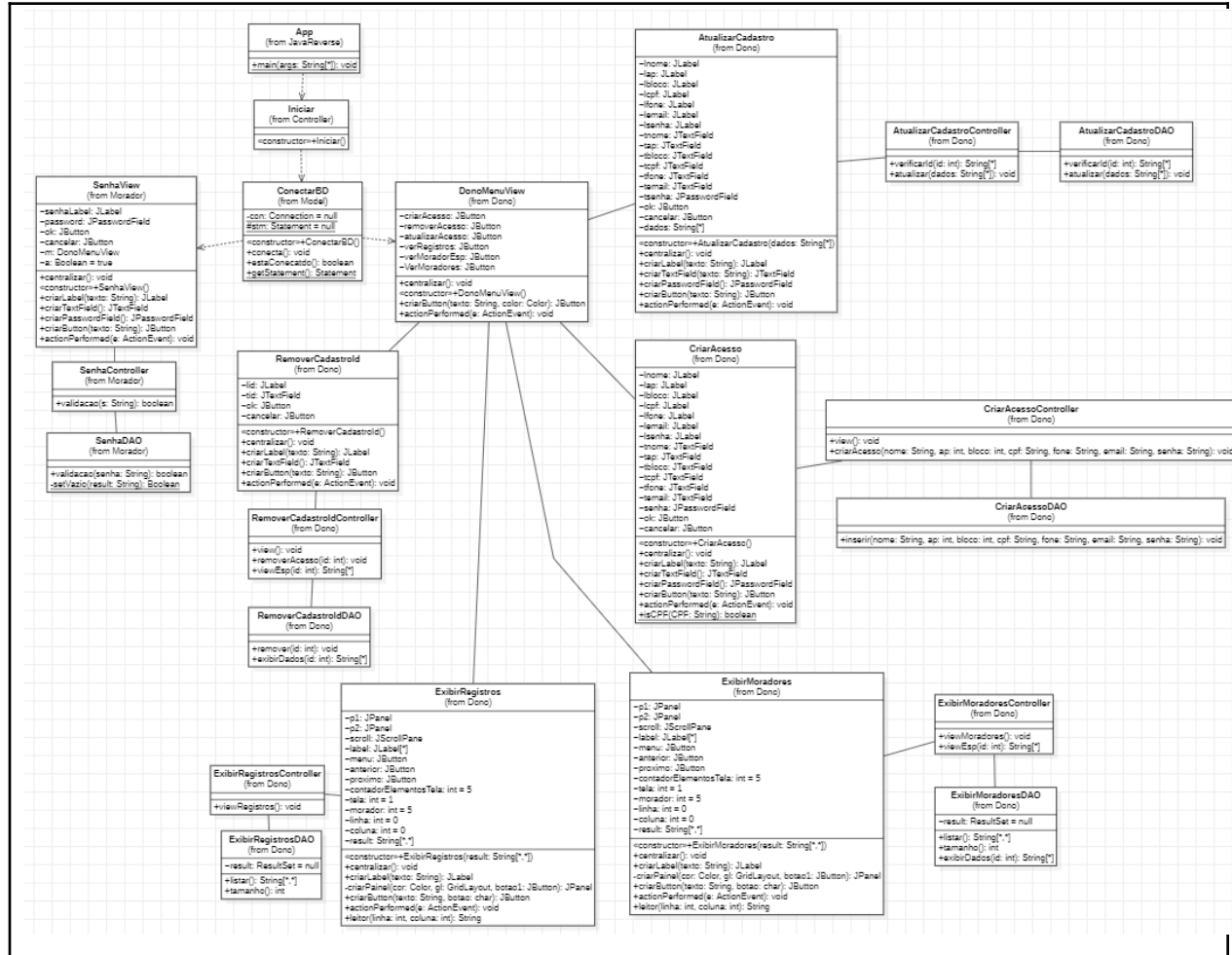
Nome do Sistema que contém o Caso de Uso: Registrador de entrada/saída

Especificação do caso de uso: 3.1 - Verificar registros de entrada e de saída

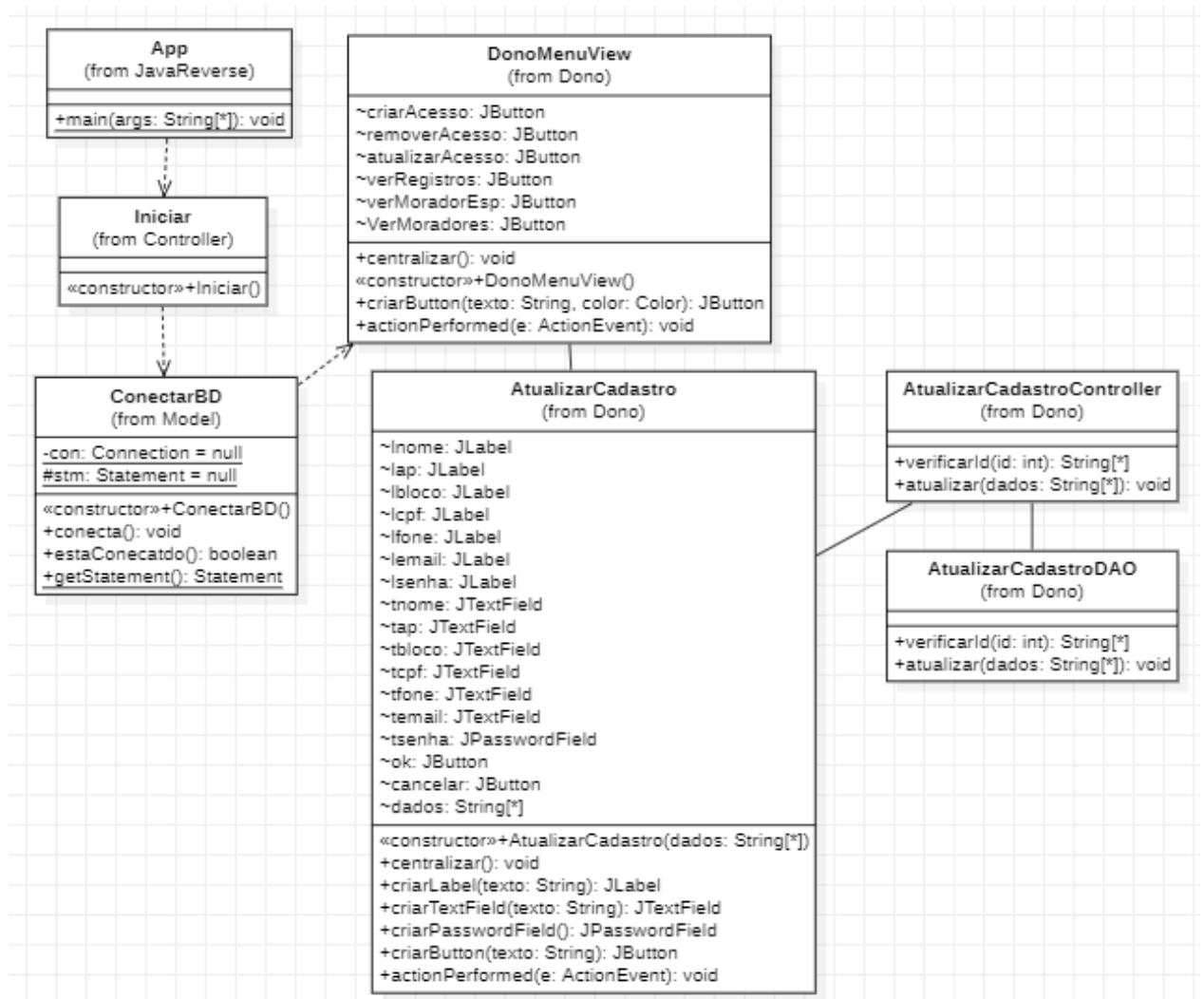
Nome do caso de uso	Verificar registros de entrada e de saída
Sumário	Caso de uso em que o dono da loja quer ver todos os acessos de entrada e saída da loja
Atores	Dono da loja
Pré-Condição	Dono da loja identificado no sistema e tiver algum registro de acesso no banco de dados
Fluxo Principal	<ol style="list-style-type: none">1- O dono da loja pede os acessos da loja2- O sistema retorna todos os acessos3- O caso de uso se encerra.
Pós-Condição	Verificação dos acessos da loja

4 Diagrama de Classe

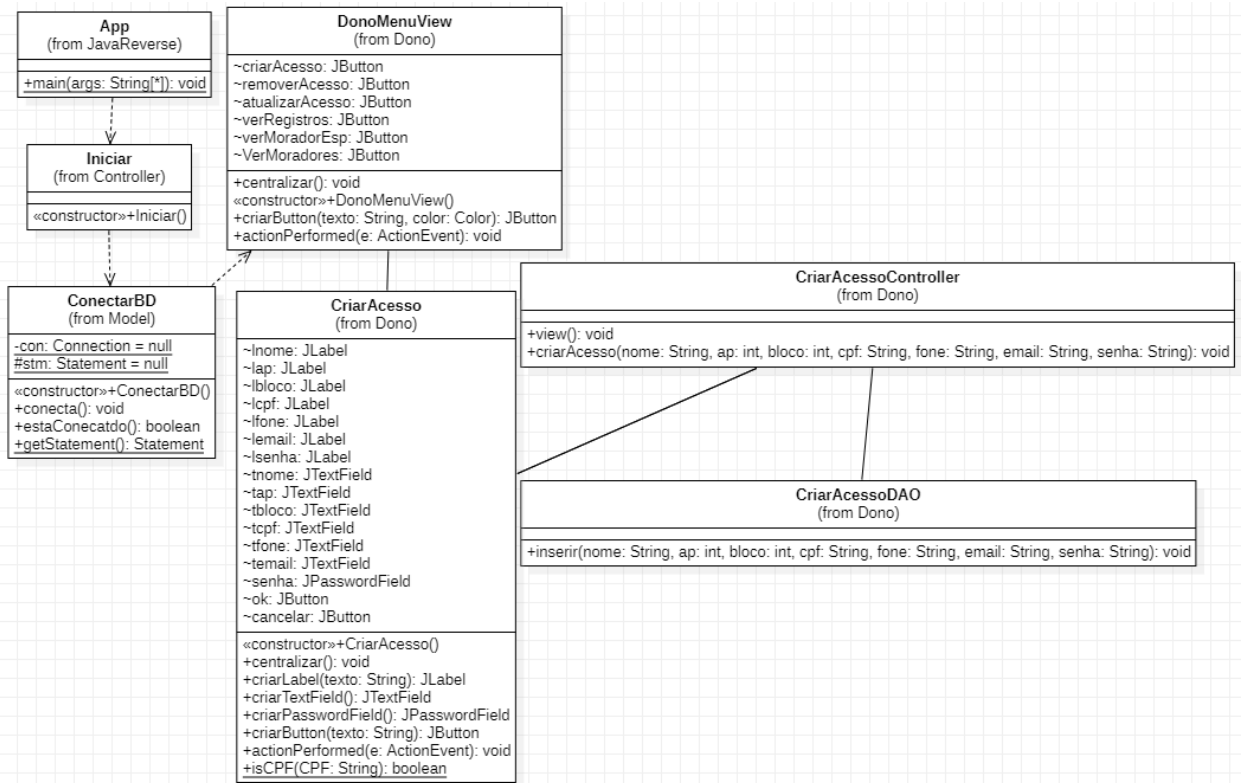
4.1 Diagrama de classe geral:



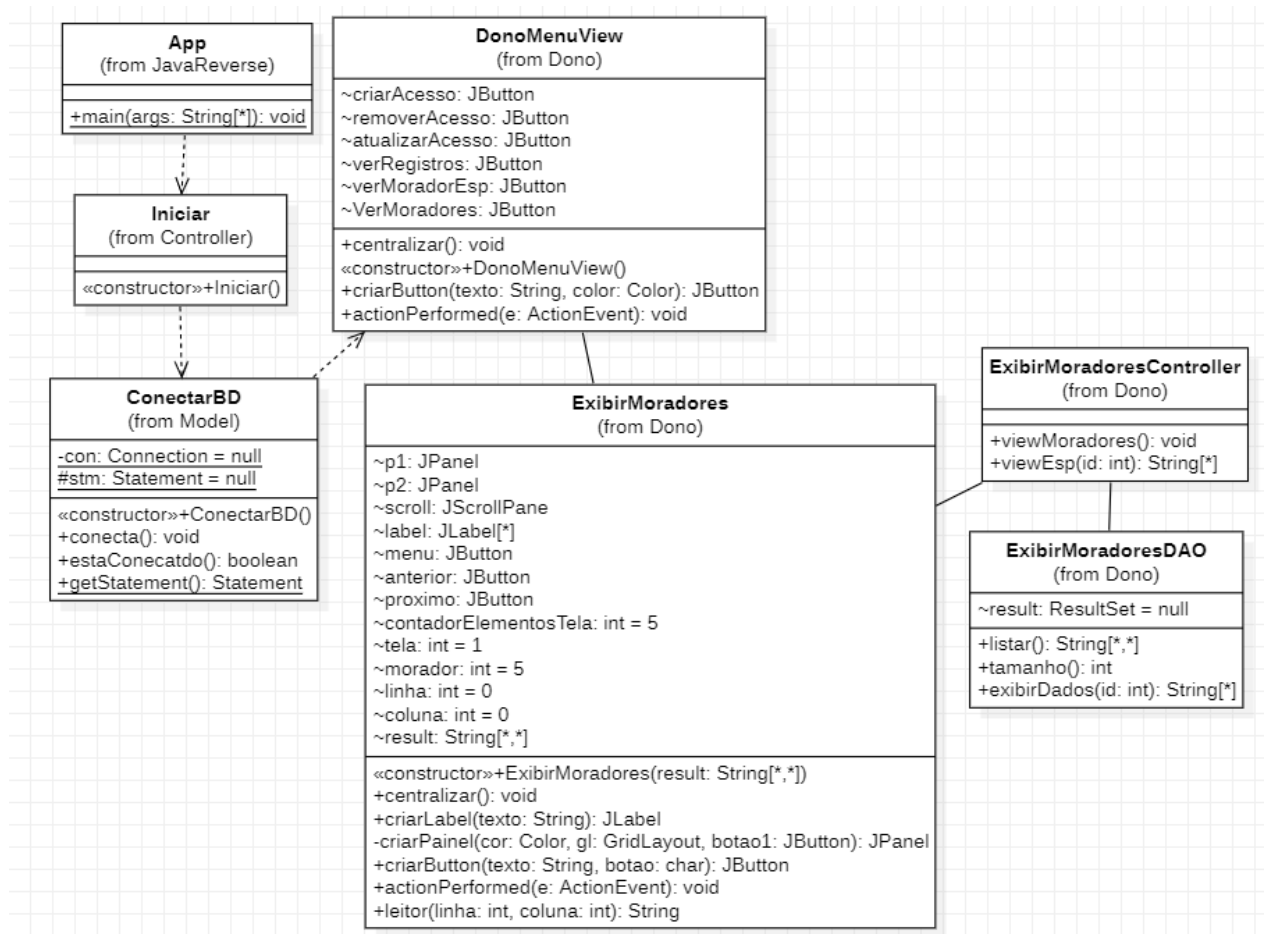
4.2 Diagrama de classe atualizar cadastro:



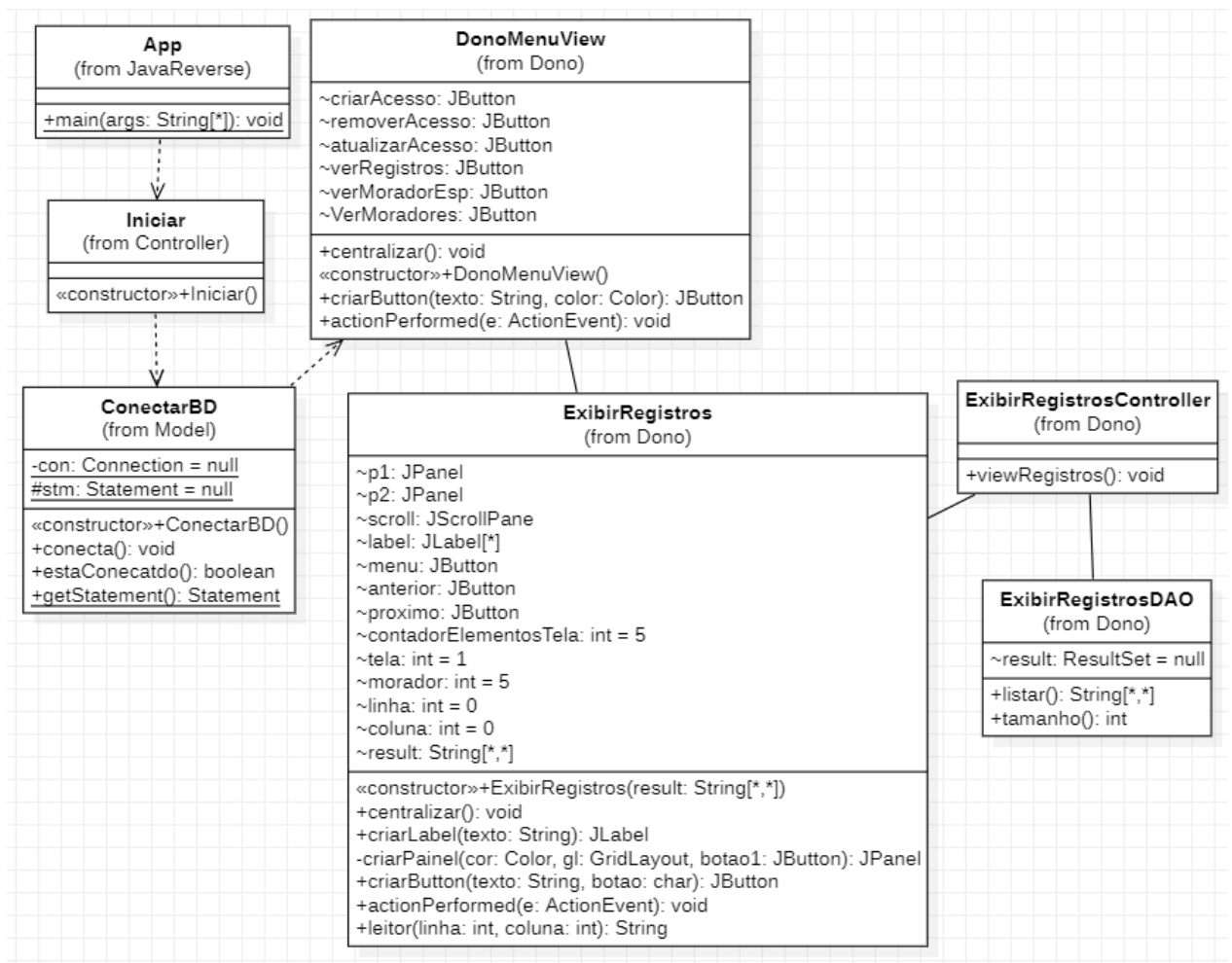
4.3 Diagrama de classe criar acesso:



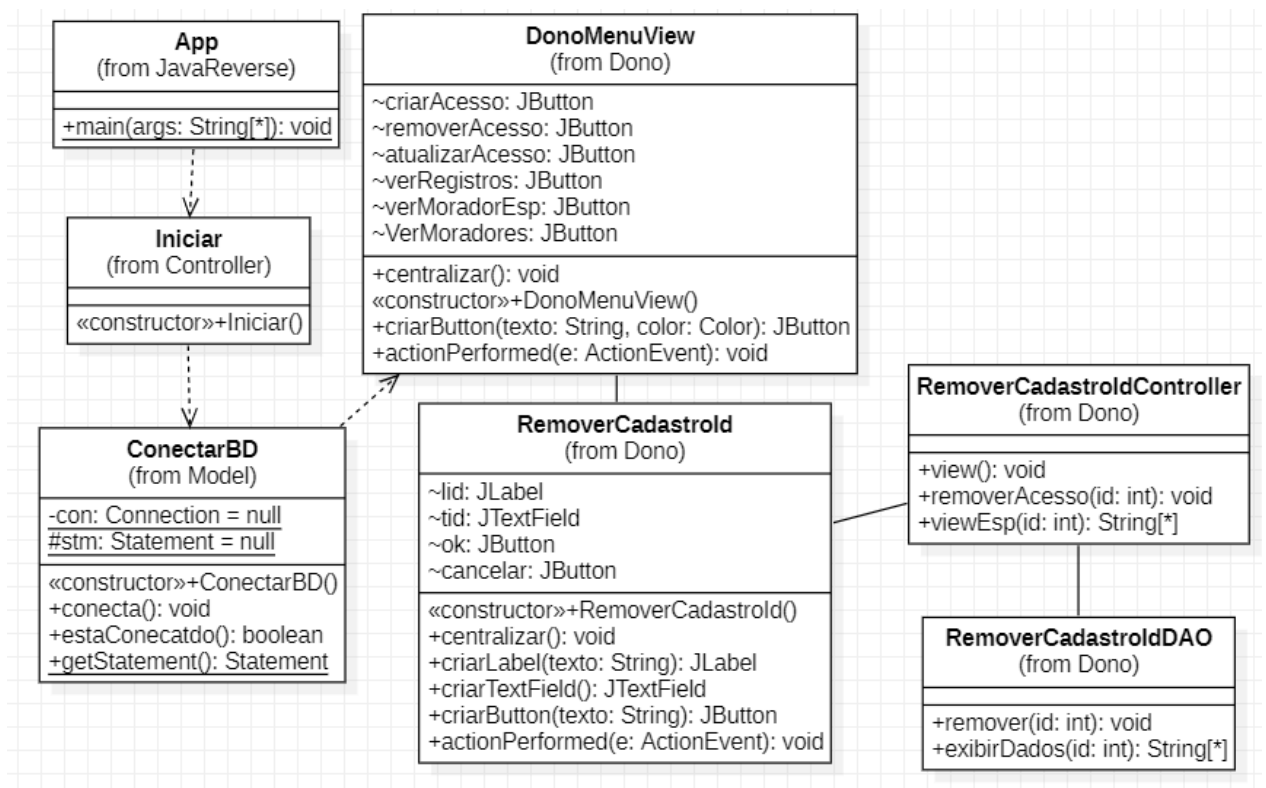
4.4 Diagrama de classe exibir moradores:



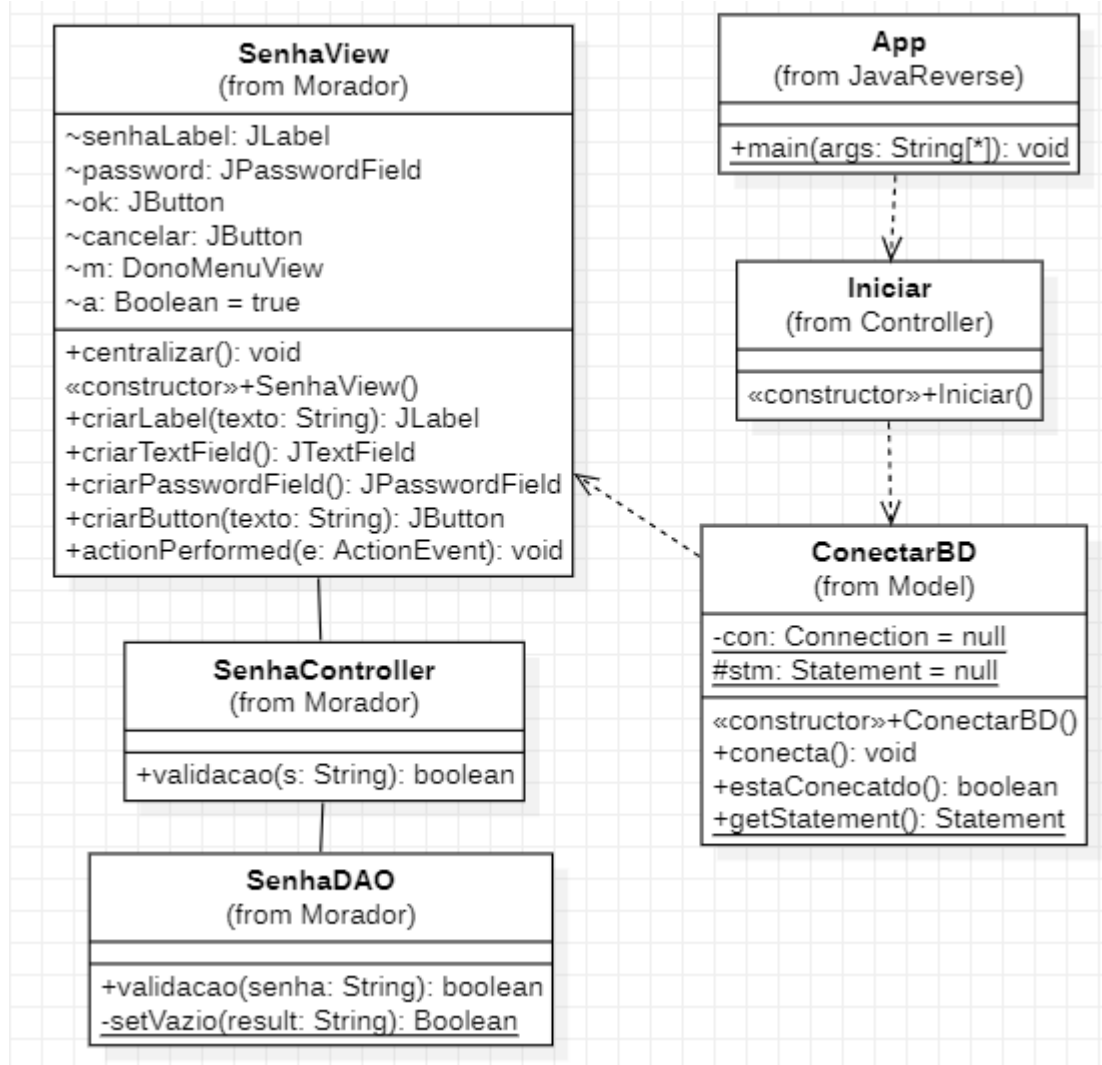
4.5 Diagrama de classe exibir registros:



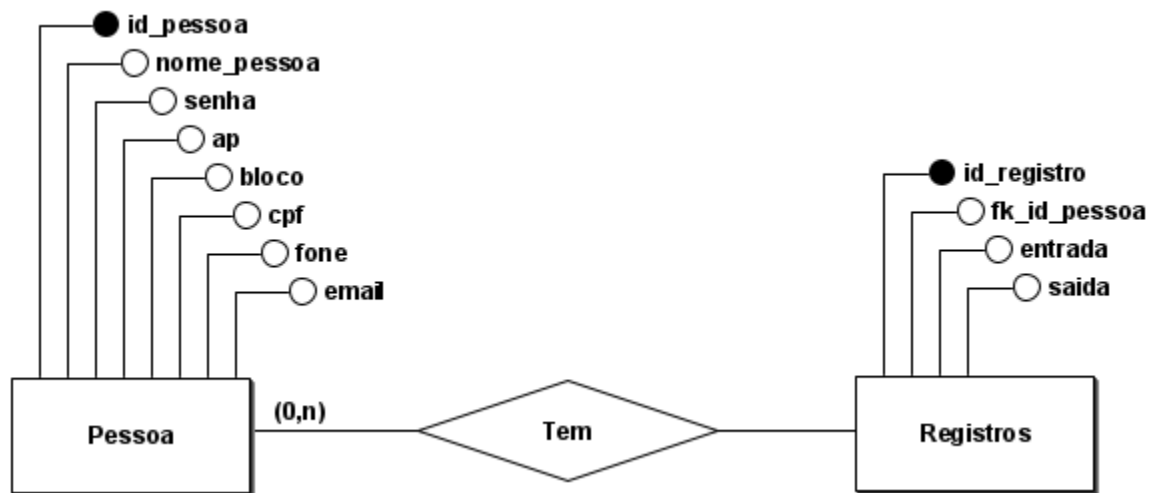
4.6 Diagrama de classe remover cadastro:



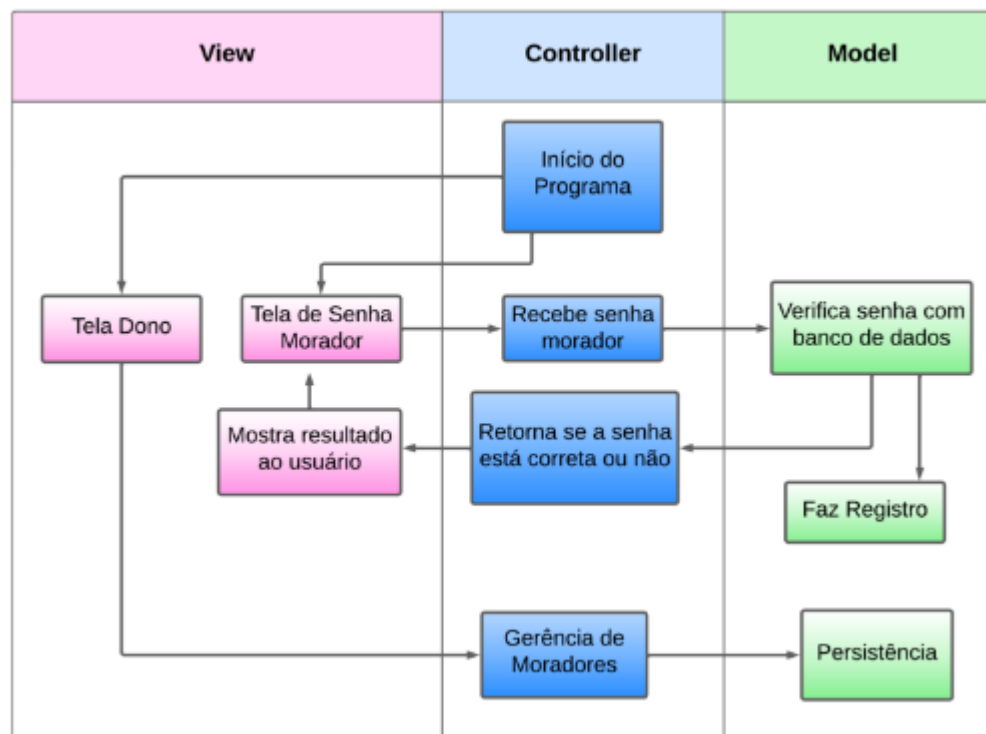
4.7 Diagrama de classe login:



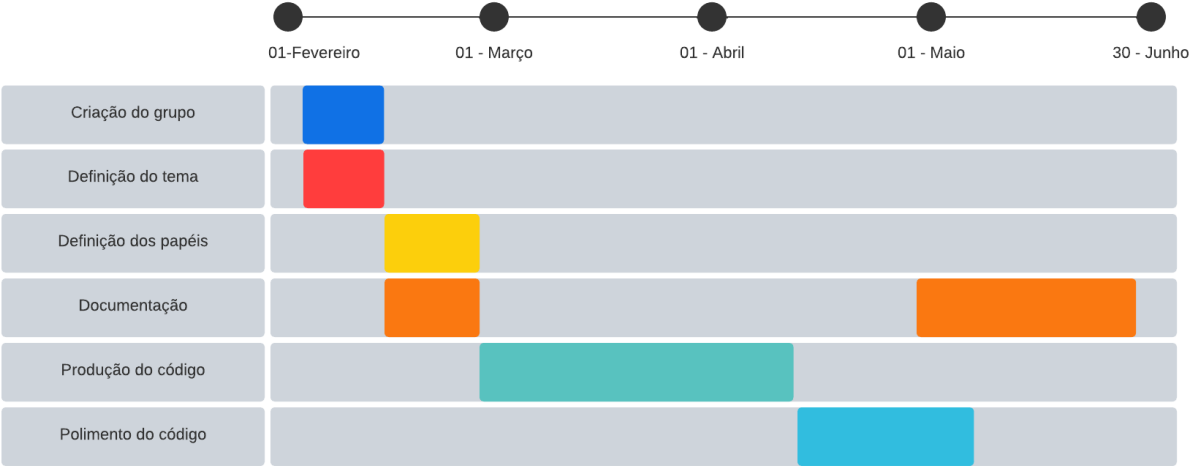
5 Banco de Dados



6 Desenho da Arquitetura do Sistema



7 Cronograma Seguido



8 Conclusão

Ao final do projeto, vimos a consolidação de tudo o que foi passado em aula, desde os pilares da Orientação a Objetos, até as ferramentas e metodologias de organização, como MVC, desenhos do sistema e outros conceitos.

O sistema foi desenvolvido buscando atender a esses conceitos o mais fielmente possível, sem deixar de entregar aquilo que foi definido pelo grupo.

O aprendizado adquirido em aula, com essas práticas já adotadas pelo mercado, com certeza fará da equipe desenvolvedores mais preparados e completos para o mundo em si.