

Sistema de Coleta de lixo

Elaborado por: Lucas Vilela Oliveira e Getulio Amorim Brunes

Visão Geral

Este sistema busca proporcionar uma experiência de usuário intuitiva e eficiente, ao mesmo tempo em que oferece uma administração robusta e controlada, com funcionalidades específicas para gerenciar usuários e agendamentos. A aplicação pode ser utilizada por qualquer pessoa que precise agendar coletas ou administrar um conjunto de usuários, garantindo flexibilidade e controle sobre os processos do sistema.

Diagrama de Caso de Uso

Atores:

Usuário: Qualquer pessoa que acessa o sistema, realizando operações básicas.

Administrador: Usuário com permissões avançadas, podendo gerenciar outros usuários.

Casos de Uso:

Login: Permite que o usuário acesse o sistema utilizando suas credenciais.

Cadastro: Usuários podem criar uma nova conta no sistema.

Agendar Coleta: Usuários podem solicitar o agendamento de coletas.

Gerar Comprovante: O sistema gera comprovantes para ações realizadas, como coletas agendadas.

Gerenciar Usuários: Caso de uso exclusivo para o Administrador, permitindo a visualização, edição e exclusão de contas de usuários.

Relações:

O Usuário pode realizar Login, Cadastro, Agendar Coleta e Gerar Comprovante.

O Administrador tem acesso ao caso de uso Gerenciar Usuários, que está relacionado a Login e Cadastro, indicando que o gerenciamento de usuários depende dessas funcionalidades.

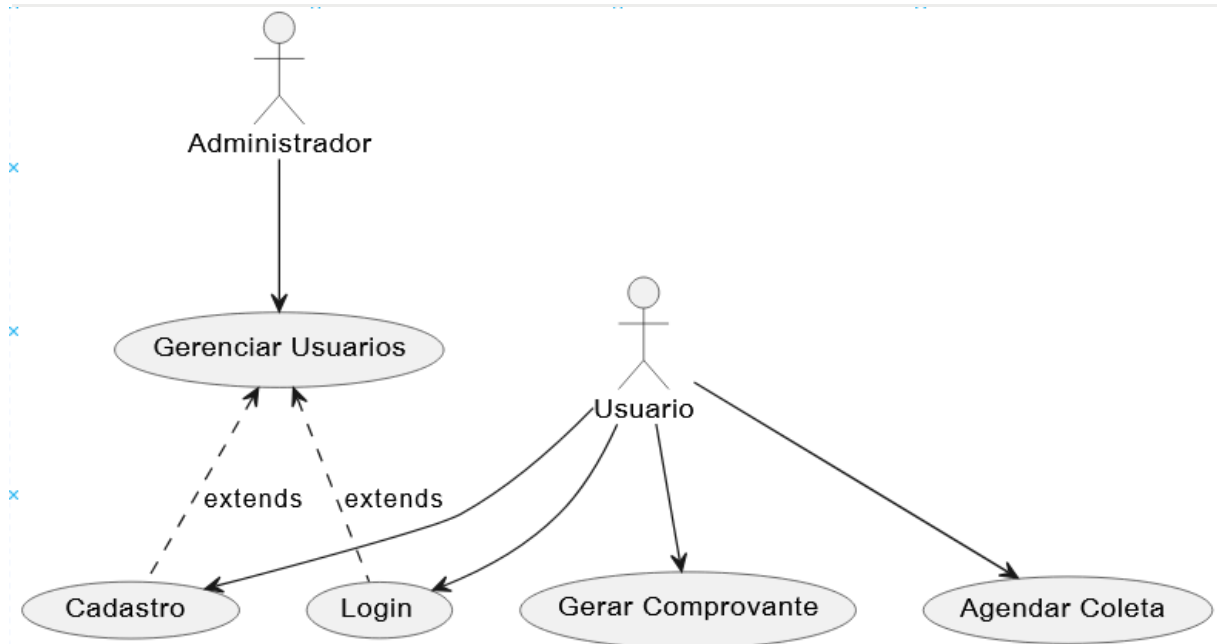


Diagrama de Classes

Classes e Detalhes:

Main:

Função: Ponto de entrada do sistema.

Métodos:

main(String[] args): Inicializa a aplicação e instancia a classe Sistema.

Sistema:

Função: Núcleo da aplicação, gerencia usuários, interface gráfica e funcionalidades Principais.

Atributos:

usuarios: Mapeia os nomes dos usuários para objetos da classe Usuario.

registros: Armazena informações sobre as atividades realizadas no sistema.

agenda: Gerencia os agendamentos de coletas.

mainFrame: Representa a janela principal da interface gráfica (Swing).

csvFile: Caminho do arquivo usuarios.csv para persistência de dados.

Métodos:

loadUsuariosFromCSV(): Carrega os dados dos usuários a partir do arquivo CSV.

saveUsuarioToCSV(Usuario user): Salva novos usuários no arquivo CSV.

criarJanelaLogin(), criarJanelaCadastro(), criarJanelaPrincipal(): Gerenciam as janelas da interface gráfica.

getCurrentDayPortuguese(): Retorna a data atual formatada em português.

criarPainelUsuario(), criarPainelAdmin(): Configuram os painéis de interface para usuários e administradores.

Usuario:

Função: Representa os usuários do sistema.

Atributos:

username: Nome do usuário.

senha: Senha do usuário.

isAdmin: Indica se o usuário possui privilégios administrativos.

Métodos:

getUsername(), getSenha(): Retornam os dados do usuário.

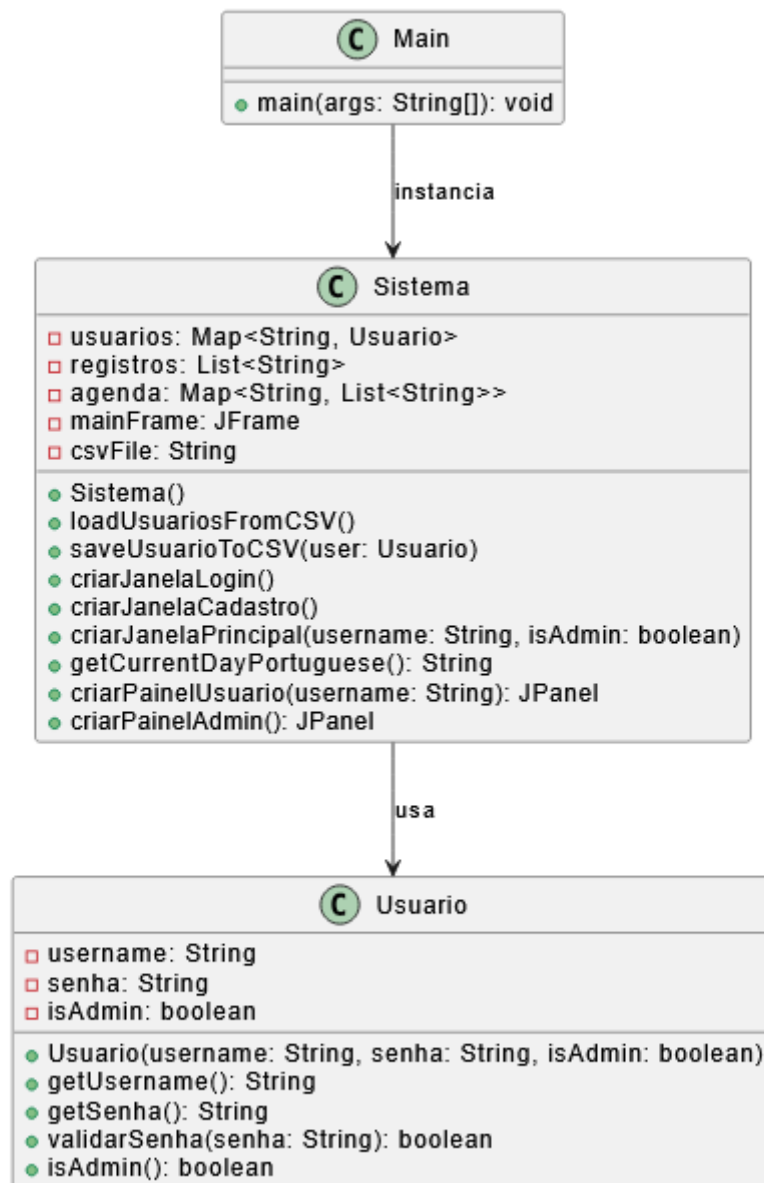
validarSenha(String senha): Verifica se a senha fornecida é válida.

isAdmin(): Retorna o status do usuário como administrador.

Relações entre as Classes:

Main → Sistema: O Main instancia o Sistema, que inicia o fluxo da aplicação.

Sistema → Usuario: O Sistema gerencia objetos da classe Usuario, principalmente para autenticação e controle de permissões.



Considerações Finais

O sistema descrito é uma aplicação voltada para a gestão de usuários e agendamentos de coletas, com diferentes níveis de permissão de acesso. Ele é composto por um conjunto de funcionalidades que permitem aos usuários interagir com o sistema de maneira simples, enquanto oferece ao administrador a capacidade de gerenciar contas e realizar ações avançadas.

