

Tema do Projeto

Localizador de Jogos com Gerenciamento de Contas

Descrição do Sistema

A ideia do projeto surgiu da necessidade de facilitar a vida de quem joga em várias plataformas. O sistema reúne jogos em um único lugar e permite abrir links, acessar portais, favoritar, ver histórico, pesquisar e cadastrar novos jogos via interface administrativa.

Problema a ser Resolvido

Jogadores lidam com diversos launchers, portais e sites. O LdJ elimina esse caos ao centralizar tudo em uma única interface.

Principais Entidades do Sistema

- Jogo
- Plataforma
- Usuário
- Favorito
- Histórico
- Jogo Customizado (NOVO)

Funcionalidades Básicas

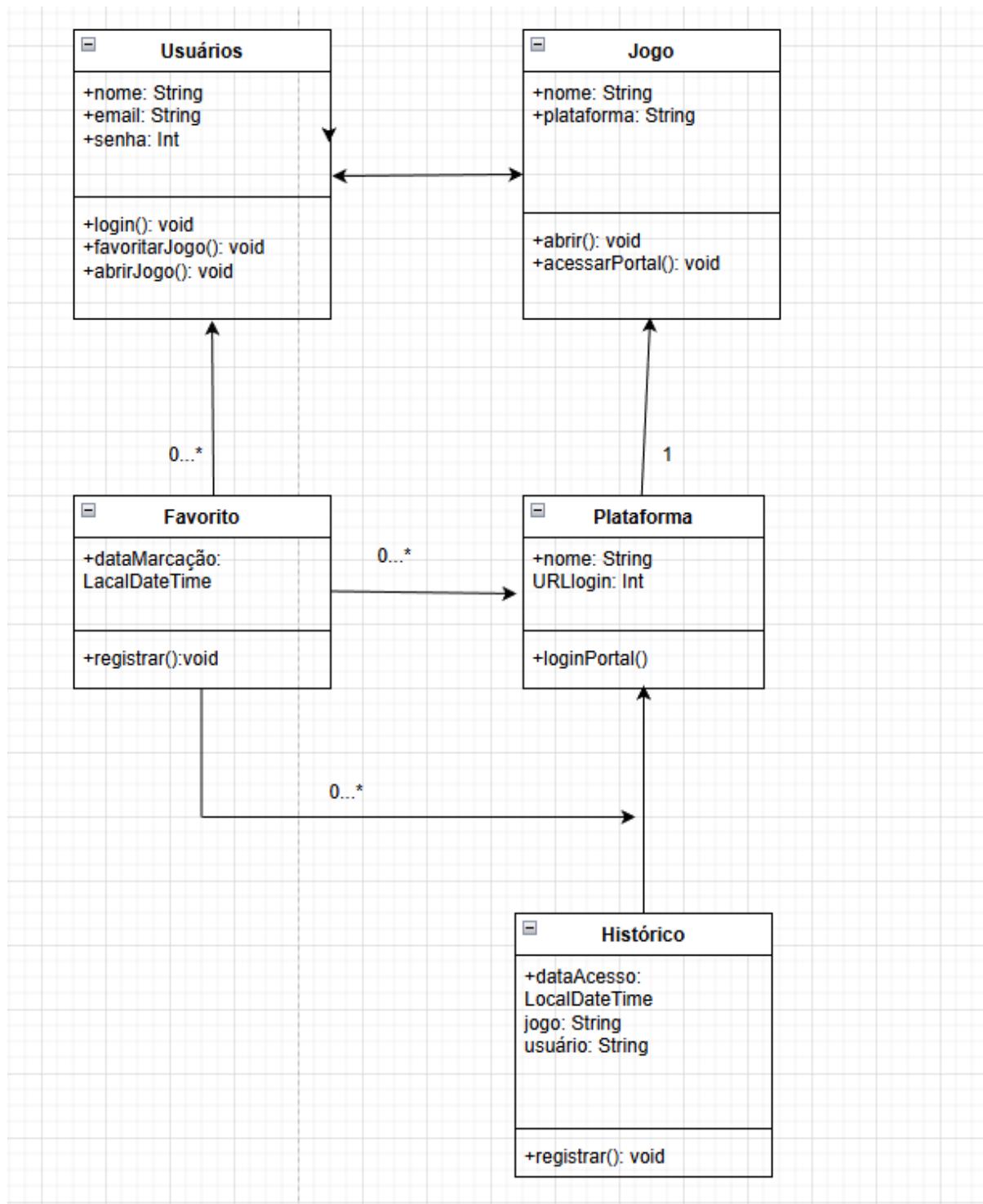
- Busca
- Abrir jogo
- Portal
- Favoritar
- Histórico
- Tema claro/escuro

- Login Admin (NOVO)
- Cadastro de Jogos (NOVO)
- Persistência de Jogos Customizados (NOVO)
- Histórico aprimorado com rolagem (NOVO)

Tecnologias Utilizadas

- Java
- Java Swing
- MVC
- Preferences API

OBJETIVO 2



Relacionamentos

Usuário ↔ Jogo

Usuário ↔ Favorito

Usuário ↔ Histórico

Jogo ↔ Plataforma

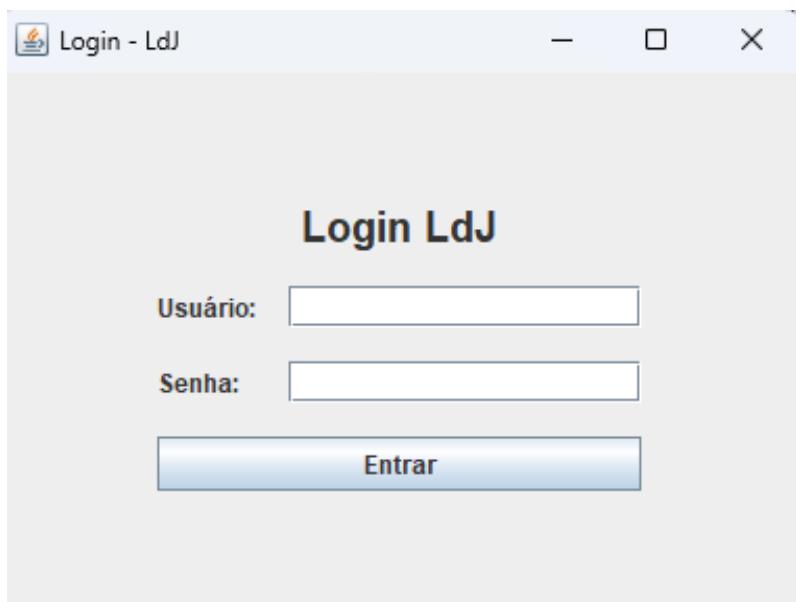
Histórico ↔ Jogo

Favorito ↔ Jogo

Administrador ↔ Jogo Customizado

Apresentação da aplicação — LdJ (Localizador de Jogos)

Tela de Login



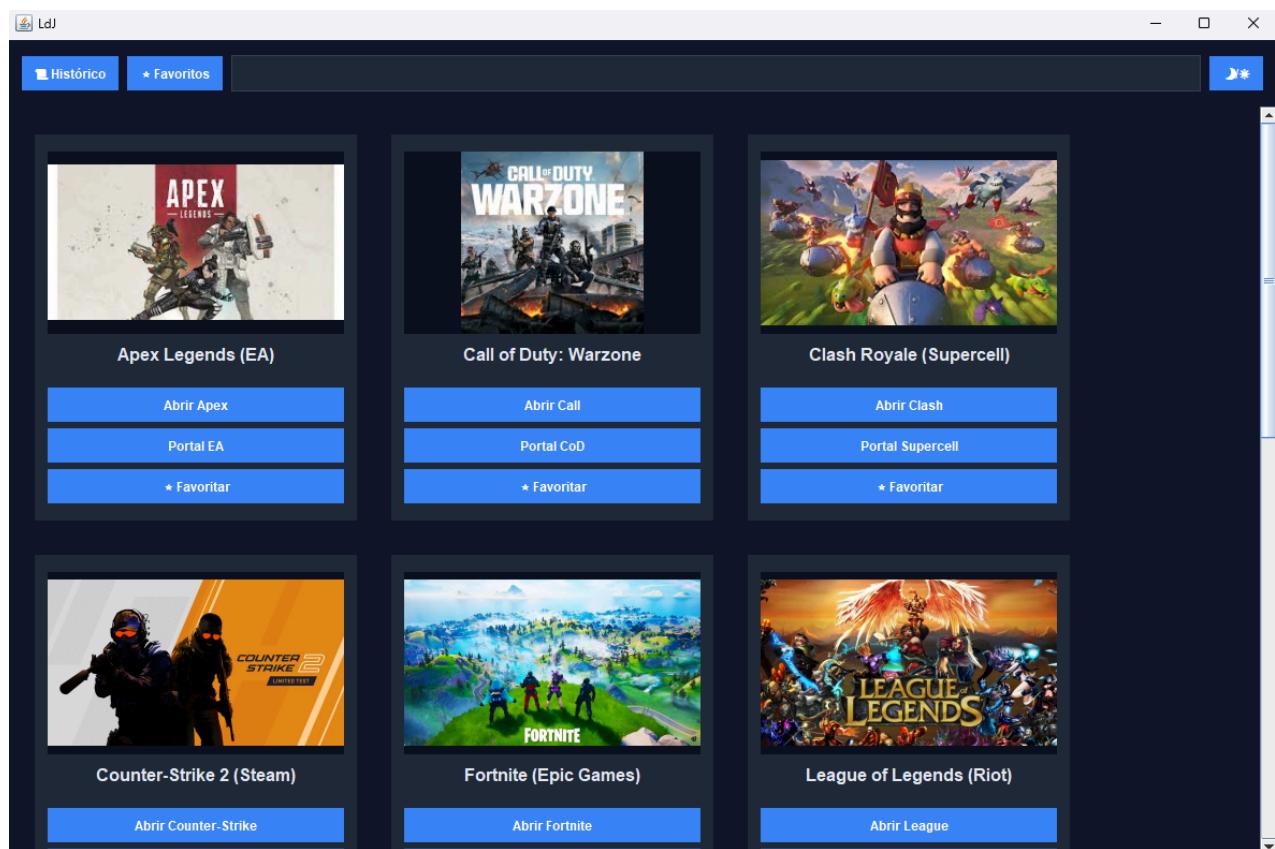
A tela de login recebe usuário e senha, válida via `AuthService`, e agora possui dois fluxos:

- Usuário comum → MainFrame
- Administrador → AdminFrame

Estrutura

- **JFrame**: janela principal da tela de login.
- **GridBagLayout**: usado para alinhar rótulos e campos verticalmente com espaçamentos.
- **JLabel**: textos como “Usuário” e “Senha”.
- **JTextField**: entrada de texto para usuário.
- **JPasswordField**: entrada oculta de senha.
- **JButton**: envia as credenciais.
- **ActionListener**: ação executada quando o usuário clica no botão "Entrar".
- **JOptionPane**: usado para alertas, erros e confirmações.
- **setDefaultCloseOperation**: define que o programa fecha ao fechar a janela.
- **SwingUtilities.invokeLater**: garante que a interface seja criada na Event Dispatch Thread.

Tela Principal – MainFrame



É o painel que mostra:

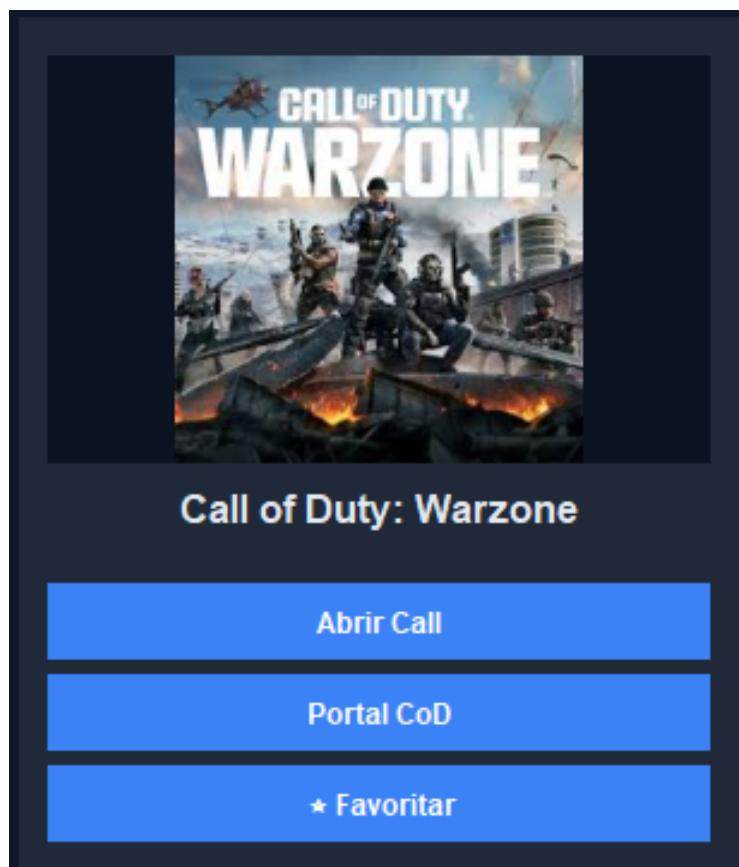
- Busca
- Botão de Tema
- Botão Favoritos

- Botão Histórico
- Cards dos jogos em grade 3×N
- **Jogos adicionados pelo admin também aparecem aqui**

Estrutura

- **JFrame**: janela principal do sistema.
- **BorderLayout**: divide a tela entre barra superior (NORTH) e área dos cards (CENTER).
- **JPanel**: usado para agrupar componentes (barra de busca, lista de cards, etc.).
- **GridBagLayout**: organiza a barra superior e serve como grade dos cards.
- **JScrollPane**: adiciona rolagem quando há muitos cards.
- **JTextField**: campo de busca, com **DocumentListener** para busca em tempo real.
- **JButton**: botões de ações (tema, favoritos, histórico).
- **ActionListener**: dispara ações dos botões.
- **Box.createGlue()**: usado na grade para manter os cards alinhados no topo.
- **SwingUtilities**: manipulação segura da UI.

Cards e Jogos — GameCard



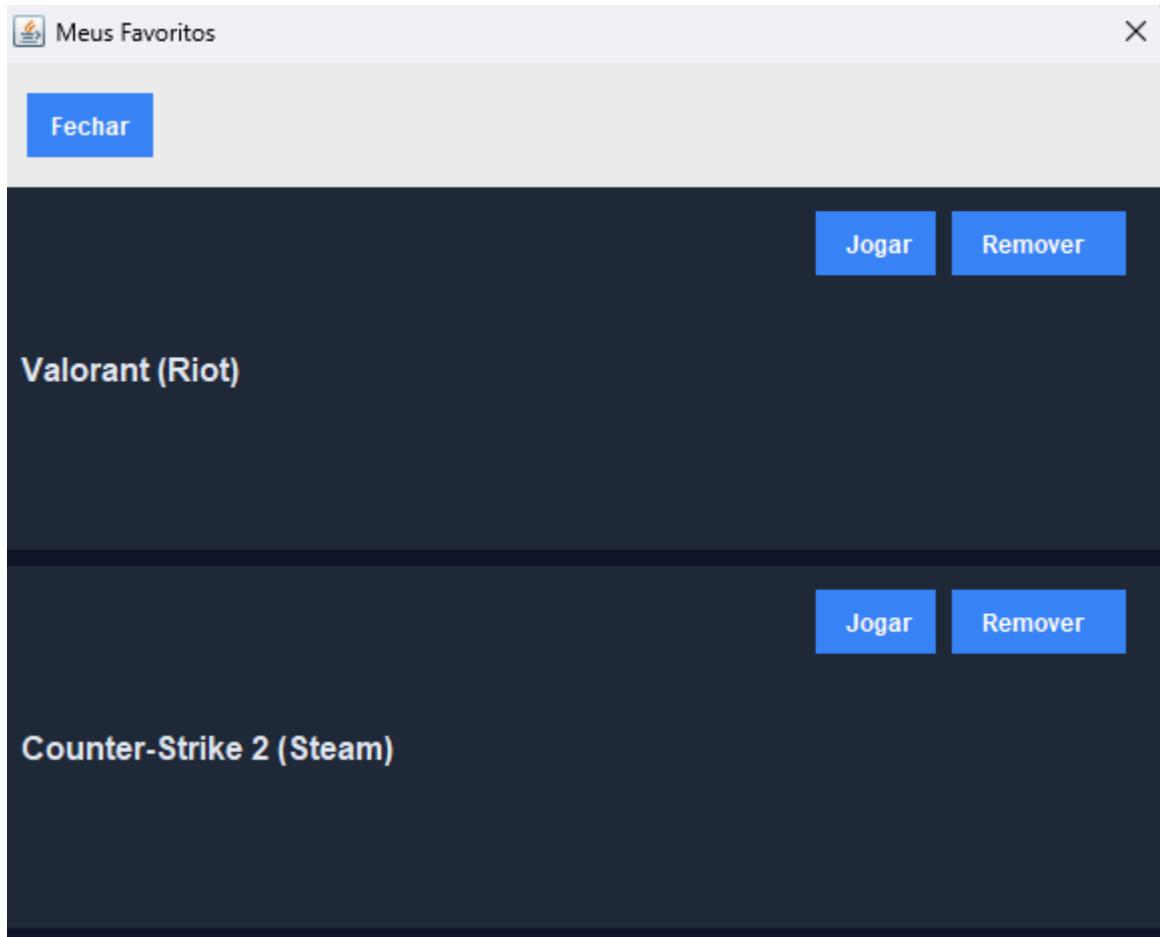
Cada card exibe:

- Imagem
- Título
- Botões
- Ações (abrir, portal, favoritar)
- Tema aplicado dinamicamente
- Carga da imagem **assíncrona**

Estrutura

- **JPanel**: representa cada card.
- **GridBagLayout**: organiza imagem, título e botões verticalmente.
- **JLabel**: mostra o título e a imagem (como **Icon**).
- **JButton**: botões Abrir, Portal, Favoritar.
- **ActionListener**: abre links, adiciona favorito, salva histórico.
- **SwingWorker**:
 - usado para carregar imagens pela internet **sem travar a UI**;
 - processa em background e atualiza o JLabel no **done()**.
- **ImageIcon**: exibe a imagem final no JLabel.
- **Graphics2D**: usado para desenhar a imagem reduzida com letterbox.
- **Desktop.getDesktop().browse()**: abre URLs.

Favoritos – FavoritesDialog



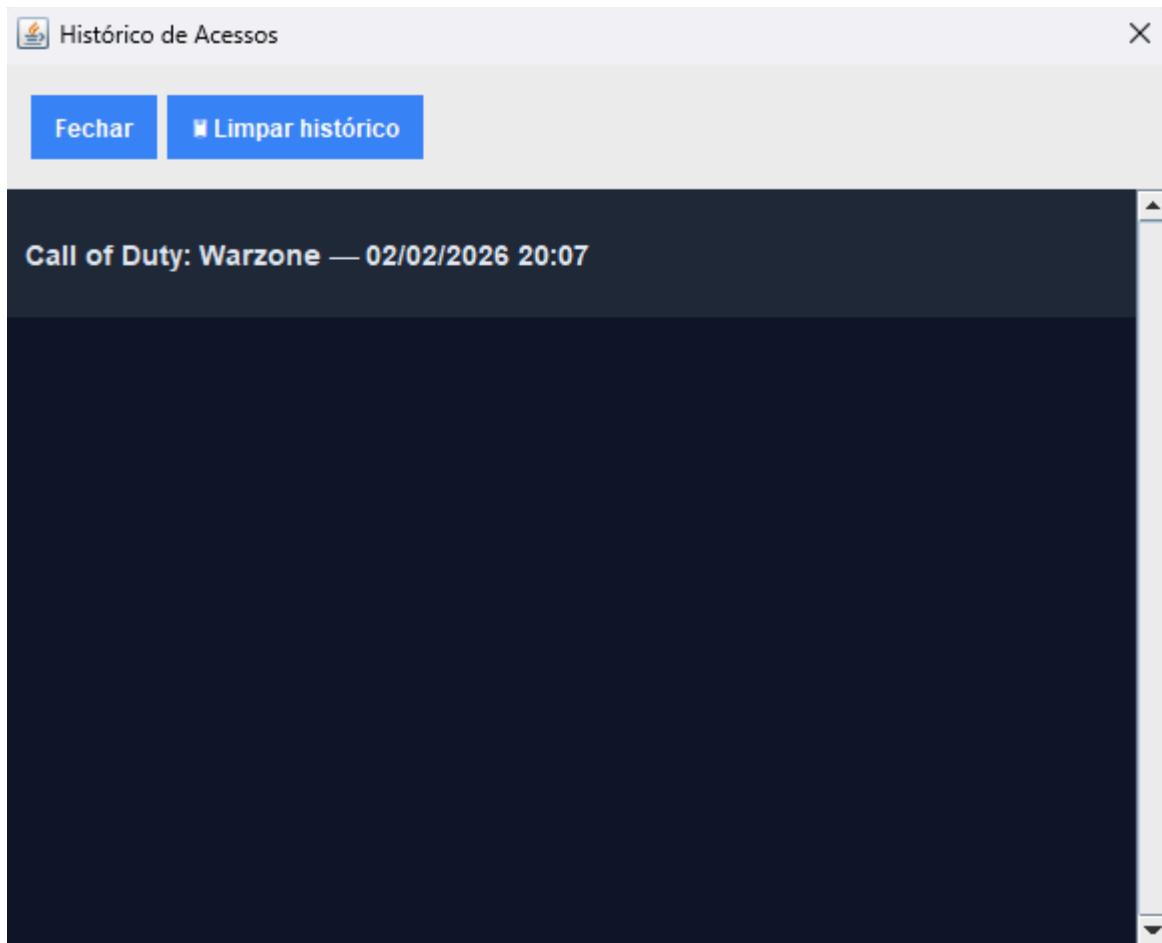
Permite visualizar, abrir e remover jogos favoritados.

Favoritos são persistentes via Preferences.

Estrutura

- **JDialog**: janela modal sobre o MainFrame.
- **BoxLayout**: posiciona cada favorito um abaixo do outro.
- **JScrollPane**: adiciona rolagem à lista.
- **JPanel**: representa cada linha da lista.
- **FlowLayout**: alinha os botões Play e Remover.
- **JButton**: executar ações.
- **ActionListener**: abre o jogo ou remove dos favoritos.
- **JOptionPane**: confirmações e alertas.

Histórico – HistoryDialog



Mostra todos os jogos abertos recentemente.

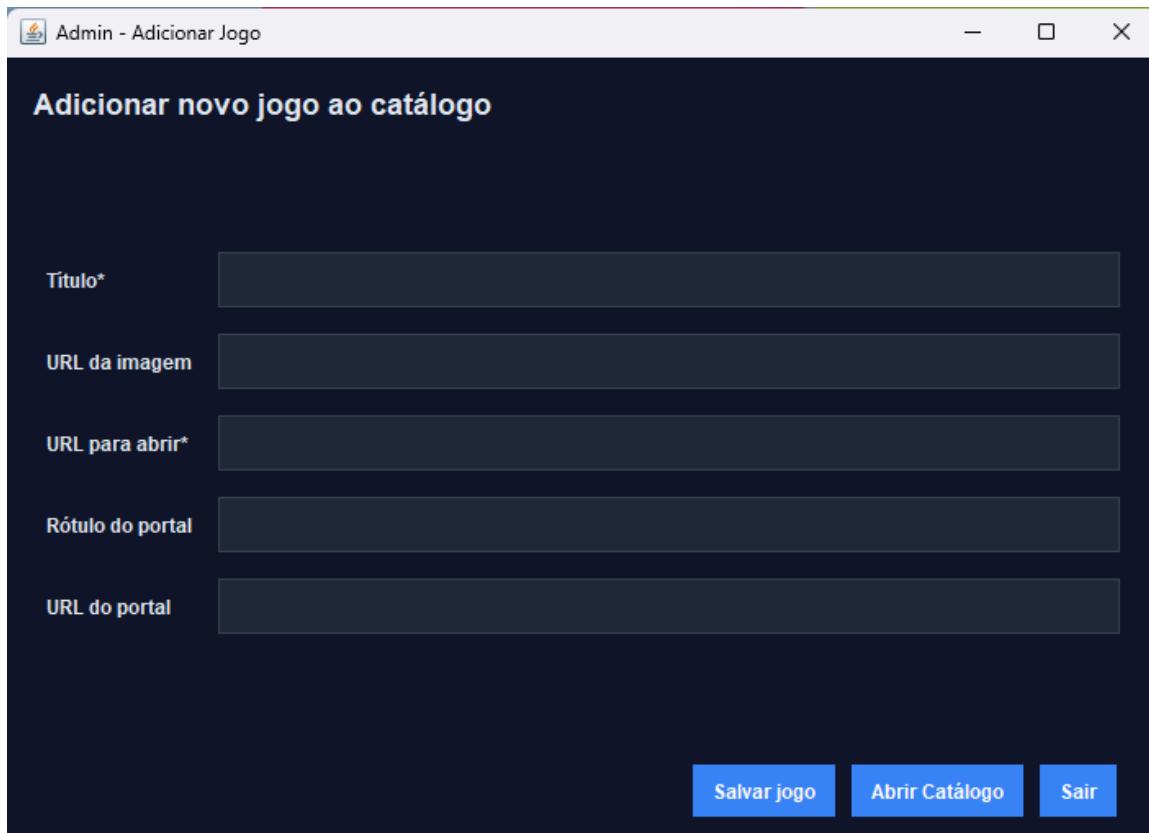
Agora possui:

- Rolagem vertical
- Linhas com altura fixa
- Botão “Limpar histórico” com confirmação

Estrutura

- **JDialog**: janela modal.
- **JScrollPane**: rolagem vertical sempre ativa.
- **BoxLayout**: organiza as entradas de histórico.
- **JPanel**: representa cada linha.
- **BorderLayout**: divide cada linha entre texto e margens.
- **JLabel**: mostra nome do jogo + horário.
- **JButton**: fechar / limpar histórico.
- **JOptionPane**: pede confirmação antes de apagar tudo.

Tela de Administração – AdminFrame



Tela usada apenas pelo administrador para cadastrar novos jogos:

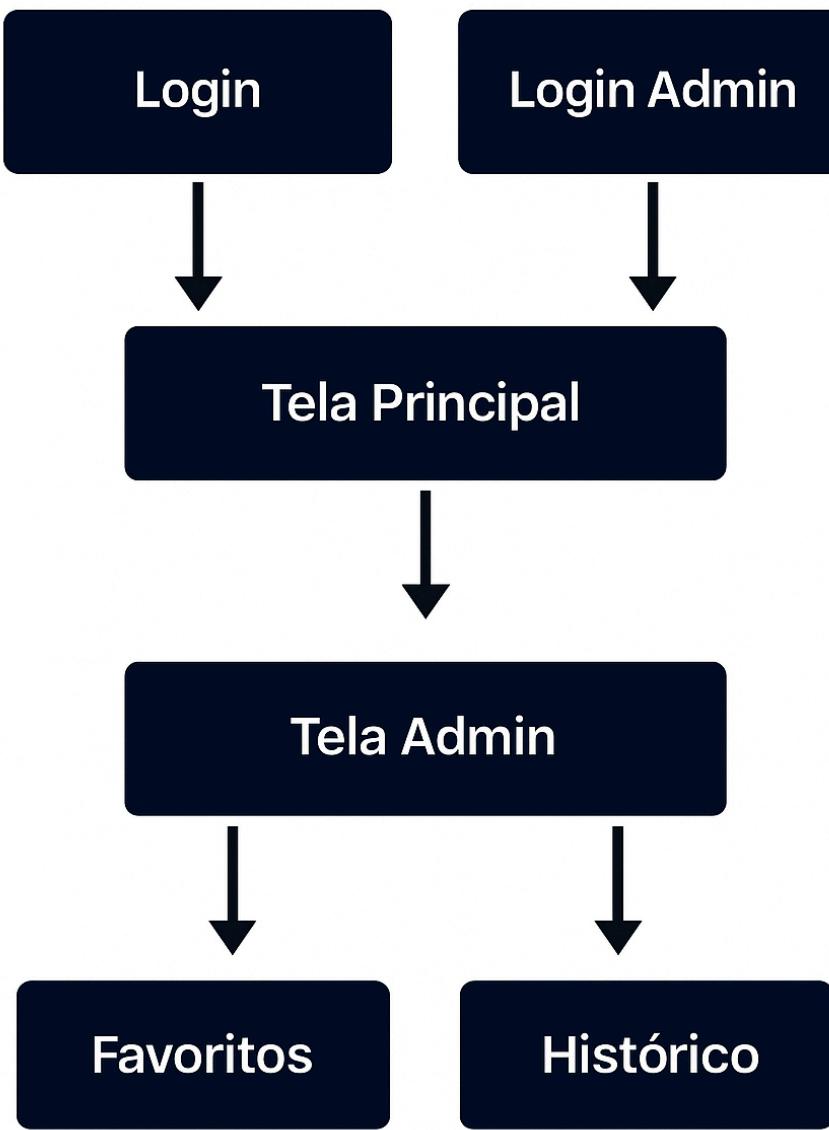
- Título
- Imagem
- URL de abertura
- Rótulo do portal
- URL do portal

Jogos cadastrados são adicionados ao catálogo automaticamente.

Estrutura

- **JFrame**: janela da área administrativa.
- **BorderLayout**: divide cabeçalho, formulário e rodapé.
- **GridBagLayout**: usado para montar o formulário de forma organizada e responsiva.
- **JTextField**: campos para entrada dos dados.
- **JLabel**: textos orientativos.
- **JButton**: ações (Salvar jogo, Abrir catálogo, Sair).
- **ActionListener**: valida dados e salva o jogo.
- **JOptionPane**: avisos, erros e confirmações.
- **applyThemeRecursive**: garante aplicação do tema a todos os componentes.

Fluxo geral do sistema



Conclusão

O LdJ continua sendo simples, rápido e eficaz, agora com:

- Login Admin
- Cadastro de jogos via interface
- Histórico aprimorado
- Persistência completa
- Cards inteligentes
- Interface Swing limpa e responsiva, com todas as propriedades aplicadas corretamente