

UM UNIVERSID

DIEGO HENRIQUE DE OLIVEIRA

RGM: 11241102717

TIAGO KUAN MELLO DURAN FERREIRA

RGM: 11241103205

NICOLY FERNANDES AMANCIO MARTINELI

RGM: 11241103083

ESPOLIO DIGITAL
MOGI DAS CRUZES
2025

MC

PROJETO FINAL: SISTEMA DE GESTÃO DE HERANÇA DIGITAL (ESPÓLIO DIGITAL)

**DIEGO HENRIQUE DE OLIVEIRA
TIAGO KUAN MELLO DURAM FERREIRA
NICOLY FERNANDES AMANCIO MARTINELI**

ESPOLIO DIGITAL

**Projeto Final (Atividade - M2) apresentado à Universidade
de Mogi das Cruzes (UMC), como requisito parcial para
obtenção de Grau acadêmico.**

**Área de Concentração: Análise e Desenvolvimento de sistemas
Matéria: Análise e Implementação de Software**

Orientador: Prof. Dr. Adilson Lima da Silva

MOGI DAS CRUZES 2025

SUMÁRIO

1. [LINK DO REPOSITÓRIO \(GITHUB\)](#)
2. [INTRODUÇÃO E OBJETIVO](#)
 - 2.1. Requisitos Funcionais (RF)
 - 2.2. Requisitos Não Funcionais (RNF)
3. [DESENHO E DESIGN DA SOLUÇÃO](#)
 - 3.1. Mapa de Navegação (Sitemap)
 - 3.2. Wireframes das Telas Principais
4. [MODELAGEM DE DADOS \(BANCO DE DADOS\)](#)
 - 4.1. Diagrama Entidade-Relacionamento (DER)
 - 4.2. Script SQL
 - 4.3. Evidências em todos os bancos de dados, com input e senhas Hash padrão de Segurança AES-256 juntamente às normas ISO 27001, 27701 ATIVAS.
 - 4.3.1 Banco de dados Espolio Digital
 - 4.3.2 Tabela Usuario
 - 4.3.3 Tabela Herdeiro
 - 4.3.4 Tabela AtivoDigital
 - 4.3.5 Tabela Solicitacao_Liberacao
5. [CÓDIGO FONTE \(NETBEANS\)](#)
 - **5.2. WebPages 1 – usuário**
 - 5.2.1 1 – tela geral.jsp
 - 5.2.2 2 – Tela Geral.jsp
 - 5.2.3 recuperar_senha.jsp
 - 5.2.4 sucesso_usuario.jsp
 - **5.3. WebPages 2 – ativodigital**
 - 5.3.1 1 – ativodigital.jsp
 - **5.4. WebPages 3 – herdeiro**
 - 5.4.1 1 – herdeiro.jsp
 - 5.4.2 sucesso_herdeiro.jsp
 - **5.5. WebPages imagens**
 - 5.5.1 login_cadastro.html
 - **5.6. Source Package Config**
 - 5.6.1 ConectaDB.java
 - 5.6.2 CryptoUtil.java
 - 5.6.3 HashUtil.java
 - **5.7. Source Package Controller**
 - 5.7.1 AtivoDigitalServlet.java
 - 5.7.2 CarregarFotoServlet.java
 - 5.7.3 EditarPerfilServlet.java
 - 5.7.4 HerdeiroServlet.java

- 5.7.5 LegadoServlet.java
- 5.7.6 LoginServlet.java
- 5.7.7 LogoutServlet.java
- **5.8. Source Package Model**
- 5.8.1 AtivoDigital.java
- 5.8.2 Herdeiro.java
- 5.8.3 LegadoAgrupado.java
- 5.8.4 LegadoDTO.java
- 5.8.5 SolicitudLiberacao.java
- 5.8.6 Usuario.java
- **5.9. Source Package ModelDAO**
- 5.9.1 AtivoDigitalDAO.java
- 5.9.2 HerdeiroDAO.java
- 5.9.3 SolicitudDAO.java
- 5.9.4 UsuarioDAO.java

6. EVIDÊNCIAS DO SISTEMA EM EXECUÇÃO (PRINTS)

- Figura 1.1: Cadastro de Usuário
- Figura 1.2: Recuperação de Senha por Palavra-Chave de Recuperação
- Figura 1.3: Tela de Login / Recuperação de Senha do Usuário / Criar Conta
- Figura 2.1: Dashboard Principal
- Figura 2.2: Editar Perfil
- Figura 3.1: Gerenciar Meus herdeiros
- Figura 3.2: Gerenciar Meus herdeiros
- Figura 3.3: Gerenciar Meus herdeiros
- Figura 4.1: Conectar Contas
- Figura 4.2: Conectar Contas
- Figura 4.3: Conectar Contas
- Figura 5.1: Meus Legados Recebidos
- Figura 5.2: Meus Legados Recebidos
- Figura 5.3: Meus Legados Recebidos

7. BIBLIOTECAS E AMBIENTE DE DESENVOLVIMENTO

8. REFERÊNCIAS BIBLIOGRÁFICAS

1. LINK DO REPOSITÓRIO (GITHUB)

O código-fonte completo e versionado deste projeto encontra-se disponível no seguinte repositório público:

<https://github.com/tiagomello06/espolio-digital>

2. INTRODUÇÃO E OBJETIVO

O **Espólio Digital** é um sistema web desenvolvido para solucionar o problema da gestão de ativos digitais post-mortem. A aplicação permite que titulares cadastrem credenciais de redes sociais, carteiras de criptomoedas e mensagens póstumas, designando herdeiros específicos. A segurança é garantida pelo bloqueio dos dados sensíveis até que o herdeiro comprove o óbito do titular mediante upload de Certidão de Óbito (PDF), validada pelo sistema.

2.1. Requisitos Funcionais (RF)

Para garantir o correto funcionamento do sistema, foram definidos os seguintes requisitos funcionais:

- [RF-001] Cadastro de Ativos Digitais: O sistema deve permitir que o Usuário (Titular) cadastre ativos digitais (como credenciais de redes sociais, carteiras de criptomoedas e mensagens póstumas), definindo login, senha, frases de recuperação ou mensagens personalizadas.
- [RF-002] Designação de Herdeiros: O sistema deve permitir que o Usuário vincule um ou mais ativos a um Herdeiro específico, utilizando o CPF como identificador único.
- [RF-003] Visualização de Legados: O sistema deve permitir que o Herdeiro visualize uma lista de ativos deixados para ele, agrupados por Titular.
- [RF-004] Proteção de Dados Sensíveis: O sistema deve ocultar dados sensíveis (ex: senhas, chaves privadas) com a marcação "Conteúdo Protegido" até que a liberação seja aprovada.
- [RF-005] Solicitação de Liberação via Upload: O sistema deve permitir que o Herdeiro solicite o desbloqueio dos ativos mediante o upload de um arquivo de Certidão de Óbito (PDF) através de uma janela modal.
- [RF-006] Feedback Visual de Upload: O sistema deve fornecer feedback visual imediato para confirmar o anexo bem-sucedido do arquivo antes do envio da solicitação.
- [RF-007] Agrupamento Inteligente de Solicitações: O sistema deve agrupar as solicitações de liberação por Titular. Novos ativos adicionados para um herdeiro que já tenha uma solicitação "APROVADA" devem ser cadastrados como "BLOQUEADO", reabilitando o botão de solicitação apenas para os novos itens.
- [RF-008] Prevenção de Duplicidade de Solicitações: O sistema não deve permitir a criação de solicitações repetidas para um mesmo ativo que já possua um status "PENDENTE".

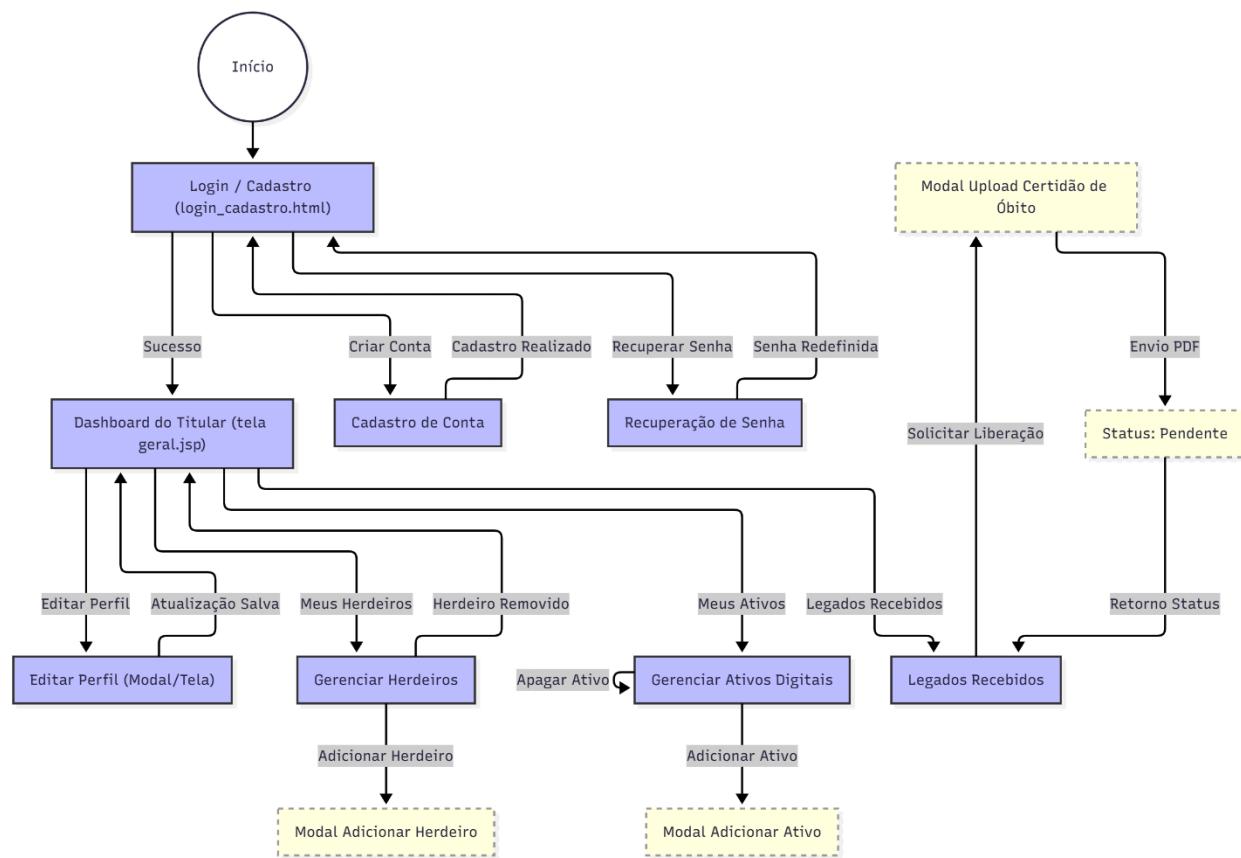
2.2. Requisitos Não Funcionais (RNF)

Para garantir a qualidade, segurança e desempenho do sistema, foram estabelecidos os seguintes requisitos não funcionais:

- [RNF-001] Formato de Arquivo para Upload: O sistema deve restringir o upload de documentos em formato PDF.
- [RNF-002] Persistência de Arquivos: Os documentos de Certidão de Óbito devem ser persistidos como dados binários (LONGBLOB) diretamente no banco de dados, e não no sistema de arquivos do servidor.
- [RNF-003] Segurança de Credenciais: As senhas dos usuários devem ser armazenadas no banco de dados utilizando criptografia de Hash SHA-256.
- [RNF-004] Interface Responsiva: A interface do usuário deve ser desenvolvida com o framework Tailwind CSS para garantir uma experiência responsiva e adaptável a diferentes tamanhos de tela (abordagem Mobile-First).
- [RNF-005] Arquitetura do Backend: O backend do sistema deve ser desenvolvido em Java, utilizando Servlets e JSP (JavaServer Pages) seguindo a arquitetura MVC, sem o uso de frameworks pesados como Spring, visando clareza didática e performance leve.

3. DESENHO E DESIGN DA SOLUÇÃO

3.1. Mapa de Navegação (Sitemap)



O fluxo do usuário foi desenhado utilizando a notação de fluxograma para garantir a naveabilidade intuitiva.

Login[Login / Cadastro]

Login -->|Sucesso| Dashboard[Dashboard do Usuário]

"Área do Titular"

Dashboard --> MeusHerdeiros[Gerenciar Herdeiros]

Dashboard --> MeusAtivos[Gerenciar Ativos]

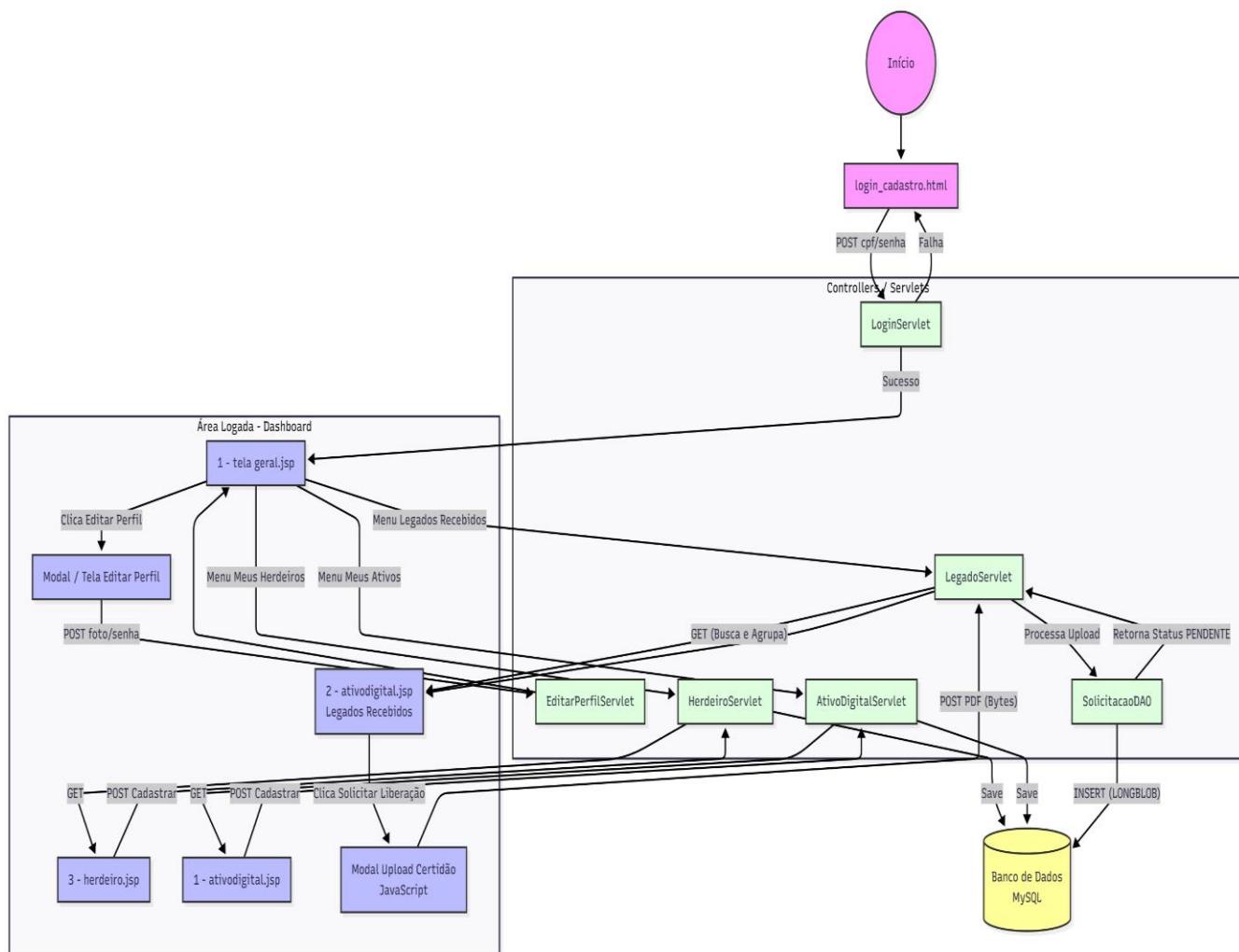
Dashboard --> EditarPerfil[Editar Foto/Senha]

"Área do Herdeiro"

Dashboard --> Legados[Legados Recebidos]

Legados -->|Status: Bloqueado| Modal[Modal Upload Certidão]

Modal -->|Envio PDF| Analise[Status: Pendente]



3.2. Wireframes das Telas Principais

O design da interface foi projetado com foco na usabilidade e responsividade (Mobile-First), utilizando o framework TailwindCSS.

Figura 1: Wireframe da Tela de Legados Recebidos (Visão do Herdeiro)

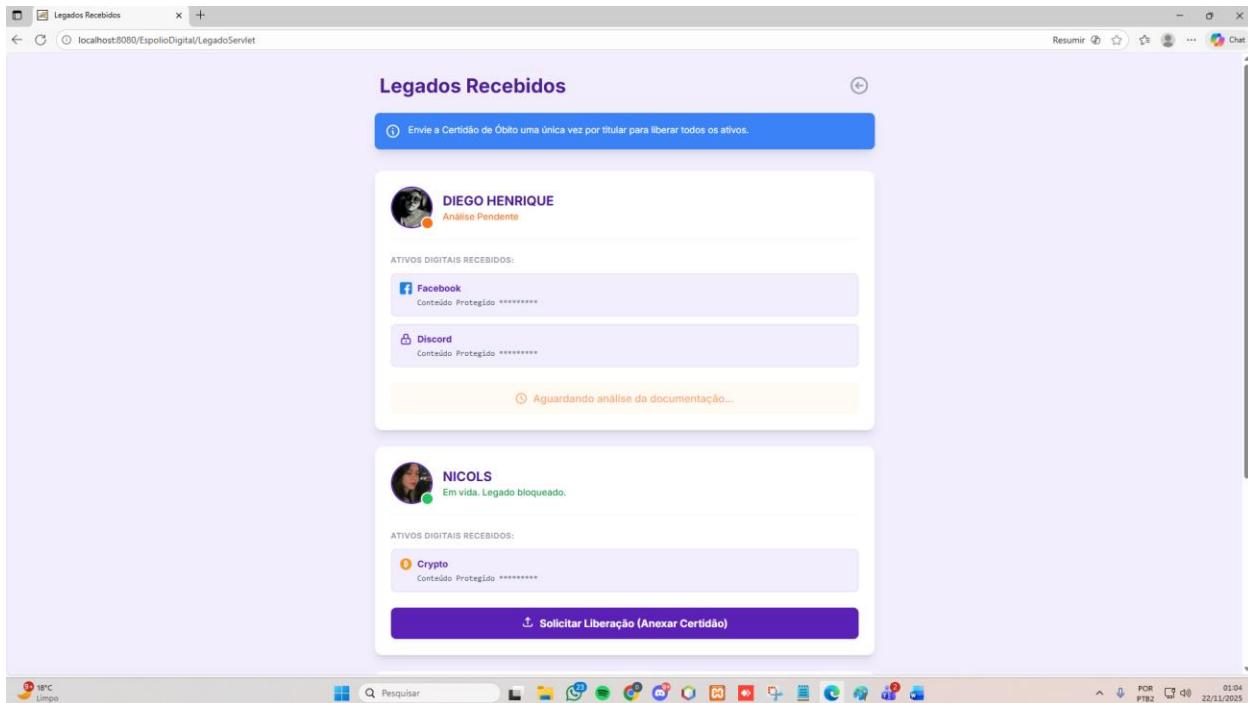
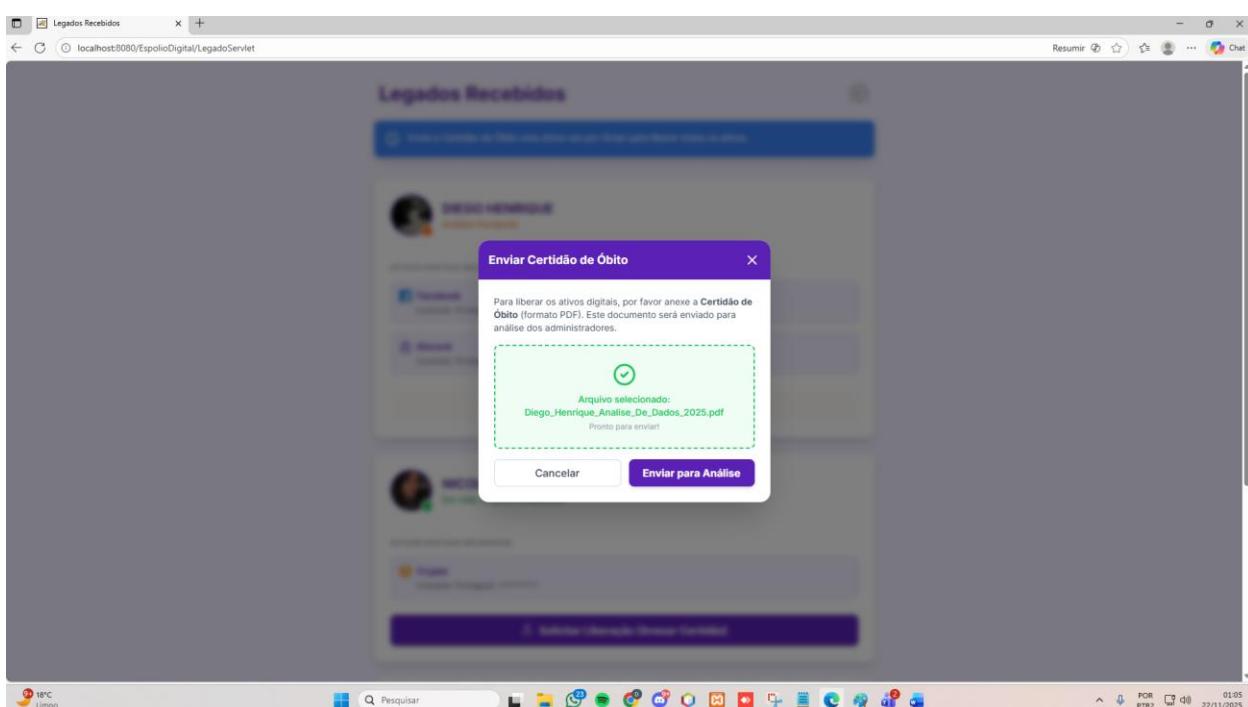


Figura 2: Wireframe do Modal de Upload



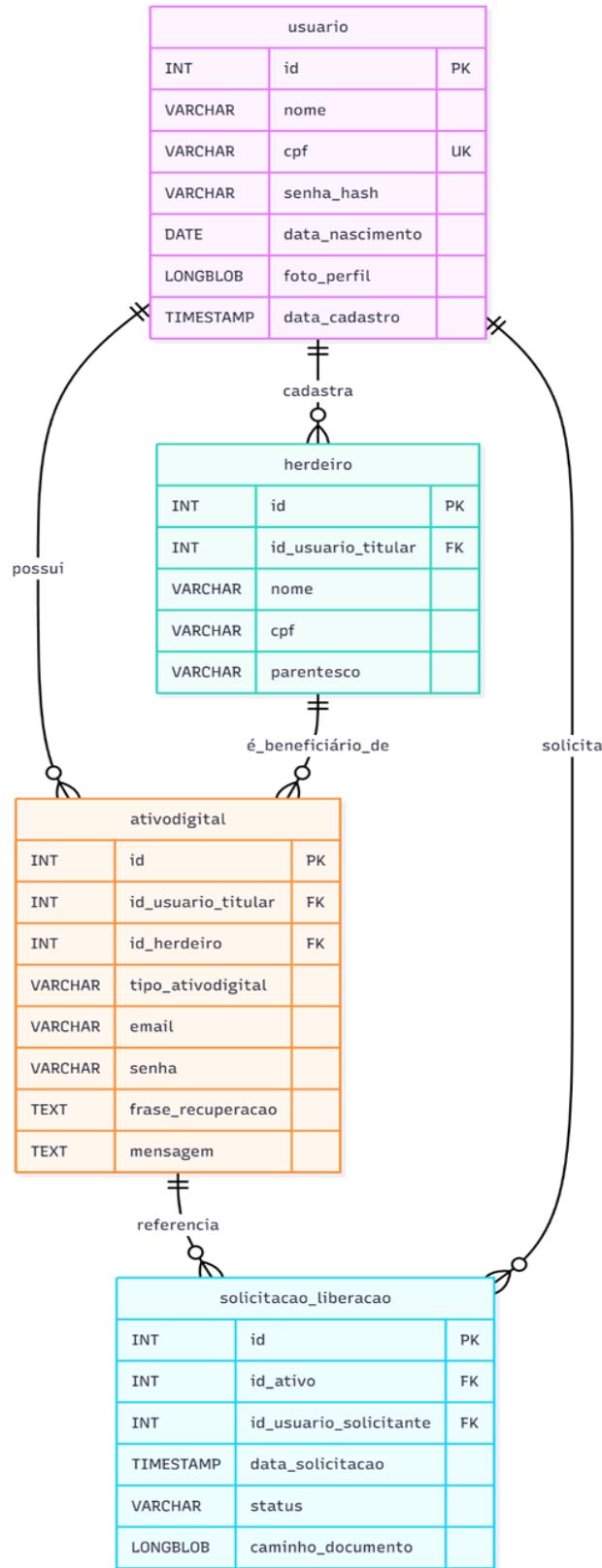
4. MODELAGEM DE DADOS (BANCO DE DADOS)

4.1. Diagrama Entidade-Relacionamento (DER)

```

1 erDiagram
2   usuario {
3     INT id PK
4     VARCHAR nome
5     VARCHAR cpf UK
6     VARCHAR senha_hash
7     DATE data_nascimento
8     LONGBLOB foto_perfil
9     TIMESTAMP data_cadastro
10  }
11
12   herdeiro {
13     INT id PK
14     INT id_usuario_titular FK
15     VARCHAR nome
16     VARCHAR cpf
17     VARCHAR parentesco
18  }
19
20   ativodigital {
21     INT id PK
22     INT id_usuario_titular FK
23     INT id_herdeiro FK
24     VARCHAR tipo_ativodigital
25     VARCHAR email
26     VARCHAR senha
27     TEXT frase_recuperacao
28     TEXT mensagem
29  }
30
31   solicitacao_liberacao {
32     INT id PK
33     INT id_ativo FK
34     INT id_usuario_solicitante FK
35     TIMESTAMP data_solicitacao
36     VARCHAR status
37     LONGBLOB caminho_documento
38  }
39
40   %% Relacionamentos
41   usuario ||--o{ herdeiro : "cadastra"
42   usuario ||--o{ ativodigital : "possui"
43   herdeiro ||--o{ ativodigital : "é_beneficiario_de"
44   ativodigital ||--o{ solicitacao_liberacao : "referencia"
45   usuario ||--o{ solicitacao_liberacao : "solicita"

```



4.2. Script SQL

```
1 -- Palavras-chave em AZUL
2 CREATE DATABASE IF NOT EXISTS espoliodigital;
3 USE espoliodigital;
4
5 -- Tabela de Usuários
6 CREATE TABLE usuario (
7     id INT AUTO_INCREMENT PRIMARY KEY,
8     nome VARCHAR(100) NOT NULL,
9     cpf VARCHAR(14) NOT NULL UNIQUE,
10    email VARCHAR(100) NOT NULL UNIQUE,
11    senha_hash VARCHAR(255) NOT NULL,
12    data_nascimento DATE,
13    foto_perfil LONGBLOB,
14    data_cadastro TIMESTAMP DEFAULT CURRENT_TIMESTAMP
15 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
16
17 -- Nomes de tabelas em VERDE
18 CREATE TABLE herdeiro (
19     id INT AUTO_INCREMENT PRIMARY KEY,
20     id_usuario_titular INT NOT NULL,
21     nome VARCHAR(100) NOT NULL,
22     cpf VARCHAR(14) NOT NULL,
23     parentesco VARCHAR(50),
24     -- Constraints em LARANJA
25     FOREIGN KEY (id_usuario_titular) REFERENCES usuario(id) ON DELETE CASCADE
26 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
27
28
29 -- Tipos de dados em ROXO
30 CREATE TABLE ativodigital (
31     id INT AUTO_INCREMENT PRIMARY KEY,
32     id_usuario_titular INT NOT NULL,
33     id_herdeiro INT NOT NULL,
34     tipo_ativodigital VARCHAR(50) NOT NULL,
35     login VARCHAR(255),
36     senha VARCHAR(255),
37     frase_recuperacao TEXT,
38     mensagem TEXT,
39     FOREIGN KEY (id_usuario_titular) REFERENCES usuario(id) ON DELETE CASCADE,
40     FOREIGN KEY (id_herdeiro) REFERENCES herdeiro(id) ON DELETE CASCADE
41 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
42
43 CREATE TABLE solicitacao_liberacao (
44     id INT AUTO_INCREMENT PRIMARY KEY,
45     id_ativo INT NOT NULL,
46     id_usuario_solicitante INT NOT NULL,
47     data_solicitacao TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
48     status VARCHAR(20) DEFAULT 'PENDENTE',
49     caminho_documento LONGBLOB,
50     FOREIGN KEY (id_ativo) REFERENCES ativodigital(id) ON DELETE CASCADE,
51     FOREIGN KEY (id_usuario_solicitante) REFERENCES usuario(id) ON DELETE CASCADE
52 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

4.3. Evidências em todos os bancos de dados, com input e senhas

Hash padrão de Segurança AES-256 juntamente às normas ISO 27001, 27701 ATIVAS.

4.3.1 Banco de dados Espolio Digital:

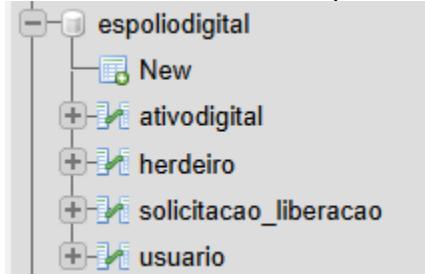


Table	Action	Rows	Type	Collation	Size	Overhead
ativodigital	Browse Structure Search Insert Drop	8	InnoDB	utf8mb4_general_ci	48.0 KIB	-
herdeiro	Browse Structure Search Insert Drop	6	InnoDB	utf8mb4_general_ci	16.0 KIB	-
solicitacao_liberacao	Browse Structure Search Insert Drop	2	InnoDB	utf8mb4_general_ci	1.5 MiB	-
usuario	Browse Structure Search Insert Drop	6	InnoDB	utf8mb4_general_ci	1.4 MiB	-
4 tables	Sum	22	InnoDB	utf8mb4_general_ci	3.0 MiB	0 B

4.3.2 Tabela Usuario:

	Edit Copy Delete	id	nome	cpf <small>Apenas os 11 números</small>	senha_hash	palavra_chave_hash	foto_perfil
1	Edit Copy Delete	1	DIEGO HENRIQUE	12121212121	240be518fabd2724ddb6f04eeb1da5967448d7e831c08c8fa8...	dc436b3294da6c880b6ce79d829ac4fc33746b454604e2bc...	[BLOB - 4.1 KiB]
2	Edit Copy Delete	2	TIAGO KAUVAN	13131313131	921443c5e72aa9f9103216252095ed5ed04ab8deec8ac0deb...	dc436b3294da6c880b6ce79d829ac4fc33746b454604e2bc...	[BLOB - 93.9 KiB]
3	Edit Copy Delete	3	NICOLY	14141414141	240be518fabd2724ddb6f04eeb1da5967448d7e831c08c8fa8...	8db59fe4d21726c79d6e76eaaf0398eb823e376bb783...	[BLOB - 62.9 KiB]
4	Edit Copy Delete	4	NICOLS	15151515151	240be518fabd2724ddb6f04eeb1da5967448d7e831c08c8fa8...	8db59fe4d21726c79d6e76eaaf0398eb823e376bb783...	[BLOB - 291.6 KiB]
5	Edit Copy Delete	5	Mayara dos Santos Maia	17171717171	240be518fabd2724ddb6f04eeb1da5967448d7e831c08c8fa8...	8db59fe4d21726c79d6e76eaaf0398eb823e376bb783...	[BLOB - 291.6 KiB]
6	Edit Copy Delete	6	Teste	212121212	240be518fabd2724ddb6f04eeb1da5967448d7e831c08c8fa8...	8db59fe4d21726c79d6e76eaaf0398eb823e376bb783...	NULL

4.3.3 Tabela Herdeiro:

Showing rows 0 - 5 (6 total, Query took 0.0002 seconds.)

SELECT * FROM `herdeiro`

	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	<input type="checkbox"/> id	<input type="checkbox"/> id_usuario_titular	<input type="checkbox"/> nome	<input type="checkbox"/> cpf	<input type="checkbox"/> parentesco
1	Edit	Copy	Delete	1	1	TIAGO	13131313131	AMIGO
2	Edit	Copy	Delete	2	1	NICOLY	15151515151	amiga
3	Edit	Copy	Delete	3	5	Tiago Amor	13131313131	Namorado
4	Edit	Copy	Delete	4	2	Diego Facul	12121212121	Amigo
5	Edit	Copy	Delete	5	2	Mayara Amor	17171717171	Amor
8	Edit	Copy	Delete	8	4	Tiago Facul	13131313131	Amigo

4.3.4 Tabela AtivoDigital:

Showing rows 0 - 7 (8 total, Query took 0.0001 seconds.)

SELECT * FROM `ativodigital`

	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	<input type="checkbox"/> id	<input type="checkbox"/> id_usuario_titular	<input type="checkbox"/> id_herdeiro	<input type="checkbox"/> tipo_ativodigital	<input type="checkbox"/> login	<input type="checkbox"/> senha	<input type="checkbox"/> frase_recuperacao	<input type="checkbox"/> mensagem	<input type="checkbox"/> data_criacao
6	Edit	Copy	Delete	1	1	facebook	1	123@123	4f90huLLhJL1g4VNA82Vw==	NULL	NULL	2025-11-21 22:13:43
7	Edit	Copy	Delete	7	1	twitter	2	123@123	4f90huLLhJL1g4VNA82Vw==	NULL	NULL	2025-11-21 22:14:19
8	Edit	Copy	Delete	8	5	instagram	3	123@123	4f90huLLhJL1g4VNA82Vw==	NULL	NULL	2025-11-21 22:39:39
9	Edit	Copy	Delete	9	5	facebook	3	123@123	4f90huLLhJL1g4VNA82Vw==	NULL	NULL	2025-11-21 22:39:51
10	Edit	Copy	Delete	10	2	instagram	5	123@123	4f90huLLhJL1g4VNA82Vw==	NULL	NULL	2025-11-21 23:28:18
11	Edit	Copy	Delete	11	2	crypto	4	123@123	4f90huLLhJL1g4VNA82Vw==	NULL	qtakW3q1K9DXsbIKo0KBRTM1VVHpmysqIK14Jen+ZqY=	2025-11-21 23:28:42
13	Edit	Copy	Delete	13	1	discord	1	123@123	4f90huLLhJL1g4VNA82Vw==	NULL	NULL	2025-11-22 01:01:31
14	Edit	Copy	Delete	14	4	crypto	8	123@123	4f90huLLhJL1g4VNA82Vw==	NULL	UqtQQ8eQ2whzg192C4qaw==	2025-11-22 01:04:02

4.3.5 Tabela Solicitacao_Liberacao:

Showing rows 0 - 1 (2 total, Query took 0.0203 seconds.)

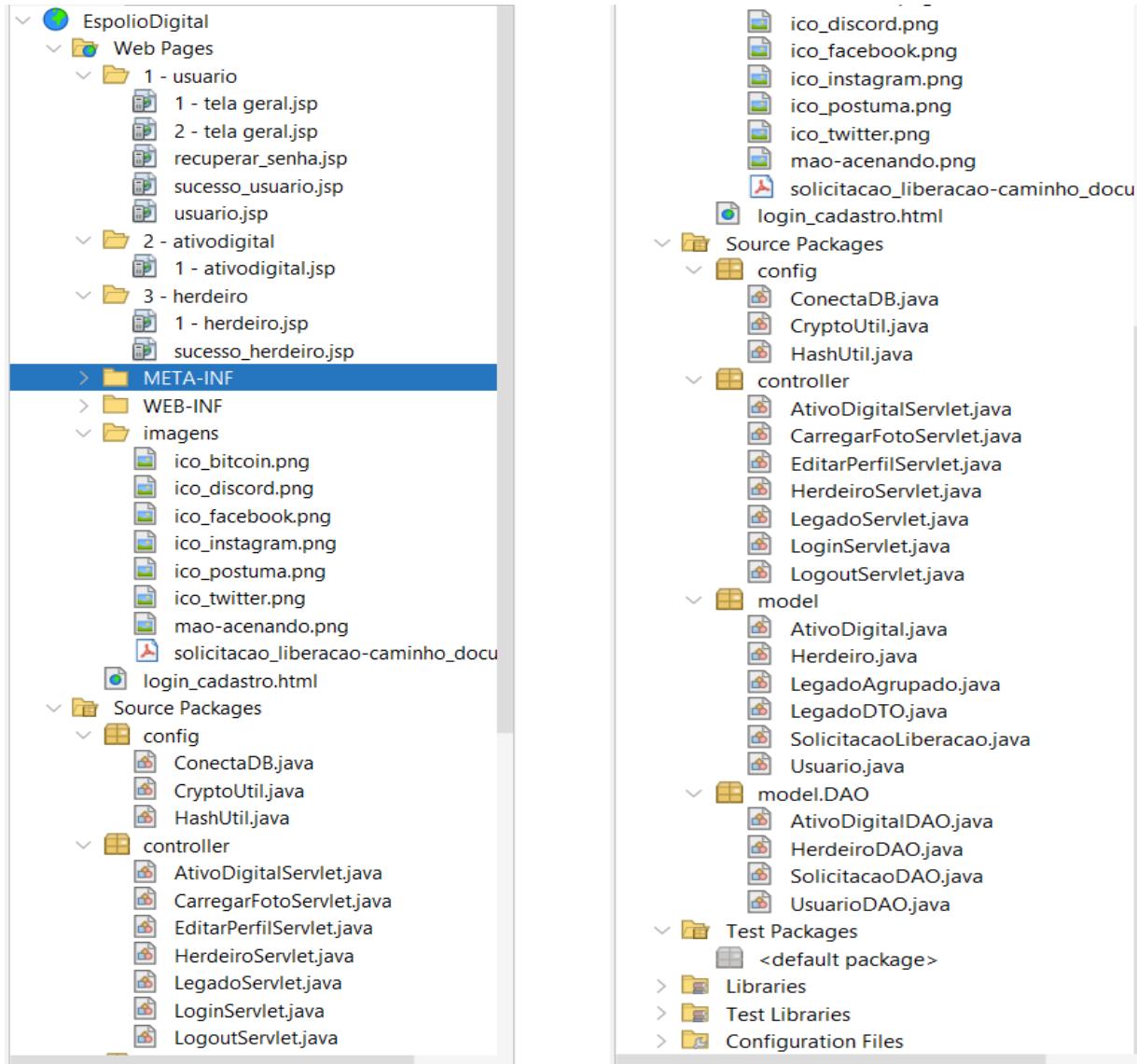
SELECT * FROM `solicitacao_liberacao`

	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	<input type="checkbox"/> id	<input type="checkbox"/> id_ativo	<input type="checkbox"/> id_usuario_solicitante	<input type="checkbox"/> data_solicitacao	<input type="checkbox"/> status	<input type="checkbox"/> caminho_documento
1	Edit	Copy	Delete	1	6	2	2025-11-22 00:07:32	PENDENTE	[BLOB - 502.2 KB]
2	Edit	Copy	Delete	2	14	2	2025-11-22 01:44:18	PENDENTE	[BLOB - 18.7 KB]

5. CÓDIGO FONTE (NETBEANS)

O projeto foi desenvolvido em Java Web utilizando a arquitetura MVC (Model-View-Controller). Abaixo, as classes principais.

5.1. Estrutura do Projeto



- **View:** Web Pages/ (JSP, HTML, CSS)
- **Controller:** Source Packages/controller (Servlets)
- **Model:** Source Packages/model (Beans)
- **DAO:** Source Packages/model/DAO (Acesso a Dados)
- **Config:** Source Packages/config (Conexão e Criptografia)

5.2. WebPages 1 – usuário

5.2.1 1 – tela geral.jsp

The screenshot shows a Java IDE interface with two tabs open, both titled "1 - tela geral.jsp".

Top Tab (Lines 1-27):

```
<%--  
1 Document : 1-telageral  
2 Created on : 22 de nov. de 2025, 11:31:04  
3 Author : TIAGO KAUAN , DIEGO HENRIQUE , NICOLY MARTINELI  
4 --%>  
5  
6 <%@page contentType="text/html" pageEncoding="UTF-8"%>  
7 <%@page import="model.Usuario"%>  
8 <%@page import="model.DAO.HerdeiroDAO"%>  
9 <%@page import="model.DAO.AtivoDigitalDAO"%>  
10 <%@page import="model.Herdeiro"%>  
11 <%@page import="model.AtivoDigital"%>  
12 <%@page import="java.util.List"%>  
13  
14  
15 <%-- -----  
16 <%-- ----- INÍCIO DO BLOCO DE SEGURANÇA E SESSÃO ----- --%>  
17 <%-- -----  
18 <%  
19 Usuario usuarioLogado = (Usuario) session.getAttribute("usuarioLogado");  
20 String nomeUsuario = "Visitante";  
21 boolean temFoto = false;  
22 int quantidadeHerdeiros = 0;  
23 int quantidadeAtivosDigitais = 0;  
24 List<Herdeiro> listaH = null;  
25 List<AtivoDigital> listaAD = null;  
26  
27 if (usuarioLogado == null) {
```

Bottom Tab (Lines 25-51):

```
25 List<AtivoDigital> listaAD = null;  
26  
27 if (usuarioLogado == null) {  
28 response.sendRedirect("../login_cadastro.html?loginErro=true");  
29 return;  
30 } else {  
31 nomeUsuario = usuarioLogado.getNome().split(" ")[0];  
32 if (usuarioLogado.getFotoPerfil() != null & usuarioLogado.getFotoPerfil().length > 0) {  
33 temFoto = true;  
34 }  
35 HerdeiroDAO hDao = new HerdeiroDAO();  
36 listaH = hDao.listarPorUsuario(usuarioLogado.getId());  
37 quantidadeHerdeiros = listaH.size();  
38  
39 AtivoDigitalDAO adDao = new AtivoDigitalDAO();  
40 listaAD = adDao.listarAtivosDigitaisDoTitular(usuarioLogado.getId());  
41 quantidadeAtivosDigitais = listaAD.size();  
42 }  
43 %>  
44 <%-- -----  
45 <%-- ----- FIM DO BLOCO DE SEGURANÇA E SESSÃO ----- --%>  
46 <%-- -----  
47  
48 <!DOCTYPE html>  
49 <html lang="pt-br">  
50 <head>  
51 <meta charset="UTF-8">
```

1 - tela geral.jsp

```
Source History < > ▾
46 <%-- -->
47
48     <!DOCTYPE html>
49     <html lang="pt-br">
50         <head>
51             <meta charset="UTF-8">
52             <meta name="viewport" content="width=device-width, initial-scale=1.0">
53             <title>Dashboard Principal</title>
54             <script src="https://cdn.tailwindcss.com"></script>
55             <link href="https://fonts.googleapis.com/css2?family=Inter:wght@400;500;700;800&display=swap" rel="stylesheet">
56         <style>
57             body { font-family: 'Inter', sans-serif; }
58             @keyframes wave {
59                 0%, 100% { transform: rotate(0deg); }
60                 25% { transform: rotate(15deg); }
61                 75% { transform: rotate(-15deg); }
62             }
63             .wave {
64                 display: inline-block;
65                 animation: wave 1s infinite;
66                 transform-origin: 70% 70%;
67             }
68         </style>
69         <script>
70             tailwind.config = {
71                 theme: {
72                     extend: {
```

1 - tela geral.jsp

```
Source History < > ▾
70             tailwind.config = {
71                 theme: {
72                     extend: {
73                         colors: {
74                             'brand-purple-light': '#f2eefe',
75                             'brand-purple': '#5b21b6',
76                             'brand-purple-medium': '#a78bfa',
77                             'brand-purple-faded': '#fede9fe',
78                             'brand-purple-text': '#4c1d95',
79                             'brand-green': '#22c55e',
80                             'brand-blue': '#3b82f6',
81                         }
82                     }
83                 }
84             }
85         </script>
86     </head>
87     <body class="bg-brand-purple-light min-h-screen">
88
89         <main class="flex items-start justify-center w-full p-4 py-8">
90
91             <div class="w-full max-w-lg md:max-w-3xl mx-auto bg-transparent rounded-2xl">
92
93                 <header class="flex justify-between items-center mb-6 px-2">
94                     <div class="flex items-center gap-4">
95                         <% if (temFoto) { %>
```

1 - tela geral.jsp

```
94 <div class="flex items-center gap-4">
95     <% if (temFoto) { %>
96         
97     <% } else { %>
98         <span class="flex items-center justify-center w-16 h-16 rounded-full bg-brand-purple-faded border-2 border-brand-purple">
99             <svg xmlns="http://www.w3.org/2000/svg" class="h-8 w-8 text-brand-purple" viewBox="0 0 20 20" fill="currentColor">
100                 <path fill-rule="evenodd" d="M10 9a3 3 0 100-6 3 3 0 000 6zm-7 9a7 7 0 1114 0H3z" clip-rule="evenodd" />
101         </span>
102     <% } %>
103     <h1 class="text-3xl font-bold text-brand-purple-text">
104         Olá, <%= nomeUsuario %>
105         <span class="wave"></span>
106     </h1>
107 </div>
108 <div class="flex items-center gap-3">
109     <a href="#" class="text-gray-400 hover:text-brand-purple transition-colors" title="Configurações">
110         <svg xmlns="http://www.w3.org/2000/svg" class="h-8 w-8" fill="none" viewBox="0 0 24 24" stroke="currentColor" stroke-linecap="round" stroke-linejoin="round" d="M10.325 4.317c.426-.1756 2.924-1.756 3.35 0a1.724 1.724 0 001.724 1.724" />
111     </a>
112     <a href="#" title="Sair" class="flex items-center gap-1.5 text-gray-500 hover:text-brand-purple font-medium">
113         <svg xmlns="http://www.w3.org/2000/svg" class="h-6 w-6" fill="none" viewBox="0 0 24 24" stroke="currentColor" stroke-linecap="round" stroke-linejoin="round" d="M17 16l4-4m0 0l-4-4m4 4H7m6 4v1a3 3 0 01-3 3H6a3 3 0 01-3 3" />
114     </a>
115     <span class="text-black hidden sm:inline-block" style="font-size: small;">Sair
116 </div>
```

1 - tela geral.jsp

```
121         </a>
122     </div>
123 </header>
124
125 <div class="grid grid-cols-2 gap-4 mb-6">
126     <div class="bg-white p-4 rounded-xl shadow-lg">
127         <span class="text-4xl font-bold text-brand-purple"><%= quantidadeAtivosDigitais %></span>
128         <p class="text-gray-600 font-medium">Contas Conectadas</p>
129     </div>
130     <div class="bg-white p-4 rounded-xl shadow-lg">
131         <span class="text-4xl font-bold text-brand-purple"><%= quantidadeHerdeiros %></span>
132         <p class="text-gray-600 font-medium">Herdeiros</p>
133     </div>
134 </div>
135 <div class="grid grid-cols-1 md:grid-cols-3 gap-4 mb-6">
136
137     <div class="bg-white p-4 rounded-xl shadow-lg flex items-center justify-between border-l-4 border-green-500">
138         <div>
139             <p class="text-xs font-bold text-gray-400 uppercase">Dólar (USD)</p>
140             <p id="cotacao-usd" class="text-lg font-bold text-gray-700">R$ ---,---</p>
141         </div>
142         <div class="p-2 bg-green-100 rounded-full text-green-600">
143             <svg xmlns="http://www.w3.org/2000/svg" class="h-6 w-6" fill="none" viewBox="0 0 24 24" stroke="currentColor"><path
```

1 - tela geral.jsp

```
Source History | 🔍 🗂️ 📁 🗃️ 🗃️ 🗃️ 🗃️ 🗃️ 🗃️ 🗃️ 🗃️ 🗃️ 🗃️ 🗃️ 🗃️ 🗃️ 🗃️ 🗃️
```

```
148 <div class="bg-white p-4 rounded-xl shadow-lg flex items-center justify-between border-1-4 border-orange-500">
149   <div>
150     <p class="text-xs font-bold text-gray-400 uppercase">Bitcoin (BTC)</p>
151     <p id="cotacao-btc" class="text-lg font-bold text-gray-700">R$ --,--</p>
152   </div>
153   <div class="p-2 bg-orange-100 rounded-full text-orange-600">
154     
155   </div>
156 </div>
157
158 <div class="bg-white p-4 rounded-xl shadow-lg flex items-center justify-between border-1-4 border-purple-500">
159   <div>
160     <p class="text-xs font-bold text-gray-400 uppercase">Ethereum (ETH)</p>
161     <p id="cotacao-eth" class="text-lg font-bold text-gray-700">R$ --,--</p>
162   </div>
163   <div class="p-2 bg-purple-100 rounded-full text-purple-600">
164     <svg xmlns="http://www.w3.org/2000/svg" class="h-6 w-6" fill="none" viewBox="0 0 24 24" stroke="currentColor"><path
```

// Função que consome o Web Service da AwesomeAPI

```
async function buscarCotacoes() {
  try {
    const response = await fetch('https://economia.awesomeapi.com.br/last/USD-BRL,BTC-BRL,ETH-BRL');
    const data = await response.json();

    const formatarMoeda = (valor) => {
      return new Intl.NumberFormat('pt-BR', { style: 'currency', currency: 'BRL' }).format(valor);
    };

    if(document.getElementById('cotacao-usd'))
      document.getElementById('cotacao-usd').innerText = formatarMoeda(data.USDBRL.bid);

    if(document.getElementById('cotacao-btc'))
      document.getElementById('cotacao-btc').innerText = formatarMoeda(data.BTCBRL.bid);

    if(document.getElementById('cotacao-eth'))
      document.getElementById('cotacao-eth').innerText = formatarMoeda(data.ETHBRL.bid);

  } catch (error) {
    console.error("Erro ao buscar cotações:", error);
  }
}

document.addEventListener('DOMContentLoaded', () => {
  buscarCotacoes();
  setInterval(buscarCotacoes, 30000);
}...)
```

1 - tela geral.jsp

```
Source History | 🔍 🗂️ 📁 🗃️ 🗃️ 🗃️ 🗃️ 🗃️ 🗃️ 🗃️ 🗃️ 🗃️ 🗃️ 🗃️ 🗃️ 🗃️ 🗃️ 🗃️
```

```
172
173
174   async function buscarCotacoes() {
175     try {
176       const response = await fetch('https://economia.awesomeapi.com.br/last/USD-BRL,BTC-BRL,ETH-BRL');
177       const data = await response.json();

178       const formatarMoeda = (valor) => {
179         return new Intl.NumberFormat('pt-BR', { style: 'currency', currency: 'BRL' }).format(valor);
180     };

181       if(document.getElementById('cotacao-usd'))
182         document.getElementById('cotacao-usd').innerText = formatarMoeda(data.USDBRL.bid);

183       if(document.getElementById('cotacao-btc'))
184         document.getElementById('cotacao-btc').innerText = formatarMoeda(data.BTCBRL.bid);

185       if(document.getElementById('cotacao-eth'))
186         document.getElementById('cotacao-eth').innerText = formatarMoeda(data.ETHBRL.bid);

187     } catch (error) {
188       console.error("Erro ao buscar cotações:", error);
189     }
190   }

191   document.addEventListener('DOMContentLoaded', () => {
192     buscarCotacoes();
193     setInterval(buscarCotacoes, 30000);
194   }...)
```

1 - tela geral.jsp

```
Source History < > □
```

```
196      buscarCotacoes();
197      setInterval(buscarCotacoes, 30000);
198  });
199  </script>
200
201  <div class="bg-white p-5 rounded-lg mb-6">
202      <h2 class="text-xl font-bold text-brand-purple-text mb-4">Suas Contas</h2>
203
204  <div class="flex flex-col space-y-3 mb-4">
205      <%
206          if (listaAD == null || listaAD.isEmpty()) {
207              %>
208                  <p class="text-gray-400 text-sm italic">Nenhuma conta conectada ainda.</p>
209          <%
210      } else {
211          for (AtivoDigital ad : listaAD) {
212              String tipo = ad.getTipoAtivoDigital();
213              String imgPath = "";
214              String nomeExibicao = tipo.substring(0, 1).toUpperCase() + tipo.substring(1);
215
216              if("facebook".equals(tipo)){ imgPath = "ico_facebook.png"; }
217              else if ("instagram".equals(tipo)) { imgPath = "ico_instagram.png"; }
218              else if ("twitter".equals(tipo)) { imgPath = "ico_twitter.png"; }
219              else if ("discord".equals(tipo)) { imgPath = "ico_discord.png"; }
220              else if ("crypto".equals(tipo)){ imgPath = "ico_bitcoin.png"; nomeExibicao = "Carteira Cripto"; }
221              else if ("mensagem".equals(tipo)){ imgPath = "ico_postuma.png"; nomeExibicao = "Mensagem Póstuma"; }
222          }
223      }
224  <%>
225
226  <div class="flex items-center justify-between p-3 bg-brand-purple-faded rounded-lg hover:bg-opacity-70 transition-all">
227      <div class="flex items-center gap-3">
228          ">
229          <span class="font-bold text-brand-purple-text"><%= nomeExibicao %></span>
230      </div>
231
232  <div class="flex items-center gap-3">
233      <div class="flex items-center gap-1 bg-green-100 text-brand-green px-2 py-1 rounded text-xs font-bold">
234          <svg xmlns="http://www.w3.org/2000/svg" class="h-3 w-3" viewBox="0 0 20 20" fill="currentColor">
235              <path fill-rule="evenodd" d="M16.707 5.293a1 0 0 1 0 1.414l-8 1 0 0-1.414 0 1-4a1 0 0 1 0 0.114a1.414-1.414 0 0 1 1.414 0z" />
236          </svg>
237          Vinculado a <%= ad.getNomeHerdeiro() %>
238      </div>
239
240      <button type="button" onclick="openDeleteModal(<%= ad.getId() %>)" class="text-gray-400 hover:text-red-500 transition-colors flex-shrink-0 title="Excluir Conta">
241          <svg xmlns="http://www.w3.org/2000/svg" class="h-6 w-6" fill="none" viewBox="0 0 24 24" stroke="currentColor">
242              <path stroke-linecap="round" stroke-linejoin="round" d="M19 71-.867 12.142a2 2 0 0 1 0 1.382 2 1H7.862a2 2 (
```

1 - tela geral.jsp

```
Source History < > □
```

```
223      String fullPath = request.getContextPath() + "/imagens/" + imgPath;
224  <%>
225
226  <div class="flex items-center justify-between p-3 bg-brand-purple-faded rounded-lg hover:bg-opacity-70 transition-all">
227      <div class="flex items-center gap-3">
228          ">
229          <span class="font-bold text-brand-purple-text"><%= nomeExibicao %></span>
230      </div>
231
232  <div class="flex items-center gap-3">
233      <div class="flex items-center gap-1 bg-green-100 text-brand-green px-2 py-1 rounded text-xs font-bold">
234          <svg xmlns="http://www.w3.org/2000/svg" class="h-3 w-3" viewBox="0 0 20 20" fill="currentColor">
235              <path fill-rule="evenodd" d="M16.707 5.293a1 0 0 1 0 1.414l-8 1 0 0-1.414 0 1-4a1 0 0 1 0 0.114a1.414-1.414 0 0 1 1.414 0z" />
236          </svg>
237          Vinculado a <%= ad.getNomeHerdeiro() %>
238      </div>
239
240      <button type="button" onclick="openDeleteModal(<%= ad.getId() %>)" class="text-gray-400 hover:text-red-500 transition-colors flex-shrink-0 title="Excluir Conta">
241          <svg xmlns="http://www.w3.org/2000/svg" class="h-6 w-6" fill="none" viewBox="0 0 24 24" stroke="currentColor">
242              <path stroke-linecap="round" stroke-linejoin="round" d="M19 71-.867 12.142a2 2 0 0 1 0 1.382 2 1H7.862a2 2 (
```


1 - tela geral.jsp

```
Source History < > □
298     <div id="modal-overlay" class="fixed inset-0 bg-black bg-opacity-60 hidden items-center justify-center p-4 z-40">
299         <div id="modal-container" class="bg-white w-full max-w-lg rounded-2xl shadow-xl p-6 relative">
300             <div class="flex justify-between items-center mb-4">
301                 <h2 class="text-2xl font-bold text-brand-purple-text">Conectar Nova Conta</h2>
302                 <button onclick="closeModal()" class="text-gray-400 hover:text-gray-700"><svg xmlns="http://www.w3.org/2000/svg" class="h-7 w-7" style="fill: currentColor; stroke: none; height: 1em; width: 1em; vertical-align: middle; transition: fill 0.2s ease; border-radius: 50%; border: 1px solid #ccc; padding: 2px; margin-right: 5px;"/></button>
303             </div>
304             <form id="form-conectar-conta" action="<%= request.getContextPath() %>/AtivoDigitalServlet" method="POST">
305                 <div class="space-y-4">
306                     <div>
307                         <label for="tipo-conta" class="block text-sm font-medium text-gray-700 mb-1">Tipo de Conta *</label>
308                         <select id="tipo-conta" name="tipo-conta" required class="w-full p-3 bg-brand-purple-faded rounded-lg border border-brand-purple">
309                             <option value="facebook">Facebook</option>
310                             <option value="instagram">Instagram</option>
311                             <option value="twitter">Twitter</option>
312                             <option value="discord">Discord</option>
313                             <option value="crypto">Carteira Cripto (Frases Chaves)</option>
314                             <option value="mensagem">Mensagem Póstuma (Segredo)</option>
315                         </select>
316                     </div>
317                     <div id="campos-padrao" class="space-y-4">
318                         <div>
319                             <label for="conta-login" class="block text-sm font-medium text-gray-700 mb-1">E-mail ou Usuário *</label>
320                             <input type="text" name="conta-login" id="conta-login" class="w-full p-3 bg-brand-purple-faded rounded-lg border border-brand-purple">
321                         </div>
322                         <div>
323                             <label for="conta-senha" class="block text-sm font-medium text-gray-700 mb-1">Senha *</label>
324                             <input type="password" name="conta-senha" id="conta-senha" class="w-full p-3 bg-brand-purple-faded rounded-lg border border-brand-purple">
325                         </div>
326                     </div>
327                     <div id="campo-frase-crypto" class="hidden">
328                         <label for="conta-frase" class="block text-sm font-medium text-gray-700 mb-1">Palavras Chaves (Frase de Recuperação) *</label>
329                         <textarea name="conta-frase" id="conta-frase" rows="4" class="w-full p-3 bg-brand-purple-faded rounded-lg border border-brand-purple"></textarea>
330                         <p class="text-xs text-gray-500 mt-1">Seus dados são criptografados antes de serem salvos.</p>
331                     </div>
332                     <div id="campo-mensagem" class="hidden">
333                         <label for="conta-mensagem" class="block text-sm font-medium text-gray-700 mb-1">Sua Mensagem Secreta *</label>
334                         <textarea name="conta-mensagem" id="conta-mensagem" rows="6" class="w-full p-3 bg-brand-purple-faded rounded-lg border border-brand-purple"></textarea>
335                         <p class="text-xs text-gray-500 mt-1">Esta mensagem será criptografada e entregue apenas ao herdeiro selecionado.</p>
336                     </div>
337                     <div>
338                         <label for="herdeiro" class="block text-sm font-medium text-gray-700 mb-1">Atribuir ao Herdeiro *</label>
339                         <select id="herdeiro" name="herdeiro" required class="w-full p-3 bg-brand-purple-faded rounded-lg border border-brand-purple">
340                             <option value="" disabled selected>Selecione um herdeiro...</option>
341                             <% if (listaH != null) { for (Herdeiro h : listaH) { %>
342                                 <option value="<%= h.getId() %>"><%= h.getNome() %> (<% if (h.getParentesco() != null ? h.getParentesco().getNome() : "Herdeiro" %>)</option>
343                             <% } } %>
344                         </select>
345                     </div>
346                 </div>
347             </form>
348         </div>
349     </div>
350 
```

1 - tela geral.jsp

```
Source History < > □
319             <label for="conta-login" class="block text-sm font-medium text-gray-700 mb-1">E-mail ou Usuário *</label>
320             <input type="text" name="conta-login" id="conta-login" class="w-full p-3 bg-brand-purple-faded rounded-lg border border-brand-purple">
321         </div>
322         <div>
323             <label for="conta-senha" class="block text-sm font-medium text-gray-700 mb-1">Senha *</label>
324             <input type="password" name="conta-senha" id="conta-senha" class="w-full p-3 bg-brand-purple-faded rounded-lg border border-brand-purple">
325         </div>
326         <div id="campo-frase-crypto" class="hidden">
327             <label for="conta-frase" class="block text-sm font-medium text-gray-700 mb-1">Palavras Chaves (Frase de Recuperação) *</label>
328             <textarea name="conta-frase" id="conta-frase" rows="4" class="w-full p-3 bg-brand-purple-faded rounded-lg border border-brand-purple"></textarea>
329             <p class="text-xs text-gray-500 mt-1">Seus dados são criptografados antes de serem salvos.</p>
330         </div>
331         <div id="campo-mensagem" class="hidden">
332             <label for="conta-mensagem" class="block text-sm font-medium text-gray-700 mb-1">Sua Mensagem Secreta *</label>
333             <textarea name="conta-mensagem" id="conta-mensagem" rows="6" class="w-full p-3 bg-brand-purple-faded rounded-lg border border-brand-purple"></textarea>
334             <p class="text-xs text-gray-500 mt-1">Esta mensagem será criptografada e entregue apenas ao herdeiro selecionado.</p>
335         </div>
336         <div>
337             <label for="herdeiro" class="block text-sm font-medium text-gray-700 mb-1">Atribuir ao Herdeiro *</label>
338             <select id="herdeiro" name="herdeiro" required class="w-full p-3 bg-brand-purple-faded rounded-lg border border-brand-purple">
339                 <option value="" disabled selected>Selecione um herdeiro...</option>
340                 <% if (listaH != null) { for (Herdeiro h : listaH) { %>
341                     <option value="<%= h.getId() %>"><%= h.getNome() %> (<% if (h.getParentesco() != null ? h.getParentesco().getNome() : "Herdeiro" %>)</option>
342                 <% } } %>
343             </select>
344         </div>
345     </div>
346 
```

1 - tela geral.jsp

```
Source History < > ▾
```

```
343     <% } } %>
344     </select>
345   </div>
346 </div>
347 <div class="flex justify-end gap-3 mt-8">
348   <button type="button" onclick="closeModal()" class="p-3 px-6 bg-gray-200 text-gray-800 rounded-lg font-bold hover:bg-gray-300 transition-all">Cancelar</button>
349   <button type="submit" class="p-3 px-6 bg-brand-green text-white rounded-lg font-bold hover:bg-opacity-90 transition-all">Salvar Conta Selec...
350 </div>
351 <m>
352 <!--
353 -->
354 <div class="fixed inset-0 bg-black bg-opacity-60 hidden flex items-center justify-center p-4 z-50">
355   <div class="bg-white w-full max-w-md rounded-2xl shadow-xl p-6 text-center">
356     <h2>Excluir Conta?</h2>
357     <p>Tem certeza que deseja remover este ativo digital? Essa ação não pode ser desfeita.</p>
358     <div class="flex justify-center gap-3">
359       <button onclick="closeDeleteModal()" class="px-5 py-3 bg-gray-200 text-gray-800 rounded-lg font-bold hover:bg-gray-300 transition-all">Cancelar</button>
360       <a id="btn-confirmar-exclusao" href="#" class="px-5 py-3 bg-red-600 text-white rounded-lg font-bold hover:bg-red-700 transition-all shadow-m...
361     </div>
362   </div>
363 </div>
364 <!--
365 -->
366 <!--
367 -->
368 <!--
369 -->
370 <!--
371   gica Modal de Cadastro ---
372   modalOverlay = document.getElementById('modal-overlay');
373   oContaSelect = document.getElementById('tipo-conta');
374   iposPadrao = document.getElementById('campos-padrao');
375   ipoFrageCrypto = document.getElementById('campo-frase-crypto');
376   ipoMensagem = document.getElementById('campo-mensagem');
377   utLogin = document.getElementById('conta-login');
378   utSenha = document.getElementById('conta-senha');
379   utFrage = document.getElementById('conta-frase');
380   utMensagem = document.getElementById('conta-mensagem');
381 <!--
382   openModal() { modalOverlay.classList.remove('hidden'); modalOverlay.classList.add('flex'); }
383   closeModal() { modalOverlay.classList.add('hidden'); modalOverlay.classList.remove('flex'); }
384 <!--
385   gica Modal de Exclusão ---
386   eteModalOverlay = document.getElementById('modal-delete-overlay');
387   eteConfirmBtn = document.getElementById('btn-confirmar-exclusao');
388 <!--
389   openDeleteModal(id) {
390     eConfirmBtn.href = "<%= request.getContextPath() %>/AtivoDigitalServlet?acao=excluir&id=" + id;
391     eModalOverlay.classList.remove('hidden');
392   }
393   closeDeleteModal() {
394     eModalOverlay.classList.add('hidden');
395   }
396 <!--
397   gertas de URL ---
398   Params = new URLSearchParams(window.location.search);
399   ams.has('erro') && urlParams.get('erro') === 'duplicado' {
400     ('ATENÇÃO: Esta conta já foi cadastrada para você! \nVocê não pode conectar a mesma conta duas vezes.');
401   }
402 <!--
403   Select.addEventListener('change', function() {
404     valor = this.value;
405     iposPadrao.classList.add('hidden');
406     ipoFrageCrypto.classList.add('hidden');
407     ipoMensagem.classList.add('hidden');
408     Login.required = false; inputSenha.required = false; inputFrage.required = false; inputMensagem.required = false;
409 <!--
```

1 - tela geral.jsp

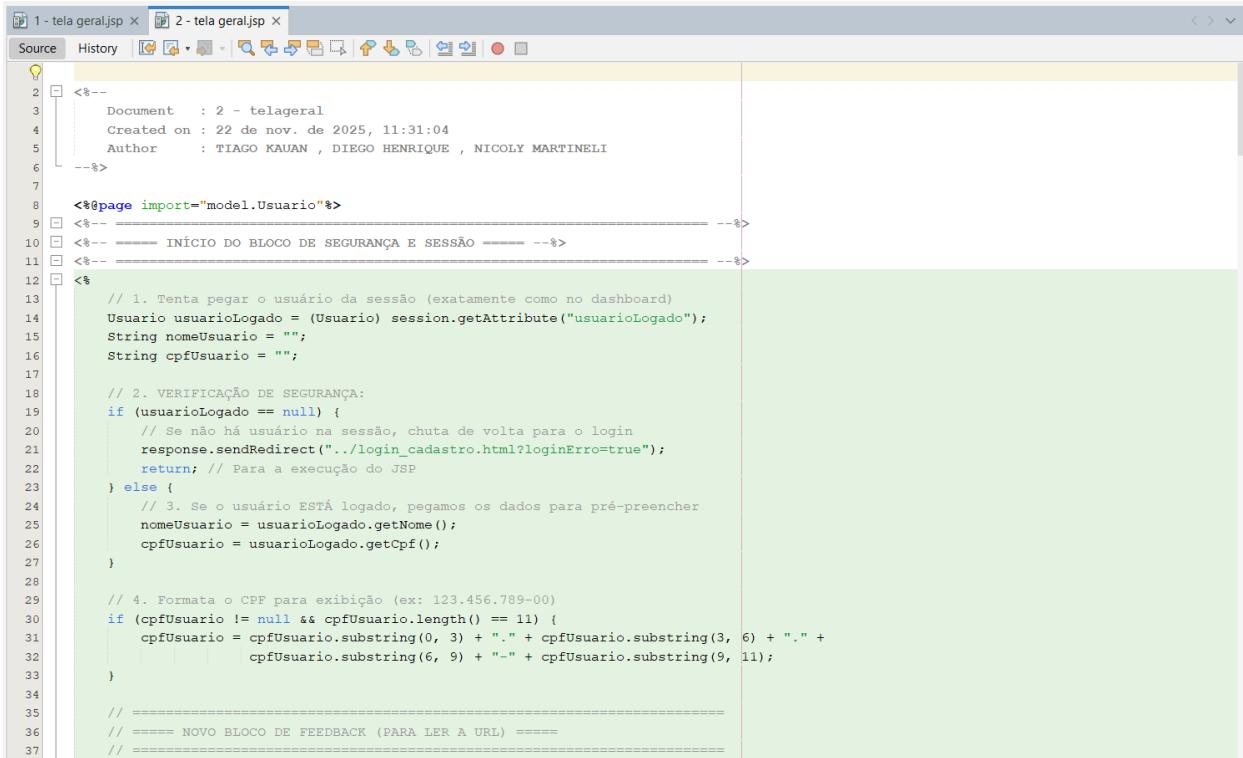
```
Source History < > ▾
```

```
373   iposPadrao = document.getElementById('campos-padrao');
374   ipoFrageCrypto = document.getElementById('campo-frase-crypto');
375   ipoMensagem = document.getElementById('campo-mensagem');
376   utLogin = document.getElementById('conta-login');
377   utSenha = document.getElementById('conta-senha');
378   utFrage = document.getElementById('conta-frase');
379   utMensagem = document.getElementById('conta-mensagem');
380
381   openModal() { modalOverlay.classList.remove('hidden'); modalOverlay.classList.add('flex'); }
382   closeModal() { modalOverlay.classList.add('hidden'); modalOverlay.classList.remove('flex'); }
383
384   <!--
385   gica Modal de Exclusão ---
386   eteModalOverlay = document.getElementById('modal-delete-overlay');
387   eteConfirmBtn = document.getElementById('btn-confirmar-exclusao');
388 <!--
389   openDeleteModal(id) {
390     eConfirmBtn.href = "<%= request.getContextPath() %>/AtivoDigitalServlet?acao=excluir&id=" + id;
391     eModalOverlay.classList.remove('hidden');
392   }
393   closeDeleteModal() {
394     eModalOverlay.classList.add('hidden');
395   }
396 <!--
397   gertas de URL ---
398   Params = new URLSearchParams(window.location.search);
399   ams.has('erro') && urlParams.get('erro') === 'duplicado' {
400     ('ATENÇÃO: Esta conta já foi cadastrada para você! \nVocê não pode conectar a mesma conta duas vezes.');
401   }
402 <!--
403   Select.addEventListener('change', function() {
404     valor = this.value;
405     iposPadrao.classList.add('hidden');
406     ipoFrageCrypto.classList.add('hidden');
407     ipoMensagem.classList.add('hidden');
408     Login.required = false; inputSenha.required = false; inputFrage.required = false; inputMensagem.required = false;
409 <!--
```

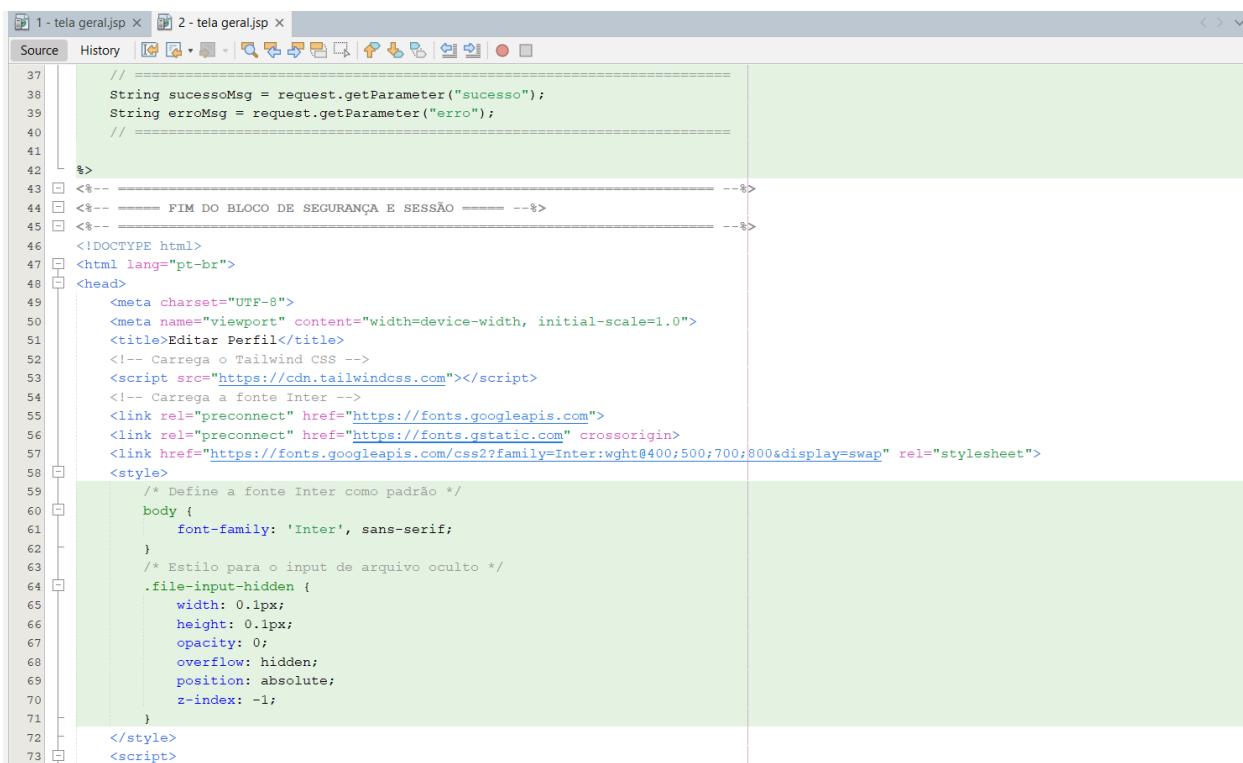
The screenshot shows a Java IDE interface with a JSP file named 'tela geral.jsp' open. The code handles various user inputs and validation rules:

```
394 eModalOverlay.classList.add('hidden');
395 
396 
397 ertas de URL ---
398 Params = new URLSearchParams(window.location.search);
399 if (params.has('erro') && urlParams.get('erro') === 'duplicado') {
400   ('ATENÇÃO: Esta conta já foi cadastrada para você! \nVocê não pode conectar a mesma conta duas vezes.');
401 }
402 
403 select.addEventListener('change', function() {
404   valor = this.value;
405   if (valor === 'padrao') {
406     inputPadrao.classList.add('hidden');
407     inputFraseCrypto.classList.add('hidden');
408     inputMensagem.classList.add('hidden');
409     login.required = false;
410     inputSenha.required = false;
411     inputFrase.required = false;
412     inputMensagem.required = false;
413   } else if (valor === 'crypto') {
414     inputFraseCrypto.classList.remove('hidden');
415     inputFrase.required = true;
416   } else if (valor === 'mensagem') {
417     inputMensagem.classList.remove('hidden');
418     inputMensagem.required = true;
419   } else {
420     inputPadrao.classList.remove('hidden');
421     login.required = true;
422     inputSenha.required = true;
423   }
424 }
425 
426 addEventListener('DOMContentLoaded', () => {
427   const select = document.querySelector('#select');
428   const modalOverlay = document.querySelector('.modal-overlay');
429 
430   select.dispatchEvent(new Event('change'));
431 
432   if (modalOverlay.classList.contains('hidden')) {
433     modalOverlay.classList.remove('flex');
434   }
435 }
```

5.2.2 2 – Tela Geral.jsp



```
1 - tela geral.jsp x 2 - tela geral.jsp x
Source History | < > v
<%--  
Document : 2 - telageral  
Created on : 22 de nov. de 2025, 11:31:04  
Author : TIAGO KAUAN , DIEGO HENRIQUE , NICOLY MARTINELI  
--%>  
<%@page import="model.Usuario"%>  
<%-- ----- INÍCIO DO BLOCO DE SEGURANÇA E SESSÃO ----- --%>  
<%-- ----- NOVO BLOCO DE FEEDBACK (PARA LER A URL) ----- --%>  
<%  
    // 1. Tenta pegar o usuário da sessão (exatamente como no dashboard)  
    Usuario usuarioLogado = (Usuario) session.getAttribute("usuarioLogado");  
    String nomeUsuario = "";  
    String cpfUsuario = "";  
  
    // 2. VERIFICAÇÃO DE SEGURANÇA:  
    if (usuarioLogado == null) {  
        // Se não há usuário na sessão, chuta de volta para o login  
        response.sendRedirect("../login_cadastro.html?loginErro=true");  
        return; // Para a execução do JSP  
    } else {  
        // 3. Se o usuário ESTÁ logado, pegamos os dados para pré-preencher  
        nomeUsuario = usuarioLogado.getNome();  
        cpfUsuario = usuarioLogado.getCpf();  
    }  
  
    // 4. Formata o CPF para exibição (ex: 123.456.789-00)  
    if (cpfUsuario != null && cpfUsuario.length() == 11) {  
        cpfUsuario = cpfUsuario.substring(0, 3) + "." + cpfUsuario.substring(3, 6) + "." +  
                    cpfUsuario.substring(6, 9) + "-" + cpfUsuario.substring(9, 11);  
    }  
  
    // ======  
    // ===== NOVO BLOCO DE FEEDBACK (PARA LER A URL) =====  
    // ======
```



```
// ======  
String sucessoMsg = request.getParameter("sucesso");  
String erroMsg = request.getParameter("erro");  
// =====  
<%>  
<%-- ----- FIM DO BLOCO DE SEGURANÇA E SESSÃO ----- --%>  
<%-- ----- NOVO BLOCO DE FEEDBACK (PARA LER A URL) ----- --%>  
<!DOCTYPE html>  
<html lang="pt-br">  
<head>  
    <meta charset="UTF-8">  
    <meta name="viewport" content="width=device-width, initial-scale=1.0">  
    <title>Editar Perfil</title>  
    <!-- Carrega o Tailwind CSS -->  
    <script src="https://cdn.tailwindcss.com"></script>  
    <!-- Carrega a fonte Inter -->  
    <link rel="preconnect" href="https://fonts.googleapis.com">  
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>  
    <link href="https://fonts.googleapis.com/css2?family=Inter:wght@400;500;700;800&display=swap" rel="stylesheet">  
    <style>  
        /* Define a fonte Inter como padrão */  
        body {  
            font-family: 'Inter', sans-serif;  
        }  
        /* Estilo para o input de arquivo oculto */  
        .file-input-hidden {  
            width: 0.1px;  
            height: 0.1px;  
            opacity: 0;  
            overflow: hidden;  
            position: absolute;  
            z-index: -1;  
        }  
    </style>  
<script>
```

1 - tela geral.jsp x 2 - tela geral.jsp x

Source History

```
73     <script>
74         // Configuração customizada do Tailwind para as cores
75         tailwind.config = {
76             theme: {
77                 extend: {
78                     colors: {
79                         'brand-purple-light': '#f2eefe', // Fundo do card
80                         'brand-purple': '#5b21b6', // Roxo escuro (botão)
81                         'brand-purple-medium': '#a78bfa', // Roxo médio
82                         'brand-purple-faded': '#ede9fe', // Fundo do input
83                         'brand-purple-text': '#4c1d95', // Texto roxo escuro
84                         'brand-green': '#22c55e', // Botão Criar Conta (verde)
85                         'brand-blue': '#3b82f6' // Box LGPD (azul)
86                     }
87                 }
88             }
89         }
90     </script>
91 </head>
92 <!-- ATUALIZADO: Fundo da página agora é roxo-claro, como o dashboard -->
93 <body class="bg-brand-purple-light min-h-screen">
94
95     <!-- Wrapper principal para centralizar e dar padding, igual ao dashboard -->
96 <main class="flex items-start justify-center w-full p-4 py-8">
97
98     <!-- Card de Edição de Perfil -->
99 <div class="w-full max-w-lg md:max-w-3xl mx-auto bg-white rounded-2xl shadow-xl overflow-hidden p-6 sm:p-10">
100
101     <!-- TELA DE EDIÇÃO -->
102     <div>
103
104         <!-- ===== INÍCIO DAS MENSAGENS DE FEEDBACK ===== -->
105         <!-- ===== INÍCIO DAS MENSAGENS DE FEEDBACK ===== -->
106         <!-- ===== INÍCIO DAS MENSAGENS DE FEEDBACK ===== -->
107         <% if (sucessoMsg != null) { %>
108             <div class="bg-green-100 border border-green-400 text-green-700 px-4 py-3 rounded relative mb-6" role="alert">
109                 <strong class="font-bold">Sucesso!</strong>
```

1 - tela geral.jsp x 2 - tela geral.jsp x

Source History

```
109                 <strong class="font-bold">Sucesso!</strong>
110                 <span class="block sm:inline">Suas alterações foram salvas.</span>
111             </div>
112         <% } %>
113
114         <% if (erroMsg != null) { %>
115             <div class="bg-red-100 border border-red-400 text-red-700 px-4 py-3 rounded relative mb-6" role="alert">
116                 <strong class="font-bold">Erro!</strong>
117                 <% if (erroMsg.equals("senhas")) { %>
118                     <span class="block sm:inline">As senhas não conferem. Tente novamente.</span>
119                 <% } else { %>
120                     <span class="block sm:inline">Ocorreu um erro ao salvar as alterações.</span>
121                 <% } %>
122             </div>
123         <% } %>
124         <!-- ===== FIM DAS MENSAGENS DE FEEDBACK ===== -->
125         <!-- ===== FIM DAS MENSAGENS DE FEEDBACK ===== -->
126         <!-- ===== FIM DAS MENSAGENS DE FEEDBACK ===== -->
127
128     <!-- Cabeçalho com Botão Voltar -->
129     <header class="flex justify-between items-center mb-6">
130         <h1 class="text-2xl font-bold text-brand-purple-text">
131             Editar Perfil
132         </h1>
133         <!-- ATUALIZADO: Link de 'Voltar' aponta para o dashboard .jsp -->
134         <a href="1 - tela geral.jsp" class="text-gray-400 hover:text-brand-purple transition-colors" title="Voltar ao Dashboard">
135             <svg xmlns="http://www.w3.org/2000/svg" class="h-8 w-8" fill="none" viewBox="0 0 24 24" stroke="currentColor" stroke-width="2">
136                 <path stroke-linecap="round" stroke-linejoin="round" d="M6 18L18 6M6 6L12 12" />
137             </svg>
138         </a>
139     </header>
140
141     <!-- Formulário de Edição ATUALIZADO -->
142     <!-- action aponta para o novo Servlet (...) para sair da pasta '1 - usuário' -->
143     <form action="../EditarPerfilServlet" method="POST" enctype="multipart/form-data" onsubmit="return validateForm(event);">
144         <div class="space-y-4">
```

1 - tela geral.jsp x 2 - tela geral.jsp x

```

142      <!-- action aponta para o novo Servlet (...) para sair da pasta '1 - usuario' -->
143      <form action="../EditarPerfilServlet" method="POST" enctype="multipart/form-data" onsubmit="return validateForm(event);">
144          <div class="space-y-4">
145
146              <!-- Campo: Foto de Perfil (Editável) -->
147              <div class="flex flex-col items-center mb-4">
148                  <label for="fotoPerfil" class="cursor-pointer">
149                      <div id="photo-preview" class="w-32 h-32 rounded-full bg-brand-purple-faded border-2 border-dashed border-brand-300">
150                          <!--
151                              NOTA: Para exibir a foto ATUAL, você precisará de um servlet
152                              separado para carregar a imagem do banco.
153                              Ex: 
154                              Por enquanto, mostramos o placeholder.
155
156                      <-->
157                      <span id="preview-text">Alterar foto de perfil</span>
158                      
159                  </div>
160                  <input type="file" name="fotoPerfil" id="fotoPerfil" accept="image/*" class="file-input-hidden">
161              </div>
162
163              <!-- Campo: Nome Completo (Bloqueado, vindo da SESSÃO) -->
164              <div>
165                  <label for="reg-nome" class="block text-sm font-medium text-gray-700 mb-1">Nome Completo (Não editável)</label>
166                  <input type="text" name="nome" id="reg-nome" value="<%& nomeUsuario %>" disabled
167                      class="w-full p-4 bg-gray-200 text-gray-500 rounded-lg border border-brand-purple-medium/30 cursor-not-allowed :>
168              </div>
169
170              <!-- Campo: CPF (Bloqueado, vindo da SESSÃO) -->
171              <div>
172                  <label for="reg-cpf" class="block text-sm font-medium text-gray-700 mb-1">CPF (Não editável)</label>
173                  <input type="text" name="cpf" id="reg-cpf" value="<%& cpfUsuario %>" disabled
174                      class="w-full p-4 bg-gray-200 text-gray-500 rounded-lg border border-brand-purple-medium/30 cursor-not-allowed :>
175              </div>
176
177              <hr class="border-t border-brand-purple-medium/20 my-4">
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214

```

1 - tela geral.jsp x 2 - tela geral.jsp x

```

178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214

```

```
214 </main> <!-- Fim do wrapper principal -->
215
216 <script>
217     // --- JavaScript para preview da foto de perfil ---
218     const photoInput = document.getElementById('fotoPerfil');
219     const previewContainer = document.getElementById('photo-preview');
220     const previewImage = document.getElementById('preview-image');
221     const previewText = document.getElementById('preview-text');
222
223     photoInput.addEventListener('change', function() {
224         const file = this.files[0];
225         if (file) {
226             const reader = new FileReader();
227
228             reader.onload = function(e) {
229                 previewImage.src = e.target.result;
230                 previewImage.classList.remove('hidden');
231                 previewText.classList.add('hidden');
232                 previewContainer.classList.remove('bg-brand-purple-faded', 'border-dashed');
233             }
234
235             reader.readAsDataURL(file);
236         }
237     });
238
239     // --- JavaScript para validar senhas ---
240     function validateForm(event) {
241         const senha = document.getElementById('reg-senha').value;
242         const confirmaSenha = document.getElementById('reg-confirm-senha').value;
243         const errorMsg = document.getElementById('password-error');
244
245         // Se ambos os campos de senha estiverem vazios, o usuário não quer mudar a senha.
246         // Isso é válido (ele pode estar mudando só a foto).
247         if (senha === '' && confirmaSenha === '') {
248             errorMsg.classList.add('hidden');
249             return true; // Permite o envio (só foto)
250         }
251
252         // Se um dos campos estiver preenchido, os dois devem ser iguais
253         if (senha !== confirmaSenha) {
254             event.preventDefault(); // Impede o envio
255             errorMsg.innerText = 'As senhas não conferem. Tente novamente.';
256             errorMsg.classList.remove('hidden'); // Mostra a mensagem de erro
257             return false;
258         }
259
260         errorMsg.classList.add('hidden');
261         return true; // Permite o envio
262     }
263
264 </script>
265
266 </body>
</html>
```

```
235         reader.readAsDataURL(file);
236     }
237 );
238
239 // --- JavaScript para validar senhas ---
240 function validateForm(event) {
241     const senha = document.getElementById('reg-senha').value;
242     const confirmaSenha = document.getElementById('reg-confirm-senha').value;
243     const errorMsg = document.getElementById('password-error');
244
245     // Se ambos os campos de senha estiverem vazios, o usuário não quer mudar a senha.
246     // Isso é válido (ele pode estar mudando só a foto).
247     if (senha === '' && confirmaSenha === '') {
248         errorMsg.classList.add('hidden');
249         return true; // Permite o envio (só foto)
250     }
251
252     // Se um dos campos estiver preenchido, os dois devem ser iguais
253     if (senha !== confirmaSenha) {
254         event.preventDefault(); // Impede o envio
255         errorMsg.innerText = 'As senhas não conferem. Tente novamente.';
256         errorMsg.classList.remove('hidden'); // Mostra a mensagem de erro
257         return false;
258     }
259
260     errorMsg.classList.add('hidden');
261     return true; // Permite o envio
262 }
263
264 </script>
265
266 </body>
</html>
```

5.2.3 recuperar_senha.jsp

The screenshot shows a code editor with the tab 'recuperar_senha.jsp' selected. The code is a JSP page with the following content:

```
<%--  
Document : recuperar_senha  
Created on : 22 de nov. de 2025, 11:31:04  
Author : TILAGO KAUAN , DIEGO HENRIQUE , NICOLY MARTINELI  
--%>  
<%@page import="config.HashUtil"%>  
<%@page import="model.DAO.UsuarioDAO"%>  
<%@page import="java.sql.SQLException"%>  
<%@page contentType="text/html" pageEncoding="UTF-8"%>  
<!DOCTYPE html>  
<html>  
<head>  
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">  
<title>Processando Recuperação</title>  
<!-- Importa o CSS e fontes para manter o estilo -->  
<script src="https://cdn.tailwindcss.com"></script>  
<link rel="preconnect" href="https://fonts.googleapis.com">  
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>  
<link href="https://fonts.googleapis.com/css2?family=Inter:wght@400;500;700;800&display=swap" rel="stylesheet">  
<style> body { font-family: 'Inter', sans-serif; } </style>  
<script>  
    tailwind.config = {  
        theme: {  
            extend: {  
                colors: {  
                    'brand-purple-light': '#f2eefe', 'brand-purple': '#5b21b6',  
                    'brand-purple-medium': '#a78bfa', 'brand-purple-faded': '#ede9fe',  
                    'brand-purple-text': '#4c1d95', 'brand-green': '#22c55e', 'brand-blue': '#3b82f6',  
                }  
            }  
        }  
    }  
</script>  
</head>  
<body class="bg-brand-purple-light flex items-center justify-center min-h-screen p-4">
```

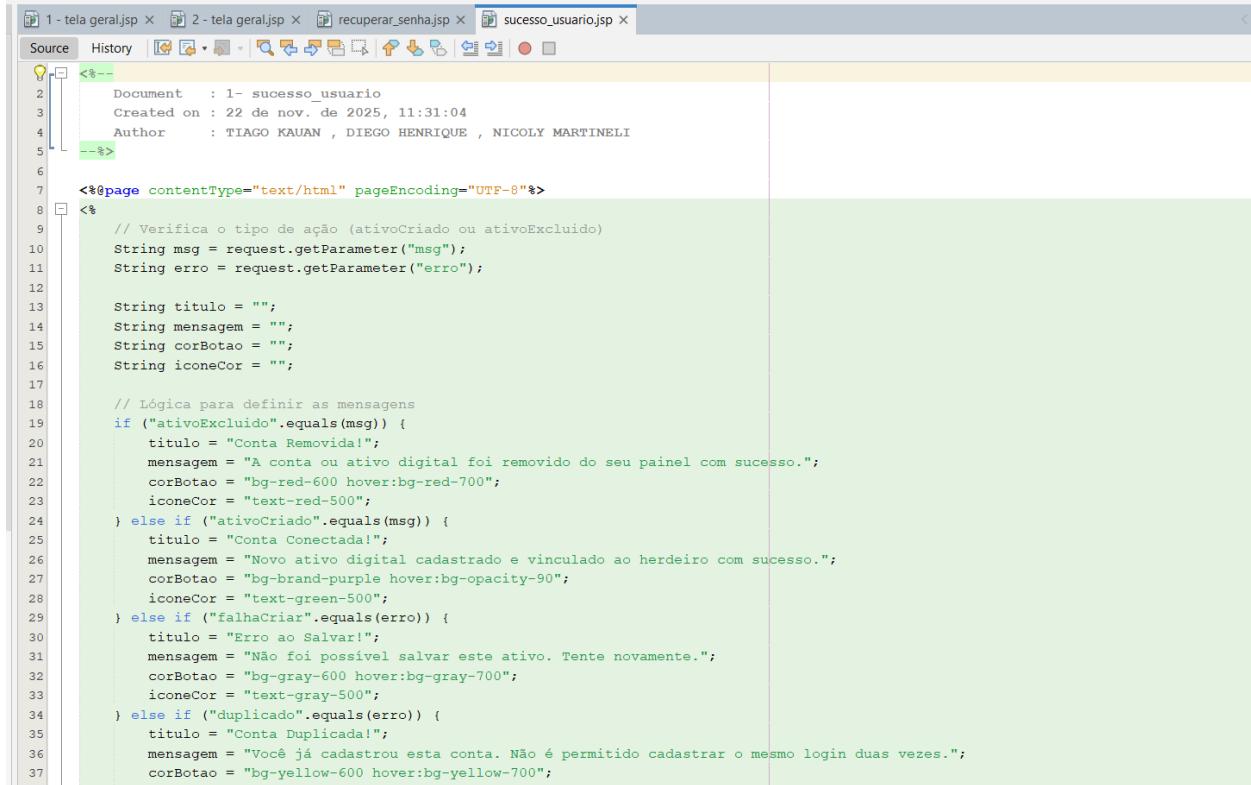
The screenshot shows a code editor with the tab 'recuperar_senha.jsp' selected. The code includes a scriptlet block and the following content:

```
</script>  
</head>  
<body class="bg-brand-purple-light flex items-center justify-center min-h-screen p-4">  
  
<div class="w-full max-w-md mx-auto bg-white p-8 rounded-2xl shadow-xl text-center">  
<%  
    String errorMessage = null;  
    boolean sucesso = false;  
  
    try {  
        // 1. Obter os dados do formulário  
        String cpf = request.getParameter("cpf");  
        String palavraChave = request.getParameter("palavra_chave");  
        String novaSenha = request.getParameter("senha_nova");  
        String confirmaSenha = request.getParameter("confirmar_senha_nova");  
  
        // 2. Validação dos campos  
        if (cpf == null || palavraChave == null || novaSenha == null || confirmaSenha == null ||  
            cpf.trim().isEmpty() || palavraChave.trim().isEmpty() ||  
            novaSenha.trim().isEmpty() || confirmaSenha.trim().isEmpty()) {  
  
            errorMessage = "Todos os campos são obrigatórios.";  
  
        } else if (!novaSenha.equals(confirmaSenha)) {  
            // Validação (extra) se as senhas não batem (o JS já deve ter feito isso)  
            errorMessage = "As novas senhas não conferem. Tente novamente.";  
  
        } else {  
  
            // 3. Criptografar (Hash) os dados  
            String cpfLimpo = cpf.replaceAll("[^0-9]", "");  
            String palavraChaveHash = HashUtil.toSHA256(palavraChave);  
            String novaSenhaHash = HashUtil.toSHA256(novaSenha);  
  
            // 4. Chamar o DAO para a operação atômica  
            UsuarioDAO usuarioDAO = new UsuarioDAO();  
            sucesso = usuarioDAO.redefinirSenha(cpfLimpo, palavraChaveHash, novaSenhaHash);  
        }  
    } catch (SQLException ex) {  
        errorMessage = "Ocorreu um erro ao processar sua solicitação. Tente novamente mais tarde.";  
    }  
%>
```

The screenshot shows a Java IDE interface with three tabs at the top: '1 - tela geral.jsp', '2 - tela geral.jsp', and 'recuperar_senha.jsp'. The 'recuperar_senha.jsp' tab is active, displaying the following JSP code:

```
String cpfLimpido = cpf.replaceAll("[^0-9]", "");  
String palavraChaveHash = HashUtil.toSHA256(palavraChave);  
String novaSenhaHash = HashUtil.toSHA256(novaSenha);  
  
// 4. Chamar o DAO para a operação atômica  
UsuarioDAO usuarioDAO = new UsuarioDAO();  
sucesso = usuarioDAO.redefinirSenha(cpfLimpido, palavraChaveHash, novaSenhaHash);  
  
if (!sucesso) {  
    errorMessage = "CPF ou Palavra-Chave de Recuperação estão incorretos.";  
}  
  
} catch (Exception e) {  
    errorMessage = "Erro no Servidor: " + e.getMessage();  
    e.printStackTrace(); // Loga o erro completo no console do Tomcat  
}  
  
// 5. Mostrar o feedback para o usuário  
if (sucesso) {  
    // SUCESSO (Parecido com a tela de "Cadastro Realizado")  
    out.println("<h2 class='text-2xl font-bold text-brand-green mb-4'>Senha Redefinida!</h2>");  
    out.println("<p class='text-gray-700'>Sua senha foi alterada com sucesso.</p>");  
    out.println("<a href='../login_cadastro.html' class='mt-6 block w-full p-3 bg-brand-purple text-white rounded-lg font-bo  
} else {  
    // ERRO  
    out.println("<h2 class='text-2xl font-bold text-red-600 mb-4'>Erro ao Redefinir</h2>");  
    out.println("<p class='text-gray-700'>" + (errorMessage != null ? errorMessage : "Erro desconhecido.") + "</p>");  
    out.println("<a href='../login_cadastro.html' onclick='history.back(); return false;' class='mt-6 block w-full p-3 bg-br  
}  
  
%>  
</div>  
</body>  
</html>
```

5.2.4 sucesso_usuario.jsp

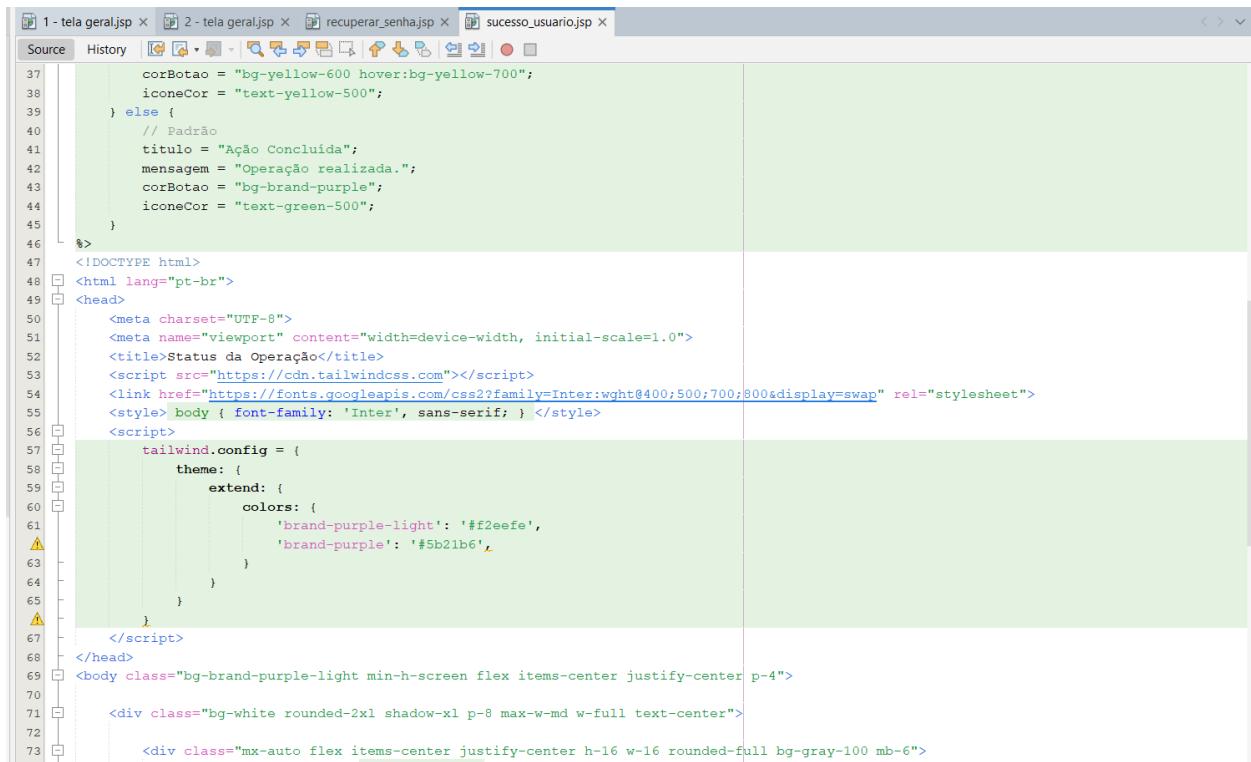


```
Document : 1- sucesso_usuario
Created on : 22 de nov. de 2025, 11:31:04
Author : TIAGO KAUAN , DIEGO HENRIQUE , NICOLY MARTINELI
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%
    // Verifica o tipo de ação (ativoCriado ou ativoExcluido)
    String msg = request.getParameter("msg");
    String erro = request.getParameter("erro");

    String titulo = "";
    String mensagem = "";
    String corBotao = "";
    String iconeCor = "";

    // Lógica para definir as mensagens
    if ("ativoExcluido".equals(msg)) {
        titulo = "Conta Removida!";
        mensagem = "A conta ou ativo digital foi removido do seu painel com sucesso.";
        corBotao = "bg-red-600 hover:bg-red-700";
        iconeCor = "text-red-500";
    } else if ("ativoCriado".equals(msg)) {
        titulo = "Conta Conectada!";
        mensagem = "Novo ativo digital cadastrado e vinculado ao herdeiro com sucesso.";
        corBotao = "bg-brand-purple hover:bg-opacity-90";
        iconeCor = "text-green-500";
    } else if ("falhaCriar".equals(erro)) {
        titulo = "Erro ao Salvar!";
        mensagem = "Não foi possível salvar este ativo. Tente novamente.";
        corBotao = "bg-gray-600 hover:bg-gray-700";
        iconeCor = "text-gray-500";
    } else if ("duplicado".equals(erro)) {
        titulo = "Conta Duplicada!";
        mensagem = "Você já cadastrou esta conta. Não é permitido cadastrar o mesmo login duas vezes.";
        corBotao = "bg-yellow-600 hover:bg-yellow-700";
    }
%>
```



```
<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Status da Operação</title>
    <script src="https://cdn.tailwindcss.com"></script>
    <link href="https://fonts.googleapis.com/css2?family=Inter:wght@400;500;700;800&display=swap" rel="stylesheet">
    <style> body { font-family: 'Inter', sans-serif; } </style>
    <script>
        tailwind.config = {
            theme: {
                extend: {
                    colors: {
                        'brand-purple-light': '#f2eefe',
                        'brand-purple': '#5b21b6',
                    }
                }
            }
        }
    </script>
</head>
<body class="bg-brand-purple-light min-h-screen flex items-center justify-center p-4">
    <div class="bg-white rounded-2xl shadow-xl p-8 max-w-md w-full text-center">
        <div class="mx-auto flex items-center justify-center h-16 w-16 rounded-full bg-gray-100 mb-6">
```

```
64      }
65
66      }
67    </script>
68  </head>
69  <body class="bg-brand-purple-light min-h-screen flex items-center justify-center p-4">
70
71    <div class="bg-white rounded-2xl shadow-xl p-8 max-w-md w-full text-center">
72
73      <div class="mx-auto flex items-center justify-center h-16 w-16 rounded-full bg-gray-100 mb-6">
74        <svg class="h-10 w-10 <%= iconeCor %>" fill="none" stroke="currentColor" viewBox="0 0 24 24">
75          <% if ("duplicado".equals(error) || "falhaCriar".equals(error)) { %>
76            <path stroke-linecap="round" stroke-linejoin="round" stroke-width="2" d="M12 9v2m0 4h.01m-6.938 4h13.856c1.54 0 2.502-1.06 2.502-1.06l-1.438-4h-2.856c-.562 0-1.06-.542-1.06-1.06z" />
77          <% } else { %>
78            <path stroke-linecap="round" stroke-linejoin="round" stroke-width="2" d="M5 13l4 4L19 7" />
79          <% } %>
80        </svg>
81      </div>
82
83      <h2 class="text-2xl font-bold text-gray-900 mb-2"><%= titulo %></h2>
84      <p class="text-gray-500 mb-8"><%= mensagem %></p>
85
86      <a href="<%= request.getContextPath() %>/1 - usuario/1 - tela geral.jsp"
87         class="block w-full py-3 px-4 rounded-lg text-white font-bold text-center transition-all shadow-md <%= corBotao %>">
88        Voltar ao Dashboard
89      </a>
90
91    </div>
92
93  </body>
94 </html>
```

5.3. WebPages 2 – ativodigital

5.3.11 – ativodigital.jsp

```
<%--  
    Document : 1- ativodigital  
    Created on : 22 de nov. de 2025, 11:31:04  
    Author : TIAGO KAUAN , DIEGO HENRIQUE , NICOLY MARTINELI  
--%>  
  
<%@page contentType="text/html" pageEncoding="UTF-8"%>  
<%@page import="java.util.List"%>  
<%@page import="model.LegadoAgrupado"%>  
<%@page import="model.LegadoDTO"%>  
<%@page import="config.CryptoUtil"%>  
  
<%  
    List<LegadoAgrupado> lista = (List<LegadoAgrupado>) request.getAttribute("listaAgrupada");  
    if (lista == null) {  
        response.sendRedirect(request.getContextPath() + "/LegadoServlet");  
        return;  
    }  
%>  
  
<!DOCTYPE html>  
<html lang="pt-br">  
<head>  
    <meta charset="UTF-8">  
    <meta name="viewport" content="width=device-width, initial-scale=1.0">  
    <title>Legados Recebidos</title>  
    <script src="https://cdn.tailwindcss.com"></script>  
    <link href="https://fonts.googleapis.com/css2?family=Inter:wght@400;500;700;800&display=swap" rel="stylesheet">  
    <style> body { font-family: 'Inter', sans-serif; } </style>  
    <script>  
        tailwind.config = {  
            theme: {  
                extend: {  
                    colors: {  
                        'brand-purple-light': '#f2eefe',  
                        'brand-purple': '#5b21b6',  
                        'brand-purple-text': '#4c1d95',  
                    }  
                }  
            }  
        }  
    </script>  
</head>
```

```
// --- CONTROLE DO MODAL ---  
function abrirModalUpload(idTitular) {  
    document.getElementById('id_titular_modal').value = idTitular;  
    document.getElementById('modalUpload').classList.remove('hidden');  
    document.getElementById('modalUpload').classList.add('flex');  
  
    // Reseta visual do input ao abrir  
    resetarInputVisual();  
}  
  
function fecharModalUpload() {  
    document.getElementById('modalUpload').classList.add('hidden');  
    document.getElementById('modalUpload').classList.remove('flex');  
}  
  
// --- NOVO: FEEDBACK VISUAL AO SELECIONAR ARQUIVO ---  
function atualizarNomeArquivo(input) {  
    const container = document.getElementById('upload-container');  
    const texto = document.getElementById('texto- arquivo');  
    const icone = document.getElementById('icone-upload');  
    const subtexto = document.getElementById('subtexto- arquivo');  
  
    if (input.files && input.files[0]) {  
        const arquivo = input.files[0];  
  
        // Muda borda para verde  
        container.classList.remove('border-gray-300');  
        container.classList.add('border-brand-green', 'bg-green-50');
```

Source History ▾ 🔍 🌐 🎯 📁 🗂️ 🏷️ 🎨 🎭 🎵 🎧 🎵 🎧 🎵 🎧

```

73     container.classList.add('border-brand-green', 'bg-green-50');
74
75     // Muda ícone e texto
76     texto.innerText = "Arquivo selecionado: " + arquivo.name;
77     texto.classList.add('text-brand-green');
78     subtexto.innerText = "Pronto para enviar!";
79
80     // Troca ícone para um Check
81     icone.innerHTML = '<path stroke-linecap="round" stroke-linejoin="round" stroke-width="2" d="M9 1212 2 4-4m6 2a9 9 0 11-18 0';
82     icone.classList.remove('text-brand-purple');
83     icone.classList.add('text-brand-green');
84   }
85
86   function resetarInputVisual() {
87     const container = document.getElementById('upload-container');
88     const texto = document.getElementById('texto-archivo');
89     const icone = document.getElementById('icone-upload');
90     const subtexto = document.getElementById('subtexto-archivo');
91     const input = document.getElementById('input-file');
92
93     input.value = ""; // Limpa o arquivo selecionado internamente
94
95     container.classList.add('border-gray-300');
96     container.classList.remove('border-brand-green', 'bg-green-50');
97
98     texto.innerText = "Clique para selecionar o PDF";
99     texto.classList.remove('text-brand-green');
100    subtexto.innerText = "Apenas arquivos .pdf";
101
102    icone.innerHTML = '<path stroke-linecap="round" stroke-linejoin="round" stroke-width="2" d="M7 16a4 4 0 0 1-.88-7.903A5 5 0 1115.';
103    icone.classList.add('text-brand-purple');
104    icone.classList.remove('text-brand-green');
105  }
106
107  </script>
108 </head>
109 <body class="bg-brand-purple-light min-h-screen relative">

```

Source History ▾ 🔍 🌐 🎯 📁 🗂️ 🏷️ 🎨 🎭 🎵 🎧 🎵 🎧 🎵 🎧

```

106
107  </script>
108 </head>
109 <body class="bg-brand-purple-light min-h-screen relative">
110
111 <main class="flex items-start justify-center w-full p-4 py-8">
112 <div class="w-full max-w-lg md:max-w-3xl mx-auto bg-transparent rounded-2xl">
113
114 <header class="flex justify-between items-center mb-6 px-2">
115   <h1 class="text-3xl font-bold text-brand-purple-text">Legados Recebidos</h1>
116   <a href="<% request.getContextPath() %>/1 - usuario/1 - tela geral.jsp" class="text-gray-400 hover:text-brand-purple transition">
117     <svg xmlns="http://www.w3.org/2000/svg" class="h-8 w-8" fill="none" viewBox="0 0 24 24" stroke="currentColor" stroke-width="2">
118       <path stroke-linecap="round" stroke-linejoin="round" d="M11 15l-3-3m0 0 11-18 0 9 9 0 0 1-18 0z" />
119   </a>
120 </header>
121
122 <div class="bg-brand-blue text-white p-4 rounded-lg flex gap-3 mb-8 shadow-lg text-sm">
123   <svg xmlns="http://www.w3.org/2000/svg" class="h-6 w-6 flex-shrink-0" fill="none" viewBox="0 0 24 24" stroke="currentColor"><pat>
124     <span>Envie a Certidão de Óbito uma única vez por titular para liberar todos os ativos.</span>
125   </div>
126
127 <div class="flex flex-col space-y-6">
128
129   <% if (lista.isEmpty()) { %>
130     <div class="text-center p-10 bg-white rounded-xl shadow text-gray-500">
131       <p>Você ainda não possui legados vinculados.</p>
132     </div>
133   <% } else {
134     for (LegadoAgrupado grupo : lista) {
135       String status = grupo.getStatusGeral();
136       String statusCor = "text-green-600";
137       String statusTexto = "Em vida. Legado bloqueado.";
138       String bolinhaCor = "bg-green-500";
139
140       if ("PENDENTE".equals(status)) {
141         statusCor = "text-brand-orange";
142       }
143     }
144   <% } %>
145
146   <table border="1" class="w-full text-sm text-left border-collapse">
147     <thead>
148       <tr>
149         <th>Nome do Legado</th>
150         <th>Status</th>
151         <th>Ações</th>
152       </tr>
153     <tbody>
154       <% for (LegadoAgrupado grupo : lista) { %>
155         <tr>
156           <td>${grupo.getNome()}</td>
157           <td>${statusCor}</td>
158           <td>
159             <a href="#" class="text-brand-purple transition">
160               <svg xmlns="http://www.w3.org/2000/svg" class="h-6 w-6" fill="none" viewBox="0 0 24 24" stroke="currentColor" stroke-width="2">
161                 <path stroke-linecap="round" stroke-linejoin="round" d="M11 15l-3-3m0 0 11-18 0 9 9 0 0 1-18 0z" />
162             </a>
163           </td>
164         </tr>
165       <% } %>
166     </tbody>
167   </table>
168
169 </div>
170 </main>
171 </div>
172 </body>
173 </html>

```

1 - tela geral.jsp x 2 - tela geral.jsp x recuperar_senha.jsp x sucesso_usuario.jsp x usuario.jsp x 1 - ativodigital.jsp x

Source History

```

136 |     String status = grupo.getStatusGeral();
137 |     String statusCor = "text-green-600";
138 |     String statusTexto = "Em vida. Legado bloqueado.";
139 |     String bolinhaCor = "bg-green-500";
140 |
141 |     if ("PENDENTE".equals(status)) {
142 |         statusCor = "text-brand-orange";
143 |         statusTexto = "Análise Pendente";
144 |         bolinhaCor = "bg-brand-orange";
145 |     } else if ("APROVADO".equals(status)) {
146 |         statusCor = "text-gray-600";
147 |         statusTexto = "Legado Liberado";
148 |         bolinhaCor = "bg-gray-600";
149 |     }
150 |
151 |    %>
152 |    <div class="bg-white p-6 rounded-lg shadow-lg flex flex-col relative overflow-hidden border border-gray-100">
153 |        <div class="flex justify-between items-center mb-6 border-b border-gray-100 pb-4">
154 |            <div class="flex items-center gap-4">
155 |                <div class="relative">
156 |                    <% if (grupo.getFotoTitularBase64() != null) { %>
157 |                        
158 |                    <% } else { %>
159 |                        <div class="w-16 h-16 rounded-full bg-brand-purple-light flex items-center justify-center border-2 border-brand-purple" style="width: 16px; height: 16px; border-radius: 50%; background-color: #e67e22; color: white; font-size: 1em; font-weight: bold;">T
```

1 - tela geral.jsp x 2 - tela geral.jsp x recuperar_senha.jsp x sucesso_usuario.jsp x usuario.jsp x 1 - ativodigital.jsp x

Source History

```

166 |
167 |    <div>
168 |        <h2 class="text-xl font-bold text-brand-purple-text"><%= grupo.getNomeTitular() %></h2>
169 |        <p class="text-sm font-medium <%= statusCor %>"><%= statusTexto %></p>
170 |    </div>
171 |
172 |
173 |
174 |    <div class="space-y-3 mb-6">
175 |        <h3 class="text-xs font-bold text-gray-400 uppercase tracking-wider mb-2">Ativos Digitais Recebidos:</h3>
176 |
177 |        <% for (LegadoDTO item : grupo.getAtivos()) {
178 |            String labelTipo = item.getTipoAtivo();
179 |            String dadosExibicao = "Conteúdo Protegido *****";
180 |
181 |            if ("APROVADO".equals(status)) {
182 |                if ("crypto".equals(labelTipo)) dadosExibicao = CryptoUtil.descriptografar(item.getFraseCripto());
183 |                else if ("mensagem".equals(labelTipo)) dadosExibicao = CryptoUtil.descriptografar(item.getMensagemCripto());
184 |                else dadosExibicao = "Login: " + item.getLogin() + " | Senha: " + CryptoUtil.descriptografar(item.getSenhaCripto());
185 |            }
186 |        %>
187 |        <div class="bg-brand-purple-light p-3 rounded-lg border border-brand-purple/10">
188 |            <div class="flex items-center gap-2 mb-1">
189 |                <% if("facebook".equals(labelTipo)) { %> 
190 |                <% else if("instagram".equals(labelTipo)) { %> 
191 |                <% else if("crypto".equals(labelTipo)) { %> 
192 |                <% else { %> <svg xmlns="http://www.w3.org/2000/svg" class="h-5 w-5 text-brand-purple" fill="none" viewBox="0 0 24 24" style="width: 16px; height: 16px; border-radius: 50%; position: relative; transition: fill 0.3s ease;"/>
193 |                    <span class="font-bold text-brand-purple capitalize text-sm" style="position: absolute; left: 50%; top: 50%; transform: translate(-50%, -50%);><%= labelTipo %></span>
194 |                </div>
195 |                <p class="text-gray-600 text-xs font-mono break-all pl-7"><%= dadosExibicao %></p>
196 |            </div>
197 |        <% } %>
198 |    </div>
199 |
200 |
201 |
202 |

```

1 - tela geral.jsp x 2 - tela geral.jsp x recuperar_senha.jsp x sucesso_usuario.jsp x usuario.jsp x 1 - ativodigital.jsp x

Source History

```

199
200
201    <% if ("BLOQUEADO".equals(status)) { %>
202        <button onclick="abrirModalUpload('<%= grupo.getIdTitular() %>')" class="w-full py-3 bg-brand-purple text-white font-bo
203            <svg xmlns="http://www.w3.org/2000/svg" class="h-5 w-5" fill="none" viewBox="0 0 24 24" stroke="currentColor"><path
204                Solicitar Liberação (Anexar Certidão)
205            </button>
206        <% } else if ("PENDENTE".equals(status)) { %>
207            <div class="flex items-center gap-2 justify-center text-brand-orange font-medium bg-orange-50 p-3 rounded-lg animate-pul
208                <svg xmlns="http://www.w3.org/2000/svg" class="h-5 w-5" fill="none" viewBox="0 0 24 24" stroke="currentColor"><path
209                    Aguardando análise da documentação...
210            </div>
211        <% } %>
212
213    </div>
214<% } } %>
215
216</div>
217</div>
218</main>
219
220<div id="modalUpload" class="hidden fixed inset-0 bg-black bg-opacity-50 z-50 items-center justify-center p-4 backdrop-blur-sm">
221    <div class="bg-white rounded-2xl shadow-2xl w-full max-w-md overflow-hidden transform transition-all scale-100">
222
223        <div class="bg-brand-purple p-4 flex justify-between items-center">
224            <h3 class="text-white font-bold text-lg">Enviar Certidão de Óbito</h3>
225            <button onclick="fecharModalUpload()" class="text-white hover:text-gray-200">
226                <svg xmlns="http://www.w3.org/2000/svg" class="h-6 w-6" fill="none" viewBox="0 0 24 24" stroke="currentColor"><path str
227            </button>
228        </div>
229
230        <div class="p-6">
231            <p class="text-gray-600 mb-4 text-sm">Para liberar os ativos digitais, por favor anexe a <strong>Certidão de Óbito</strong>
232
233            <form action="<% request.getContextPath() %>/LegadoServlet" method="POST" enctype="multipart/form-data" class="space-y-4">
234
235                <input type="hidden" name="id_titular" id="id_titular_modal">

```

1 - tela geral.jsp x 2 - tela geral.jsp x recuperar_senha.jsp x sucesso_usuario.jsp x usuario.jsp x 1 - ativodigital.jsp x

Source History

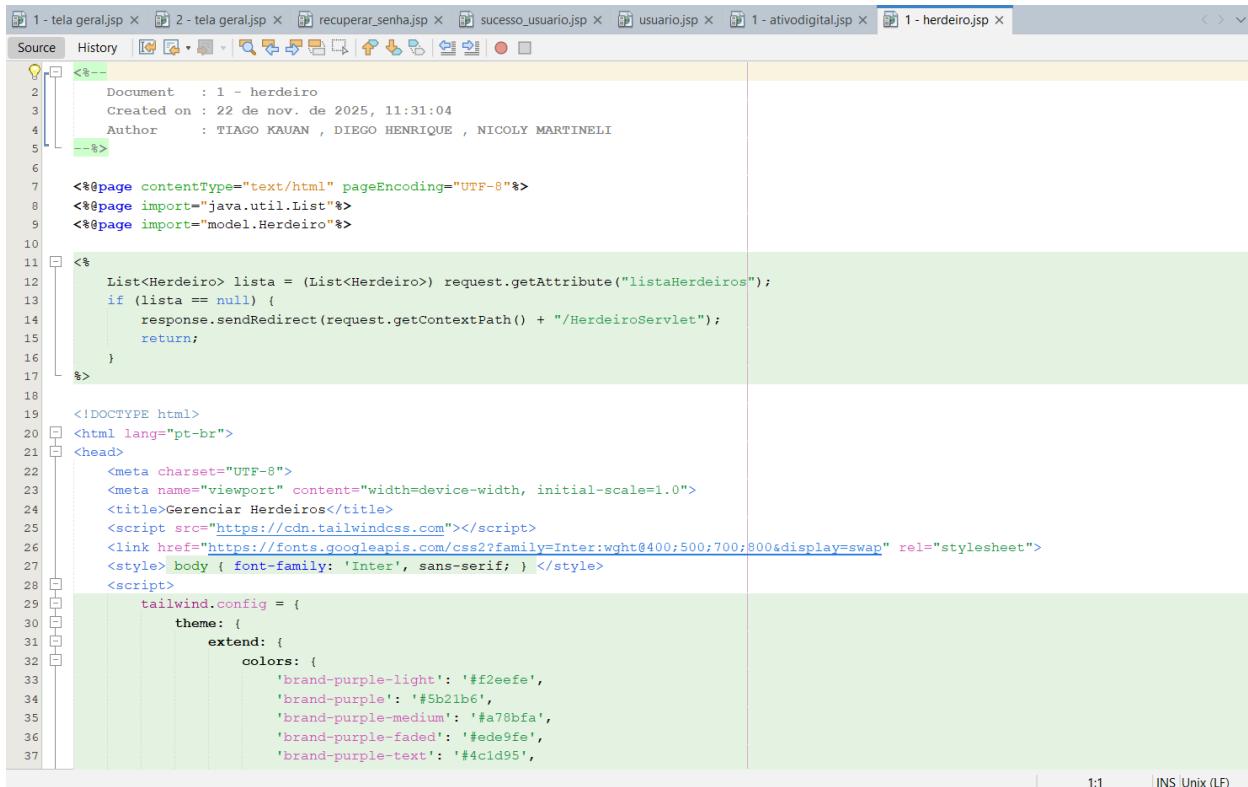
```

229
230
231    <div class="p-6">
232        <p class="text-gray-600 mb-4 text-sm">Para liberar os ativos digitais, por favor anexe a <strong>Certidão de Óbito</strong>
233
234        <form action="<% request.getContextPath() %>/LegadoServlet" method="POST" enctype="multipart/form-data" class="space-y-4">
235
236            <input type="hidden" name="id_titular" id="id_titular_modal">
237
238            <div id="upload-container" class="border-2 border-dashed border-gray-300 rounded-lg p-6 text-center hover:bg-gray-50 tr
239                <input type="file" id="input-file" name="certidao" accept="application/pdf" required
240                    class="absolute inset-0 w-full h-full opacity-0 cursor-pointer z-10"
241                    onchange="atualizarNomeArquivo(this)">
242
242                <img alt="Upload icon" data-bbox="288 615 315 628" />
243                <p id="texto- arquivo" class="text-sm text-gray-500 font-medium">Clique para selecionar o PDF</p>
244                <p id="subtexto- arquivo" class="text-xs text-gray-400 mt-1">Apenas arquivos .pdf</p>
245
246            </div>
247
248            <div class="flex gap-3 mt-6">
249                <button type="button" onclick="fecharModalUpload()" class="flex-1 py-2 border border-gray-300 text-gray-700 rounded
250                    <button type="submit" class="flex-1 py-2 bg-brand-purple text-white rounded-lg font-bold hover:bg-brand-purple-text
251                </div>
252
253            </div>
254
255        </div>
256
257    </body>
    </html>

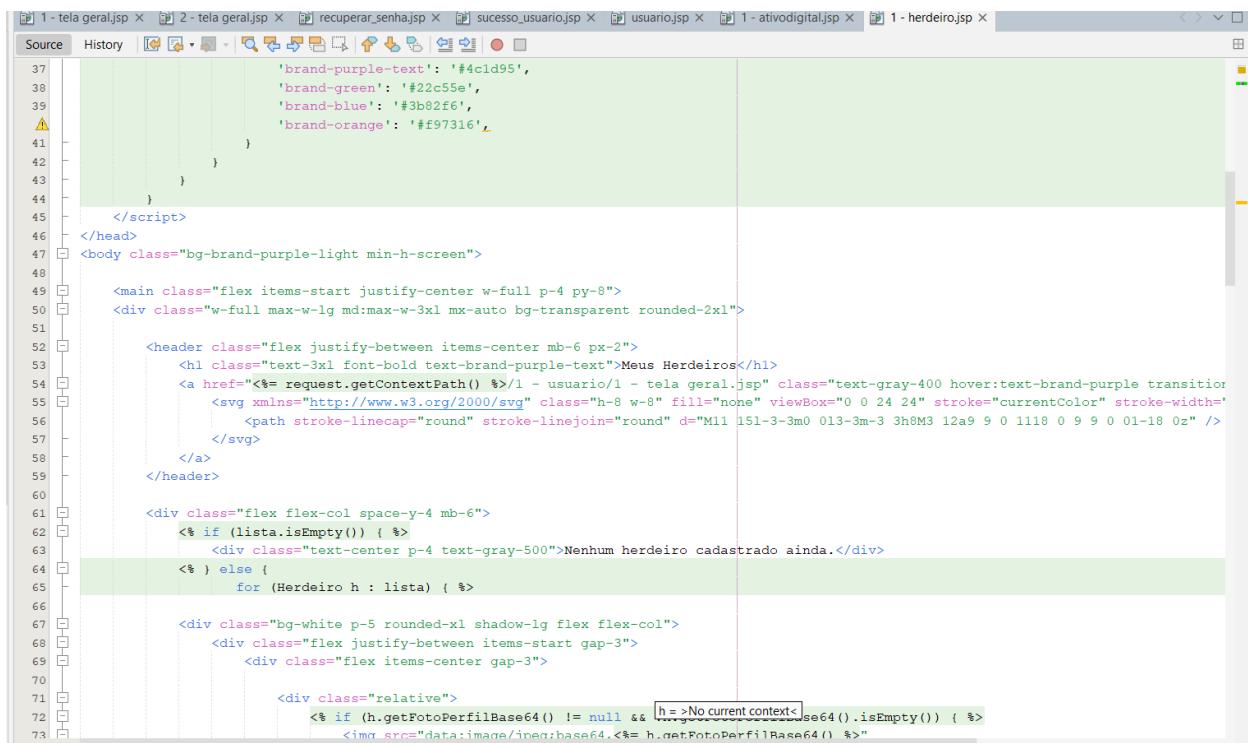
```

5.4. WebPages 3 – herdeiro

5.4.11 – herdeiro.jsp



```
<%--  
1 Document : 1 - herdeiro  
2 Created on : 22 de nov. de 2025, 11:31:04  
3 Author : TIAGO KAUAN , DIEGO HENRIQUE , NICOLY MARTINELI  
4-->  
5  
6 <%@page contentType="text/html" pageEncoding="UTF-8"%>  
7 <%@page import="java.util.List"%>  
8 <%@page import="model.Herdeiro"%>  
9  
10 <%  
11     List<Herdeiro> lista = (List<Herdeiro>) request.getAttribute("listaHerdeiros");  
12     if (lista == null) {  
13         response.sendRedirect(request.getContextPath() + "/HerdeiroServlet");  
14         return;  
15     }  
16 %>  
17  
18 <!DOCTYPE html>  
19 <html lang="pt-br">  
20     <head>  
21         <meta charset="UTF-8">  
22         <meta name="viewport" content="width=device-width, initial-scale=1.0">  
23         <title>Gerenciar Herdeiros</title>  
24         <script src="https://cdn.tailwindcss.com"></script>  
25         <link href="https://fonts.googleapis.com/css2?family=Inter:wght@400;500;700;800&display=swap" rel="stylesheet">  
26         <style> body { font-family: 'Inter', sans-serif; } </style>  
27         <script>  
28             tailwind.config = {  
29                 theme: {  
30                     extend: {  
31                         colors: {  
32                             'brand-purple-light': '#f2eefe',  
33                             'brand-purple': '#5b2l6b',  
34                             'brand-purple-medium': '#a78bfa',  
35                             'brand-purple-faded': '#ede9fe',  
36                             'brand-purple-text': '#4c1d95',  
37                         }  
38                     }  
39                 }  
40             }  
41         </script>  
42     </head>  
43     <body class="bg-brand-purple-light min-h-screen">  
44  
45         <main class="flex items-start justify-center w-full p-4 py-8">  
46             <div class="w-full max-w-lg md:max-w-3xl mx-auto bg-transparent rounded-2xl">  
47  
48                 <header class="flex justify-between items-center mb-6 px-2">  
49                     <h1 class="text-3xl font-bold text-brand-purple-text">Meus Herdeiros</h1>  
50                     <a href="<% request.getContextPath() %>/1 - usuario/1 - tela geral.jsp" class="text-gray-400 hover:text-brand-purple transition duration-200 ease-in-out">  
51                         <img alt="User icon" data-bbox="114 118 879 491"/>  
52                     </a>  
53                 </header>  
54  
55                 <div class="flex flex-col space-y-4 mb-6">  
56                     <% if (lista.isEmpty()) { %>  
57                         <div class="text-center p-4 text-gray-500">Nenhum herdeiro cadastrado ainda.</div>  
58                     <% } else {  
59                         for (Herdeiro h : lista) { %>  
60  
61                             <div class="bg-white p-5 rounded-xl shadow-lg flex flex-col">  
62                                 <div class="flex justify-between items-start gap-3">  
63                                     <div class="flex items-center gap-3">  
64                                         <div class="relative">  
65                                             <% if (h.getFotoPerfilBase64() != null && h.getFotoPerfilBase64().isEmpty()) { %>  
66                                                 <img alt="Placeholder image" data-bbox="114 491 879 850"/>  
67                                             <% } else { %>  
68                                                 <img alt="User profile picture" data-bbox="114 491 879 850"/>  
69                                             <% } %>  
70                                         <div class="flex flex-col justify-between items-start gap-3">  
71                                             <div class="text-gray-500 text-sm">Nome:</div>  
72                                             <div class="text-gray-500 text-sm">  
73                                                 <% if (h.getNome() != null) { %>  
74                                                     <div>...</div>  
75                                                 <% } else { %>  
76                                                     <div>...</div>  
77                                                 <% } %>  
78                                         </div>  
79                                     </div>  
80                                 </div>  
81                             </div>  
82                         <% } %>  
83                     </div>  
84                 </div>  
85             </div>  
86         </main>  
87     </body>  
88 </html>
```



```
37             <div class="relative">  
38                 <% if (h.getFotoPerfilBase64() != null && h.getFotoPerfilBase64().isEmpty()) { %>  
39                     <img alt="Placeholder image" data-bbox="114 491 879 850"/>  
40                 <% } else { %>  
41                     <img alt="User profile picture" data-bbox="114 491 879 850"/>  
42                 <% } %>  
43             </div>  
44         </div>  
45     </div>  
46     </div>  
47 </div>
```


Screenshot of a code editor showing the source code for '1 - herdeiro.jsp'. The code is a JSP page with embedded JavaScript for a modal overlay.

```

142 <div id="modal-overlay" class="fixed inset-0 bg-black bg-opacity-60 hidden flex items-center justify-center p-4 z-50">
143   <div id="modal-container" class="bg-white w-full max-w-lg rounded-2xl shadow-xl p-6 relative">
144     <div class="flex justify-between items-center mb-6">
145       <h2 class="text-2xl font-bold text-brand-purple-text">Adicionar Herdeiro</h2>
146       <button onclick="closeModal()" class="text-gray-400 hover:text-gray-700">
147         <svg xmlns="http://www.w3.org/2000/svg" class="h-7 w-7" fill="none" viewBox="0 0 24 24" stroke="currentColor" stroke-width="2">
148           <path stroke-linecap="round" stroke-linejoin="round" d="M6 18L18 6M6 6L12 12" />
149         </svg>
150       </button>
151     </div>
152 
153   <form action="<%= request.getContextPath() %>/HerdeiroServlet" method="POST">
154     <div class="space-y-4">
155       <div>
156         <label for="herdeiro-nome" class="block text-sm font-medium text-gray-700 mb-1">Nome do Herdeiro *</label>
157         <input type="text" name="herdeiro-nome" id="herdeiro-nome" required
158           class="w-full p-3 bg-brand-purple-faded rounded-lg border border-brand-purple-medium/30 placeholder-brand-purple-medium/30" placeholder="Nome do Herdeiro" value=""></input>
159       </div>
160 
161       <div>
162         <label for="herdeiro-cpf" class="block text-sm font-medium text-gray-700 mb-1">CPF do Herdeiro *</label>
163         <input type="text" name="herdeiro-cpf" id="herdeiro-cpf" placeholder="Apenas números (11 dígitos)" required
164           class="w-full p-3 bg-brand-purple-faded rounded-lg border border-brand-purple-medium/30 placeholder-brand-purple-medium/30" value=""></input>
165         <script>
166           oninput="this.value = this.value.replace(/[^0-9]/g, '')"
167         </script>
168       </div>
169 
170       <div>
171         <label for="herdeiro-parentesco" class="block text-sm font-medium text-gray-700 mb-1">Parentesco (opcional)</label>
172         <input type="text" name="herdeiro-parentesco" id="herdeiro-parentesco" placeholder="Ex: Mãe, Irmão, etc." required
173           class="w-full p-3 bg-brand-purple-faded rounded-lg border border-brand-purple-medium/30 placeholder-brand-purple-medium/30" value=""></input>
174       </div>
175 
176     <div class="bg-brand-blue text-white p-3 rounded-lg flex items-start gap-3 my-6 text-sm">
177       <span><strong>Notificação:</strong> Se o CPF informado já possuir cadastro, o usuário será notificado no aplicativo.</span>
178       <img alt="Information icon" data-bbox="288 515 318 535" style="vertical-align: middle;"/>
179     </div>
180   </div>
181 
182   <div class="flex justify-end gap-3 mt-8">
183     <button type="button" onclick="closeModal()" class="p-3 px-6 bg-gray-200 text-gray-800 rounded-lg font-bold hover:bg-gray-300 transition-all" style="outline: none; border: none; font-size: 1em; padding: 0.5em 1.5em; border-radius: 0.25em; font-weight: bold; color: inherit; background-color: inherit; text-decoration: none; cursor: pointer; white-space: nowrap; text-align: center; width: fit-content; height: 1.5em; line-height: 1.5em; vertical-align: middle; margin-right: 0.5em;">Cancelar
184     <button type="submit" class="p-3 px-6 bg-brand-green text-white rounded-lg font-bold hover:bg-opacity-90 transition-all" style="outline: none; border: none; font-size: 1em; padding: 0.5em 1.5em; border-radius: 0.25em; font-weight: bold; color: inherit; background-color: inherit; text-decoration: none; cursor: pointer; white-space: nowrap; text-align: center; width: fit-content; height: 1.5em; line-height: 1.5em; vertical-align: middle; margin-right: 0.5em;">Salvar
185   </div>
186 </div>
187 
188 <script>
189   const modalOverlay = document.getElementById('modal-overlay');
190   function openModal() {
191     modalOverlay.classList.remove('hidden');
192     modalOverlay.classList.add('flex');
193   }
194   function closeModal() {
195     modalOverlay.classList.add('hidden');
196     modalOverlay.classList.remove('flex');
197   }
198 </script>
199 
200 </body>
201 </html>

```

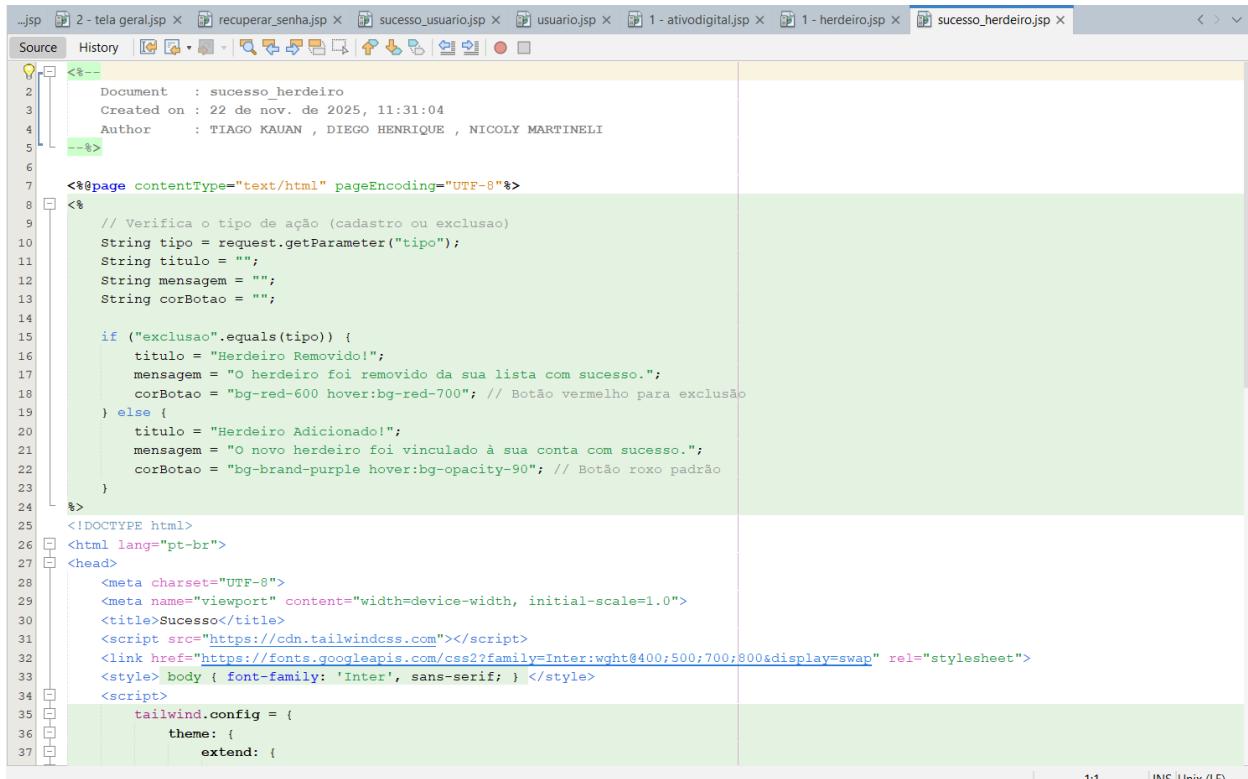
Screenshot of a code editor showing the source code for '1 - herdeiro.jsp' (continued). The code includes a modal overlay script at the bottom.

```

178   <path d="M2.003 5.884l10 9.88217.997-3.998A2 2 0 0016 4H4a2 2 0 00-1.997 1.884z" />
179   <path d="M18 8.118l-8 4-8-4V14a2 2 0 002 2h12a2 2 0 002-2V8.118z" />
180 </svg>
181 <span><strong>Notificação:</strong> Se o CPF informado já possuir cadastro, o usuário será notificado no aplicativo.</span>
182 </div>
183 
184 <div class="flex justify-end gap-3 mt-8">
185   <button type="button" onclick="closeModal()" class="p-3 px-6 bg-gray-200 text-gray-800 rounded-lg font-bold hover:bg-gray-300 transition-all" style="outline: none; border: none; font-size: 1em; padding: 0.5em 1.5em; border-radius: 0.25em; font-weight: bold; color: inherit; background-color: inherit; text-decoration: none; cursor: pointer; white-space: nowrap; text-align: center; width: fit-content; height: 1.5em; line-height: 1.5em; vertical-align: middle; margin-right: 0.5em;">Cancelar
186   <button type="submit" class="p-3 px-6 bg-brand-green text-white rounded-lg font-bold hover:bg-opacity-90 transition-all" style="outline: none; border: none; font-size: 1em; padding: 0.5em 1.5em; border-radius: 0.25em; font-weight: bold; color: inherit; background-color: inherit; text-decoration: none; cursor: pointer; white-space: nowrap; text-align: center; width: fit-content; height: 1.5em; line-height: 1.5em; vertical-align: middle; margin-right: 0.5em;">Salvar
187 </div>
188 
189 <script>
190   const modalOverlay = document.getElementById('modal-overlay');
191   function openModal() {
192     modalOverlay.classList.remove('hidden');
193     modalOverlay.classList.add('flex');
194   }
195   function closeModal() {
196     modalOverlay.classList.add('hidden');
197     modalOverlay.classList.remove('flex');
198   }
199 </script>
200 
201 </body>
202 </html>

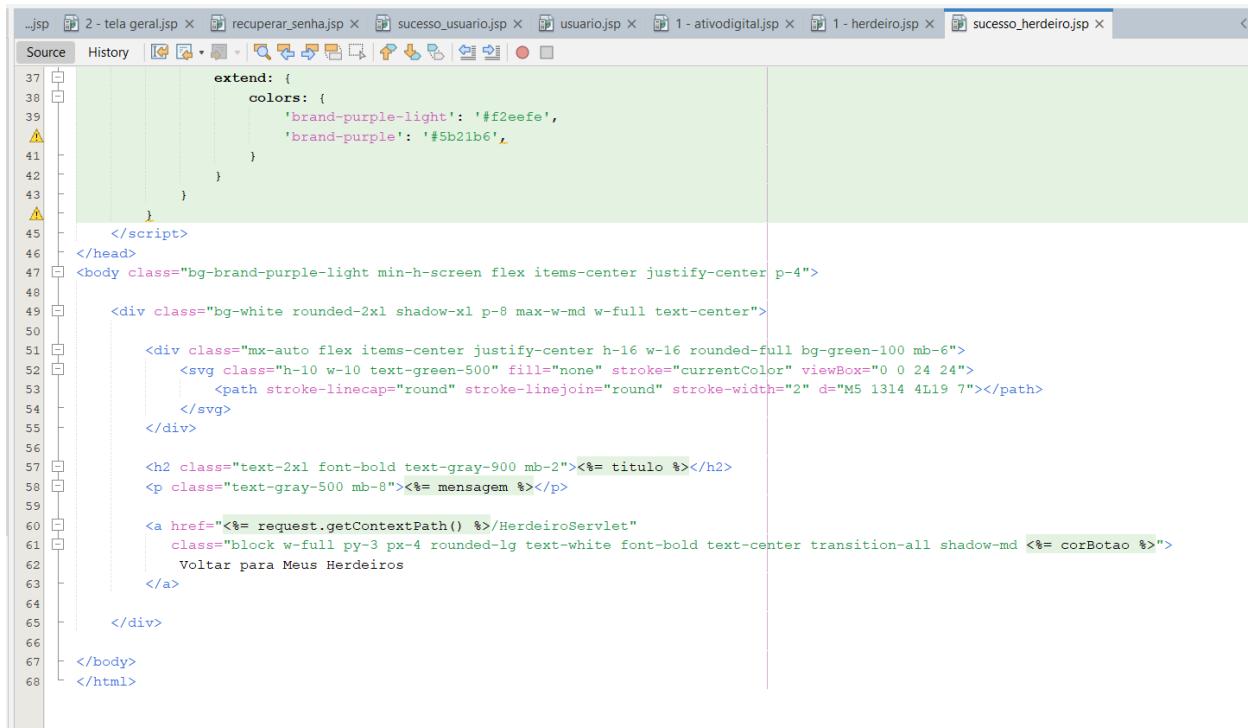
```

5.4.2 sucesso_herdeiro.jsp



The screenshot shows the Eclipse IDE interface with the JSP file "sucesso_herdeiro.jsp" selected in the tab bar. The code editor displays the JSP code, which includes scriptlets, declarations, and HTML output. The code handles different types of actions (exclusão or adição) and sets titles, messages, and button colors accordingly. It also includes meta tags for mobile devices, a title, and external CSS and font links. A tailwind.js configuration block is present at the bottom.

```
<%--  
Document : sucesso_herdeiro  
Created on : 22 de nov. de 2025, 11:31:04  
Author : TIAGO KAUAN , DIEGO HENRIQUE , NICOLY MARTINELI  
--%>  
  
<%@page contentType="text/html" pageEncoding="UTF-8"%>  
<%  
    // Verifica o tipo de ação (cadastro ou exclusão)  
    String tipo = request.getParameter("tipo");  
    String titulo = "";  
    String mensagem = "";  
    String corBotao = "";  
  
    if ("exclusao".equals(tipo)) {  
        titulo = "Herdeiro Removido!";  
        mensagem = "O herdeiro foi removido da sua lista com sucesso.";  
        corBotao = "bg-red-600 hover:bg-red-700"; // Botão vermelho para exclusão  
    } else {  
        titulo = "Herdeiro Adicionado!";  
        mensagem = "O novo herdeiro foi vinculado à sua conta com sucesso.";  
        corBotao = "bg-brand-purple hover:bg-opacity-90"; // Botão roxo padrão  
    }  
%>  
<!DOCTYPE html>  
<html lang="pt-br">  
<head>  
    <meta charset="UTF-8">  
    <meta name="viewport" content="width=device-width, initial-scale=1.0">  
    <title>Sucesso</title>  
    <script src="https://cdn.tailwindcss.com"></script>  
    <link href="https://fonts.googleapis.com/css2?family=Inter:wght@400;500;700;800&display=swap" rel="stylesheet">  
    <style> body { font-family: 'Inter', sans-serif; } </style>  
    <script>  
        tailwind.config = {  
            theme: {  
                extend: {  
                    colors: {  
                        'brand-purple-light': '#f2eefe',  
                        'brand-purple': '#5b21b6'  
                    }  
                }  
            }  
        }  
    </script>  
</head>  
<body class="bg-brand-purple-light min-h-screen flex items-center justify-center p-4">  
  
    <div class="bg-white rounded-2xl shadow-xl p-8 max-w-md w-full text-center">  
  
        <div class="mx-auto flex items-center justify-center h-16 w-16 rounded-full bg-green-100 mb-6">  
            <svg class="h-10 w-10 text-green-500" fill="none" stroke="currentColor" viewBox="0 0 24 24">  
                <path stroke-linecap="round" stroke-linejoin="round" stroke-width="2" d="M5 13l4 7L19 7" />  
            </svg>  
        </div>  
  
        <h2 class="text-2xl font-bold text-gray-900 mb-2"><%= titulo %></h2>  
        <p class="text-gray-500 mb-8"><%= mensagem %></p>  
  
        <a href="<%= request.getContextPath() %>/HerdeiroServlet"  
            class="block w-full py-3 px-4 rounded-lg text-white font-bold text-center transition-all shadow-md <%= corBotao %>">  
            Voltar para Meus Herdeiros  
        </a>  
  
    </div>  
    </body>  
</html>
```

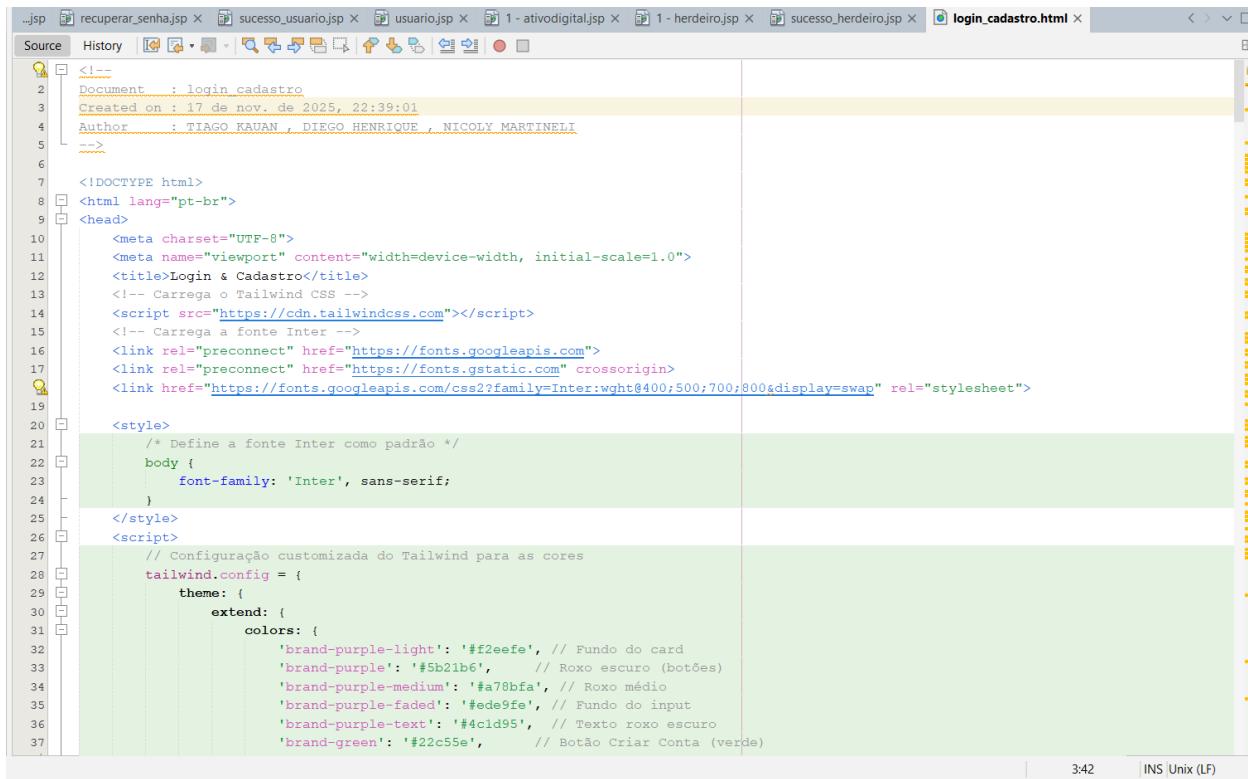


This screenshot shows the same JSP file "sucesso_herdeiro.jsp" in the Eclipse IDE, but with syntax highlighting applied. The code is color-coded to distinguish between different language elements: Java scriptlets in green, declarations in blue, HTML in grey, and CSS/JavaScript in purple. The syntax highlighting makes the code structure and components more visually distinct.

```
<%--  
Document : sucesso_herdeiro  
Created on : 22 de nov. de 2025, 11:31:04  
Author : TIAGO KAUAN , DIEGO HENRIQUE , NICOLY MARTINELI  
--%>  
  
<%@page contentType="text/html" pageEncoding="UTF-8"%>  
<%  
    // Verifica o tipo de ação (cadastro ou exclusão)  
    String tipo = request.getParameter("tipo");  
    String titulo = "";  
    String mensagem = "";  
    String corBotao = "";  
  
    if ("exclusao".equals(tipo)) {  
        titulo = "Herdeiro Removido!";  
        mensagem = "O herdeiro foi removido da sua lista com sucesso.";  
        corBotao = "bg-red-600 hover:bg-red-700"; // Botão vermelho para exclusão  
    } else {  
        titulo = "Herdeiro Adicionado!";  
        mensagem = "O novo herdeiro foi vinculado à sua conta com sucesso.";  
        corBotao = "bg-brand-purple hover:bg-opacity-90"; // Botão roxo padrão  
    }  
%>  
<!DOCTYPE html>  
<html lang="pt-br">  
<head>  
    <meta charset="UTF-8">  
    <meta name="viewport" content="width=device-width, initial-scale=1.0">  
    <title>Sucesso</title>  
    <script src="https://cdn.tailwindcss.com"></script>  
    <link href="https://fonts.googleapis.com/css2?family=Inter:wght@400;500;700;800&display=swap" rel="stylesheet">  
    <style> body { font-family: 'Inter', sans-serif; } </style>  
    <script>  
        tailwind.config = {  
            theme: {  
                extend: {  
                    colors: {  
                        'brand-purple-light': '#f2eefe',  
                        'brand-purple': '#5b21b6'  
                    }  
                }  
            }  
        }  
    </script>  
</head>  
<body class="bg-brand-purple-light min-h-screen flex items-center justify-center p-4">  
  
    <div class="bg-white rounded-2xl shadow-xl p-8 max-w-md w-full text-center">  
  
        <div class="mx-auto flex items-center justify-center h-16 w-16 rounded-full bg-green-100 mb-6">  
            <svg class="h-10 w-10 text-green-500" fill="none" stroke="currentColor" viewBox="0 0 24 24">  
                <path stroke-linecap="round" stroke-linejoin="round" stroke-width="2" d="M5 13l4 7L19 7" />  
            </svg>  
        </div>  
  
        <h2 class="text-2xl font-bold text-gray-900 mb-2"><%= titulo %></h2>  
        <p class="text-gray-500 mb-8"><%= mensagem %></p>  
  
        <a href="<%= request.getContextPath() %>/HerdeiroServlet"  
            class="block w-full py-3 px-4 rounded-lg text-white font-bold text-center transition-all shadow-md <%= corBotao %>">  
            Voltar para Meus Herdeiros  
        </a>  
  
    </div>  
    </body>  
</html>
```

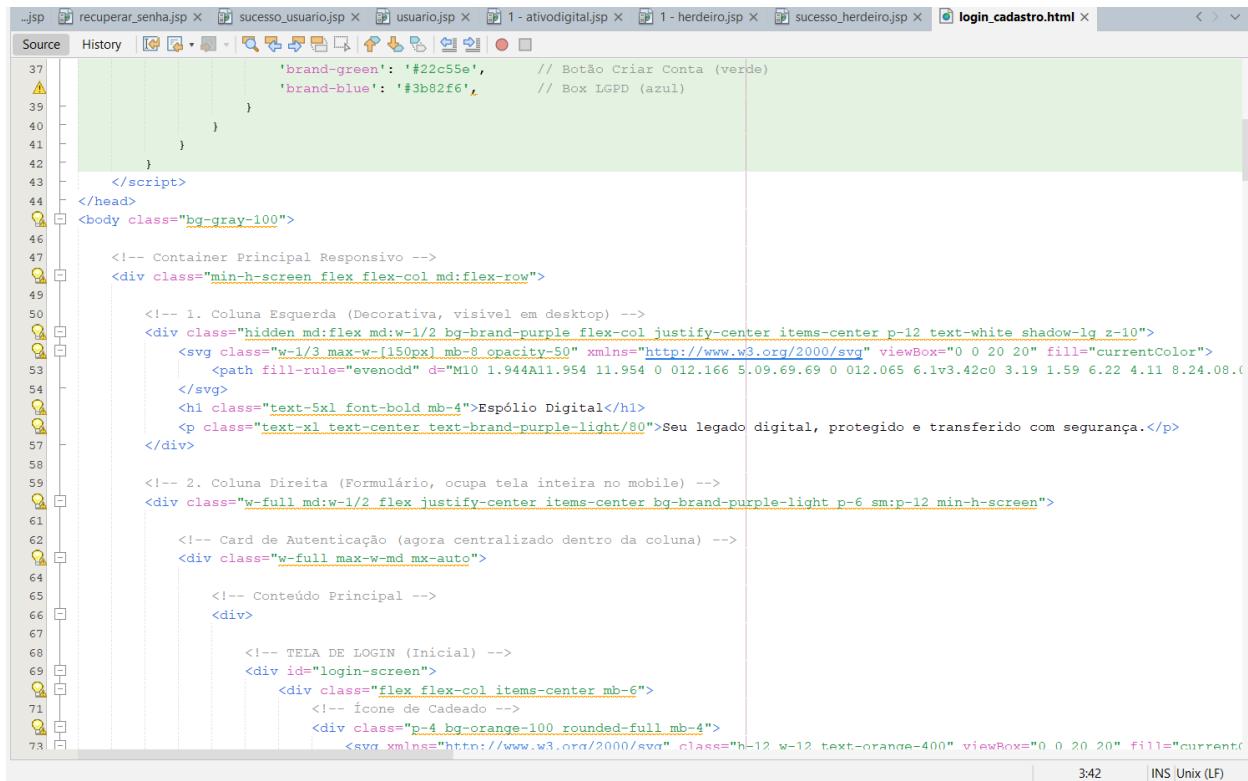
5.5. WebPages imagens

5.5.1 login_cadastro.html



```
<!--
Document : login cadastro
Created on : 17 de nov. de 2025, 22:39:01
Author : TIAGO KAUAN , DIEGO HENRIQUE , NICOLY MARTINELI
-->

<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login & Cadastro</title>
    <!-- Carrega o Tailwind CSS -->
    <script src="https://cdn.tailwindcss.com"></script>
    <!-- Carrega a fonte Inter -->
    <link rel="preconnect" href="https://fonts.googleapis.com">
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
    <link href="https://fonts.googleapis.com/css2?family=Inter:wght@400;500;700;800&display=swap" rel="stylesheet">
<style>
    /* Define a fonte Inter como padrão */
    body {
        font-family: 'Inter', sans-serif;
    }
</style>
<script>
    // Configuração customizada do Tailwind para as cores
    tailwind.config = {
        theme: {
            extend: {
                colors: {
                    'brand-purple-light': '#f2eefe', // Fundo do card
                    'brand-purple': '#5b21b6', // Roxo escuro (botões)
                    'brand-purple-medium': '#a78bfa', // Roxo médio
                    'brand-purple-faded': '#edede9e', // Fundo do input
                    'brand-purple-text': '#4c1d95', // Texto roxo escuro
                    'brand-green': '#22c55e', // Botão Criar Conta (verde)
                }
            }
        }
    }
</script>
</head>
<body class="bg-gray-100">
```



```

    <!-- Container Principal Responsivo -->
    <div class="min-h-screen flex flex-col md:flex-row">

        <!-- 1. Coluna Esquerda (Decorativa, visível em desktop) -->
        <div class="hidden md:flex md:w-1/2 bg-brand-purple flex-col justify-center items-center p-12 text-white shadow-lg z-10">
            <svg class="w-1/3 max-w-[150px] mb-8 opacity-50" xmlns="http://www.w3.org/2000/svg" viewBox="0 0 20 20" fill="currentColor">
                <path fill-rule="evenodd" d="M10 1.944A11.954 11.954 0 012.166 5.0969.69 0 012.065 6.1v3.42c0 3.19 1.59 6.22 4.11 8.24.08.08.08.11 0 1.944z" />
            </svg>
            <h1 class="text-5xl font-bold mb-4">Espólio Digital</h1>
            <p class="text-xl text-center text-brand-purple-light/80">Seu legado digital, protegido e transferido com segurança.</p>
        </div>

        <!-- 2. Coluna Direita (Formulário, ocupa tela inteira no mobile) -->
        <div class="w-full md:w-1/2 flex justify-center items-center bg-brand-purple-light p-6 sm:p-12 min-h-screen">

            <!-- Card de Autenticação (agora centralizado dentro da coluna) -->
            <div class="w-full max-w-md mx-auto">

                <!-- Conteúdo Principal -->
                <div>

                    <!-- TELA DE LOGIN (Inicial) -->
                    <div id="login-screen">
                        <div class="flex flex-col items-center mb-6">
                            <!-- ícone de Cadeado -->
                            <div class="p-4 bg-orange-100 rounded-full mb-4">
                                <svg xmlns="http://www.w3.org/2000/svg" class="h-12 w-12 text-orange-400" viewBox="0 0 20 20" fill="currentColor">
                                    <path fill-rule="evenodd" d="M10 1.944A11.954 11.954 0 012.166 5.0969.69 0 012.065 6.1v3.42c0 3.19 1.59 6.22 4.11 8.24.08.08.08.11 0 1.944z" />
                                </svg>
                            </div>
                            <div class="text-2xl font-bold mb-2">Crie sua conta</div>
                            <div class="text-sm text-gray-500">Já tenho uma conta</div>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>

```

```
...jsp recuperar_senha.jsp x suceso_usuario.jsp x usuario.jsp x 1 - ativodigital.jsp x 1 - herdeiro.jsp x suceso_herdeiro.jsp x login_cadastro.html x
Source History < > □
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
```

76 </svg>
77 </div>
78 <h1 class="text-3xl font-bold text-brand-purple-text">
79 Bem-vindo
80 </h1>
81 <!-- NOVO: Mensagem de Erro (controlada por JS) -->
82 <p id="login-error" class="text-red-600 bg-red-100 p-3 rounded-lg text-sm hidden w-full text-center mt-4">
83 <!-- Mensagem de erro aparecerá aqui -->
84 </p>
85 </div>
86
87 <!--
88 =====
89 ATUALIZADO: Formulário de login agora aponta para "LoginServlet"
90 e 'onsubmit' foi removido.
91 =====
92 -->
93 <form action="LoginServlet" method="POST">
94 <div class="space-y-4">
95 <div>
96 <label for="login-cpf" class="block text-sm font-medium text-brand-purple-text mb-1">CPF</label>
97 <input type="text" name="cpf" id="login-cpf" placeholder="Apenas números" required
98 maxLength="11" oninput="this.value = this.value.replace(/[^0-9]/g, '') .slice(0, 11)">
99 class="w-full p-4 bg-brand-purple-faded rounded-lg border border-brand-purple-medium/30 placeholder-100">
100 </div>
101 <div>
102 <div class="flex justify-between items-center mb-1">
103 <label for="login-senha" class="block text-sm font-medium text-brand-purple-text">Senha</label>
104 <button type="button" onclick="showRecovery()" class="text-sm font-medium text-brand-purple hover:underline underline-offset-2">
105 Esqueceu a senha?
106 </button>
107 </div>
108 <input type="password" name="senha" id="login-senha" placeholder="Sua senha" required
109 class="w-full p-4 bg-brand-purple-faded rounded-lg border border-brand-purple-medium/30 placeholder-110">
110 </div>
111 </div>

3:42 | INS | Unix (LF)

```
...jsp recuperar_senha.jsp x suceso_usuario.jsp x usuario.jsp x 1 - ativodigital.jsp x 1 - herdeiro.jsp x suceso_herdeiro.jsp x login_cadastro.html x
Source History < > □
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
```

110 <div class="flex items-center my-6">
111 <div class="flex items-center">
112 <input type="checkbox" id="lembra-me" name="lembra-me" checked="" value="checked" class="h-4 w-4 text-brand-purple rounded border border-brand-purple-medium/30 placeholder-113">
113 <label for="lembra-me" class="ml-2 block text-sm text-brand-purple-text">
114 Lembrar-me
115 </label>
116 </div>
117 <div class="space-y-3">
118 <button type="submit" class="w-full p-4 bg-brand-purple text-white rounded-lg font-bold text-lg hover:bg-opacity-90 transition duration-200 ease-in-out">
119 Entrar
120 </button>
121 <button type="button" onclick="showRegister()" class="w-full p-4 bg-brand-purple-faded text-brand-purple-text rounded-lg font-bold text-lg hover:bg-br-122 ase-in-out">
122 Criar Conta
123 </button>
124 </div>
125
126 <p class="text-center text-brand-purple-medium text-sm mt-6 flex items-center justify-center gap-1">
127
128 Login protegido com biometria
129 </p>
130 </div>
131
132 <!-- TELA DE CADASTRO (Oculta e ATUALIZADA) -->
133 <div id="register-screen" class="hidden">
134 <h1 class="text-2xl font-bold text-brand-purple-text mb-6 text-center">
135 Criar Conta Segura
136 </h1>

...jsp recuperar_senha.jsp x sucesso_usuario.jsp x usuario.jsp x 1 - ativodigital.jsp x 1 - herdeiro.jsp x sucesso_herdeiro.jsp x login_cadastro.html x

Source History

```

145 |         Criar Conta Segura
146 |
147 |         <!-- Formulário de cadastro ATUALIZADO -->
148 |         <!-- 1. 'action' aponta para o novo JSP -->
149 |         <!-- 2. 'method' é POST -->
150 |         <!-- 3. 'onsubmit' chama nossa nova função de validação -->
151 |         <form id="form-cadastro" action="1 - usuario/usuario.jsp" method="POST" onsubmit="return validarCadastro(event);">
152 |             <!-- CAMPOS ATUALIZADOS -->
153 |             <div class="space-y-4">
154 |
155 |                 <!-- Campo: Nome Completo -->
156 |                 <div>
157 |                     <label for="reg-nome" class="block text-sm font-medium text-brand-purple-text mb-1">Nome Completo *</label>
158 |                     <input type="text" name="nome" id="reg-nome" placeholder="Seu nome completo" required
159 |                         class="w-full p-4 bg-brand-purple-faded rounded-lg border border-brand-purple-medium/30 placeholder-white">
160 |                 </div>
161 |
162 |
163 |                 <!-- Campo: CPF (Novo) -->
164 |                 <div>
165 |                     <label for="reg-cpf" class="block text-sm font-medium text-brand-purple-text mb-1">CPF *</label>
166 |                     <input type="text" name="cpf" id="reg-cpf" placeholder="Apenas números" required
167 |                         maxlength="11" oninput="this.value = this.value.replace(/[^0-9]/g, '').slice(0, 11)"
168 |                         class="w-full p-4 bg-brand-purple-faded rounded-lg border border-brand-purple-medium/30 placeholder-white">
169 |                 </div>
170 |
171 |
172 |                 <!-- Campo: Senha -->
173 |                 <div>
174 |                     <label for="reg-senha" class="block text-sm font-medium text-brand-purple-text mb-1">Senha Segura *</label>
175 |                     <input type="password" name="senha" id="reg-senha" placeholder="Crie uma senha forte" required
176 |                         class="w-full p-4 bg-brand-purple-faded rounded-lg border border-brand-purple-medium/30 placeholder-white">
177 |                 </div>
178 |
179 |
180 |                 <!-- Campo: Confirmar Senha -->
181 |                 <div>
182 |                     <label for="reg-confirm-senha" class="block text-sm font-medium text-brand-purple-text mb-1">Confirmar S
183 |                     <input type="password" name="confirmar_senha" id="reg-confirm-senha" placeholder="Confirme sua senha" re
184 |                 </div>
185 |
186 |
187 |             <!-- NOVO: Mensagem de erro de validação -->
188 |             <p id="reg-error" class="text-red-600 text-sm -mt-2 hidden"></p>
189 |
190 |
191 |             <!-- CAMPO PALAVRA-CHAVE -->
192 |             <div>
193 |                 <label for="reg-palavra-chave" class="block text-sm font-medium text-brand-purple-text mb-1">Palavra-Chave
194 |                 <input type="text" name="palavra_chave" id="reg-palavra-chave" placeholder="Uma palavra única para recuperar sua conta" required
195 |                     class="w-full p-4 bg-brand-purple-faded rounded-lg border border-brand-purple-medium/30 placeholder-white">
196 |                 <p class="text-xs text-brand-purple-text/70 mt-1">Guarde bem esta palavra. Ela será essencial caso você precise recup
197 |             </div>
198 |
199 |
200 |             <!-- Box LGPD -->
201 |             <div class="bg-brand-blue text-white p-3 rounded-lg flex items-start gap-3 my-6 text-sm">
202 |                 <img alt="LGPD logo" data-bbox="318 668 408 728" style="width: 100px; height: 60px; margin-right: 10px;" />
203 |                 <span>LGPD:</span> Dados criptografados AES-256 e usados apenas para gestão do legado.
204 |             </div>
205 |
206 |
207 |             <!-- Checkbox Termos -->
208 |             <div class="flex items-center mb-6">
209 |                 <input id="termos" name="termos" type="checkbox" required
210 |                     class="h-4 w-4 text-brand-purple rounded border-gray-300 focus:ring-brand-purple">
211 |                 <label for="termos" class="ml-2 block text-sm text-brand-purple-text">
212 |                     Aceito os termos e política de privacidade
213 |                 </label>
214 |             </div>
215 |
216 |
217 |             <!-- Botões Ação -->

```

...jsp recuperar_senha.jsp x sucesso_usuario.jsp x usuario.jsp x 1 - ativodigital.jsp x 1 - herdeiro.jsp x sucesso_herdeiro.jsp x login_cadastro.html x

Source History

```

181 |             <input type="password" name="confirmar_senha" id="reg-confirm-senha" placeholder="Confirme sua senha" required
182 |                         class="w-full p-4 bg-brand-purple-faded rounded-lg border border-brand-purple-medium/30 placeholder-white">
183 |         </div>
184 |
185 |
186 |         <!-- NOVO: Mensagem de erro de validação -->
187 |         <p id="reg-error" class="text-red-600 text-sm -mt-2 hidden"></p>
188 |
189 |
190 |         <!-- CAMPO PALAVRA-CHAVE -->
191 |         <div>
192 |             <label for="reg-palavra-chave" class="block text-sm font-medium text-brand-purple-text mb-1">Palavra-Chave
193 |             <input type="text" name="palavra_chave" id="reg-palavra-chave" placeholder="Uma palavra única para recuperar sua conta" required
194 |                 class="w-full p-4 bg-brand-purple-faded rounded-lg border border-brand-purple-medium/30 placeholder-white">
195 |             <p class="text-xs text-brand-purple-text/70 mt-1">Guarde bem esta palavra. Ela será essencial caso você precise recup
196 |         </div>
197 |
198 |
199 |         <!-- Box LGPD -->
200 |         <div class="bg-brand-blue text-white p-3 rounded-lg flex items-start gap-3 my-6 text-sm">
201 |             <img alt="LGPD logo" data-bbox="318 668 408 728" style="width: 100px; height: 60px; margin-right: 10px;" />
202 |             <span>LGPD:</span> Dados criptografados AES-256 e usados apenas para gestão do legado.
203 |         </div>
204 |
205 |
206 |         <!-- Checkbox Termos -->
207 |         <div class="flex items-center mb-6">
208 |             <input id="termos" name="termos" type="checkbox" required
209 |                 class="h-4 w-4 text-brand-purple rounded border-gray-300 focus:ring-brand-purple">
210 |             <label for="termos" class="ml-2 block text-sm text-brand-purple-text">
211 |                 Aceito os termos e política de privacidade
212 |             </label>
213 |         </div>
214 |
215 |
216 |         <!-- Botões Ação -->

```

```
...Jsp recuperar_senha.jsp x sucesso_usuario.jsp x usuario.jsp x 1-ativodigital.jsp x 1-herdeiro.jsp x sucesso_herdeiro.jsp x login_cadastro.html x
Source History < > □
217
219
221
222
223
225
226
227
229
230
231
232
233
235
236
237
238
239
240
241
242
243
245
247
248
249
251
252
254
255
256
257
259
261
262
264
265
266
268
269
271
272
273
274
276
277
279
280
281
283
284
285
286
288
<!-- Botões Ação -->
<div class="space-y-3">
    <button type="submit"
        class="w-full p-4 bg-brand-green text-white rounded-lg font-bold text-lg hover:bg-opacity-90 transition-all border border-brand-green rounded-lg border-brand-green-medium/30 placeholder-white">Criar Conta</button>
    <button type="button" onclick="showLogin()"
        class="w-full p-4 bg-brand-purple text-white rounded-lg font-bold text-lg hover:bg-opacity-90 transition-all border border-brand-purple rounded-lg border-brand-purple-medium/30 placeholder-white">Já tenho conta</button>
</div>
</form>
</div>

<!-- ===== TELA DE RECUPERAÇÃO (ATUALIZADA) ===== -->
<div id="recovery-screen" class="hidden">
    <h1 class="text-2xl font-bold text-brand-purple-text mb-6 text-center">Recuperar Senha</h1>

    <!-- Formulário de Recuperação ATUALIZADO -->
    <!-- 1. 'action' aponta para o novo JSP de recuperação -->
    <!-- 2. 'method' é POST -->
    <!-- 3. 'onsubmit' chama a nova validação 'validarRecuperacao' -->
    <form id="form-recuperacao" action="1 - usuario/recuperar_senha.jsp" method="POST" onsubmit="return validarRecuperacao()">
        <div class="space-y-4">
            <p class="text-sm text-brand-purple-text/80 text-center">Insira seu CPF e sua palavra-chave de recuperação para recuperar sua senha!</p>
            <!-- Campo: CPF (ATUALIZADO) -->
            <div>
                <label for="rec-cpf" class="block text-sm font-medium text-brand-purple-text mb-1">CPF *</label>
                <input type="text" name="cpf" id="rec-cpf" placeholder="Apenas números" required
                    maxlength="11" oninput="this.value = this.value.replace(/[^0-9]/g, '').slice(0, 11)">
                <div class="w-full p-4 bg-brand-purple-faded rounded-lg border border-brand-purple-medium/30 placeholder-white"></div>
            </div>
            <!-- Campo: Palavra-Chave -->
            <div>
                <label for="rec-palavra-chave" class="block text-sm font-medium text-brand-purple-text mb-1">Palavra-Chave *</label>
                <input type="text" name="palavra_chave" id="rec-palavra-chave" placeholder="Sua palavra-chave secreta" required
                    class="w-full p-4 bg-brand-purple-faded rounded-lg border border-brand-purple-medium/30 placeholder-white">
                <div class="w-full p-4 bg-brand-purple-faded rounded-lg border border-brand-purple-medium/30 placeholder-white"></div>
            </div>
            <hr class="border-t border-brand-purple-medium/20 my-4">
            <!-- Campo: Nova Senha -->
            <div>
                <label for="rec-senha" class="block text-sm font-medium text-brand-purple-text mb-1">Nova Senha *</label>
                <!-- ATUALIZADO: 'name' para 'senha_nova' -->
                <input type="password" name="senha_nova" id="rec-senha" placeholder="Crie uma nova senha forte" required
                    class="w-full p-4 bg-brand-purple-faded rounded-lg border border-brand-purple-medium/30 placeholder-white">
                <div class="w-full p-4 bg-brand-purple-faded rounded-lg border border-brand-purple-medium/30 placeholder-white"></div>
            </div>
            <!-- Campo: Confirmar Nova Senha -->
            <div>
                <label for="rec-confirm-senha" class="block text-sm font-medium text-brand-purple-text mb-1">Confirmar Nova Senha *</label>
                <!-- ATUALIZADO: 'name' para 'confirmar_senha_nova' -->
                <input type="password" name="confirmar_senha_nova" id="rec-confirm-senha" placeholder="Confirme sua nova senha" required
                    class="w-full p-4 bg-brand-purple-faded rounded-lg border border-brand-purple-medium/30 placeholder-white">
                <div class="w-full p-4 bg-brand-purple-faded rounded-lg border border-brand-purple-medium/30 placeholder-white"></div>
            </div>
            <!-- NOVO: Mensagem de erro de validação -->
            <p id="rec-error" class="text-red-600 text-sm mt-2 hidden"></p>
        </div>
        <!-- Botões Ação -->
        <div class="space-y-3 mt-8">
            <button type="submit"
                class="w-full p-4 bg-brand-green text-white rounded-lg font-bold text-lg hover:bg-opacity-90 transition-all border border-brand-green rounded-lg border-brand-green-medium/30 placeholder-white">Recuperar Senha</button>
        </div>
    </form>
</div>
```

```
...Jsp recuperar_senha.jsp x sucesso_usuario.jsp x usuario.jsp x 1-ativodigital.jsp x 1-herdeiro.jsp x sucesso_herdeiro.jsp x login_cadastro.html x
Source History < > □
254
255
256
257
259
261
262
264
265
266
268
269
271
272
273
274
276
277
279
280
281
283
284
285
286
288
<!-- Campo: CPF (ATUALIZADO) -->
<div>
    <label for="rec-cpf" class="block text-sm font-medium text-brand-purple-text mb-1">CPF *</label>
    <input type="text" name="cpf" id="rec-cpf" placeholder="Apenas números" required
        maxlength="11" oninput="this.value = this.value.replace(/[^0-9]/g, '').slice(0, 11)">
    <div class="w-full p-4 bg-brand-purple-faded rounded-lg border border-brand-purple-medium/30 placeholder-white"></div>
</div>
<!-- Campo: Palavra-Chave -->
<div>
    <label for="rec-palavra-chave" class="block text-sm font-medium text-brand-purple-text mb-1">Palavra-Chave *</label>
    <input type="text" name="palavra_chave" id="rec-palavra-chave" placeholder="Sua palavra-chave secreta" required
        class="w-full p-4 bg-brand-purple-faded rounded-lg border border-brand-purple-medium/30 placeholder-white">
    <div class="w-full p-4 bg-brand-purple-faded rounded-lg border border-brand-purple-medium/30 placeholder-white"></div>
</div>
<hr class="border-t border-brand-purple-medium/20 my-4">
<!-- Campo: Nova Senha -->
<div>
    <label for="rec-senha" class="block text-sm font-medium text-brand-purple-text mb-1">Nova Senha *</label>
    <!-- ATUALIZADO: 'name' para 'senha_nova' -->
    <input type="password" name="senha_nova" id="rec-senha" placeholder="Crie uma nova senha forte" required
        class="w-full p-4 bg-brand-purple-faded rounded-lg border border-brand-purple-medium/30 placeholder-white">
    <div class="w-full p-4 bg-brand-purple-faded rounded-lg border border-brand-purple-medium/30 placeholder-white"></div>
</div>
<!-- Campo: Confirmar Nova Senha -->
<div>
    <label for="rec-confirm-senha" class="block text-sm font-medium text-brand-purple-text mb-1">Confirmar Nova Senha *</label>
    <!-- ATUALIZADO: 'name' para 'confirmar_senha_nova' -->
    <input type="password" name="confirmar_senha_nova" id="rec-confirm-senha" placeholder="Confirme sua nova senha" required
        class="w-full p-4 bg-brand-purple-faded rounded-lg border border-brand-purple-medium/30 placeholder-white">
    <div class="w-full p-4 bg-brand-purple-faded rounded-lg border border-brand-purple-medium/30 placeholder-white"></div>
</div>
<!-- NOVO: Mensagem de erro de validação -->
<p id="rec-error" class="text-red-600 text-sm mt-2 hidden"></p>
<!-- Botões Ação -->
<div class="space-y-3 mt-8">
    <button type="submit"
        class="w-full p-4 bg-brand-green text-white rounded-lg font-bold text-lg hover:bg-opacity-90 transition-all border border-brand-green rounded-lg border-brand-green-medium/30 placeholder-white">Recuperar Senha</button>
</div>
```

The screenshot shows the 'Source' tab of a browser's developer tools. The file being viewed is 'login_cadastro.html'. The code is a combination of HTML and JavaScript. The HTML part includes a button with the onclick event set to 'showLogin()'. The JavaScript part handles error messages and screen transitions. A yellow warning icon is present near the start of the script block.

```
292         <button type="button" onclick="showLogin()" class="w-full p-4 bg-brand-purple-faded text-brand-purple-text rounded-lg font-bold text-lg hover:bg-bran...
294             Cancelar
295         </button>
296     </div>
297     </form>
298   </div>
299   <!-- ===== FIM DA TELA DE RECUPERAÇÃO ===== -->
300 
301   </div>
302   </div>
303 </div>
304 
305 </div>
306 
307 </div>
308 
309 <script>
310     // --- ATUALIZADO: Script para mostrar erro de login ---
311     document.addEventListener('DOMContentLoaded', (event) => {
312         const params = new URLSearchParams(window.location.search);
313         if (params.has('erro')) {
314             const errorMsg = document.getElementById("login-error");
315             if(params.get('erro') == '1') {
316                 errorMsg.innerText = 'CPF ou senha inválidos. Tente novamente.';
317             } else {
318                 errorMsg.innerText = 'Ocorreu um erro inesperado no servidor.';
319             }
320             errorMsg.classList.remove('hidden');
321         }
322     });
323 
324 
325     // --- JavaScript para alternar telas (ATUALIZADO) ---
326 
327     const loginScreen = document.getElementById('login-screen');
328     const registerScreen = document.getElementById('register-screen');
```

This screenshot shows the same 'Source' tab as the previous one, but with a different portion of the code visible. It contains several functions: 'showLogin()', 'showRegister()', 'showRecovery()', and 'validarCadastro(event)'. The 'validarCadastro' function includes validation logic for password matching and prevents form submission if the passwords don't match. A yellow warning icon is located near the top of the script block.

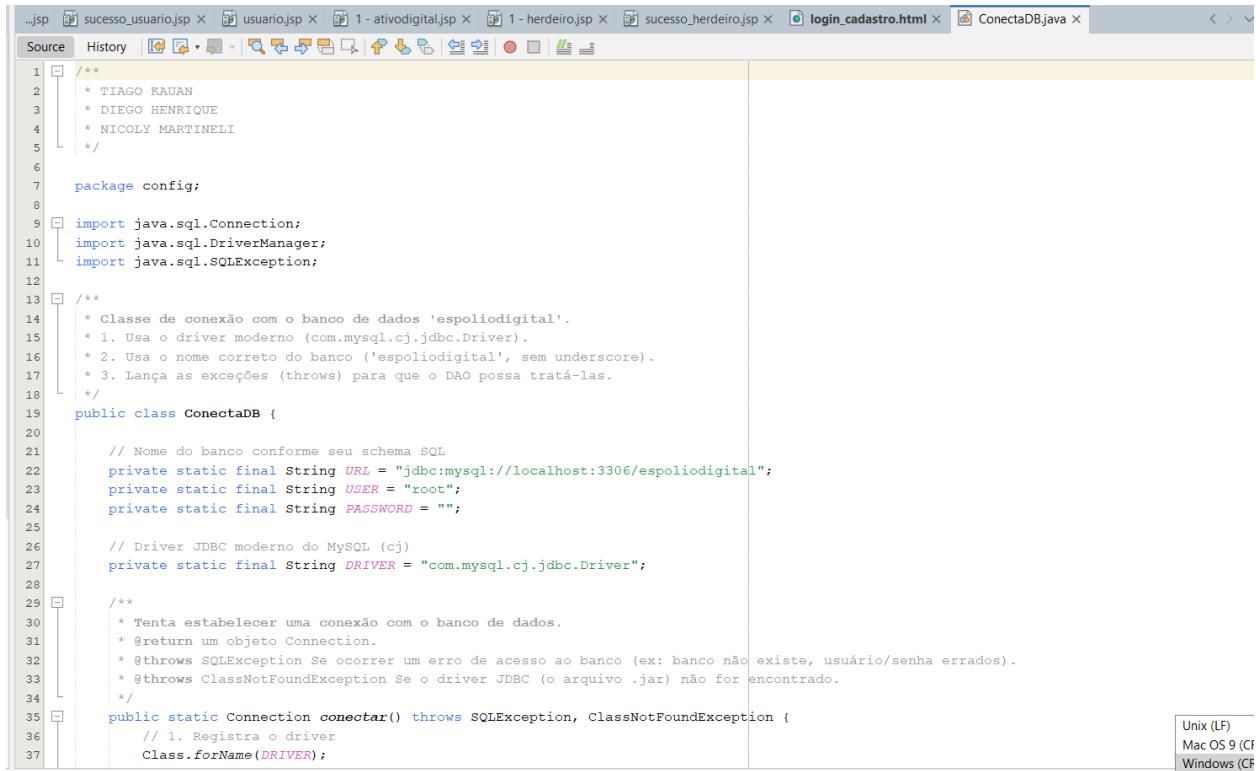
```
328     const registerScreen = document.getElementById('register-screen');
329     const recoveryScreen = document.getElementById('recovery-screen'); // Nova tela
330 
331     function showLogin() {
332         loginScreen.classList.remove('hidden');
333         registerScreen.classList.add('hidden');
334         recoveryScreen.classList.add('hidden'); // Oculta recuperação
335     }
336 
337     function showRegister() {
338         loginScreen.classList.add('hidden');
339         registerScreen.classList.remove('hidden');
340         recoveryScreen.classList.add('hidden'); // Oculta recuperação
341     }
342 
343     function showRecovery() {
344         loginScreen.classList.add('hidden');
345 
346         registerScreen.classList.add('hidden');
347         recoveryScreen.classList.remove('hidden'); // Mostra recuperação
348     }
349 
350     /**
351      * SCRIPT DE VALIDAÇÃO DE CADASTRO
352      * Valida o formulário de cadastro antes de enviar para o JSP.
353      * Impede o envio se as senhas não baterem.
354      */
355     function validarCadastro(event) {
356         const senha = document.getElementById('reg-senha').value;
357         const confirmaSenha = document.getElementById('reg-confirm-senha').value;
358         const errorMsg = document.getElementById('reg-error');
359 
360         if (senha !== confirmaSenha) {
361             // Impede o envio do formulário
362             event.preventDefault();
363             errorMsg.innerText = 'As senhas não conferem. Tente novamente.';
364             errorMsg.classList.remove('hidden'); // Mostra a mensagem de erro
```

```
358     const errorMsg = document.getElementById('reg-error');
359
360     if (senha !== confirmaSenha) {
361         // Impede o envio do formulário
362         event.preventDefault();
363         errorMsg.innerText = 'As senhas não conferem. Tente novamente.';
364         errorMsg.classList.remove('hidden'); // Mostra a mensagem de erro
365         return false;
366     }
367
368     // Se as senhas forem iguais
369     errorMsg.classList.add('hidden'); // Oculta a mensagem de erro
370     return true; // Permite o envio
371 }
372
373 /**
374 * NOVO SCRIPT
375 * Valida o formulário de RECUPERAÇÃO antes de enviar para o JSP.
376 * Impede o envio se as NOVAS senhas não baterem.
377 */
378
379 function validarRecuperacao(event) {
380     const senha = document.getElementById('rec-senha').value;
381     const confirmaSenha = document.getElementById('rec-confirm-senha').value;
382     const errorMsg = document.getElementById('rec-error');
383
384     if (senha !== confirmaSenha) {
385         // Impede o envio do formulário
386         event.preventDefault();
387         errorMsg.innerText = 'As novas senhas não conferem. Tente novamente.';
388         errorMsg.classList.remove('hidden'); // Mostra a mensagem de erro
389         return false;
390     }
391
392     // Se as senhas forem iguais
393     errorMsg.classList.add('hidden'); // Oculta a mensagem de erro
394     return true;
395 }
```

```
376     * Impede o envio se as NOVAS senhas não baterem.
377 */
378
379 function validarRecuperacao(event) {
380     const senha = document.getElementById('rec-senha').value;
381     const confirmaSenha = document.getElementById('rec-confirm-senha').value;
382     const errorMsg = document.getElementById('rec-error');
383
384     if (senha !== confirmaSenha) {
385         // Impede o envio do formulário
386         event.preventDefault();
387         errorMsg.innerText = 'As novas senhas não conferem. Tente novamente.';
388         errorMsg.classList.remove('hidden'); // Mostra a mensagem de erro
389         return false;
390     }
391
392     // Se as senhas forem iguais
393     errorMsg.classList.add('hidden'); // Oculta a mensagem de erro
394     return true;
395 }
396
397 </script>
398
399 </body>
</html>
```

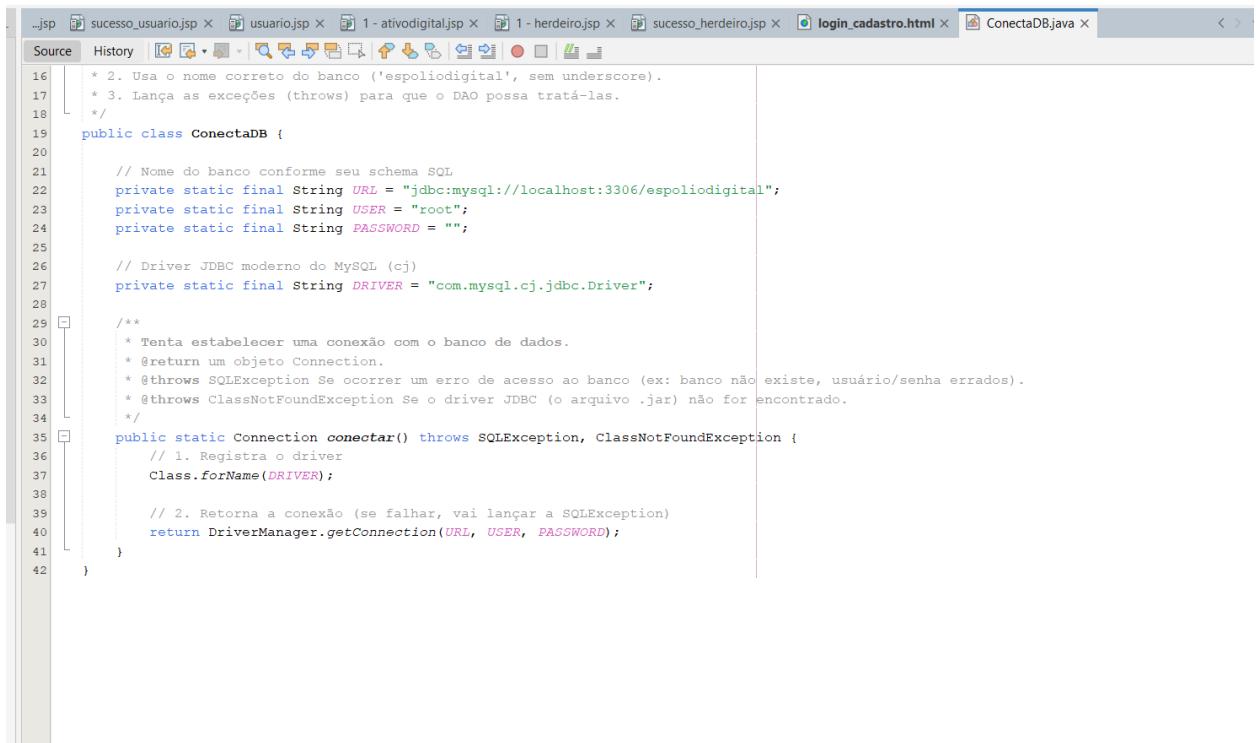
5.6. Source Package Config

5.6.1 ConectaDB.java



```
1 /**
2  * TIAGO KAUAN
3  * DIEGO HENRIQUE
4  * NICOLY MARTINELI
5  */
6
7 package config;
8
9 import java.sql.Connection;
10 import java.sql.DriverManager;
11 import java.sql.SQLException;
12
13 /**
14  * Classe de conexão com o banco de dados 'espoliodigital'.
15  * 1. Usa o driver moderno (com.mysql.cj.jdbc.Driver).
16  * 2. Usa o nome correto do banco ('espoliodigital', sem underscore).
17  * 3. Lança as exceções (throws) para que o DAO possa tratá-las.
18 */
19 public class ConectaDB {
20
21     // Nome do banco conforme seu schema SQL
22     private static final String URL = "jdbc:mysql://localhost:3306/espoliodigital";
23     private static final String USER = "root";
24     private static final String PASSWORD = "";
25
26     // Driver JDBC moderno do MySQL (cj)
27     private static final String DRIVER = "com.mysql.cj.jdbc.Driver";
28
29 /**
30  * Tenta estabelecer uma conexão com o banco de dados.
31  * @return um objeto Connection.
32  * @throws SQLException Se ocorrer um erro de acesso ao banco (ex: banco não existe, usuário/senha errados).
33  * @throws ClassNotFoundException Se o driver JDBC (o arquivo .jar) não for encontrado.
34 */
35 public static Connection conectar() throws SQLException, ClassNotFoundException {
36     // 1. Registra o driver
37     Class.forName(DRIVER);
38
39     // 2. Retorna a conexão (se falhar, vai lançar a SQLException)
40     return DriverManager.getConnection(URL, USER, PASSWORD);
41 }
42 }
```

Unix (LF)
Mac OS 9 (C)
Windows (C)



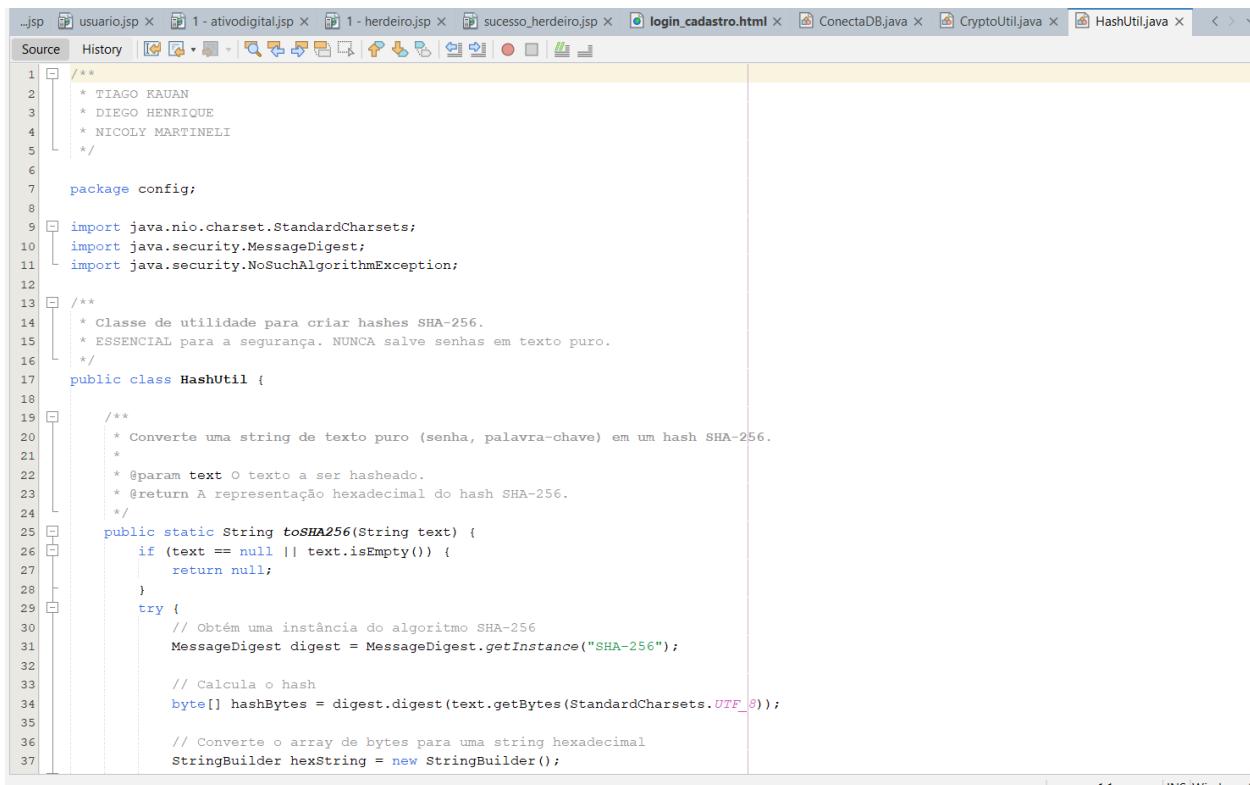
```
16 /**
17  * 2. Usa o nome correto do banco ('espoliodigital', sem underscore).
18  * 3. Lança as exceções (throws) para que o DAO possa tratá-las.
19 */
20
21 public class ConectaDB {
22
23     // Nome do banco conforme seu schema SQL
24     private static final String URL = "jdbc:mysql://localhost:3306/espoliodigital";
25     private static final String USER = "root";
26     private static final String PASSWORD = "";
27
28     // Driver JDBC moderno do MySQL (cj)
29     private static final String DRIVER = "com.mysql.cj.jdbc.Driver";
30
31 /**
32  * Tenta estabelecer uma conexão com o banco de dados.
33  * @return um objeto Connection.
34  * @throws SQLException Se ocorrer um erro de acesso ao banco (ex: banco não existe, usuário/senha errados).
35  * @throws ClassNotFoundException Se o driver JDBC (o arquivo .jar) não for encontrado.
36 */
37 public static Connection conectar() throws SQLException, ClassNotFoundException {
38     // 1. Registra o driver
39     Class.forName(DRIVER);
40
41     // 2. Retorna a conexão (se falhar, vai lançar a SQLException)
42     return DriverManager.getConnection(URL, USER, PASSWORD);
43 }
```

5.6.2 CryptoUtil.java

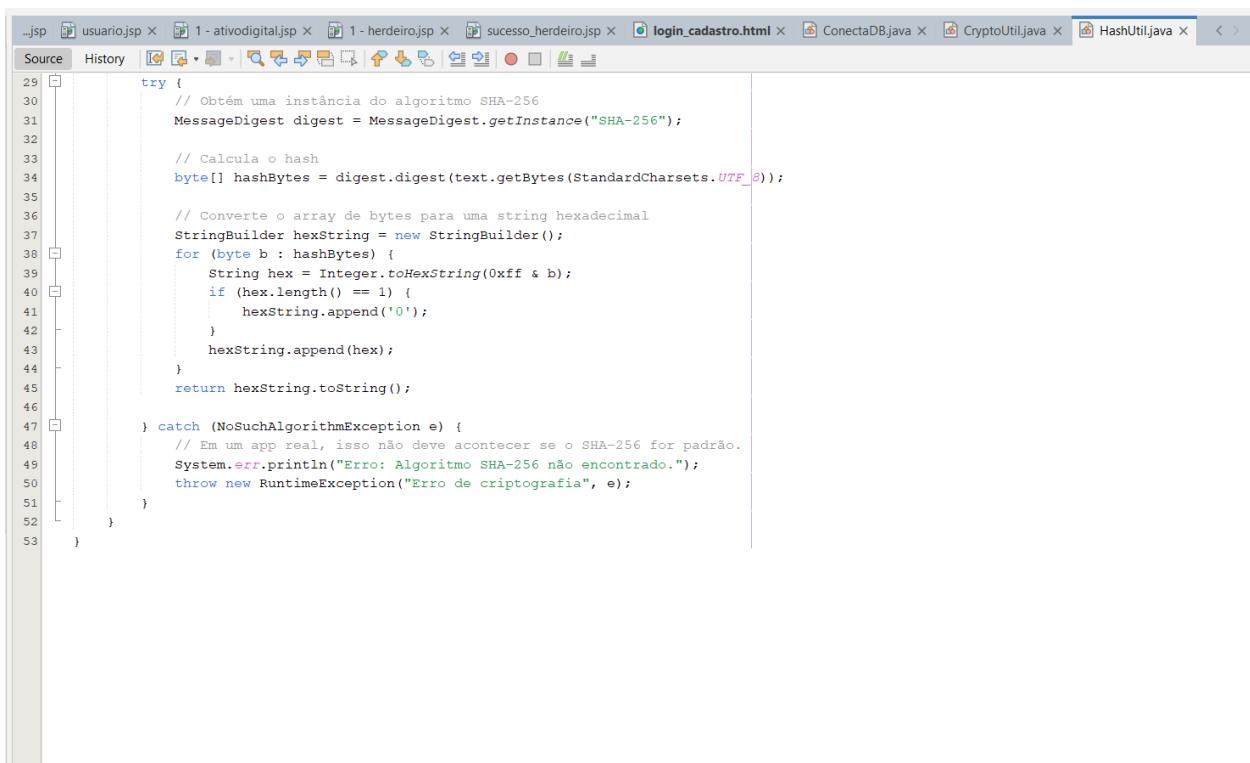
```
1  /**
2   * TIAGO KAUAN
3   * DIEGO HENRIQUE
4   * NICOLY MARTINELI
5   */
6
7  package config;
8
9  import java.util.Base64;
10 import javax.crypto.Cipher;
11 import javax.crypto.spec.SecretKeySpec;
12
13 public class CryptoUtil {
14
15     // CHAVE SECRETA DO SISTEMA (Deve ter 16 caracteres exatos para AES-128)
16     // Em produção, isso ficaria em uma variável de ambiente, mas para o projeto acadêmico pode ficar aqui.
17     private static final String CHAVE_MESTRA = "EspolioDigital125"; // 16 chars
18
19     public static String criptografar(String valorOriginal) {
20         if (valorOriginal == null || valorOriginal.isEmpty()) {
21             return null;
22         }
23         try {
24             SecretKeySpec secretKey = new SecretKeySpec(CHAVE_MESTRA.getBytes(), "AES");
25             Cipher cipher = Cipher.getInstance("AES");
26             cipher.init(Cipher.ENCRYPT_MODE, secretKey);
27
28             byte[] dadosCriptografados = cipher.doFinal(valorOriginal.getBytes());
29             // Retorna em Base64 para poder salvar no banco como String/Varchar
30             return Base64.getEncoder().encodeToString(dadosCriptografados);
31         } catch (Exception e) {
32             e.printStackTrace();
33             return null;
34         }
35     }
36
37     public static String descriptografar(String valorCriptografado) {
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53 }
```

```
25     Cipher cipher = Cipher.getInstance("AES");
26     cipher.init(Cipher.ENCRYPT_MODE, secretKey);
27
28     byte[] dadosCriptografados = cipher.doFinal(valorOriginal.getBytes());
29     // Retorna em Base64 para poder salvar no banco como String/Varchar
30     return Base64.getEncoder().encodeToString(dadosCriptografados);
31 } catch (Exception e) {
32     e.printStackTrace();
33     return null;
34 }
35
36
37     public static String descriptografar(String valorCriptografado) {
38         if (valorCriptografado == null || valorCriptografado.isEmpty()) {
39             return null;
40         }
41         try {
42             SecretKeySpec secretKey = new SecretKeySpec(CHAVE_MESTRA.getBytes(), "AES");
43             Cipher cipher = Cipher.getInstance("AES");
44             cipher.init(Cipher.DECRYPT_MODE, secretKey);
45
46             byte[] dadosOriginais = cipher.doFinal(Base64.getDecoder().decode(valorCriptografado));
47             return new String(dadosOriginais);
48         } catch (Exception e) {
49             e.printStackTrace();
50             return null;
51         }
52     }
53 }
```

5.6.3 HashUtil.java



```
1  /**
2  * TIAGO KAUAN
3  * DIEGO HENRIQUE
4  * NICOLY MARTINELI
5  */
6
7  package config;
8
9  import java.nio.charset.StandardCharsets;
10 import java.security.MessageDigest;
11 import java.security.NoSuchAlgorithmException;
12
13 /**
14 * Classe de utilidade para criar hashes SHA-256.
15 * ESSENCIAL para a segurança. NUNCA salve senhas em texto puro.
16 */
17 public class HashUtil {
18
19     /**
20      * Converte uma string de texto puro (senha, palavra-chave) em um hash SHA-256.
21      *
22      * @param text O texto a ser hasheado.
23      * @return A representação hexadecimal do hash SHA-256.
24     */
25     public static String toSHA256(String text) {
26         if (text == null || text.isEmpty()) {
27             return null;
28         }
29         try {
30             // Obtém uma instância do algoritmo SHA-256
31             MessageDigest digest = MessageDigest.getInstance("SHA-256");
32
33             // Calcula o hash
34             byte[] hashBytes = digest.digest(text.getBytes(StandardCharsets.UTF_8));
35
36             // Converte o array de bytes para uma string hexadecimal
37             StringBuilder hexString = new StringBuilder();
38
39             for (byte b : hashBytes) {
40                 String hex = Integer.toHexString(0xff & b);
41                 if (hex.length() == 1) {
42                     hexString.append('0');
43                 }
44                 hexString.append(hex);
45             }
46             return hexString.toString();
47
48         } catch (NoSuchAlgorithmException e) {
49             // Em um app real, isso não deve acontecer se o SHA-256 for padrão.
50             System.err.println("Erro: Algoritmo SHA-256 não encontrado.");
51             throw new RuntimeException("Erro de criptografia", e);
52         }
53     }
54 }
```



```
29     try {
30         // Obtém uma instância do algoritmo SHA-256
31         MessageDigest digest = MessageDigest.getInstance("SHA-256");
32
33         // Calcula o hash
34         byte[] hashBytes = digest.digest(text.getBytes(StandardCharsets.UTF_8));
35
36         // Converte o array de bytes para uma string hexadecimal
37         StringBuilder hexString = new StringBuilder();
38         for (byte b : hashBytes) {
39             String hex = Integer.toHexString(0xff & b);
40             if (hex.length() == 1) {
41                 hexString.append('0');
42             }
43             hexString.append(hex);
44         }
45         return hexString.toString();
46
47     } catch (NoSuchAlgorithmException e) {
48         // Em um app real, isso não deve acontecer se o SHA-256 for padrão.
49         System.err.println("Erro: Algoritmo SHA-256 não encontrado.");
50         throw new RuntimeException("Erro de criptografia", e);
51     }
52 }
```

5.7. Source Package Controller

5.7.1 AtivoDigitalServlet.java

```

1 /**
2  * TIAGO KAUAN
3  * DIEGO HENRIQUE
4  * NICOLY MARTINELI
5 */
6
7 package controller;
8
9 import model.DAO.AativoDigitalDAO;
10 import model.AativoDigital;
11 import model.Usuario;
12 import java.io.IOException;
13 import java.util.List;
14 import javax.servlet.ServletException;
15 import javax.servlet.annotation.WebServlet;
16 import javax.servlet.http.HttpServlet;
17 import javax.servlet.http.HttpServletRequest;
18 import javax.servlet.http.HttpServletResponse;
19 import javax.servlet.http.HttpSession;
20
21 @WebServlet(name = "AtivoDigitalServlet", urlPatterns = {"AtivoDigitalServlet"})
22 public class AtivoDigitalServlet extends HttpServlet {
23
24     // MÉTODO GET: Listar OU Excluir
25     @Override
26     protected void doGet(HttpServletRequest request, HttpServletResponse response)
27         throws ServletException, IOException {
28
29         HttpSession session = request.getSession();
30         Usuario usuarioLogado = (Usuario) session.getAttribute("usuarioLogado");
31
32         if (usuarioLogado == null) {
33             response.sendRedirect("index.html");
34             return;
35         }
36
37         String acao = request.getParameter("acao");

```

```

37     String acao = request.getParameter("acao");
38     AativoDigitalDAO dao = new AativoDigitalDAO();
39
40     // --- LÓGICA DE EXCLUSÃO ---
41     if ("excluir".equals(acao)) {
42         String idParam = request.getParameter("id");
43         if (idParam != null && !idParam.isEmpty()) {
44             int idAtivo = Integer.parseInt(idParam);
45
46             // TODO: Seria bom verificar se esse ativo pertence mesmo ao usuário logado por segurança
47             boolean excluiu = dao.excluirAativoDigital(idAtivo); // Você precisará criar este método no DAO (veja Passo 3)
48
49             if (excluiu) {
50                 // REDIRECIONA PARA A NOVA TELA DE SUCESSO
51                 response.sendRedirect("1 - usuario/sucesso_usuario.jsp?msg=ativoExcluido");
52                 return;
53             }
54         }
55     }
56
57
58     // Lógica padrão de listagem (usada pela tela 2 - ativodigital)
59     List<AativoDigital> lista = dao.listarAitivosDigitaisDoTitular(usuarioLogado.getId());
60     request.setAttribute("listaAtivos", lista);
61     request.getRequestDispatcher("2 - ativodigital/1 - ativodigital.jsp").forward(request, response);
62 }
63
64 // MÉTODO POST: Cadastrar
65 @Override
66 protected void doPost(HttpServletRequest request, HttpServletResponse response)
67     throws ServletException, IOException {
68
69     request.setCharacterEncoding("UTF-8");
70     HttpSession session = request.getSession();
71     Usuario usuarioLogado = (Usuario) session.getAttribute("usuarioLogado");
72
73     if (usuarioLogado == null) {

```

...jsp 1 - herdeiro.jsp x sucessoso_herdeiro.jsp x login_cadastro.html x ConectaDB.java x CryptoUtil.java x HashUtil.java x AtivoDigitalServlet.java x

Source History

```

73     if (usuarioLogado == null) {
74         response.sendRedirect("index.html");
75         return;
76     }
77
78     String tipo = request.getParameter("tipo-conta");
79     String idHerdeiroStr = request.getParameter("herdeiro");
80
81     if(idHerdeiroStr == null || idHerdeiroStr.isEmpty()) {
82         // Erro de validação - redireciona para tela de erro (opcional)
83         response.sendRedirect("1 - usuario/sucesso_usuario.jsp?erro=falhaCriar");
84         return;
85     }
86
87     int idHerdeiro = Integer.parseInt(idHerdeiroStr);
88
89     AtivoDigital ad = new AtivoDigital();
90     ad.setIdUsuarioTitular(usuarioLogado.getId());
91     ad.setIdHerdeiro(idHerdeiro);
92     ad.setTipoAtivoDigital(tipo);
93
94     if ("crypto".equals(tipo)) {
95         ad.setFraseRecuperacao(request.getParameter("conta-frase"));
96     } else if ("mensagem".equals(tipo)) {
97         ad.setMensagem(request.getParameter("conta-mensagem"));
98     } else {
99         ad.setLogin(request.getParameter("conta-login"));
100        ad.setSenha(request.getParameter("conta-senha"));
101    }
102
103    AtivoDigitalDAO dao = new AtivoDigitalDAO();
104
105    // Validação de Duplicidade
106    if (ad.getLogin() != null && !ad.getLogin().isEmpty()) {
107        if (dao.verificarDuplicidade(usuarioLogado.getId(), ad.getTipoAtivoDigital(), ad.getLogin())) {
108            // REDIRECIONA PARA A NOVA TELA COM ERRO DE DUPLICIDADE
109            response.sendRedirect("1 - usuario/sucesso_usuario.jsp?erro=duplicado");
110        }
111    }

```

...jsp 1 - herdeiro.jsp x sucessoso_herdeiro.jsp x login_cadastro.html x ConectaDB.java x CryptoUtil.java x HashUtil.java x AtivoDigitalServlet.java x

Source History

```

91     ad.setIdHerdeiro(idHerdeiro);
92     ad.setTipoAtivoDigital(tipo);
93
94     if ("crypto".equals(tipo)) {
95         ad.setFraseRecuperacao(request.getParameter("conta-frase"));
96     } else if ("mensagem".equals(tipo)) {
97         ad.setMensagem(request.getParameter("conta-mensagem"));
98     } else {
99         ad.setLogin(request.getParameter("conta-login"));
100        ad.setSenha(request.getParameter("conta-senha"));
101    }
102
103    AtivoDigitalDAO dao = new AtivoDigitalDAO();
104
105    // Validação de Duplicidade
106    if (ad.getLogin() != null && !ad.getLogin().isEmpty()) {
107        if (dao.verificarDuplicidade(usuarioLogado.getId(), ad.getTipoAtivoDigital(), ad.getLogin())) {
108            // REDIRECIONA PARA A NOVA TELA COM ERRO DE DUPLICIDADE
109            response.sendRedirect("1 - usuario/sucesso_usuario.jsp?erro=duplicado");
110            return;
111        }
112    }
113
114    boolean sucesso = dao.cadastrarAtivoDigital(ad);
115
116    if (sucesso) {
117        // REDIRECIONA PARA A NOVA TELA DE SUCESSO (CADASTRO)
118        response.sendRedirect("1 - usuario/sucesso_usuario.jsp?msg=ativoCriado");
119    } else {
120        response.sendRedirect("1 - usuario/sucesso_usuario.jsp?erro=falhaCriar");
121    }
122}

```

5.7.2 CarregarFotoServlet.java

```
1  /*
2   * TIAGO KAUAN
3   * DIEGO HENRIQUE
4   * NICOLY MARTINELI
5   */
6
7 package controller;
8
9 import java.io.IOException;
10 import java.io.OutputStream;
11 import javax.servlet.ServletException;
12 import javax.servlet.annotation.WebServlet;
13 import javax.servlet.http.HttpServlet;
14 import javax.servlet.http.HttpServletRequest;
15 import javax.servlet.http.HttpServletResponse;
16 import javax.servlet.http.HttpSession;
17 import model.Usuario;
18
19 /**
20  * Servlet dedicado a carregar e exibir a foto de perfil do usuário
21  * que está logado na sessão.
22  */
23 @WebServlet(name = "CarregarFotoServlet", urlPatterns = {" /CarregarFotoServlet"})
24 public class CarregarFotoServlet extends HttpServlet {
25
26     protected void processRequest(HttpServletRequest request, HttpServletResponse response)
27         throws ServletException, IOException {
28
29     try {
30         HttpSession session = request.getSession(false); // Pega a sessão, não cria uma nova
31
32         // 1. Guarda de Segurança: Verifica se o usuário está logado
33         if (session == null || session.getAttribute("usuarioLogado") == null) {
34             // Se não estiver logado, não há foto para mostrar
35             response.setStatus(HttpServletResponse.SC_NOT_FOUND);
36             return;
37         }
38
39         // 2. Pega o usuário da sessão
40         Usuario usuarioLogado = (Usuario) session.getAttribute("usuarioLogado");
41         byte[] fotoBytes = usuarioLogado.getFotoPerfil();
42
43         // 3. Verifica se o usuário TEM uma foto
44         if (fotoBytes != null && fotoBytes.length > 0) {
45
46             // 4. Configura a Resposta
47             response.setContentType("image/jpeg"); // Ou image/png, dependendo do upload
48             response.setContentLength(fotoBytes.length);
49
50             // Define cabecalhos para evitar cache (para que a foto mude se o usuário atualizar)
51             response.setHeader("Cache-Control", "no-cache, no-store, must-revalidate");
52             response.setHeader("Pragma", "no-cache");
53             response.setDateHeader("Expires", 0);
54
55             // 5. Escreve os bytes da imagem diretamente na resposta
56             try (OutputStream out = response.getOutputStream()) {
57                 out.write(fotoBytes);
58             }
59
60         } else {
61             // 6. Se o usuário não tem foto, envia um 404 (Não Encontrado)
62             // O navegador mostrará o 'alt' text ou o ícone de imagem quebrada.
63             // (No JSP, nós tratamos isso mostrando um ícone padrão)
64             response.setStatus(HttpServletResponse.SC_NOT_FOUND);
65         }
66     } catch (Exception e) {
67         System.err.println("Erro ao carregar foto: " + e.getMessage());
68         response.setStatus(HttpServletResponse.SC_INTERNAL_SERVER_ERROR);
69     }
70 }
```

```
34         // Se não estiver logado, não há foto para mostrar
35         response.setStatus(HttpServletResponse.SC_NOT_FOUND);
36         return;
37     }
38
39     // 2. Pega o usuário da sessão
40     Usuario usuarioLogado = (Usuario) session.getAttribute("usuarioLogado");
41     byte[] fotoBytes = usuarioLogado.getFotoPerfil();
42
43     // 3. Verifica se o usuário TEM uma foto
44     if (fotoBytes != null && fotoBytes.length > 0) {
45
46         // 4. Configura a Resposta
47         response.setContentType("image/jpeg"); // Ou image/png, dependendo do upload
48         response.setContentLength(fotoBytes.length);
49
50         // Define cabecalhos para evitar cache (para que a foto mude se o usuário atualizar)
51         response.setHeader("Cache-Control", "no-cache, no-store, must-revalidate");
52         response.setHeader("Pragma", "no-cache");
53         response.setDateHeader("Expires", 0);
54
55         // 5. Escreve os bytes da imagem diretamente na resposta
56         try (OutputStream out = response.getOutputStream()) {
57             out.write(fotoBytes);
58         }
59
60     } else {
61         // 6. Se o usuário não tem foto, envia um 404 (Não Encontrado)
62         // O navegador mostrará o 'alt' text ou o ícone de imagem quebrada.
63         // (No JSP, nós tratamos isso mostrando um ícone padrão)
64         response.setStatus(HttpServletResponse.SC_NOT_FOUND);
65     }
66 } catch (Exception e) {
67     System.err.println("Erro ao carregar foto: " + e.getMessage());
68     response.setStatus(HttpServletResponse.SC_INTERNAL_SERVER_ERROR);
69 }
```

The screenshot shows a Java IDE interface with the file `CarregarFotoServlet.java` open. The code implements a servlet for handling file uploads, specifically for profile pictures. It includes logic to handle both GET and POST requests, validate file types (image/jpg, image/jpeg), and save the uploaded file to a specified directory. Error handling is provided for cases where no file is uploaded or an error occurs during processing.

```
55     // 5. Escreve os bytes da imagem diretamente na resposta
56     try (OutputStream out = response.getOutputStream()) {
57         out.write(fotoBytes);
58     }
59
60     } else {
61         // 6. Se o usuário não tem foto, envia um 404 (Não Encontrado)
62         // O navegador mostrará o 'alt' text ou o ícone de imagem quebrada.
63         // (No JSP, nós tratamos isso mostrando um ícone padrão)
64         response.setStatus(HttpServletResponse.SC_NOT_FOUND);
65     }
66 } catch (Exception e) {
67     System.err.println("Erro ao carregar foto: " + e.getMessage());
68     response.setStatus(HttpServletResponse.SC_INTERNAL_SERVER_ERROR);
69 }
70
71
72 @Override
73 protected void doGet(HttpServletRequest request, HttpServletResponse response)
74     throws ServletException, IOException {
75     processRequest(request, response);
76 }
77
78 @Override
79 protected void doPost(HttpServletRequest request, HttpServletResponse response)
80     throws ServletException, IOException {
81     processRequest(request, response);
82 }
83 }
```

5.7.3 EditarPerfilServlet.java

```
1 /**
2  * TIAGO KAUAN
3  * DIEGO HENRIQUE
4  * NICOLY MARTINELI
5 */
6
7 package controller;
8
9 import config.HashUtil;
10 import java.io.IOException;
11 import java.io.InputStream;
12 import javax.servlet.ServletException;
13 import javax.servlet.annotation.MultipartConfig;
14 import javax.servlet.annotation.WebServlet;
15 import javax.servlet.http.HttpServlet;
16 import javax.servlet.http.HttpServletRequest;
17 import javax.servlet.http.HttpServletResponse;
18 import javax.servlet.http.HttpSession;
19 import javax.servlet.http.Part;
20 import model.DAO.UsuarioDAO;
21 import model.Usuario;
22 // Você precisa adicionar a biblioteca 'Apache Commons IO' ao projeto
23 import org.apache.commons.io.IOUtils;
24
25 /**
26  * Servlet para processar a ATUALIZAÇÃO do perfil do usuário.
27  * 1. Recebe a foto (se houver)
28  * 2. Recebe a nova senha (se houver)
29  * 3. Atualiza o banco de dados
30  * 4. Atualiza a sessão
31  * @MultipartConfig é OBRIGATÓRIO para formulários que enviam arquivos (fotos).
32 */
33 @WebServlet(name = "EditarPerfilServlet", urlPatterns = {"/EditarPerfilServlet"})
34 @MultipartConfig(
35     fileSizeThreshold = 1024 * 1024 * 2, // 2MB
36     maxFileSize = 1024 * 1024 * 10, // 10MB
37     maxRequestSize = 1024 * 1024 * 50 // 50MB
38 )
```

```
39 public class EditarPerfilServlet extends HttpServlet {
40
41     protected void processRequest(HttpServletRequest request, HttpServletResponse response)
42             throws ServletException, IOException {
43
44         response.setContentType("text/html;charset=UTF-8");
45         HttpSession session = request.getSession(false); // Pega a sessão existente
46
47         // 1. Guarda de Segurança: Verifica se o usuário está logado
48         if (session == null || session.getAttribute("usuarioLogado") == null) {
49             response.sendRedirect("login_cadastro.html?loginErro=true");
50             return;
51         }
52
53         // 2. Pega o usuário da sessão
54         Usuario usuarioLogado = (Usuario) session.getAttribute("usuarioLogado");
55         int idUsuario = usuarioLogado.getId();
56
57         // 3. Prepara variáveis para o DAO
58         byte[] fotoBytes = null;
59         String novaSenhaHash = null;
56
59         // Flag para sabermos se a atualização no DAO foi bem-sucedida
60         boolean sucessoDAO = false;
61         boolean algoParaAtualizar = false;
62
63         try {
64             // 4. Processa a FOTO (se ela foi enviada)
65             Part filePart = request.getPart("fotoPerfil"); // 'fotoPerfil' é o 'name' do input
66             if (filePart != null & filePart.getSize() > 0) {
67                 algoParaAtualizar = true; // Marcamos que temos algo para salvar
68                 try (InputStream inputStream = filePart.getInputStream()) {
```

...jsp login_cadastro.html ConectaDB.java CryptoUtil.java HashUtil.java AtivoDigitalServlet.java CarregarFotoServlet.java EditarPerfilServlet.java

Source History

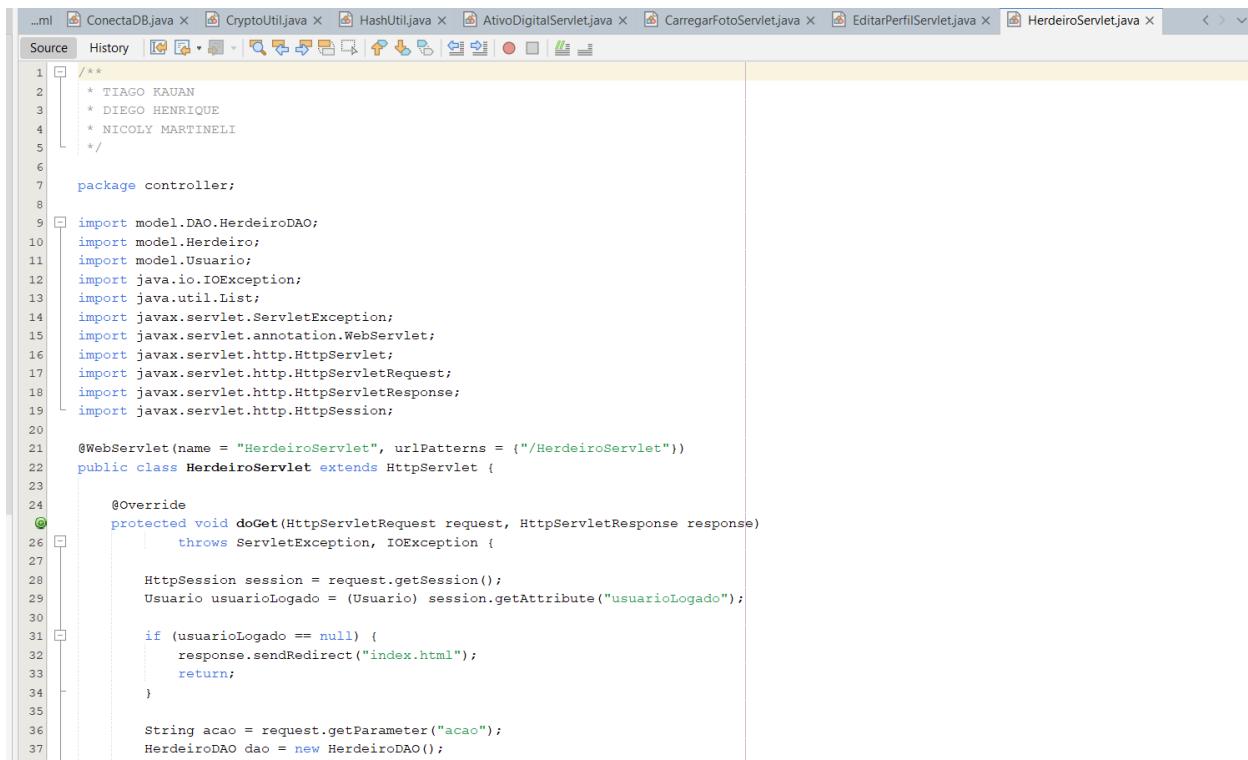
```
67     Part filePart = request.getPart("fotoPerfil"); // "fotoPerfil" é o 'name' do input
68     if (filePart != null && filePart.getSize() > 0) {
69         algoParaAtualizar = true; // Marcamos que temos algo para salvar
70         try (InputStream inputStream = filePart.getInputStream()) {
71             // Usamos a biblioteca Apache Commons IO para converter InputStream para byte[]
72             fotoBytes = IOUtils.toByteArray(inputStream);
73         }
74     }
75
76     // 5. Processa a SENHA (se ela foi enviada)
77     String novaSenha = request.getParameter("senha");
78     if (novaSenha != null && !novaSenha.trim().isEmpty()) {
79         algoParaAtualizar = true; // Marcamos que temos algo para salvar
80         String confirmaSenha = request.getParameter("confirmar_senha");
81
82         // Validação de confirmação (o JS já fez, mas checamos de novo)
83         if (novaSenha.equals(confirmaSenha)) {
84             novaSenhaHash = HashUtil.toSHA256(novaSenha);
85         } else {
86             System.err.println("Tentativa de alterar senha falhou: senhas não conferem.");
87             // Se as senhas não batem, redireciona com erro
88             response.sendRedirect("1 - usuario/2 - tela geral.jsp?erro=senhas");
89             return;
90         }
91     }
92
93     // 6. Chama o DAO para atualizar (apenas se houver algo para mudar)
94     if (algoParaAtualizar) {
95         UsuarioDAO usuarioDAO = new UsuarioDAO();
96         sucessoDAO = usuarioDAO.updatePerfil(idUsuario, fotoBytes, novaSenhaHash);
97
98         if (sucessoDAO) {
99             // 7. IMPORTANTE: Atualizar a sessão!
100            // Buscamos o usuário atualizado do banco
101            Usuario usuarioAtualizado = usuarioDAO.getUsuarioPorId(idUsuario);
102            session.setAttribute("usuarioLogado", usuarioAtualizado);
103        }
104    }
105
106    // 8. Redireciona de volta para a tela de Edição com feedback
107    if (sucessoDAO) {
108        response.sendRedirect("1 - usuario/2 - tela geral.jsp?sucesso=true");
109    } else {
110        response.sendRedirect("1 - usuario/2 - tela geral.jsp?erro=true");
111    }
112
113    } catch (Exception e) {
114        System.err.println("Erro no EditarPerfilServlet: " + e.getMessage());
115        e.printStackTrace();
116        // Em caso de erro, manda para a tela de edição com feedback de erro
117        response.sendRedirect("1 - usuario/2 - tela geral.jsp?erro=true");
118    }
119
120    @Override
121    protected void doGet(HttpServletRequest request, HttpServletResponse response)
122        throws ServletException, IOException {
123        processRequest(request, response);
124    }
125
126    @Override
127    protected void doPost(HttpServletRequest request, HttpServletResponse response)
128        throws ServletException, IOException {
129        processRequest(request, response);
130    }
131
132    @Override
133    public String getServletInfo() {
134        return "Servlet para processar a edição do perfil do usuário.";
135    }
136
137
138
139 }
```

...jsp login_cadastro.html ConectaDB.java CryptoUtil.java HashUtil.java AtivoDigitalServlet.java CarregarFotoServlet.java EditarPerfilServlet.java

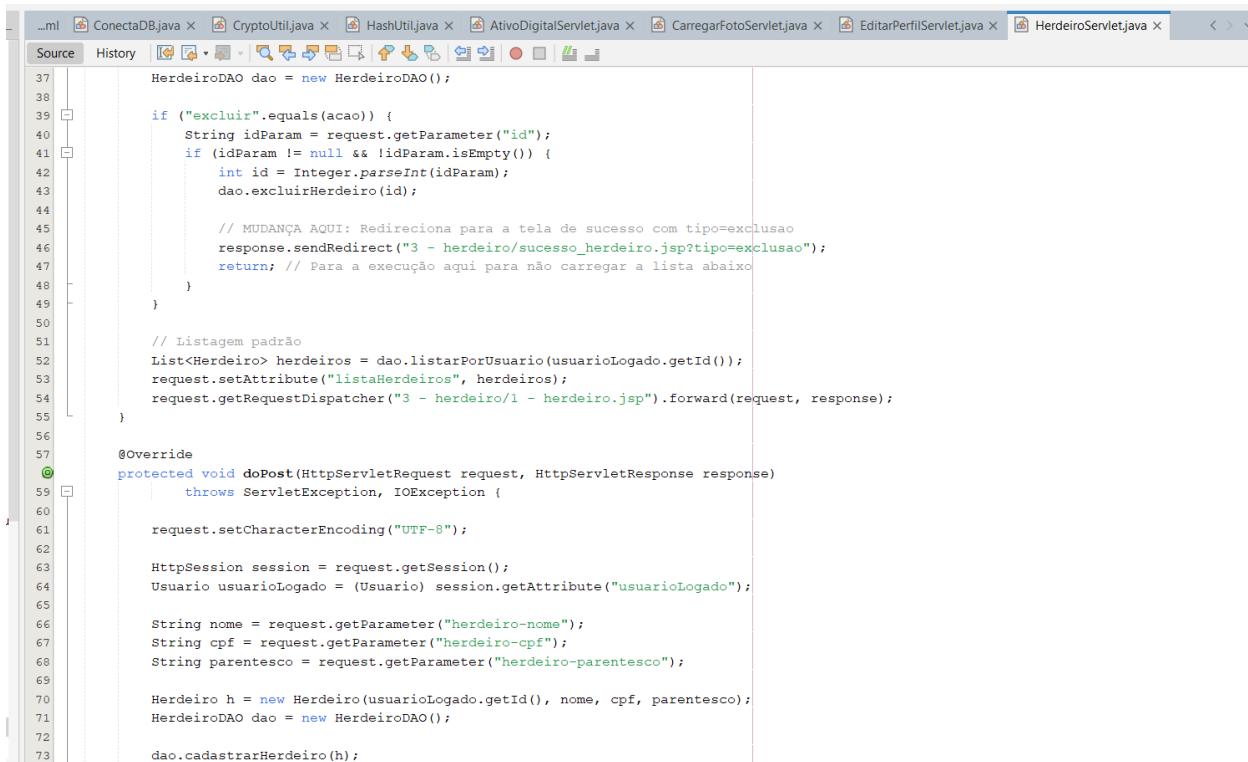
Source History

```
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139 }
```

5.7.4 HerdeiroServlet.java



```
1 /**
2  * TIAGO KAUAN
3  * DIEGO HENRIQUE
4  * NICOLY MARTINELI
5 */
6
7 package controller;
8
9 import model.DAO.HerdeiroDAO;
10 import model.Herdeiro;
11 import model.Usuario;
12 import java.io.IOException;
13 import java.util.List;
14 import javax.servlet.ServletException;
15 import javax.servlet.annotation.WebServlet;
16 import javax.servlet.http.HttpServlet;
17 import javax.servlet.http.HttpServletRequest;
18 import javax.servlet.http.HttpServletResponse;
19 import javax.servlet.http.HttpSession;
20
21 @WebServlet(name = "HerdeiroServlet", urlPatterns = {"/*HerdeiroServlet"})
22 public class HerdeiroServlet extends HttpServlet {
23
24     @Override
25     protected void doGet(HttpServletRequest request, HttpServletResponse response)
26         throws ServletException, IOException {
27
28         HttpSession session = request.getSession();
29         Usuario usuarioLogado = (Usuario) session.getAttribute("usuarioLogado");
30
31         if (usuarioLogado == null) {
32             response.sendRedirect("index.html");
33             return;
34         }
35
36         String acao = request.getParameter("acao");
37         HerdeiroDAO dao = new HerdeiroDAO();
```



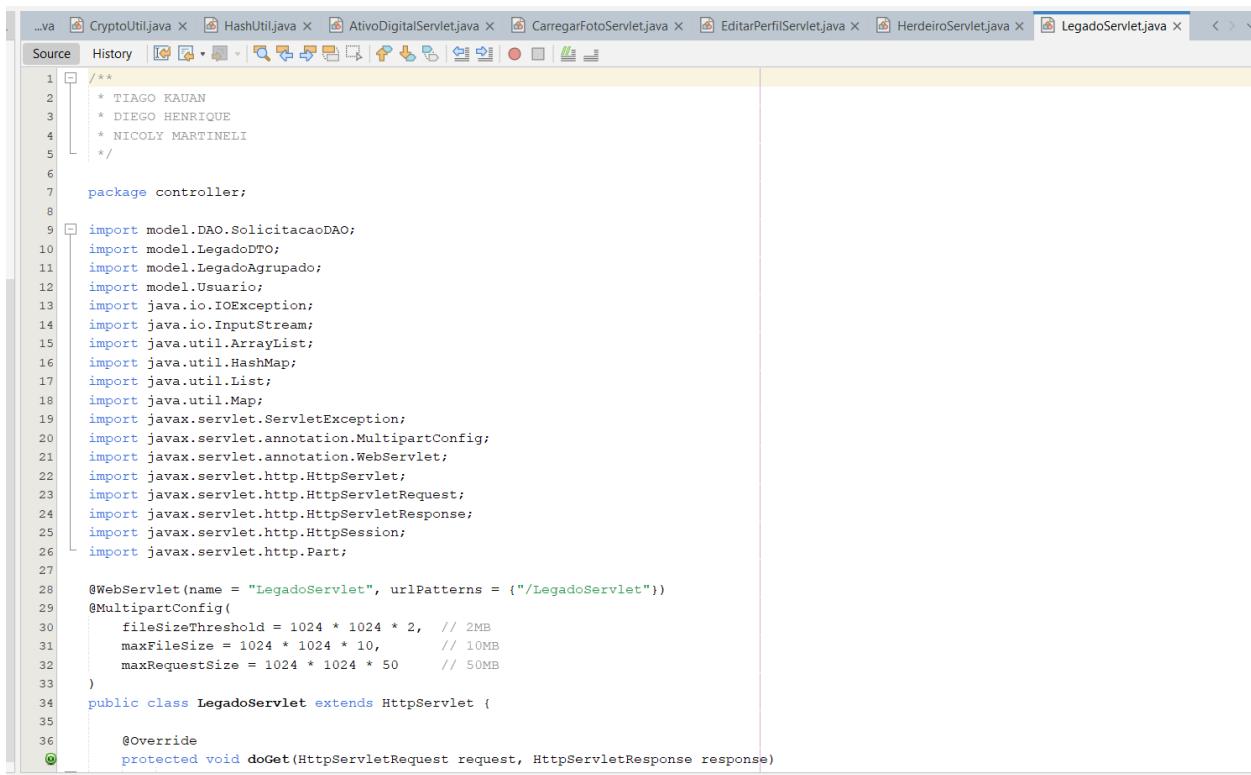
```
37         HerdeiroDAO dao = new HerdeiroDAO();
38
39         if ("excluir".equals(acao)) {
40             String idParam = request.getParameter("id");
41             if (idParam != null & !idParam.isEmpty()) {
42                 int id = Integer.parseInt(idParam);
43                 dao.excluirHerdeiro(id);
44
45                 // MUDANÇA AQUI: Redireciona para a tela de sucesso com tipo=exclusao
46                 response.sendRedirect("3 - herdeiro/sucesso_herdeiro.jsp?tipo=exclusao");
47                 return; // Para a execução aqui para não carregar a lista abaixo
48             }
49         }
50
51         // Listagem padrão
52         List<Herdeiro> herdeiros = dao.listarPorUsuario(usuarioLogado.getId());
53         request.setAttribute("listaHerdeiros", herdeiros);
54         request.getRequestDispatcher("3 - herdeiro/1 - herdeiro.jsp").forward(request, response);
55     }
56
57     @Override
58     protected void doPost(HttpServletRequest request, HttpServletResponse response)
59         throws ServletException, IOException {
60
61         request.setCharacterEncoding("UTF-8");
62
63         HttpSession session = request.getSession();
64         Usuario usuarioLogado = (Usuario) session.getAttribute("usuarioLogado");
65
66         String nome = request.getParameter("herdeiro-nome");
67         String cpf = request.getParameter("herdeiro-cpf");
68         String parentesco = request.getParameter("herdeiro-parentesco");
69
70         Herdeiro h = new Herdeiro(usuarioLogado.getId(), nome, cpf, parentesco);
71         HerdeiroDAO dao = new HerdeiroDAO();
72
73         dao.cadastrarHerdeiro(h);
```

The screenshot shows a Java IDE interface with the following details:

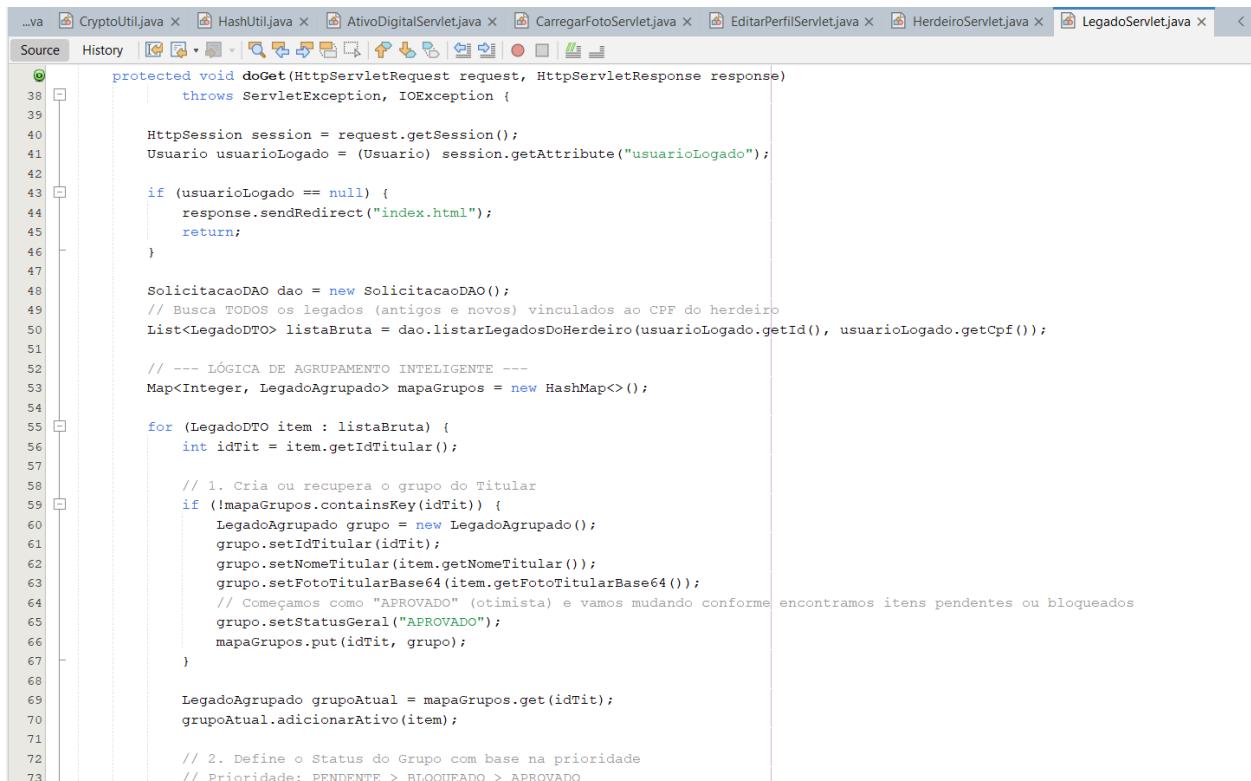
- Title Bar:** The title bar displays several file names: ...ml, ConectaDB.java, CryptoUtil.java, HashUtil.java, AtivoDigitalServlet.java, CarregarFotoServlet.java, EditarPerfilServlet.java, and HerdeiroServlet.java.
- Toolbar:** A standard toolbar with icons for file operations like Open, Save, Print, and others.
- Source Tab:** The "Source" tab is selected, indicating the current view is the Java code.
- Code Area:** The main area contains the Java code for the HerdeiroServlet class. The code is annotated with line numbers (52-78) and includes comments and imports.

```
52     List<Herdeiro> herdeiros = dao.listarPorUsuario(usuarioLogado.getId());
53     request.setAttribute("listaHerdeiros", herdeiros);
54     request.getRequestDispatcher("3 - herdeiro/1 - herdeiro.jsp").forward(request, response);
55   }
56
57   @Override
58   protected void doPost(HttpServletRequest request, HttpServletResponse response)
59       throws ServletException, IOException {
60
61     request.setCharacterEncoding("UTF-8");
62
63     HttpSession session = request.getSession();
64     Usuario usuarioLogado = (Usuario) session.getAttribute("usuarioLogado");
65
66     String nome = request.getParameter("herdeiro-nome");
67     String cpf = request.getParameter("herdeiro-cpf");
68     String parentesco = request.getParameter("herdeiro-parentesco");
69
70     Herdeiro h = new Herdeiro(usuarioLogado.getId(), nome, cpf, parentesco);
71     HerdeiroDAO dao = new HerdeiroDAO();
72
73     dao.cadastrarHerdeiro(h);
74
75     // MUDANÇA AQUI: Redireciona para a tela de sucesso com tipo=cadastro
76     response.sendRedirect("3 - herdeiro/sucesso_herdeiro.jsp?tipo=cadastro");
77   }
78 }
```

5.7.5 LegadoServlet.java



```
1 /**
2  * TIAGO KAUAN
3  * DIEGO HENRIQUE
4  * NICOLY MARTINELI
5 */
6
7 package controller;
8
9 import model.DAO.SolicitacaoDAO;
10 import model.LegadoDTO;
11 import model.LegadoAgrupado;
12 import model.Usuario;
13 import java.io.IOException;
14 import java.io.InputStream;
15 import java.util.ArrayList;
16 import java.util.HashMap;
17 import java.util.List;
18 import java.util.Map;
19 import javax.servlet.ServletException;
20 import javax.servlet.annotation.MultipartConfig;
21 import javax.servlet.annotation.WebServlet;
22 import javax.servlet.http.HttpServlet;
23 import javax.servlet.http.HttpServletRequest;
24 import javax.servlet.http.HttpServletResponse;
25 import javax.servlet.http.HttpSession;
26 import javax.servlet.http.Part;
27
28 @WebServlet(name = "LegadoServlet", urlPatterns = {" /LegadoServlet"})
29 @MultipartConfig(
30     fileSizeThreshold = 1024 * 1024 * 2, // 2MB
31     maxFileSize = 1024 * 1024 * 10, // 10MB
32     maxRequestSize = 1024 * 1024 * 50 // 50MB
33 )
34 public class LegadoServlet extends HttpServlet {
35
36     @Override
37     protected void doGet(HttpServletRequest request, HttpServletResponse response)
```



```
38     protected void doGet(HttpServletRequest request, HttpServletResponse response)
39         throws ServletException, IOException {
40
41     HttpSession session = request.getSession();
42     Usuario usuarioLogado = (Usuario) session.getAttribute("usuarioLogado");
43
44     if (usuarioLogado == null) {
45         response.sendRedirect("index.html");
46         return;
47     }
48
49     SolicitacaoDAO dao = new SolicitacaoDAO();
50     // Busca TODOS os legados (antigos e novos) vinculados ao CPF do herdeiro
51     List<LegadoDTO> listaBruta = dao.listarLegadosDoHerdeiro(usuarioLogado.getId(), usuarioLogado.getCpf());
52
53     // --- LÓGICA DE AGRUPAMENTO INTELIGENTE ---
54     Map<Integer, LegadoAgrupado> mapaGrupos = new HashMap<>();
55
56     for (LegadoDTO item : listaBruta) {
57         int idTit = item.getIdTitular();
58
59         // 1. Cria ou recupera o grupo do Titular
60         if (!mapaGrupos.containsKey(idTit)) {
61             LegadoAgrupado grupo = new LegadoAgrupado();
62             grupo.setIdTitular(idTit);
63             grupo.setNomeTitular(item.getNomeTitular());
64             grupo.setFotoTitularBase64(item.getFotoTitularBase64());
65             grupo.setStatusGeral("APROVADO");
66             mapaGrupos.put(idTit, grupo);
67         }
68
69         LegadoAgrupado grupoAtual = mapaGrupos.get(idTit);
70         grupoAtual.adicionarAtivo(item);
71
72         // 2. Define o Status do Grupo com base na prioridade
73         // Prioridade: PENDENTE > BLOQUEADO > APROVADO
```


The screenshot shows a Java code editor interface with the file `LegadoServlet.java` open. The code is a servlet that handles file uploads and interacts with a DAO. The code is annotated with line numbers and includes several comments explaining its functionality.

```
115     Part filePart = request.getPart("certidao");
116     byte[] certidaoBytes = null;
117
118     if (filePart != null && filePart.getSize() > 0) {
119         // Conversão nativa do PDF para bytes
120         try (InputStream inputStream = filePart.getInputStream();
121              java.io.ByteArrayOutputStream buffer = new java.io.ByteArrayOutputStream()) {
122
123             int nRead;
124             byte[] data = new byte[4096];
125             while ((nRead = inputStream.read(data, 0, data.length)) != -1) {
126                 buffer.write(data, 0, nRead);
127             }
128             certidaoBytes = buffer.toByteArray();
129         }
130     }
131
132     // 3. Envia para o DAO
133     SolicitacaoDAO dao = new SolicitacaoDAO();
134
135     // O DAO vai identificar quais itens desse titular ainda estão bloqueados e criar solicitação APENAS para eles.
136     // Os itens antigos (já aprovados) serão ignorados pela query "NOT IN".
137     dao.solicitarDesbloqueioPorTitular(idTitular, usuarioLogado.getId(), certidaoBytes);
138
139     // Recarrega a página
140     response.sendRedirect("LegadoServlet");
141
142     } catch (Exception e) {
143         e.printStackTrace();
144         response.sendRedirect("LegadoServlet?erro=upload");
145     }
146 }
147 }
```

5.7.6 LoginServlet.java

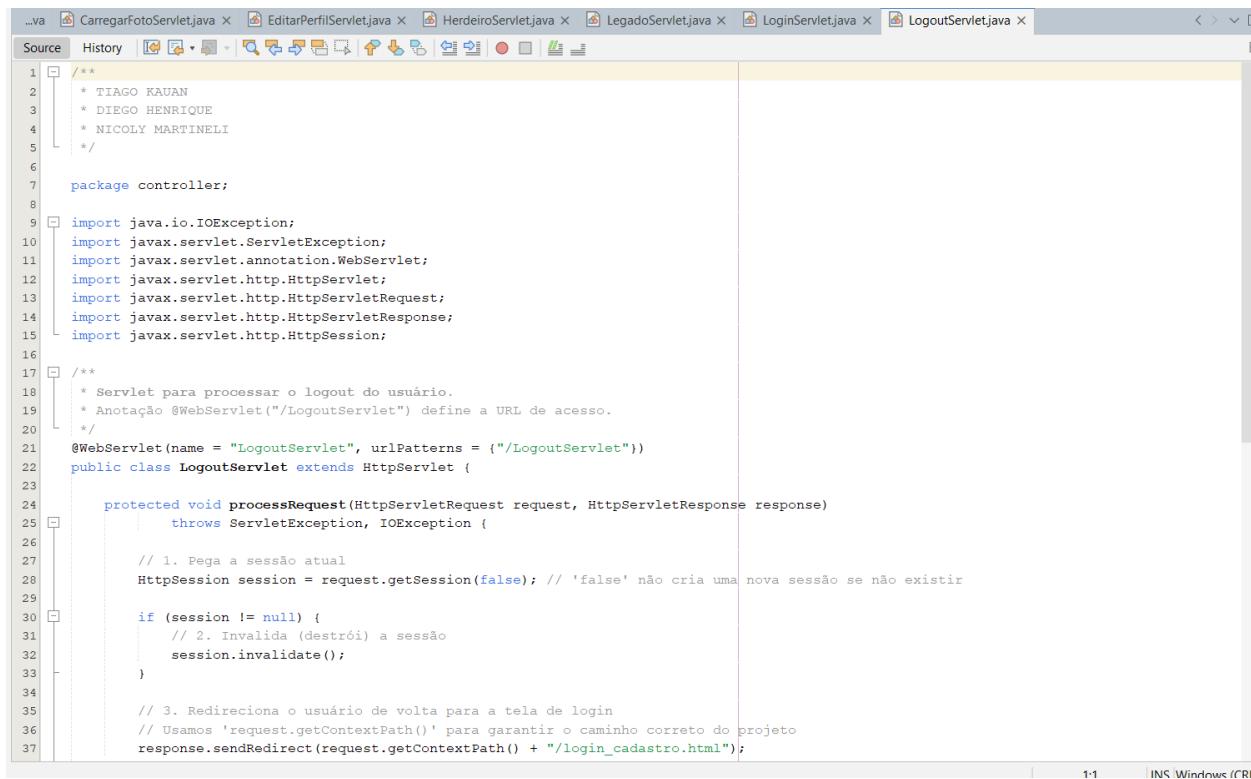
```
1  /*
2   * TIAGO KAUAN
3   * DIEGO HENRIQUE
4   * NICOLY MARTINELI
5   */
6
7 package controller;
8
9 import config.HashUtil;
10 import model.DAO.UsuarioDAO;
11 import model.Usuario;
12
13 import java.io.IOException;
14 import javax.servlet.ServletException;
15 import javax.servlet.annotation.WebServlet;
16 import javax.servlet.http.HttpServlet;
17 import javax.servlet.http.HttpServletRequest;
18 import javax.servlet.http.HttpServletResponse;
19 import javax.servlet.http.HttpSession;
20
21 /**
22  * Servlet (Controller) para processar o login do usuário.
23  * Mapeado para a URL "/LoginServlet", que é o 'action' do formulário de login.
24  */
25 @WebServlet(name = "LoginServlet", urlPatterns = {"/LoginServlet"})
26 public class LoginServlet extends HttpServlet {
27
28     protected void processRequest(HttpServletRequest request, HttpServletResponse response)
29             throws ServletException, IOException {
30
31         response.setContentType("text/html;charset=UTF-8");
32
33         try {
34             // 1. Obter os dados do formulário (da tela login_cadastro.html)
35             String cpf = request.getParameter("cpf");
36             String senhaDigitada = request.getParameter("senha");
37
38             // Validação simples
39             if (cpf == null || senhaDigitada == null || cpf.trim().isEmpty() || senhaDigitada.trim().isEmpty()) {
40                 // Redireciona de volta com erro
41                 response.sendRedirect("login_cadastro.html?erro=1");
42                 return;
43             }
44
45             // 2. Criptografar a senha digitada para comparar com o banco
46             String senhaHashDigitada = HashUtil.toSHA256(senhaDigitada);
47
48             // 3. Chamar o DAO para verificar o usuário
49             UsuarioDAO usuarioDAO = new UsuarioDAO();
50             Usuario usuarioDoBanco = usuarioDAO.getUsuarioPorCPF(cpf);
51
52             // 4. Lógica de Verificação
53             // Verifica se o usuário existe E se a senha hasheada do banco é igual à senha hasheada digitada
54             if (usuarioDoBanco != null && usuarioDoBanco.getSenhaHash().equals(senhaHashDigitada)) {
55
56                 // SUCESSO NO LOGIN
57
58                 // 5. Criar uma Sessão para "manter o usuário logado"
59                 HttpSession session = request.getSession();
60                 session.setAttribute("usuarioLogado", usuarioDoBanco); // Salva o objeto 'Usuario' inteiro na sessão
61                 session.setAttribute("usuarioNome", usuarioDoBanco.getNome()); // Salva só o nome (para facilitar o "Olá, João!")
62
63                 // 6. Redirecionar para o Dashboard Principal
64                 // (Caminho baseado na sua estrutura de arquivos)
65                 response.sendRedirect("1 - usuario/1 - tela geral.jsp");
66
67             } else {
68
69                 // FALHA NO LOGIN
70
71                 // 7. Redirecionar de volta para a tela de login com uma mensagem de erro
72                 // O '?erro=1' será lido pelo JavaScript no HTML para mostrar a mensagem
73                 response.sendRedirect("login_cadastro.html?erro=1");
74
75         }
76
77     }
78
79 }
```

```
37
38
39             // Validação simples
40             if (cpf == null || senhaDigitada == null || cpf.trim().isEmpty() || senhaDigitada.trim().isEmpty()) {
41                 // Redireciona de volta com erro
42                 response.sendRedirect("login_cadastro.html?erro=1");
43                 return;
44             }
45
46             // 2. Criptografar a senha digitada para comparar com o banco
47             String senhaHashDigitada = HashUtil.toSHA256(senhaDigitada);
48
49             // 3. Chamar o DAO para verificar o usuário
50             UsuarioDAO usuarioDAO = new UsuarioDAO();
51             Usuario usuarioDoBanco = usuarioDAO.getUsuarioPorCPF(cpf);
52
53             // 4. Lógica de Verificação
54             // Verifica se o usuário existe E se a senha hasheada do banco é igual à senha hasheada digitada
55             if (usuarioDoBanco != null && usuarioDoBanco.getSenhaHash().equals(senhaHashDigitada)) {
56
57                 // SUCESSO NO LOGIN
58
59                 // 5. Criar uma Sessão para "manter o usuário logado"
60                 HttpSession session = request.getSession();
61                 session.setAttribute("usuarioLogado", usuarioDoBanco); // Salva o objeto 'Usuario' inteiro na sessão
62                 session.setAttribute("usuarioNome", usuarioDoBanco.getNome()); // Salva só o nome (para facilitar o "Olá, João!")
63
64                 // 6. Redirecionar para o Dashboard Principal
65                 // (Caminho baseado na sua estrutura de arquivos)
66                 response.sendRedirect("1 - usuario/1 - tela geral.jsp");
67
68             } else {
69
70                 // FALHA NO LOGIN
71
72                 // 7. Redirecionar de volta para a tela de login com uma mensagem de erro
73                 // O '?erro=1' será lido pelo JavaScript no HTML para mostrar a mensagem
74                 response.sendRedirect("login_cadastro.html?erro=1");
75
76         }
77
78     }
79
80 }
```

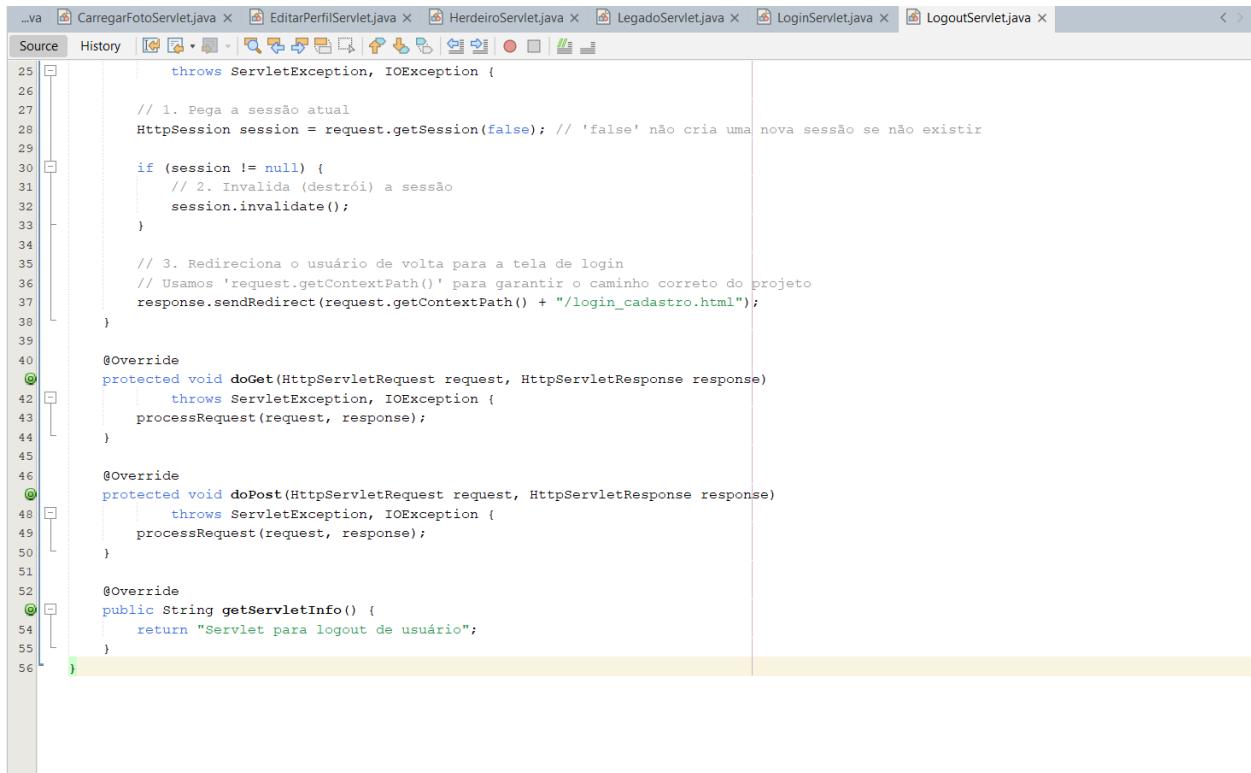
```
...va HashUtil.java X AtivoDigitalServlet.java X CarregarFotoServlet.java X EditarPerfilServlet.java X HerdeiroServlet.java X LegadoServlet.java X LoginServlet.java X < > 
Source History [ ] 🔍 🗂️ 📁 🎯 🏷️ 🚪 🛡️ 🌐 🌐 🌐 
61         session.setAttribute("usuarioNome", usuarioDoBanco.getNome()); // Salva só o nome (para facilitar o "Olá, João!")
62
63         // 6. Redirecionar para o Dashboard Principal
64         // (Caminho baseado na sua estrutura de arquivos)
65         response.sendRedirect("1 - usuario/1 - tela geral.jsp");
66
67     } else {
68
69         // FALHA NO LOGIN
70
71         // 7. Redirecionar de volta para a tela de login com uma mensagem de erro
72         // O '?erro=1' será lido pelo JavaScript no HTML para mostrar a mensagem
73         response.sendRedirect("login_cadastro.html?erro=1");
74     }
75
76 } catch (Exception e) {
77     // Lidar com erros inesperados
78     System.err.println("Erro grave no LoginServlet: " + e.getMessage());
79     e.printStackTrace();
80     response.sendRedirect("login_cadastro.html?erro=2"); // Erro genérico
81 }
82
83
84 @Override
85 protected void doGet(HttpServletRequest request, HttpServletResponse response)
86     throws ServletException, IOException {
87     processRequest(request, response);
88 }
89
90 @Override
91 protected void doPost(HttpServletRequest request, HttpServletResponse response)
92     throws ServletException, IOException {
93     processRequest(request, response);
94 }
95
96 @Override
97 public String getServletInfo() {
98
99 }
```

```
...va HashUtil.java X AtivoDigitalServlet.java X CarregarFotoServlet.java X EditarPerfilServlet.java X HerdeiroServlet.java X LegadoServlet.java X LoginServlet.java X < > 
Source History [ ] 🔍 🗂️ 📁 🎯 🏷️ 🚪 🛡️ 🌐 🌐 🌐 
73         response.sendRedirect("login_cadastro.html?erro=1");
74
75 } catch (Exception e) {
76     // Lidar com erros inesperados
77     System.err.println("Erro grave no LoginServlet: " + e.getMessage());
78     e.printStackTrace();
79     response.sendRedirect("login_cadastro.html?erro=2"); // Erro genérico
80 }
81
82
83
84 @Override
85 protected void doGet(HttpServletRequest request, HttpServletResponse response)
86     throws ServletException, IOException {
87     processRequest(request, response);
88 }
89
90 @Override
91 protected void doPost(HttpServletRequest request, HttpServletResponse response)
92     throws ServletException, IOException {
93     processRequest(request, response);
94 }
95
96 @Override
97 public String getServletInfo() {
98     return "Servlet para autenticação de usuário";
99 }
100 }
```

5.7.7 LogoutServlet.java



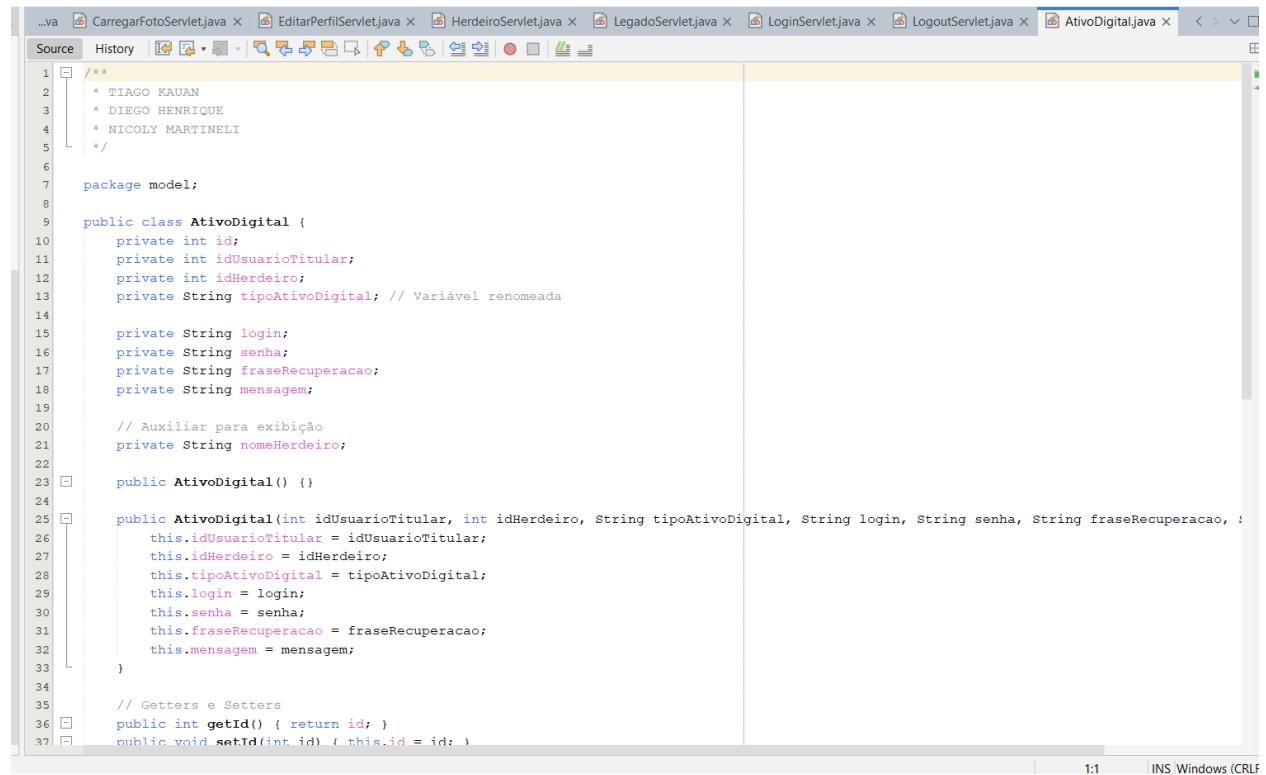
```
1  /**
2  * TIAGO KAUAN
3  * DIEGO HENRIQUE
4  * NICOLY MARTINELI
5  */
6
7 package controller;
8
9 import java.io.IOException;
10 import javax.servlet.ServletException;
11 import javax.servlet.annotation.WebServlet;
12 import javax.servlet.http.HttpServlet;
13 import javax.servlet.http.HttpServletRequest;
14 import javax.servlet.http.HttpServletResponse;
15 import javax.servlet.http.HttpSession;
16
17 /**
18 * Servlet para processar o logout do usuário.
19 * Anotação @WebServlet("/LogoutServlet") define a URL de acesso.
20 */
21 @WebServlet(name = "LogoutServlet", urlPatterns = {"LogoutServlet"})
22 public class LogoutServlet extends HttpServlet {
23
24     protected void processRequest(HttpServletRequest request, HttpServletResponse response)
25         throws ServletException, IOException {
26
27         // 1. Pega a sessão atual
28         HttpSession session = request.getSession(false); // 'false' não cria uma nova sessão se não existir
29
30         if (session != null) {
31             // 2. Invalida (destrói) a sessão
32             session.invalidate();
33         }
34
35         // 3. Redireciona o usuário de volta para a tela de login
36         // Usamos 'request.getContextPath()' para garantir o caminho correto do projeto
37         response.sendRedirect(request.getContextPath() + "/login_cadastro.html");
38     }
39
40     @Override
41     protected void doGet(HttpServletRequest request, HttpServletResponse response)
42         throws ServletException, IOException {
43         processRequest(request, response);
44     }
45
46     @Override
47     protected void doPost(HttpServletRequest request, HttpServletResponse response)
48         throws ServletException, IOException {
49         processRequest(request, response);
50     }
51
52     @Override
53     public String getServletInfo() {
54         return "Servlet para logout de usuário";
55     }
56 }
```



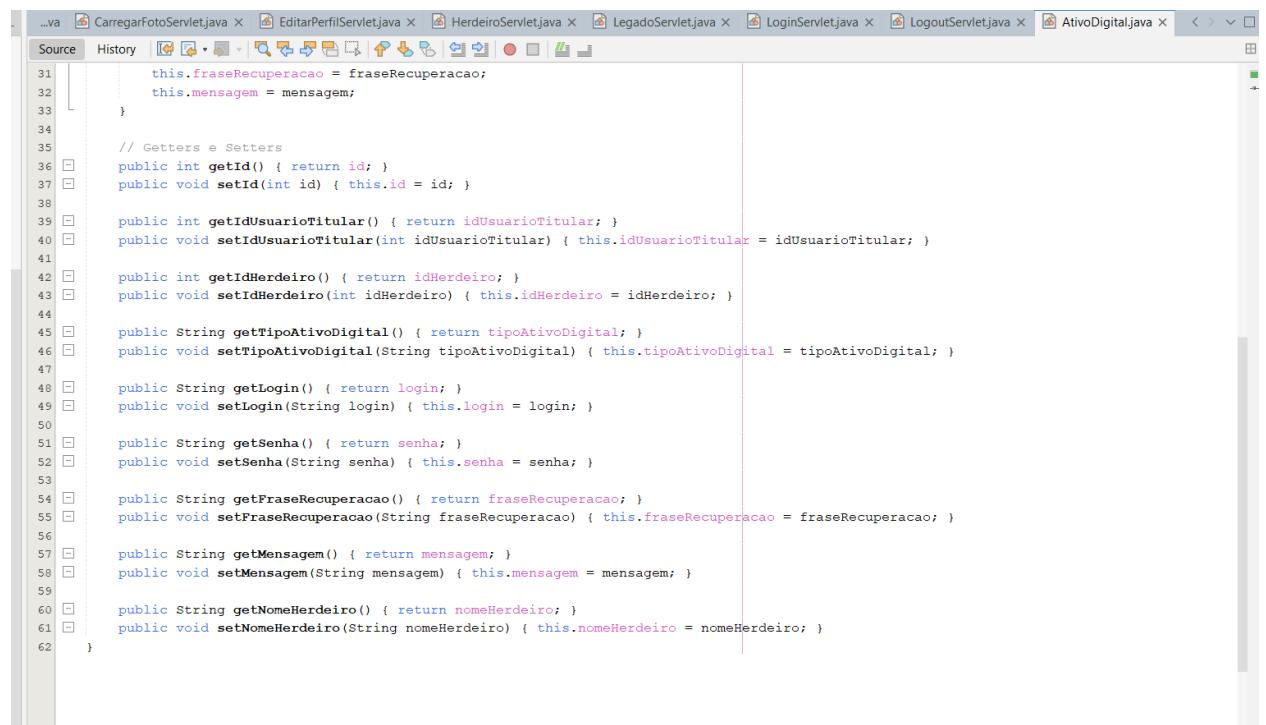
```
25     throws ServletException, IOException {
26
27     // 1. Pega a sessão atual
28     HttpSession session = request.getSession(false); // 'false' não cria uma nova sessão se não existir
29
30     if (session != null) {
31         // 2. Invalida (destrói) a sessão
32         session.invalidate();
33     }
34
35     // 3. Redireciona o usuário de volta para a tela de login
36     // Usamos 'request.getContextPath()' para garantir o caminho correto do projeto
37     response.sendRedirect(request.getContextPath() + "/login_cadastro.html");
38 }
39
40     @Override
41     protected void doGet(HttpServletRequest request, HttpServletResponse response)
42         throws ServletException, IOException {
43         processRequest(request, response);
44     }
45
46     @Override
47     protected void doPost(HttpServletRequest request, HttpServletResponse response)
48         throws ServletException, IOException {
49         processRequest(request, response);
50     }
51
52     @Override
53     public String getServletInfo() {
54         return "Servlet para logout de usuário";
55     }
56 }
```

5.8. Source Package Model

5.8.1 AtivoDigital.java

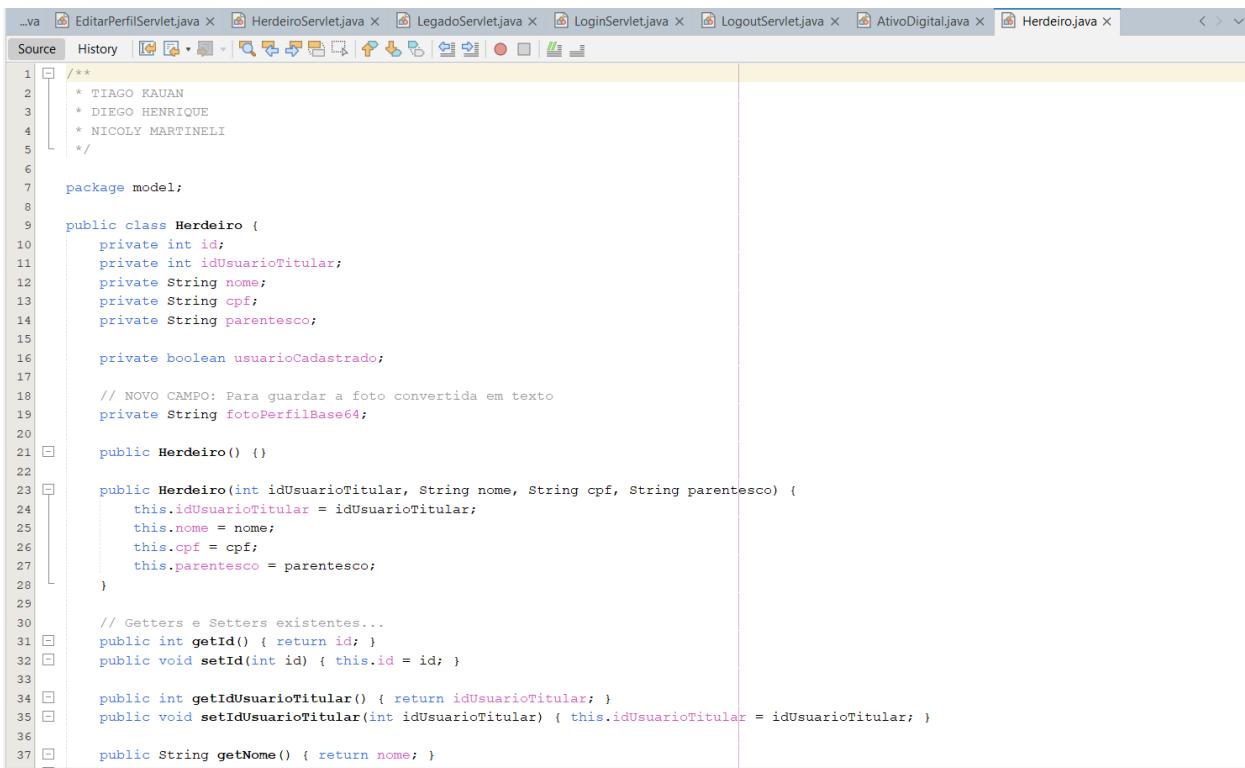


```
1 /**
2  * TIAGO KAUAN
3  * DIEGO HENRIQUE
4  * NICOLY MARTINELI
5 */
6
7 package model;
8
9 public class AtivoDigital {
10     private int id;
11     private int idUsuarioTitular;
12     private int idHerdeiro;
13     private String tipoAtivoDigital; // Variável renomeada
14
15     private String login;
16     private String senha;
17     private String fraseRecuperacao;
18     private String mensagem;
19
20     // Auxiliar para exibição
21     private String nomeHerdeiro;
22
23     public AtivoDigital() {}
24
25     public AtivoDigital(int idUsuarioTitular, int idHerdeiro, String tipoAtivoDigital, String login, String senha, String fraseRecuperacao, String mensagem) {
26         this.idUsuarioTitular = idUsuarioTitular;
27         this.idHerdeiro = idHerdeiro;
28         this.tipoAtivoDigital = tipoAtivoDigital;
29         this.login = login;
30         this.senha = senha;
31         this.fraseRecuperacao = fraseRecuperacao;
32         this.mensagem = mensagem;
33     }
34
35     // Getters e Setters
36     public int getId() { return id; }
37     public void setId(int id) { this.id = id; }
38
39     public int getIdUsuarioTitular() { return idUsuarioTitular; }
40     public void setIdUsuarioTitular(int idUsuarioTitular) { this.idUsuarioTitular = idUsuarioTitular; }
41
42     public int getIdHerdeiro() { return idHerdeiro; }
43     public void setIdHerdeiro(int idHerdeiro) { this.idHerdeiro = idHerdeiro; }
44
45     public String getTipoAtivoDigital() { return tipoAtivoDigital; }
46     public void setTipoAtivoDigital(String tipoAtivoDigital) { this.tipoAtivoDigital = tipoAtivoDigital; }
47
48     public String getLogin() { return login; }
49     public void setLogin(String login) { this.login = login; }
50
51     public String getSenha() { return senha; }
52     public void setSenha(String senha) { this.senha = senha; }
53
54     public String getFraseRecuperacao() { return fraseRecuperacao; }
55     public void setFraseRecuperacao(String fraseRecuperacao) { this.fraseRecuperacao = fraseRecuperacao; }
56
57     public String getMensagem() { return mensagem; }
58     public void setMensagem(String mensagem) { this.mensagem = mensagem; }
59
60     public String getNomeHerdeiro() { return nomeHerdeiro; }
61     public void setNomeHerdeiro(String nomeHerdeiro) { this.nomeHerdeiro = nomeHerdeiro; }
62 }
```

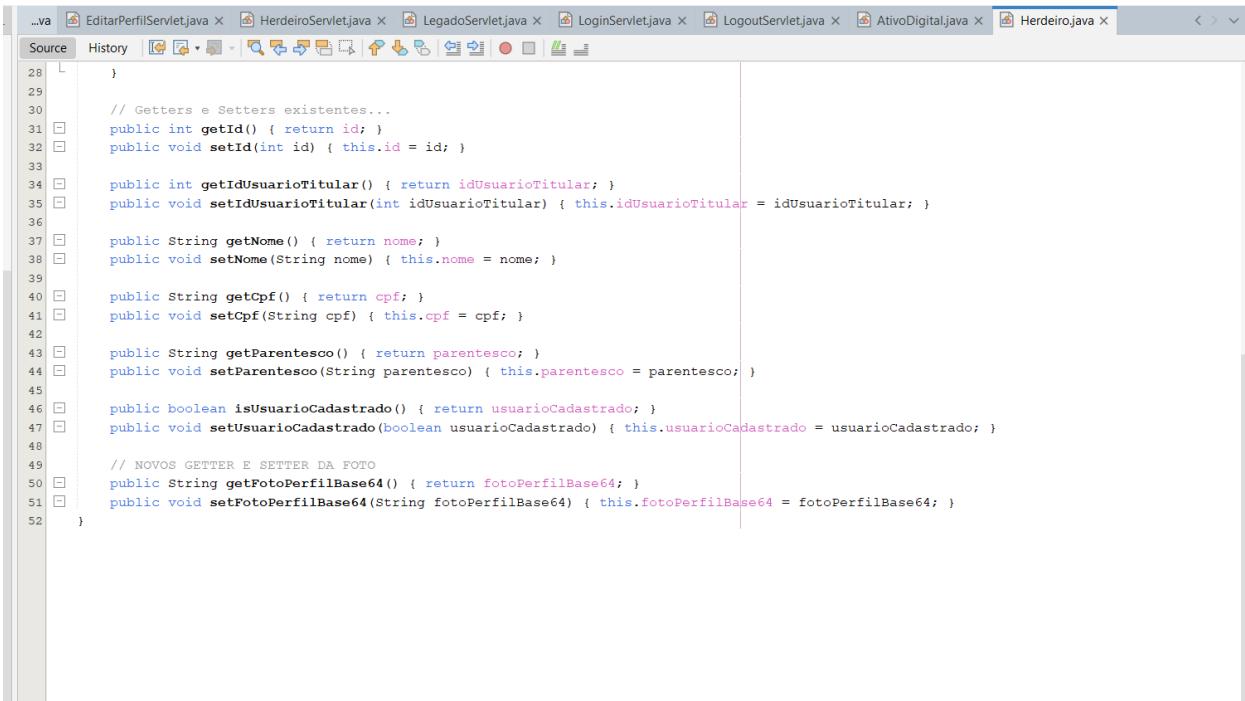


```
31     this.fraseRecuperacao = fraseRecuperacao;
32     this.mensagem = mensagem;
33 }
34
35     // Getters e Setters
36     public int getId() { return id; }
37     public void setId(int id) { this.id = id; }
38
39     public int getIdUsuarioTitular() { return idUsuarioTitular; }
40     public void setIdUsuarioTitular(int idUsuarioTitular) { this.idUsuarioTitular = idUsuarioTitular; }
41
42     public int getIdHerdeiro() { return idHerdeiro; }
43     public void setIdHerdeiro(int idHerdeiro) { this.idHerdeiro = idHerdeiro; }
44
45     public String getTipoAtivoDigital() { return tipoAtivoDigital; }
46     public void setTipoAtivoDigital(String tipoAtivoDigital) { this.tipoAtivoDigital = tipoAtivoDigital; }
47
48     public String getLogin() { return login; }
49     public void setLogin(String login) { this.login = login; }
50
51     public String getSenha() { return senha; }
52     public void setSenha(String senha) { this.senha = senha; }
53
54     public String getFraseRecuperacao() { return fraseRecuperacao; }
55     public void setFraseRecuperacao(String fraseRecuperacao) { this.fraseRecuperacao = fraseRecuperacao; }
56
57     public String getMensagem() { return mensagem; }
58     public void setMensagem(String mensagem) { this.mensagem = mensagem; }
59
60     public String getNomeHerdeiro() { return nomeHerdeiro; }
61     public void setNomeHerdeiro(String nomeHerdeiro) { this.nomeHerdeiro = nomeHerdeiro; }
62 }
```

5.8.2 Herdeiro.java

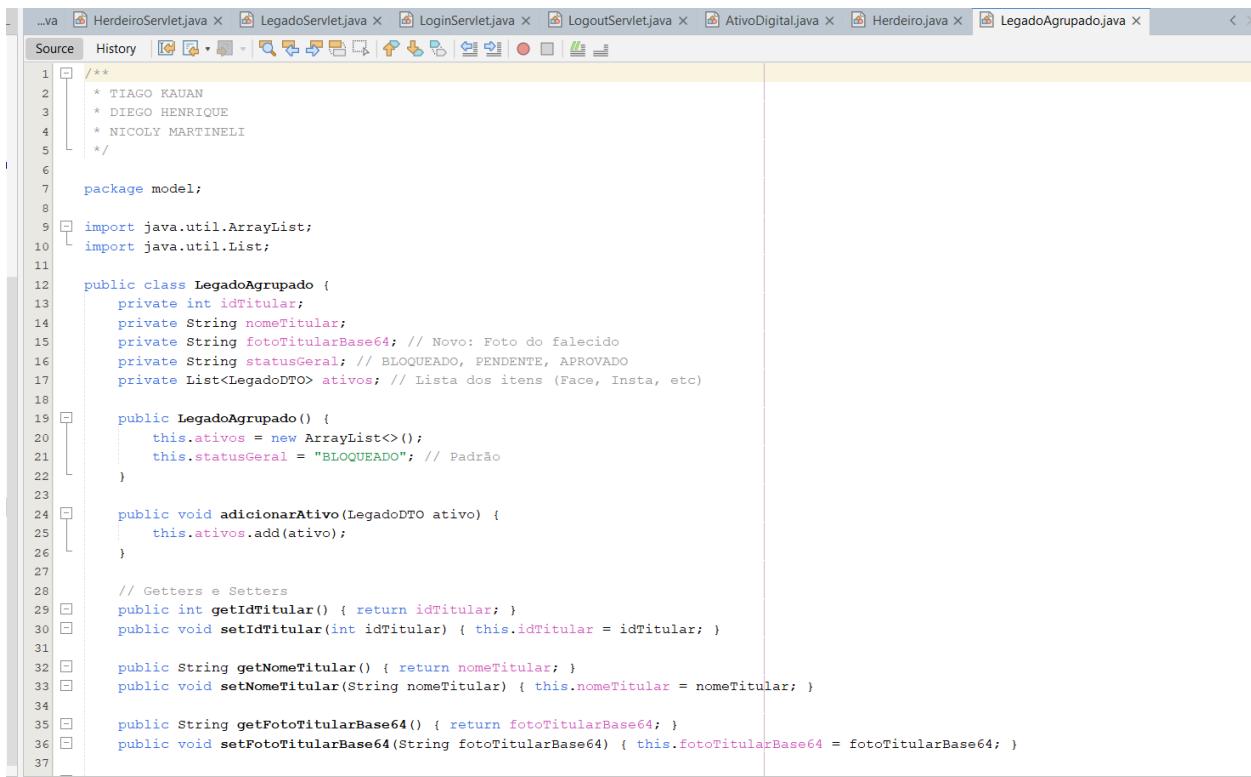


```
1 /**
2  * TIAGO KAUAN
3  * DIEGO HENRIQUE
4  * NICOLY MARTINELI
5 */
6
7 package model;
8
9 public class Herdeiro {
10     private int id;
11     private int idUsuarioTitular;
12     private String nome;
13     private String cpf;
14     private String parentesco;
15
16     private boolean usuarioCadastrado;
17
18     // NOVO CAMPO: Para guardar a foto convertida em texto
19     private String fotoPerfilBase64;
20
21     public Herdeiro() {}
22
23     public Herdeiro(int idUsuarioTitular, String nome, String cpf, String parentesco) {
24         this.idUsuarioTitular = idUsuarioTitular;
25         this.nome = nome;
26         this.cpf = cpf;
27         this.parentesco = parentesco;
28     }
29
30     // Getters e Setters existentes...
31     public int getId() { return id; }
32     public void setId(int id) { this.id = id; }
33
34     public int getIdUsuarioTitular() { return idUsuarioTitular; }
35     public void setIdUsuarioTitular(int idUsuarioTitular) { this.idUsuarioTitular = idUsuarioTitular; }
36
37     public String getNome() { return nome; }
38
39 }
```

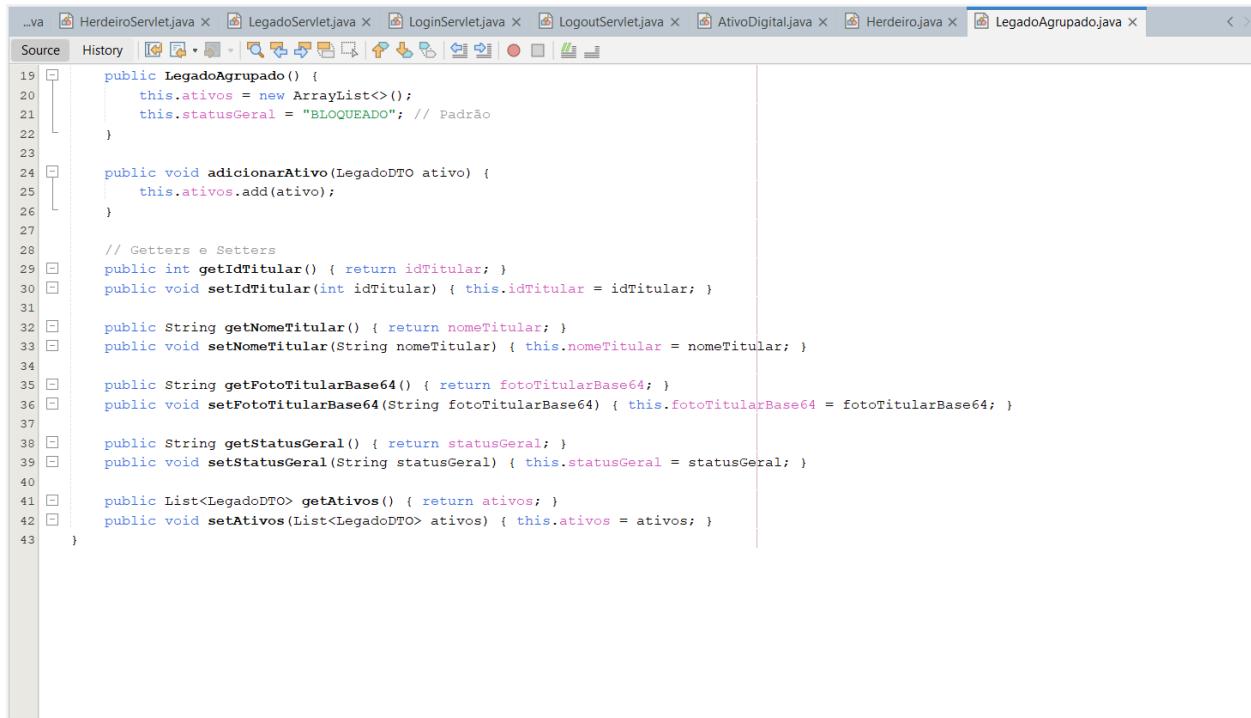


```
28 }
29
30     // Getters e Setters existentes...
31     public int getId() { return id; }
32     public void setId(int id) { this.id = id; }
33
34     public int getIdUsuarioTitular() { return idUsuarioTitular; }
35     public void setIdUsuarioTitular(int idUsuarioTitular) { this.idUsuarioTitular = idUsuarioTitular; }
36
37     public String getNome() { return nome; }
38     public void setNome(String nome) { this.nome = nome; }
39
40     public String getCpf() { return cpf; }
41     public void setCpf(String cpf) { this.cpf = cpf; }
42
43     public String getParentesco() { return parentesco; }
44     public void setParentesco(String parentesco) { this.parentesco = parentesco; }
45
46     public boolean isUsuarioCadastrado() { return usuarioCadastrado; }
47     public void setUsuarioCadastrado(boolean usuarioCadastrado) { this.usuarioCadastrado = usuarioCadastrado; }
48
49     // NOVOS GETTER E SETTER DA FOTO
50     public String getFotoPerfilBase64() { return fotoPerfilBase64; }
51     public void setFotoPerfilBase64(String fotoPerfilBase64) { this.fotoPerfilBase64 = fotoPerfilBase64; }
52 }
```

5.8.3 LegadoAgrupado.java

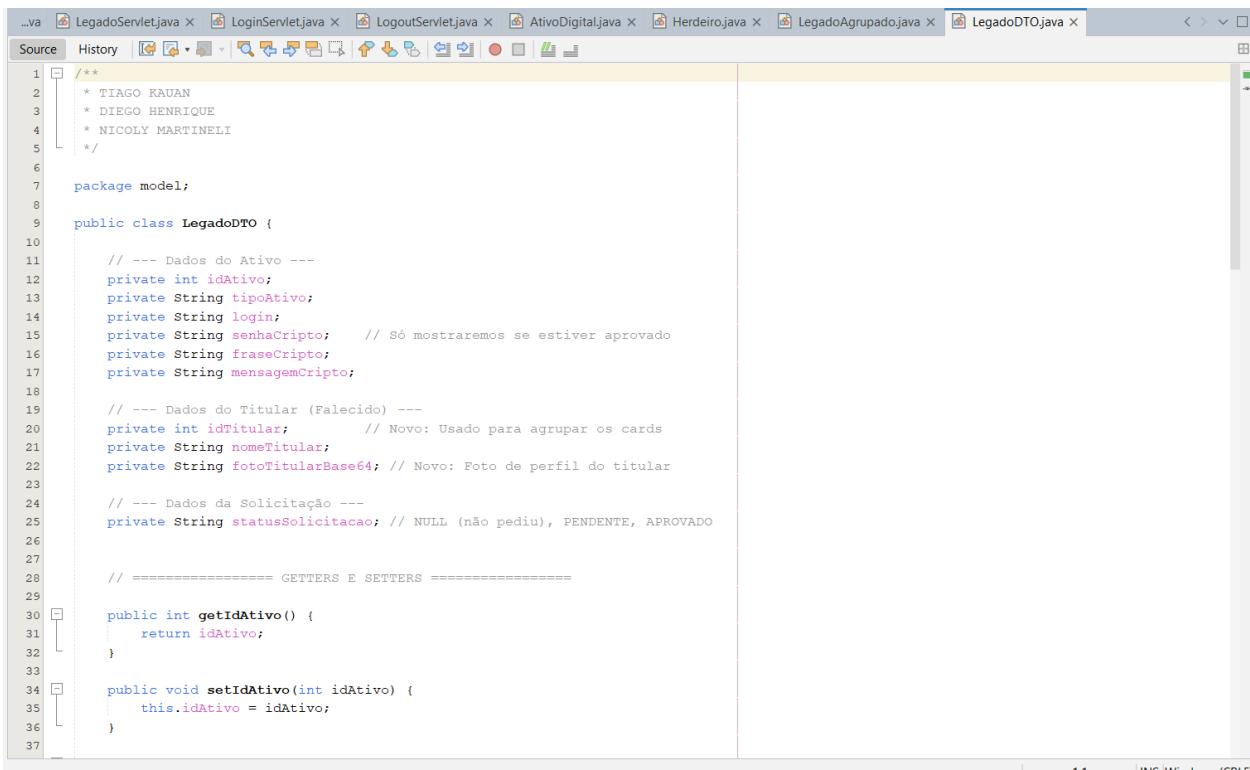


```
1  /**
2  * TIAGO KAUAN
3  * DIEGO HENRIQUE
4  * NICOLY MARTINELI
5  */
6
7 package model;
8
9 import java.util.ArrayList;
10 import java.util.List;
11
12 public class LegadoAgrupado {
13     private int idTitular;
14     private String nomeTitular;
15     private String fotoTitularBase64; // Novo: Foto do falecido
16     private String statusGeral; // BLOQUEADO, PENDENTE, APROVADO
17     private List<LegadoDTO> ativos; // Lista dos itens (Face, Insta, etc)
18
19     public LegadoAgrupado() {
20         this.ativos = new ArrayList<>();
21         this.statusGeral = "BLOQUEADO"; // Padrão
22     }
23
24     public void adicionarAtivo(LegadoDTO ativo) {
25         this.ativos.add(ativo);
26     }
27
28     // Getters e Setters
29     public int getIdTitular() { return idTitular; }
30     public void setIdTitular(int idTitular) { this.idTitular = idTitular; }
31
32     public String getNomeTitular() { return nomeTitular; }
33     public void setNomeTitular(String nomeTitular) { this.nomeTitular = nomeTitular; }
34
35     public String getFotoTitularBase64() { return fotoTitularBase64; }
36     public void setFotoTitularBase64(String fotoTitularBase64) { this.fotoTitularBase64 = fotoTitularBase64; }
37
38 }
```

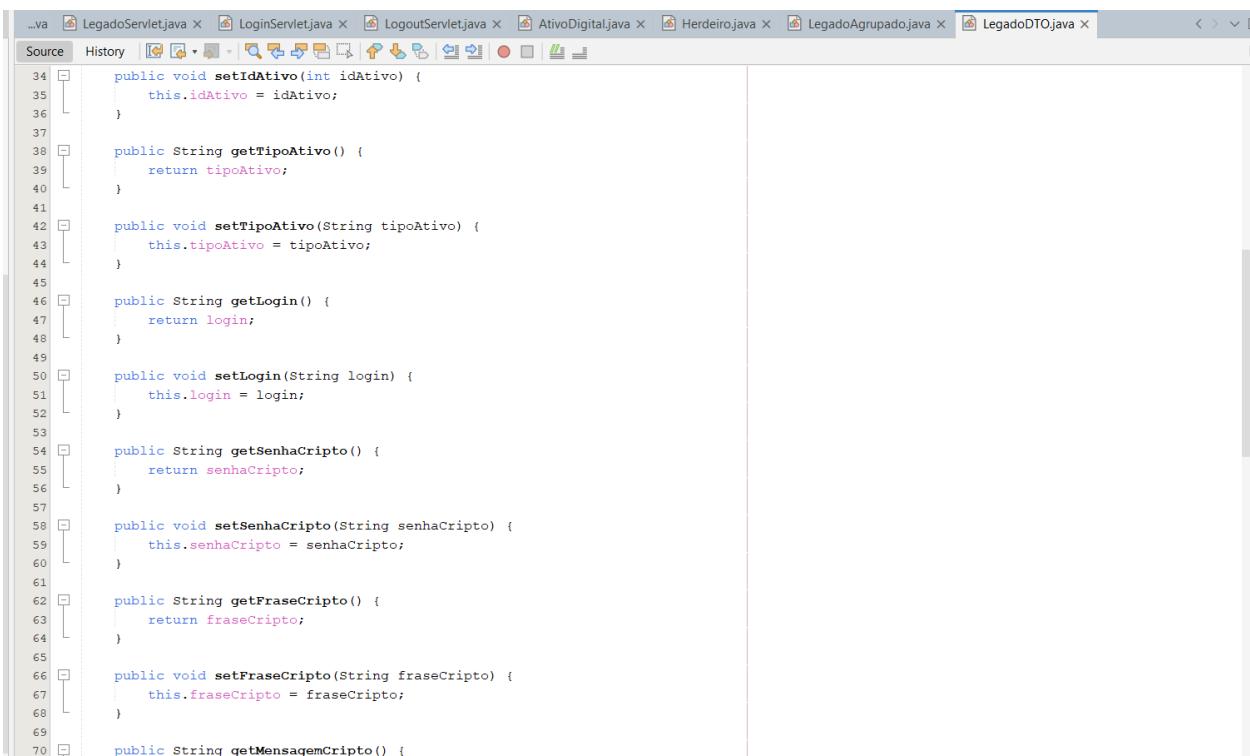


```
19     public LegadoAgrupado() {
20         this.ativos = new ArrayList<>();
21         this.statusGeral = "BLOQUEADO"; // Padrão
22     }
23
24     public void adicionarAtivo(LegadoDTO ativo) {
25         this.ativos.add(ativo);
26     }
27
28     // Getters e Setters
29     public int getIdTitular() { return idTitular; }
30     public void setIdTitular(int idTitular) { this.idTitular = idTitular; }
31
32     public String getNomeTitular() { return nomeTitular; }
33     public void setNomeTitular(String nomeTitular) { this.nomeTitular = nomeTitular; }
34
35     public String getFotoTitularBase64() { return fotoTitularBase64; }
36     public void setFotoTitularBase64(String fotoTitularBase64) { this.fotoTitularBase64 = fotoTitularBase64; }
37
38     public String getStatusGeral() { return statusGeral; }
39     public void setStatusGeral(String statusGeral) { this.statusGeral = statusGeral; }
40
41     public List<LegadoDTO> getAtivos() { return ativos; }
42     public void setAtivos(List<LegadoDTO> ativos) { this.ativos = ativos; }
43 }
```

5.8.4 LegadoDTO.java



```
1 /**
2  * TIAGO KAUAN
3  * DIEGO HENRIQUE
4  * NICOLY MARTINELI
5 */
6
7 package model;
8
9 public class LegadoDTO {
10
11     // --- Dados do Ativo ---
12     private int idAtivo;
13     private String tipoAtivo;
14     private String login;
15     private String senhaCripto; // Só mostraremos se estiver aprovado
16     private String fraseCripto;
17     private String mensagemCripto;
18
19     // --- Dados do Titular (Falecido) ---
20     private int idTitular; // Novo: Usado para agrupar os cards
21     private String nomeTitular;
22     private String fotoTitularBase64; // Novo: Foto de perfil do titular
23
24     // --- Dados da Solicitação ---
25     private String statusSolicitacao; // NULL (não pediu), PENDENTE, APROVADO
26
27
28     // ===== GETTERS E SETTERS =====
29
30     public int getIdAtivo() {
31         return idAtivo;
32     }
33
34     public void setIdAtivo(int idAtivo) {
35         this.idAtivo = idAtivo;
36     }
37
38     public String getTipoAtivo() {
39         return tipoAtivo;
40     }
41
42     public void setTipoAtivo(String tipoAtivo) {
43         this.tipoAtivo = tipoAtivo;
44     }
45
46     public String getLogin() {
47         return login;
48     }
49
50     public void setLogin(String login) {
51         this.login = login;
52     }
53
54     public String getSenhaCripto() {
55         return senhaCripto;
56     }
57
58     public void setSenhaCripto(String senhaCripto) {
59         this.senhaCripto = senhaCripto;
60     }
61
62     public String getFraseCripto() {
63         return fraseCripto;
64     }
65
66     public void setFraseCripto(String fraseCripto) {
67         this.fraseCripto = fraseCripto;
68     }
69
70     public String getMensagemCripto() {
71 }
```



```
34     public void setIdAtivo(int idAtivo) {
35         this.idAtivo = idAtivo;
36     }
37
38     public String getTipoAtivo() {
39         return tipoAtivo;
40     }
41
42     public void setTipoAtivo(String tipoAtivo) {
43         this.tipoAtivo = tipoAtivo;
44     }
45
46     public String getLogin() {
47         return login;
48     }
49
50     public void setLogin(String login) {
51         this.login = login;
52     }
53
54     public String getSenhaCripto() {
55         return senhaCripto;
56     }
57
58     public void setSenhaCripto(String senhaCripto) {
59         this.senhaCripto = senhaCripto;
60     }
61
62     public String getFraseCripto() {
63         return fraseCripto;
64     }
65
66     public void setFraseCripto(String fraseCripto) {
67         this.fraseCripto = fraseCripto;
68     }
69
70     public String getMensagemCripto() {
71 }
```

...va LegadoServlet.java X LoginServlet.java X LogoutServlet.java X AtivoDigital.java X Herdeiro.java X LegadoAgrupado.java X LegadoDTO.java X

Source History

```

70     public String getMensagemCripto() {
71         return mensagemCripto;
72     }
73
74     public void setMensagemCripto(String mensagemCripto) {
75         this.mensagemCripto = mensagemCripto;
76     }
77
78     // ---- Getters/Setters do Titular ---
79
80     public int getIdTitular() {
81         return idTitular;
82     }
83
84     public void setIdTitular(int idTitular) {
85         this.idTitular = idTitular;
86     }
87
88     public String getNomeTitular() {
89         return nomeTitular;
90     }
91
92     public void setNomeTitular(String nomeTitular) {
93         this.nomeTitular = nomeTitular;
94     }
95
96     public String getFotoTitularBase64() {
97         return fotoTitularBase64;
98     }
99
100    public void setFotoTitularBase64(String fotoTitularBase64) {
101        this.fotoTitularBase64 = fotoTitularBase64;
102    }
103
104    // ---- Getters/Setters do Status ---
105
106    public String getStatusSolicitacao() {
107        return statusSolicitacao;
108    }
109
110    public void setStatusSolicitacao(String statusSolicitacao) {
111        this.statusSolicitacao = statusSolicitacao;
112    }
113

```

...va LegadoServlet.java X LoginServlet.java X LogoutServlet.java X AtivoDigital.java X Herdeiro.java X LegadoAgrupado.java X LegadoDTO.java X

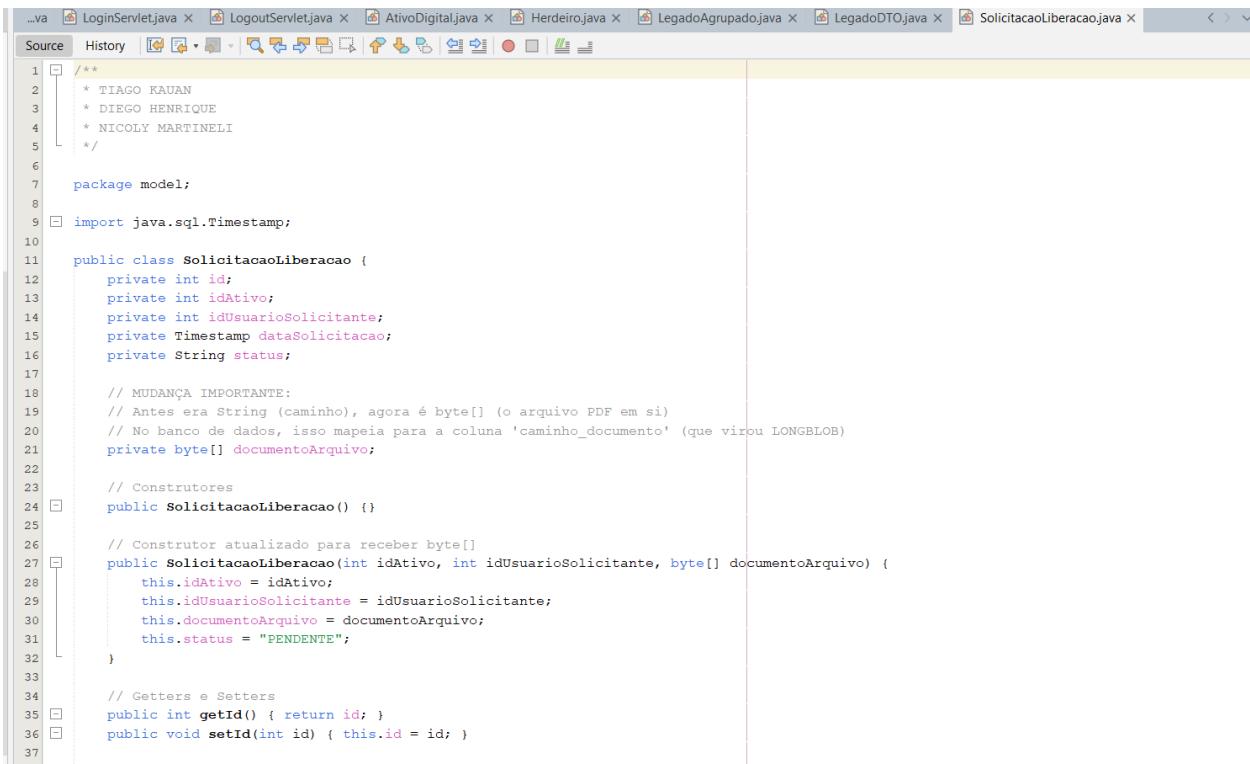
Source History

```

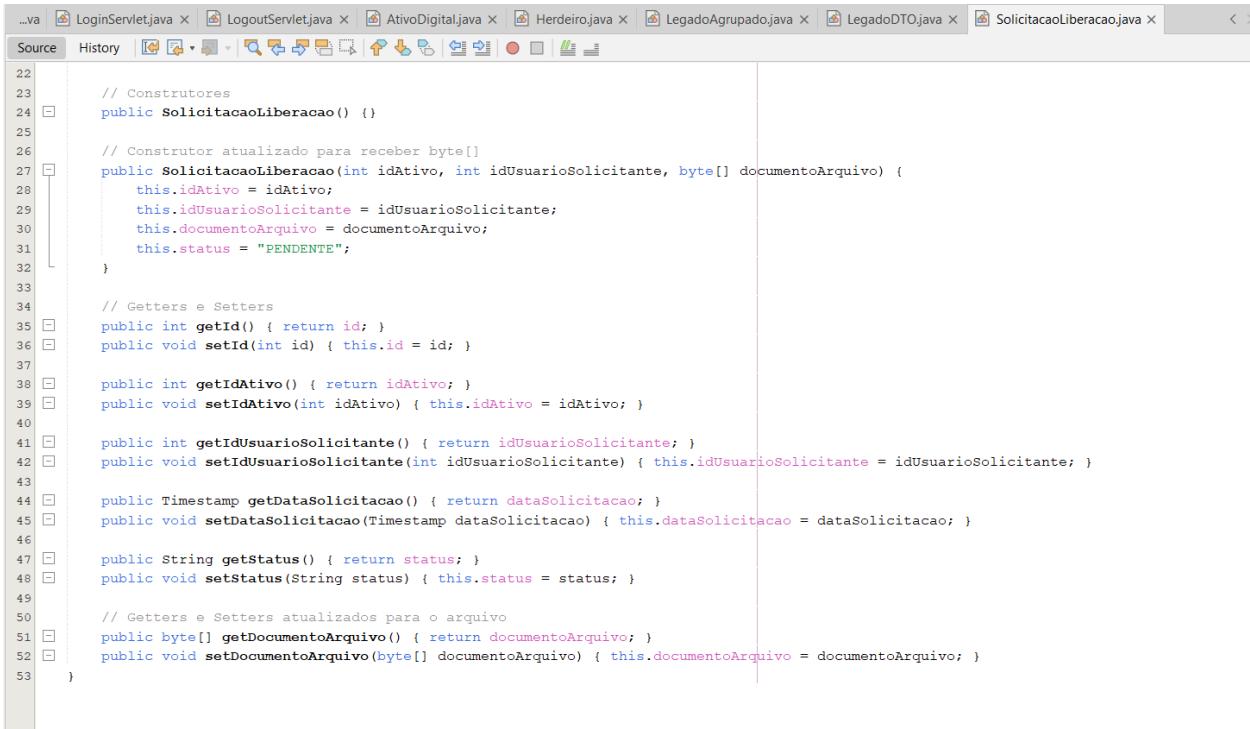
82     }
83
84     public void setIdTitular(int idTitular) {
85         this.idTitular = idTitular;
86     }
87
88     public String getNomeTitular() {
89         return nomeTitular;
90     }
91
92     public void setNomeTitular(String nomeTitular) {
93         this.nomeTitular = nomeTitular;
94     }
95
96     public String getFotoTitularBase64() {
97         return fotoTitularBase64;
98     }
99
100    public void setFotoTitularBase64(String fotoTitularBase64) {
101        this.fotoTitularBase64 = fotoTitularBase64;
102    }
103
104    // ---- Getters/Setters do Status ---
105
106    public String getStatusSolicitacao() {
107        return statusSolicitacao;
108    }
109
110    public void setStatusSolicitacao(String statusSolicitacao) {
111        this.statusSolicitacao = statusSolicitacao;
112    }
113

```

5.8.5 SolicitudLiberacao.java



```
1 /**
2  * TIAGO KAUAN
3  * DIEGO HENRIQUE
4  * NICOLY MARTINELI
5 */
6
7 package model;
8
9 import java.sql.Timestamp;
10
11 public class SolicitudLiberacao {
12     private int id;
13     private int idAtivo;
14     private int idUsuarioSolicitante;
15     private Timestamp dataSolicitacao;
16     private String status;
17
18     // MUDANÇA IMPORTANTE:
19     // Antes era String (caminho), agora é byte[] (o arquivo PDF em si)
20     // No banco de dados, isso mapeia para a coluna 'caminho_documento' (que virou LONGBLOB)
21     private byte[] documentoArquivo;
22
23     // Construtores
24     public SolicitudLiberacao() {}
25
26     // Construtor atualizado para receber byte[]
27     public SolicitudLiberacao(int idAtivo, int idUsuarioSolicitante, byte[] documentoArquivo) {
28         this.idAtivo = idAtivo;
29         this.idUsuarioSolicitante = idUsuarioSolicitante;
30         this.documentoArquivo = documentoArquivo;
31         this.status = "PENDENTE";
32     }
33
34     // Getters e Setters
35     public int getId() { return id; }
36     public void setId(int id) { this.id = id; }
37 }
```



```
22
23
24     // Construtores
25     public SolicitudLiberacao() {}
26
27     // Construtor atualizado para receber byte[]
28     public SolicitudLiberacao(int idAtivo, int idUsuarioSolicitante, byte[] documentoArquivo) {
29         this.idAtivo = idAtivo;
30         this.idUsuarioSolicitante = idUsuarioSolicitante;
31         this.documentoArquivo = documentoArquivo;
32         this.status = "PENDENTE";
33
34     // Getters e Setters
35     public int getId() { return id; }
36     public void setId(int id) { this.id = id; }
37
38     public int getIdAtivo() { return idAtivo; }
39     public void setIdAtivo(int idAtivo) { this.idAtivo = idAtivo; }
40
41     public int getIdUsuarioSolicitante() { return idUsuarioSolicitante; }
42     public void setIdUsuarioSolicitante(int idUsuarioSolicitante) { this.idUsuarioSolicitante = idUsuarioSolicitante; }
43
44     public Timestamp getDataSolicitacao() { return dataSolicitacao; }
45     public void setDataSolicitacao(Timestamp dataSolicitacao) { this.dataSolicitacao = dataSolicitacao; }
46
47     public String getStatus() { return status; }
48     public void setStatus(String status) { this.status = status; }
49
50     // Getters e Setters atualizados para o arquivo
51     public byte[] getDocumentoArquivo() { return documentoArquivo; }
52     public void setDocumentoArquivo(byte[] documentoArquivo) { this.documentoArquivo = documentoArquivo; }
53 }
```

5.8.6 Usuario.java

The screenshot shows the beginning of the `Usuario.java` class definition. It includes package declarations, class annotations, and constructor definitions.

```
1  /**
2   * TIAGO KAUAN
3   * DIEGO HENRIQUE
4   * NICOLY MARTINELI
5   */
6
7  package model;
8
9  /**
10  * Classe de Modelo (POJO) que representa a entidade 'usuario' do banco de dados.
11  * Esta classe é baseada nos campos do formulário de cadastro
12  * (Nome, CPF, Senha, Palavra-Chave) e na tabela 'usuario' do schema SQL.
13  */
14 public class Usuario {
15
16     private int id;
17     private String nome;
18     private String cpf; // Usado como login e identificador único
19     private String senhaHash; // Armazena a senha com hash (ex: SHA-256)
20     private String palavraChaveHash; // Armazena a palavra-chave com hash
21     private byte[] fotoPerfil; // Armazena os bytes da imagem (MEDIUMBLOB)
22
23     /**
24      * Construtor padrão (vazio).
25      */
26     public Usuario() {
27
28     }
29
30     /**
31      * Construtor para INSERIR um novo usuário (sem id, pois é AUTO_INCREMENT).
32      * A foto é nula no cadastro inicial.
33      */
34     public Usuario(String nome, String cpf, String senhaHash, String palavraChaveHash) {
35         this.nome = nome;
36         this.cpf = cpf;
37         this.senhaHash = senhaHash;
38         this.palavraChaveHash = palavraChaveHash;
```

The screenshot shows the completion of the `Usuario.java` class definition, including all fields and their corresponding getters and setters.

```
37         this.palavraChaveHash = palavraChaveHash;
38         // fotoPerfil começa como null
39     }
40
41     /**
42      * Construtor completo para LER um usuário do banco de dados.
43      */
44     public Usuario(int id, String nome, String cpf, String senhaHash, String palavraChaveHash, byte[] fotoPerfil) {
45         this.id = id;
46         this.nome = nome;
47         this.cpf = cpf;
48         this.senhaHash = senhaHash;
49         this.palavraChaveHash = palavraChaveHash;
50         this.fotoPerfil = fotoPerfil;
51     }
52
53     // --- Getters e Setters ---
54
55     public int getId() {
56         return id;
57     }
58
59     public void setId(int id) {
60         this.id = id;
61     }
62
63     public String getNome() {
64         return nome;
65     }
66
67     public void setNome(String nome) {
68         this.nome = nome;
69     }
70
71     public String getCpf() {
72         return cpf;
73     }
```

The screenshot shows a Java IDE interface with the 'Source' tab selected. The code editor displays the 'Usuario.java' file, which contains the following Java code:

```
73 }
74
75     public void setCpf(String cpf) {
76         this.cpf = cpf;
77     }
78
79     public String getSenhaHash() {
80         return senhaHash;
81     }
82
83     public void setSenhaHash(String senhaHash) {
84         this.senhaHash = senhaHash;
85     }
86
87     public String getPalavraChaveHash() {
88         return palavraChaveHash;
89     }
90
91     public void setPalavraChaveHash(String palavraChaveHash) {
92         this.palavraChaveHash = palavraChaveHash;
93     }
94
95     public byte[] getFotoPerfil() {
96         return fotoPerfil;
97     }
98
99     public void setFotoPerfil(byte[] fotoPerfil) {
100        this.fotoPerfil = fotoPerfil;
101    }
102 }
```

5.9. Source Package ModelDAO

5.9.1 AtivoDigitalDAO.java

```
1 /**
2  * TIAGO KAUAN
3  * DIEGO HENRIQUE
4  * NICOLY MARTINELI
5 */
6
7 package model.DAO;
8
9 import config.ConectaDB;
10 import config.CryptoUtil; // Importamos a nova classe
11 import model.AtkivoDigital;
12 import java.sql.*;
13 import java.util.ArrayList;
14 import java.util.List;
15
16 public class AtivoDigitalDAO {
17
18     // Cadastrar (COM CRIPTOGRAFIA)
19     public boolean cadastrarAtivoDigital(AtivoDigital ad) {
20         String sql = "INSERT INTO ativodigital (id_usuario_titular, id_herdeiro, tipo_ativodigital, login, senha, frase_recuperacao, mensagem_criptografada) " +
21             + "VALUES (?, ?, ?, ?, ?, ?, ?)";
22
23         try (Connection conn = ConectaDB.conectar();
24              PreparedStatement ps = conn.prepareStatement(sql)) {
25
26             ps.setInt(1, ad.getIdUsuarioTitular());
27             ps.setInt(2, ad.getIdHerdeiro());
28             ps.setString(3, ad.getTipoAtivoDigital());
29
30             // Dados normais
31             ps.setString(4, ad.getLogin());
32
33             // --- CRIPTOGRAFIA APLICADA AQUI ---
34             // Só criptografa se o dado existir
35             String senhaCripto = (ad.getSenha() != null) ? CryptoUtil.criptografar(ad.getSenha()) : null;
36             String fraseCripto = (ad.getFraseRecuperacao() != null) ? CryptoUtil.criptografar(ad.getFraseRecuperacao()) : null;
37             String msgCripto = (ad.getMensagem() != null) ? CryptoUtil.criptografar(ad.getMensagem()) : null;
38
39             ps.setString(6, fraseCripto);
40             ps.setString(7, msgCripto);
41
42             // -----
43
44             return ps.executeUpdate() > 0;
45         } catch (Exception e) {
46             e.printStackTrace();
47             return false;
48         }
49     }
50
51     // Verificar Duplicidade (Mantém igual, compara string normal com banco)
52     // Obs: Se quiser verificar duplicidade de senha criptografada, teria que criptografar a entrada antes de comparar.
53     // Mas como login geralmente não é criptografado, o método abaixo continua funcionando para login.
54     public boolean verificarDuplicidade(int idTitular, String tipo, String login) {
55         if (login == null || login.isEmpty()) { return false; }
56         String sql = "SELECT count(*) FROM ativodigital WHERE id_usuario_titular = ? AND tipo_ativodigital = ? AND login = ?";
57         try (Connection conn = ConectaDB.conectar(); PreparedStatement ps = conn.prepareStatement(sql)) {
58             ps.setInt(1, idTitular);
59             ps.setString(2, tipo);
60             ps.setString(3, login);
61             ResultSet rs = ps.executeQuery();
62             if (rs.next()) { return rs.getInt(1) > 0; }
63         } catch (Exception e) { e.printStackTrace(); }
64         return false;
65     }
66
67     // Listar (Traz os dados criptografados do banco, o Java não precisa ler eles na dashboard)
68     public List<AtivoDigital> listarAtivosDigitaisDoTitular(int idTitular) {
69         List<AtivoDigital> lista = new ArrayList<>();
70         String sql = "SELECT a.* , h.nome as nome_herdeiro " +
71             "FROM ativodigital a " +
72             "JOIN herdeiro h ON a.id_herdeiro = h.id " +
73             "WHERE a.id_usuario_titular = ?";
74
75         try (Connection conn = ConectaDB.conectar();
76              PreparedStatement ps = conn.prepareStatement(sql)) {
```

```
77             ps.setInt(1, idTitular);
78             ResultSet rs = ps.executeQuery();
79             while (rs.next()) {
80                 AtivoDigital ad = new AtivoDigital();
81                 ad.setId(rs.getInt("id"));
82                 ad.setIdUsuarioTitular(rs.getInt("id_usuario_titular"));
83                 ad.setIdHerdeiro(rs.getInt("id_herdeiro"));
84                 ad.setTipoAtivoDigital(rs.getString("tipo_ativodigital"));
85                 ad.setLogin(rs.getString("login"));
86                 ad.setSenha(rs.getString("senha"));
87                 ad.setFraseRecuperacao(rs.getString("frase_recuperacao"));
88                 ad.setMensagem(rs.getString("mensagem_criptografada"));
89                 ad.setNomeHerdado(rs.getString("nome_herdeiro"));
90
91                 lista.add(ad);
92             }
93         } catch (Exception e) { e.printStackTrace(); }
94     }
95 }
```

```
79         ResultSet rs = ps.executeQuery();
80
81         while (rs.next()) {
82             AtivoDigital ad = new AtivoDigital();
83             ad.setId(rs.getInt("id"));
84             ad.setIdUsuarioTitular(rs.getInt("id_usuario_titular"));
85             ad.setIdHerdeiro(rs.getInt("id_herdeiro"));
86             ad.setTipoAtivoDigital(rs.getString("tipo_ativodigital"));
87             ad.setLogin(rs.getString("login")); // Login continua legivel
88
89             // Os campos abaixo virão "embaralhados" (criptografados) do banco
90             // O dashboard só verifica se é != null, então tudo bem vir embaralhado.
91             ad.setSenha(rs.getString("senha"));
92             ad.setFraseRecuperacao(rs.getString("frase_recuperacao"));
93             ad.setMensagem(rs.getString("mensagem"));
94
95             ad.setNomeHerdeiro(rs.getString("nome_herdeiro"));
96
97             lista.add(ad);
98         }
99     } catch (Exception e) {
100        e.printStackTrace();
101    }
102    return lista;
103}
104
105 // Excluir (Mantém igual)
106 public boolean excluirAtivoDigital(int idAtivo) {
107     String sql = "DELETE FROM atividigital WHERE id = ?";
108     try (Connection conn = ConectaDB.conectar();
109          PreparedStatement ps = conn.prepareStatement(sql)) {
110         ps.setInt(1, idAtivo);
111         return ps.executeUpdate() > 0;
112     } catch (Exception e) {
113        e.printStackTrace();
114    }
115    return false;
116}
```

```
88
89         // Os campos abaixo virão "embaralhados" (criptografados) do banco
90         // O dashboard só verifica se é != null, então tudo bem vir embaralhado.
91         ad.setSenha(rs.getString("senha"));
92         ad.setFraseRecuperacao(rs.getString("frase_recuperacao"));
93         ad.setMensagem(rs.getString("mensagem"));
94
95         ad.setNomeHerdeiro(rs.getString("nome_herdeiro"));
96
97         lista.add(ad);
98     } catch (Exception e) {
99        e.printStackTrace();
100    }
101    return lista;
102}
103
104
105 // Excluir (Mantém igual)
106 public boolean excluirAtivoDigital(int idAtivo) {
107     String sql = "DELETE FROM atividigital WHERE id = ?";
108     try (Connection conn = ConectaDB.conectar();
109          PreparedStatement ps = conn.prepareStatement(sql)) {
110         ps.setInt(1, idAtivo);
111         return ps.executeUpdate() > 0;
112     } catch (Exception e) {
113        e.printStackTrace();
114    }
115    return false;
116}
```

5.9.2 HerdeiroDAO.java

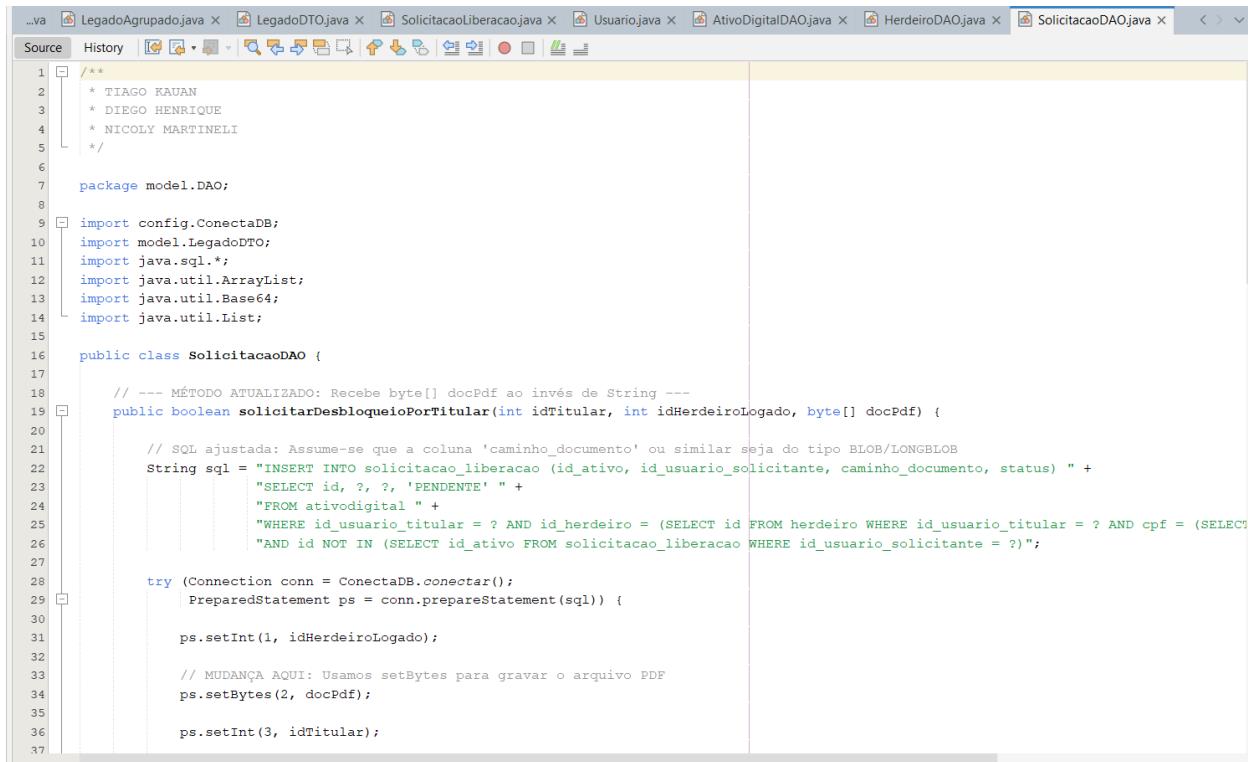
```
1  /**
2  * * TIAGO KAUAN
3  * * DIEGO HENRIQUE
4  * * NICOLY MARTINELI
5  */
6
7 package model.DAO;
8
9 import config.ConectaDB;
10 import model.Herdeiro;
11 import java.sql.*;
12 import java.util.ArrayList;
13 import java.util.Base64; // Importante para converter a foto
14 import java.util.List;
15
16 public class HerdeiroDAO {
17
18     public boolean cadastrarHerdeiro(Herdeiro h) {
19         String sql = "INSERT INTO herdeiro (id_usuario_titular, nome, cpf, parentesco) VALUES (?, ?, ?, ?)";
20         try (Connection conn = ConectaDB.conectar();) {
21             PreparedStatement ps = conn.prepareStatement(sql);
22             ps.setInt(1, h.getIdUsuarioTitular());
23             ps.setString(2, h.getNome());
24             ps.setString(3, h.getCpf());
25             ps.setString(4, h.getParentesco());
26             return ps.executeUpdate() > 0;
27         } catch (Exception e) {
28             e.printStackTrace();
29             return false;
30         }
31     }
32
33     public List<Herdeiro> listarPorUsuario(int idTitular) {
34         List<Herdeiro> lista = new ArrayList<>();
35         String sql = "SELECT * FROM herdeiro WHERE id_usuario_titular = ?";
36
37         try (Connection conn = ConectaDB.conectar();) {
38             PreparedStatement ps = conn.prepareStatement(sql);
39             ps.setInt(1, idTitular);
40             ResultSet rs = ps.executeQuery();
41
42             while (rs.next()) {
43                 Herdeiro h = new Herdeiro();
44                 h.setId(rs.getInt("id"));
45                 h.setIdUsuarioTitular(rs.getInt("id_usuario_titular"));
46                 h.setNome(rs.getString("nome"));
47                 h.setCpf(rs.getString("cpf"));
48                 h.setParentesco(rs.getString("parentesco"));
49
50                 // CHAMA O MÉTODO AUXILIAR QUE PREENCHE FOTO E STATUS
51                 buscarDadosDoUsuarioNaPlataforma(h, conn);
52
53                 lista.add(h);
54             }
55         } catch (Exception e) {
56             e.printStackTrace();
57         }
58         return lista;
59     }
60
61     // MÉTODO NOVO: Verifica se existe e já pega a foto se tiver
62     private void buscarDadosDoUsuarioNaPlataforma(Herdeiro h, Connection conn) {
63         String sql = "SELECT foto_perfil FROM usuario WHERE cpf = ?";
64         try (PreparedStatement ps = conn.prepareStatement(sql)) {
65             ps.setString(1, h.getCpf());
66             ResultSet rs = ps.executeQuery();
67
68             if (rs.next()) {
69                 // Se entrou aqui, o usuário existe
70             }
71         }
72     }
73 }
```

```
34     List<Herdeiro> lista = new ArrayList<>();
35     String sql = "SELECT * FROM herdeiro WHERE id_usuario_titular = ?";
36
37     try (Connection conn = ConectaDB.conectar();) {
38         PreparedStatement ps = conn.prepareStatement(sql);
39
40         ps.setInt(1, idTitular);
41         ResultSet rs = ps.executeQuery();
42
43         while (rs.next()) {
44             Herdeiro h = new Herdeiro();
45             h.setId(rs.getInt("id"));
46             h.setIdUsuarioTitular(rs.getInt("id_usuario_titular"));
47             h.setNome(rs.getString("nome"));
48             h.setCpf(rs.getString("cpf"));
49             h.setParentesco(rs.getString("parentesco"));
50
51             // CHAMA O MÉTODO AUXILIAR QUE PREENCHE FOTO E STATUS
52             buscarDadosDoUsuarioNaPlataforma(h, conn);
53
54             lista.add(h);
55         }
56     } catch (Exception e) {
57         e.printStackTrace();
58     }
59     return lista;
60 }
61
62     // MÉTODO NOVO: Verifica se existe e já pega a foto se tiver
63     private void buscarDadosDoUsuarioNaPlataforma(Herdeiro h, Connection conn) {
64         String sql = "SELECT foto_perfil FROM usuario WHERE cpf = ?";
65         try (PreparedStatement ps = conn.prepareStatement(sql)) {
66             ps.setString(1, h.getCpf());
67             ResultSet rs = ps.executeQuery();
68
69             if (rs.next()) {
70                 // Se entrou aqui, o usuário existe
71             }
72         }
73     }
74 }
```

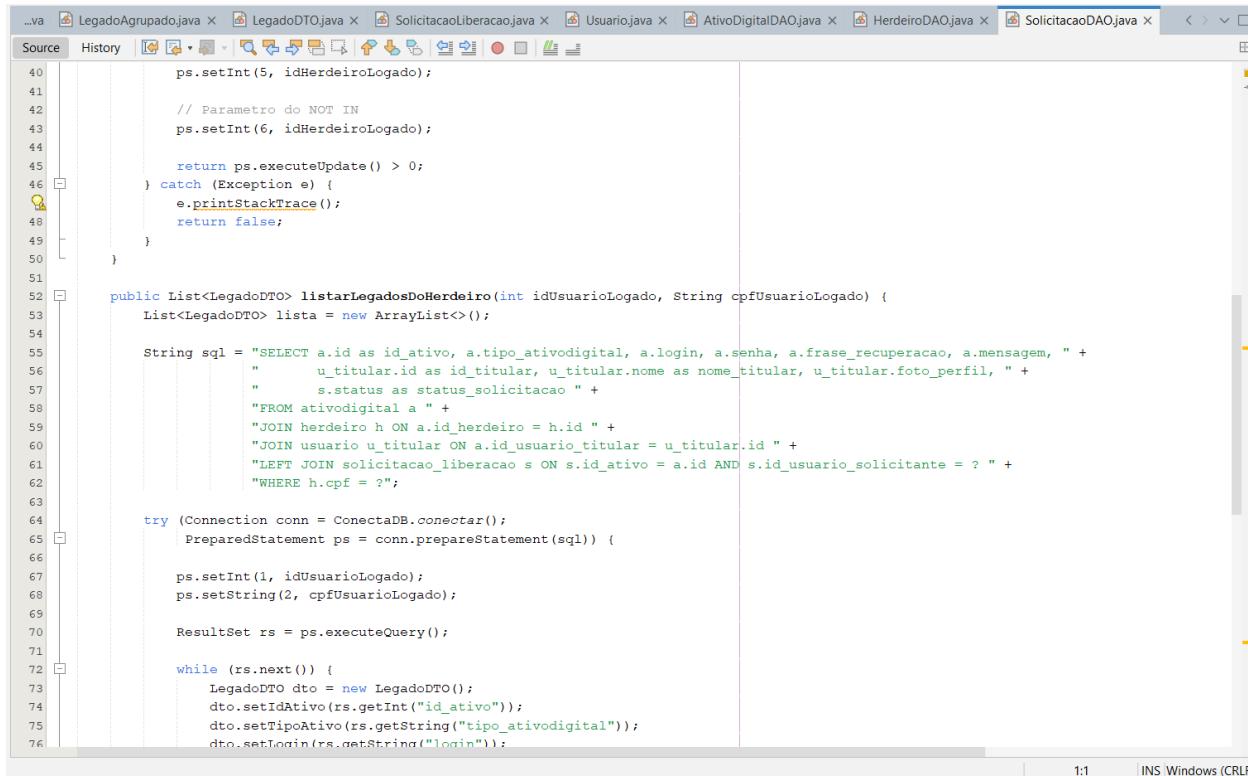
The screenshot shows a Java IDE interface with the 'Source' tab selected. The code editor displays the `HerdeiroDAO.java` file. The code implements methods for managing heirs in a database, including updating user information and deleting heirs.

```
70     // Se entrou aqui, o usuário existe
71     h.setUsuarioCadastrado(true);
72
73     // Tenta pegar a foto (BLOB)
74     Blob blob = rs.getBlob("foto_perfil");
75     if (blob != null) {
76         byte[] imageBytes = blob.getBytes(1, (int) blob.length());
77         if (imageBytes != null && imageBytes.length > 0) {
78             // Converte bytes para String Base64 para exibir no HTML
79             String base64Image = Base64.getEncoder().encodeToString(imageBytes);
80             h.setFotoPerfilBase64(base64Image);
81         }
82     } else {
83         h.setUsuarioCadastrado(false);
84     }
85 } catch (SQLException e) {
86     e.printStackTrace();
87 }
88
89 public boolean excluirHerdeiro(int idHerdeiro) {
90     String sql = "DELETE FROM herdeiro WHERE id = ?";
91     try (Connection conn = ConectaDB.conectar();
92          PreparedStatement ps = conn.prepareStatement(sql)) {
93         ps.setInt(1, idHerdeiro);
94         return ps.executeUpdate() > 0;
95     } catch (Exception e) {
96         e.printStackTrace();
97         return false;
98     }
99 }
100 }
```

5.9.3 SolicitudDAO.java



```
1  /*
2   * TIAGO KAUAN
3   * DIEGO HENRIQUE
4   * NICOLY MARTINELI
5   */
6
7  package model.DAO;
8
9  import config.ConectaDB;
10 import model.LegadoDTO;
11 import java.sql.*;
12 import java.util.ArrayList;
13 import java.util.Base64;
14 import java.util.List;
15
16 public class SolicitudDAO {
17
18     // --- MÉTODO ATUALIZADO: Recebe byte[] docPdf ao invés de String ---
19     public boolean solicitarDesbloqueioPorTitular(int idTitular, int idHerdeiroLogado, byte[] docPdf) {
20
21         // SQL ajustada: Assume-se que a coluna 'caminho_documento' ou similar seja do tipo BLOB/LONGBLOB
22         String sql = "INSERT INTO solicitacao_liberacao (id_ativo, id_usuario_solicitante, caminho_documento, status) " +
23             "SELECT id, ?, ?, 'PENDENTE' " +
24             "FROM atividigital " +
25             "WHERE id_usuario_titular = ? AND id_herdeiro = (SELECT id FROM herdeiro WHERE id_usuario_titular = ? AND cpf = (SELECT
26             "AND id NOT IN (SELECT id_ativo FROM solicitacao_liberacao WHERE id_usuario_solicitante = ?))";
27
28         try (Connection conn = ConectaDB.conectar();
29              PreparedStatement ps = conn.prepareStatement(sql)) {
30
31             ps.setInt(1, idHerdeiroLogado);
32
33             // MUDANÇA AQUI: Usamos setBytes para gravar o arquivo PDF
34             ps.setBytes(2, docPdf);
35
36             ps.setInt(3, idTitular);
37
38         } catch (Exception e) {
39             e.printStackTrace();
40             return false;
41         }
42
43         return ps.executeUpdate() > 0;
44     }
45
46     public List<LegadoDTO> listarLegadosDoHerdado(int idUsuarioLogado, String cpfUsuarioLogado) {
47
48         List<LegadoDTO> lista = new ArrayList<>();
49
50         String sql = "SELECT a.id as id_ativo, a.tipo_atividigital, a.login, a.senha, a.frase_recuperacao, a.mensagem, " +
51             "u_titular.id as id_titular, u_titular.nome as nome_titular, u_titular.foto_perfil, " +
52             "s.status as status_solicitacao " +
53             "FROM atividigital a " +
54             "JOIN herdeiro h ON a.id_herdeiro = h.id " +
55             "JOIN usuario u_titular ON a.id_usuario_titular = u_titular.id " +
56             "LEFT JOIN solicitacao_liberacao s ON s.id_ativo = a.id AND s.id_usuario_solicitante = ? " +
57             "WHERE h.cpf = ?";
58
59         try (Connection conn = ConectaDB.conectar();
60              PreparedStatement ps = conn.prepareStatement(sql)) {
61
62             ps.setInt(1, idUsuarioLogado);
63             ps.setString(2, cpfUsuarioLogado);
64
65             ResultSet rs = ps.executeQuery();
66
67             while (rs.next()) {
68                 LegadoDTO dto = new LegadoDTO();
69                 dto.setIdAtivo(rs.getInt("id_ativo"));
70                 dto.setTipoAtivo(rs.getString("tipo_atividigital"));
71                 dto.setLogin(rs.getString("login"));
72             }
73
74         } catch (Exception e) {
75             e.printStackTrace();
76         }
77
78     }
79
80 }
```

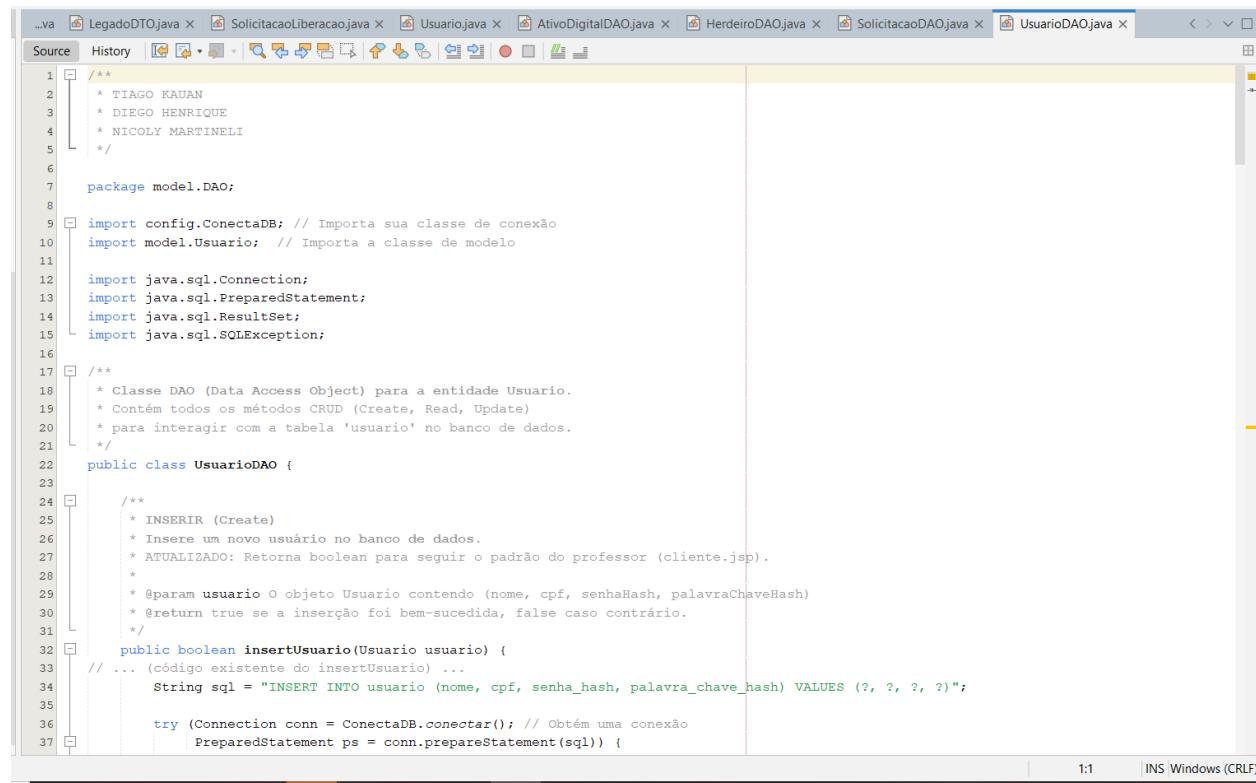


```
40         ps.setInt(5, idHerdeiroLogado);
41
42         // Parametro do NOT IN
43         ps.setInt(6, idHerdeiroLogado);
44
45         return ps.executeUpdate() > 0;
46     } catch (Exception e) {
47         e.printStackTrace();
48         return false;
49     }
50 }
51
52     public List<LegadoDTO> listarLegadosDoHerdado(int idUsuarioLogado, String cpfUsuarioLogado) {
53
54         List<LegadoDTO> lista = new ArrayList<>();
55
56         String sql = "SELECT a.id as id_ativo, a.tipo_atividigital, a.login, a.senha, a.frase_recuperacao, a.mensagem, " +
57             "u_titular.id as id_titular, u_titular.nome as nome_titular, u_titular.foto_perfil, " +
58             "s.status as status_solicitacao " +
59             "FROM atividigital a " +
60             "JOIN herdeiro h ON a.id_herdeiro = h.id " +
61             "JOIN usuario u_titular ON a.id_usuario_titular = u_titular.id " +
62             "LEFT JOIN solicitacao_liberacao s ON s.id_ativo = a.id AND s.id_usuario_solicitante = ? " +
63             "WHERE h.cpf = ?";
64
65         try (Connection conn = ConectaDB.conectar();
66              PreparedStatement ps = conn.prepareStatement(sql)) {
67
68             ps.setInt(1, idUsuarioLogado);
69             ps.setString(2, cpfUsuarioLogado);
70
71             ResultSet rs = ps.executeQuery();
72
73             while (rs.next()) {
74                 LegadoDTO dto = new LegadoDTO();
75                 dto.setIdAtivo(rs.getInt("id_ativo"));
76                 dto.setTipoAtivo(rs.getString("tipo_atividigital"));
77                 dto.setLogin(rs.getString("login"));
78             }
79
80         } catch (Exception e) {
81             e.printStackTrace();
82         }
83
84     }
85
86 }
```

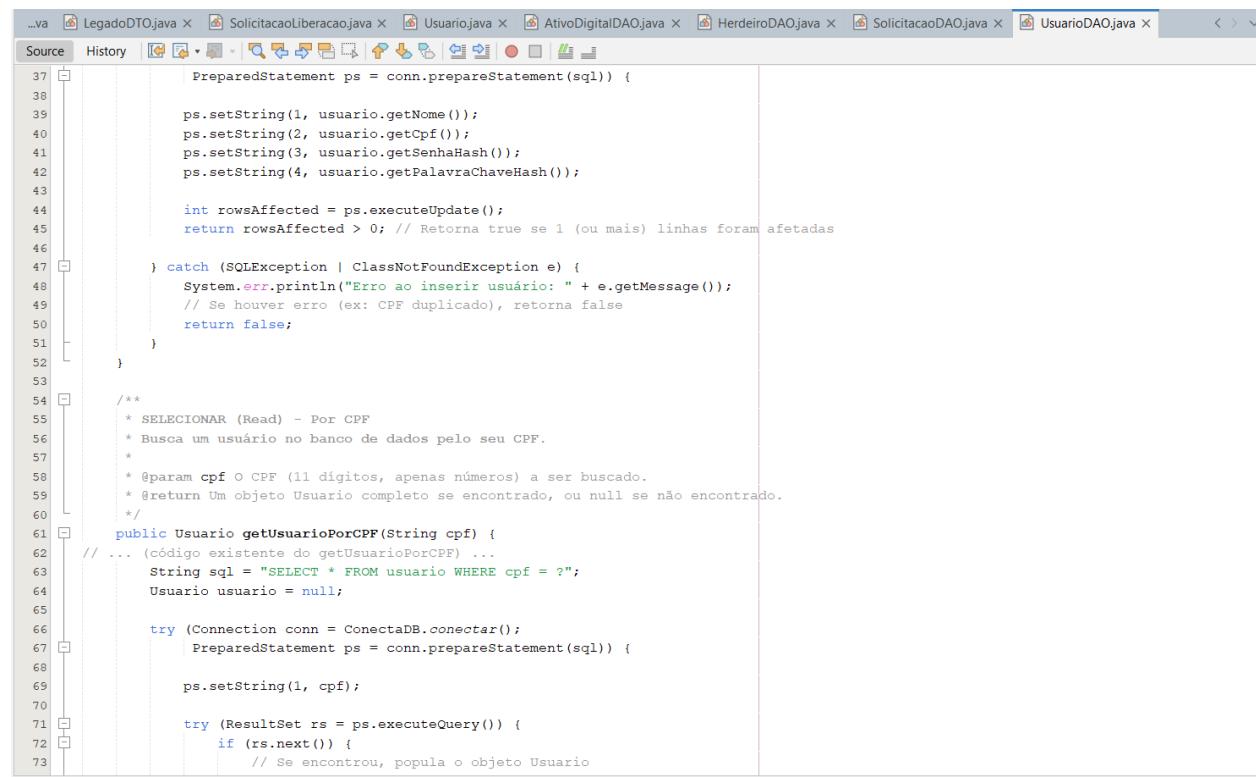
The screenshot shows a Java code editor with the file `SolicitacaoDAO.java` open. The code is a method that retrieves a list of `LegadoDTO` objects from a database. The code uses JDBC to execute a query and map the results to `LegadoDTO` objects. It handles the conversion of a `Blob` object representing a profile photo into a Base64 string. Any exceptions are caught and printed to the stack trace.

```
70     ResultSet rs = ps.executeQuery();
71
72     while (rs.next()) {
73         LegadoDTO dto = new LegadoDTO();
74         dto.setIdAtivo(rs.getInt("id_ativo"));
75         dto.setTipoAtivo(rs.getString("tipo_ativodigital"));
76         dto.setLogin(rs.getString("login"));
77         dto.setSenhaCripto(rs.getString("senha"));
78         dto.setFraseCripto(rs.getString("frase_recuperacao"));
79         dto.setMensagemCripto(rs.getString("mensagem"));
80         dto.setNomeTitular(rs.getString("nome_titular"));
81
82         // --- TRATAMENTO DA FOTO ---
83         Blob blob = rs.getBlob("foto_perfil");
84         if (blob != null) {
85             byte[] imageBytes = blob.getBytes(1, (int) blob.length());
86             String base64 = Base64.getEncoder().encodeToString(imageBytes);
87             dto.setFotoTitularBase64(base64);
88         }
89
90         dto.setIdTitular(rs.getInt("id_titular"));
91
92         String st = rs.getString("status_solicitacao");
93         dto.setStatusSolicitacao(st == null ? "BLOQUEADO" : st);
94
95         lista.add(dto);
96     }
97     catch (Exception e) {
98         e.printStackTrace();
99     }
100    return lista;
101}
102}
```

5.9.4 UsuarioDAO.java



```
1  /**
2   * TIAGO KAUAN
3   * DIEGO HENRIQUE
4   * NICOLY MARTINELI
5   */
6
7  package model.DAO;
8
9  import config.ConectaDB; // Importa sua classe de conexão
10 import model.Usuario; // Importa a classe de modelo
11
12 import java.sql.Connection;
13 import java.sql.PreparedStatement;
14 import java.sql.ResultSet;
15 import java.sql.SQLException;
16
17 /**
18  * Classe DAO (Data Access Object) para a entidade Usuario.
19  * Contém todos os métodos CRUD (Create, Read, Update)
20  * para interagir com a tabela 'usuario' no banco de dados.
21  */
22 public class UsuarioDAO {
23
24     /**
25      * INSERIR (Create)
26      * Insere um novo usuário no banco de dados.
27      * ATUALIZADO: Retorna boolean para seguir o professor (cliente.jsp).
28      *
29      * @param usuario O objeto Usuario contendo (nome, cpf, senhaHash, palavraChaveHash)
30      * @return true se a inserção foi bem-sucedida, false caso contrário.
31      */
32     public boolean insertUsuario(Usuario usuario) {
33         // ... (código existente do insertUsuario) ...
34         String sql = "INSERT INTO usuario (nome, cpf, senha_hash, palavra_chave_hash) VALUES (?, ?, ?, ?)";
35
36         try (Connection conn = ConectaDB.conectar(); // Obtém uma conexão
37              PreparedStatement ps = conn.prepareStatement(sql)) {
38             ps.setString(1, usuario.getNome());
39             ps.setString(2, usuario.getCpf());
40             ps.setString(3, usuario.getSenhaHash());
41             ps.setString(4, usuario.getPalavraChaveHash());
42
43             int rowsAffected = ps.executeUpdate();
44             return rowsAffected > 0; // Retorna true se 1 (ou mais) linhas foram afetadas
45         } catch (SQLException | ClassNotFoundException e) {
46             System.err.println("Erro ao inserir usuário: " + e.getMessage());
47             // Se houver erro (ex: CPF duplicado), retorna false
48             return false;
49         }
50     }
51
52
53     /**
54      * SELECIONAR (Read) - Por CPF
55      * Busca um usuário no banco de dados pelo seu CPF.
56      *
57      * @param cpf O CPF (11 dígitos, apenas números) a ser buscado.
58      * @return Um objeto Usuario completo se encontrado, ou null se não encontrado.
59      */
60     public Usuario getUsuarioPorCPF(String cpf) {
61         // ... (código existente do getUsuarioPorCPF) ...
62         String sql = "SELECT * FROM usuario WHERE cpf = ?";
63         Usuario usuario = null;
64
65         try (Connection conn = ConectaDB.conectar();
66              PreparedStatement ps = conn.prepareStatement(sql)) {
67             ps.setString(1, cpf);
68
69             try (ResultSet rs = ps.executeQuery()) {
70                 if (rs.next()) {
71                     // Se encontrou, popula o objeto Usuario
72                 }
73             }
74         }
75     }
76 }
```



```
75
76     try (ResultSet rs = ps.executeQuery()) {
77         if (rs.next()) {
78             // Se encontrou, popula o objeto Usuario
79         }
80     }
81 }
82 }
```

This screenshot shows a Java code editor with multiple tabs open at the top. The active tab contains Java code for a DAO class. The code includes several try-catch blocks and database interaction logic using JDBC.

```
73     // Se encontrou, popula o objeto Usuario
74     usuario = new Usuario();
75     usuario.setId(rs.getInt("id"));
76     usuario.setNome(rs.getString("nome"));
77     usuario.setCpf(rs.getString("cpf"));
78     usuario.setSenhaHash(rs.getString("senha_hash"));
79     usuario.setPalavraChaveHash(rs.getString("palavra_chave_hash"));
80     usuario.setFotoPerfil(rs.getBytes("foto_perfil"));
81   }
82 }
83
84 } catch (SQLException | ClassNotFoundException e) {
85   System.err.println("Erro ao buscar usuário por CPF: " + e.getMessage());
86 }
87
88 return usuario; // Retorna o usuário (ou null se não achou)
89 }
90
91 /**
92 * SELECONAR (Read) - Por ID
93 * Busca um usuário no banco de dados pelo seu ID (chave primária).
94 *
95 * @param id O ID (int) do usuário.
96 * @return Um objeto Usuario completo se encontrado, ou null se não encontrado.
97 */
98 public Usuario getUsuarioPorId(int id) {
99 // ... (código existente de getUsuarioPorId) ...
100 String sql = "SELECT * FROM usuario WHERE id = ?";
101 Usuario usuario = null;
102
103 try (Connection conn = ConectaDB.conectar();
104      PreparedStatement ps = conn.prepareStatement(sql)) {
105
106   ps.setInt(1, id);
107
108   try (ResultSet rs = ps.executeQuery()) {
109     if (rs.next()) {
110       ps.setInt(1, id);
111
112       try (ResultSet rs = ps.executeQuery()) {
113         if (rs.next()) {
114           usuario = new Usuario();
115           usuario.setId(rs.getInt("id"));
116           usuario.setNome(rs.getString("nome"));
117           usuario.setCpf(rs.getString("cpf"));
118           usuario.setSenhaHash(rs.getString("senha_hash"));
119           usuario.setPalavraChaveHash(rs.getString("palavra_chave_hash"));
120           usuario.setFotoPerfil(rs.getBytes("foto_perfil"));
121         }
122       }
123     }
124   }
125
126 /**
127 * ATUALIZAR (Update) - Recuperação de Senha (versão antiga)
128 * Este método não é seguro para recuperação, pois não valida a palavra-chave.
129 * Vamos manter, mas criar um novo e melhor.
130 */
131 public boolean updateSenhaPorCPF(String cpf, String novaSenhaHash) {
132 // ... (código existente de updateSenhaPorCPF) ...
133 String sql = "UPDATE usuario SET senha_hash = ? WHERE cpf = ?";
134
135 try (Connection conn = ConectaDB.conectar();
136      PreparedStatement ps = conn.prepareStatement(sql)) {
137
138   ps.setString(1, novaSenhaHash);
139   ps.setString(2, cpf);
140
141   int rowsAffected = ps.executeUpdate();
142 }
```

This screenshot shows a Java code editor with multiple tabs open at the top. The active tab contains Java code for updating a user's password. It uses JDBC to execute an UPDATE query.

```
106   ps.setInt(1, id);
107
108   try (ResultSet rs = ps.executeQuery()) {
109     if (rs.next()) {
110       usuario = new Usuario();
111       usuario.setId(rs.getInt("id"));
112       usuario.setNome(rs.getString("nome"));
113       usuario.setCpf(rs.getString("cpf"));
114       usuario.setSenhaHash(rs.getString("senha_hash"));
115       usuario.setPalavraChaveHash(rs.getString("palavra_chave_hash"));
116       usuario.setFotoPerfil(rs.getBytes("foto_perfil"));
117     }
118   }
119 } catch (SQLException | ClassNotFoundException e) {
120   System.err.println("Erro ao buscar usuário por ID: " + e.getMessage());
121 }
122
123 return usuario;
124 }
125
126 /**
127 * ATUALIZAR (Update) - Recuperação de Senha (versão antiga)
128 * Este método não é seguro para recuperação, pois não valida a palavra-chave.
129 * Vamos manter, mas criar um novo e melhor.
130 */
131 public boolean updateSenhaPorCPF(String cpf, String novaSenhaHash) {
132 // ... (código existente de updateSenhaPorCPF) ...
133 String sql = "UPDATE usuario SET senha_hash = ? WHERE cpf = ?";
134
135 try (Connection conn = ConectaDB.conectar();
136      PreparedStatement ps = conn.prepareStatement(sql)) {
137
138   ps.setString(1, novaSenhaHash);
139   ps.setString(2, cpf);
140
141   int rowsAffected = ps.executeUpdate();
142 }
```


The screenshot shows a Java code editor with the file `UsuarioDAO.java` open. The code implements a method to update a user's password securely. It uses prepared statements to prevent SQL injection and handles exceptions.

```
208     // =====
209     // ===== NOVO MÉTODO PARA RECUPERAÇÃO DE SENHA =====
210     // =====
211
212
213     /**
214      * ATUALIZAR (Update) - Redefinição de Senha Segura
215      * Atualiza a senha APENAS SE o CPF e a Palavra Chave baterem.
216      *
217      * @param cpf O CPF do usuário.
218      * @param palavraChaveHash A palavra-chave (já com hash)
219      * @param novaSenhaHash A nova senha (já com hash)
220      * @return true se a atualização foi bem-sucedida, false caso contrário.
221     */
222     public boolean redefinirSenha(String cpf, String palavraChaveHash, String novaSenhaHash) {
223         // Este SQL é atômico: ele só vai atualizar (e retornar 1) se
224         // AMBAS as condições (cpf E palavra_chave_hash) forem verdadeiras.
225         String sql = "UPDATE usuario SET senha_hash = ? WHERE cpf = ? AND palavra_chave_hash = ?";
226
227         try (Connection conn = ConectaDB.conectar();
228              PreparedStatement ps = conn.prepareStatement(sql)) {
229
230             ps.setString(1, novaSenhaHash);
231             ps.setString(2, cpf);
232             ps.setString(3, palavraChaveHash);
233
234             int rowsAffected = ps.executeUpdate();
235
236             // Se rowsAffected > 0, significa que o CPF e a Palavra-Chave bateram
237             return rowsAffected > 0;
238
239         } catch (SQLException | ClassNotFoundException e) {
240             System.err.println("Erro ao redefinir senha: " + e.getMessage());
241             return false;
242         }
243     }
244 }
```

6. EVIDÊNCIAS DO SISTEMA EM EXECUÇÃO (PRINTS)

Figura 1.1: Cadastro de Usuário

The figure consists of two screenshots of a web application for user registration.

Screenshot 1: User Registration Form (localhost:8080/EspolioDigital/login_cadastro.html)

This screenshot shows the user registration form titled "Criar Conta Segura". The form fields include:

- Nome Completo *: "Seu nome completo"
- CPF *: "Apenas números"
- Senha Segura *: "Crie uma senha forte"
- Confirmar Senha *: "Confirme sua senha"
- Palavra-Chave de Recuperação *: "Uma palavra única para recuperar sua conta". Below it, a note says: "Guarde bem esta palavra. Ela será essencial caso você perca sua senha."
- A small note at the bottom left of the form area states: "LOPD: Dados criptografados AES-256 e usados apenas para gestão do legado."
- A checkbox labeled "Aceito os termos e política de privacidade".
- Two buttons at the bottom: "Criar Conta" (green) and "Já tenho conta" (purple).

Screenshot 2: Confirmation Message (localhost:8080/EspolioDigital/1%20-%20usuario/usuario.jsp)

This screenshot shows a confirmation message box titled "Cadastro Realizado!" with the message "Sua conta foi criada com sucesso." and a button "Ir para o Login".

Figura 1.2: Recuperação de Senha por Palavra-Chave de Recuperação

The figure consists of two screenshots of a web application interface.

Screenshot 1: Recuperar Senha (Reset Password) Page

This page is titled "Recuperar Senha". It contains the following fields:

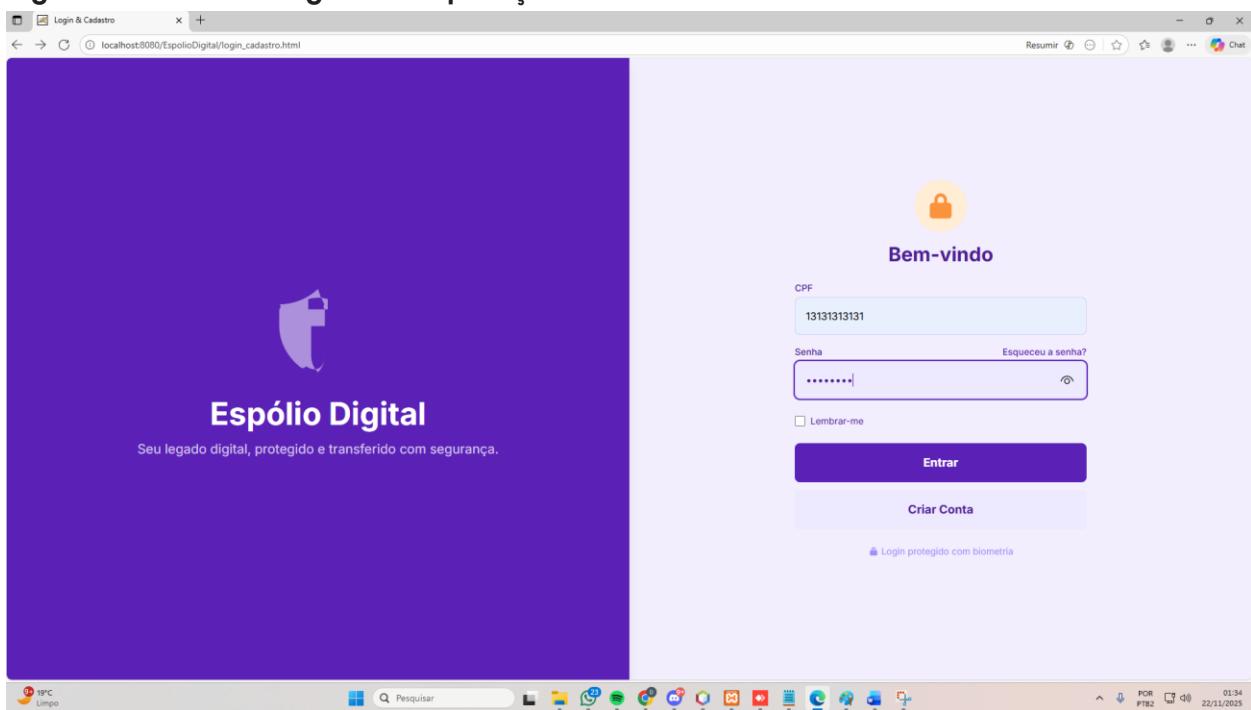
- CPF *: Input field containing "13131313131"
- Palavra-Chave de Recuperação *: Input field containing "lua"
- Nova Senha *: Input field showing "*****"
- Confirmar Nova Senha *: Input field showing "*****" with an eye icon to its right

At the bottom are two buttons: a green "Redefinir Senha" (Reset Password) button and a grey "Cancelar" (Cancel) button.

Screenshot 2: Senha Redefinida! (Password Reset!) Message

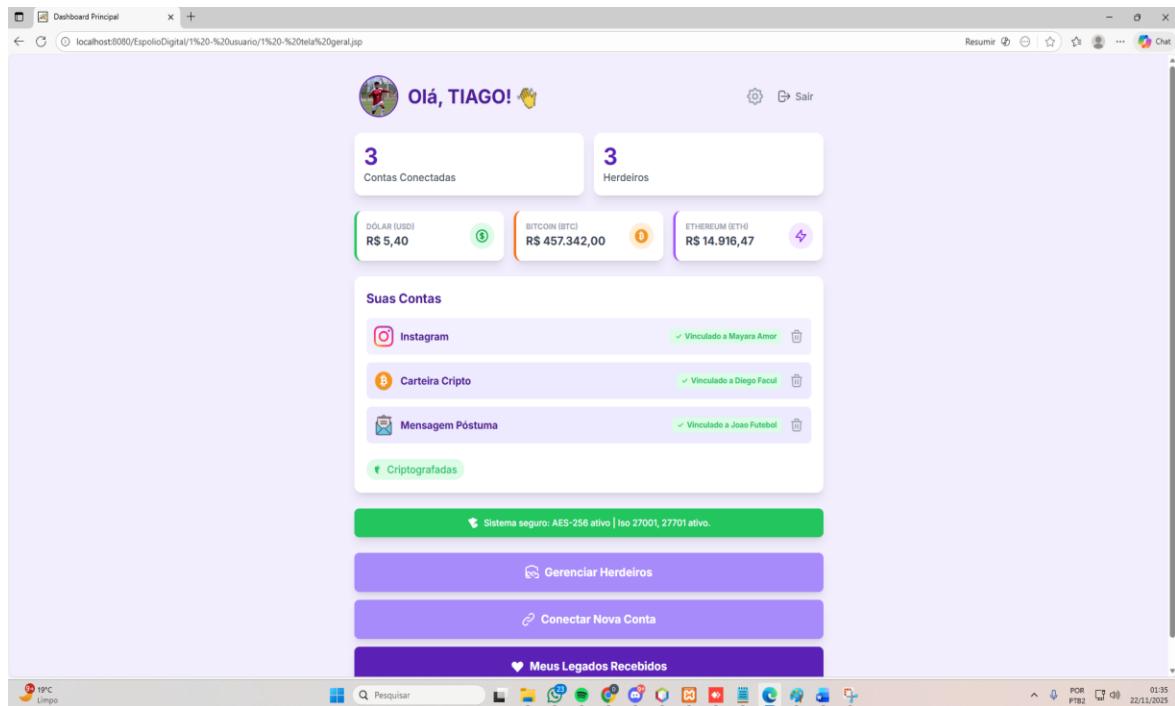
This page displays a success message: "Senha Redefinida!" (Password Reset!) and "Sua senha foi alterada com sucesso." (Your password was successfully changed). A purple "Ir para o Login" (Go to Login) button is present.

Figura 1.3: Tela de Login / Recuperação de Senha do Usuário / Criar Conta



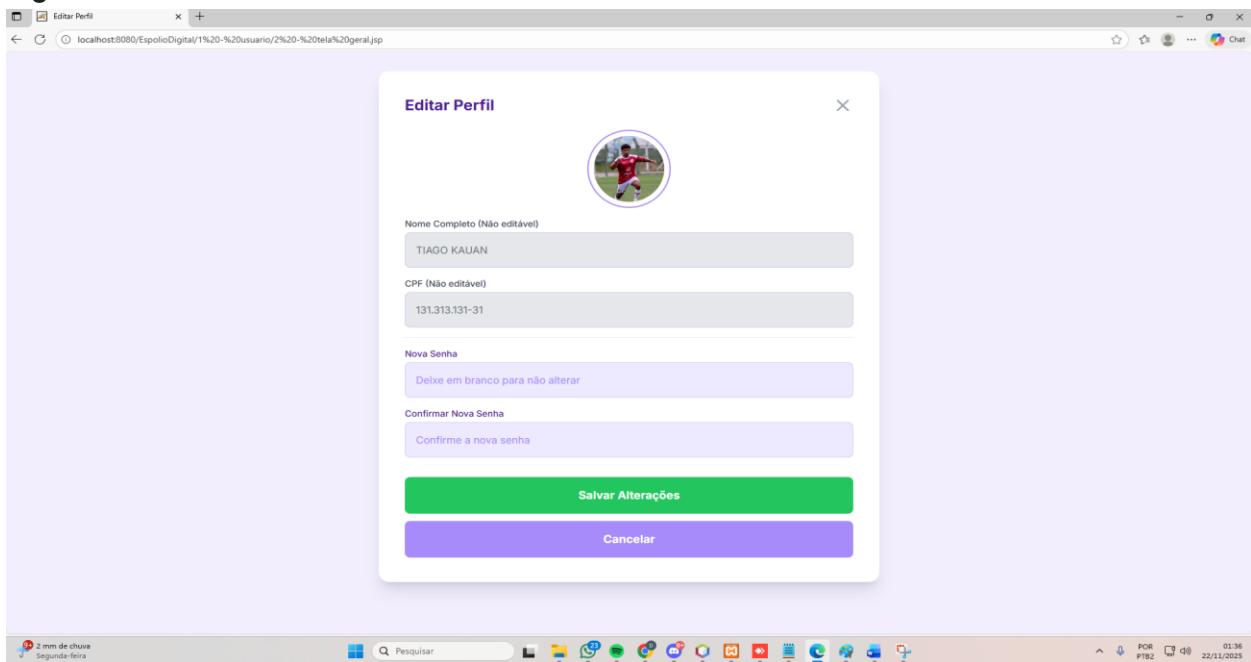
Descrição: Tela inicial de autenticação do usuário.

Figura 2.1: Dashboard Principal



Descrição: Painel de controle do usuário titular.

Figura 2.2: Editar Perfil



Descrição: Painel de controle de edição do perfil do usuário titular.

Figura 3.1: Gerenciar Meus herdeiros

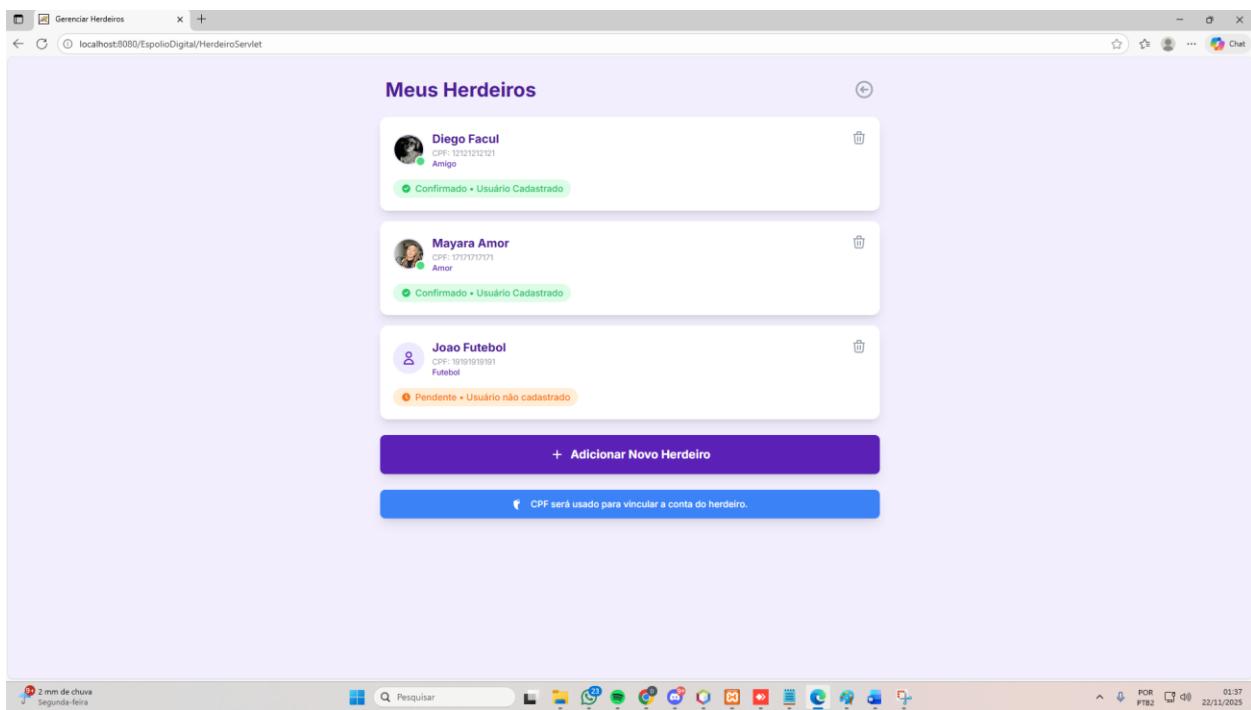


Figura 3.2: Gerenciar Meus herdeiros

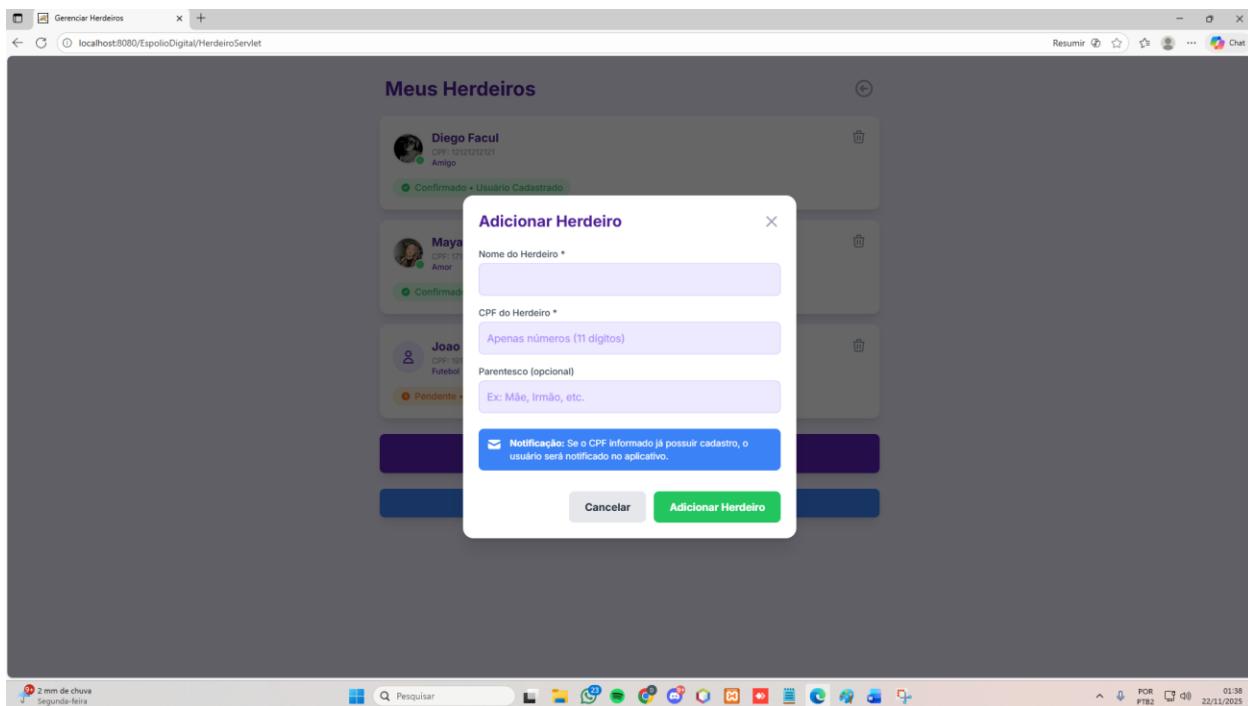
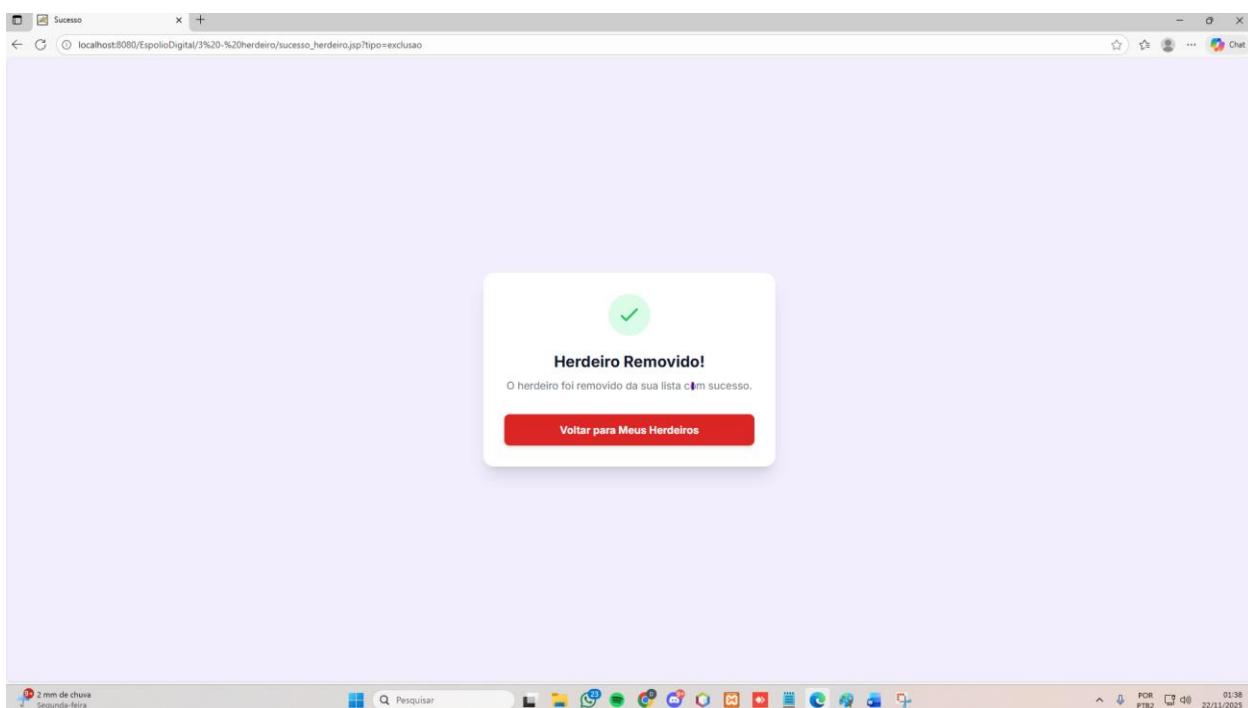


Figura 3.3: Gerenciar Meus herdeiros



Descrição: Interface onde usuário cadasra seus herdeiros.

Figura 4.1: Conectar Contas

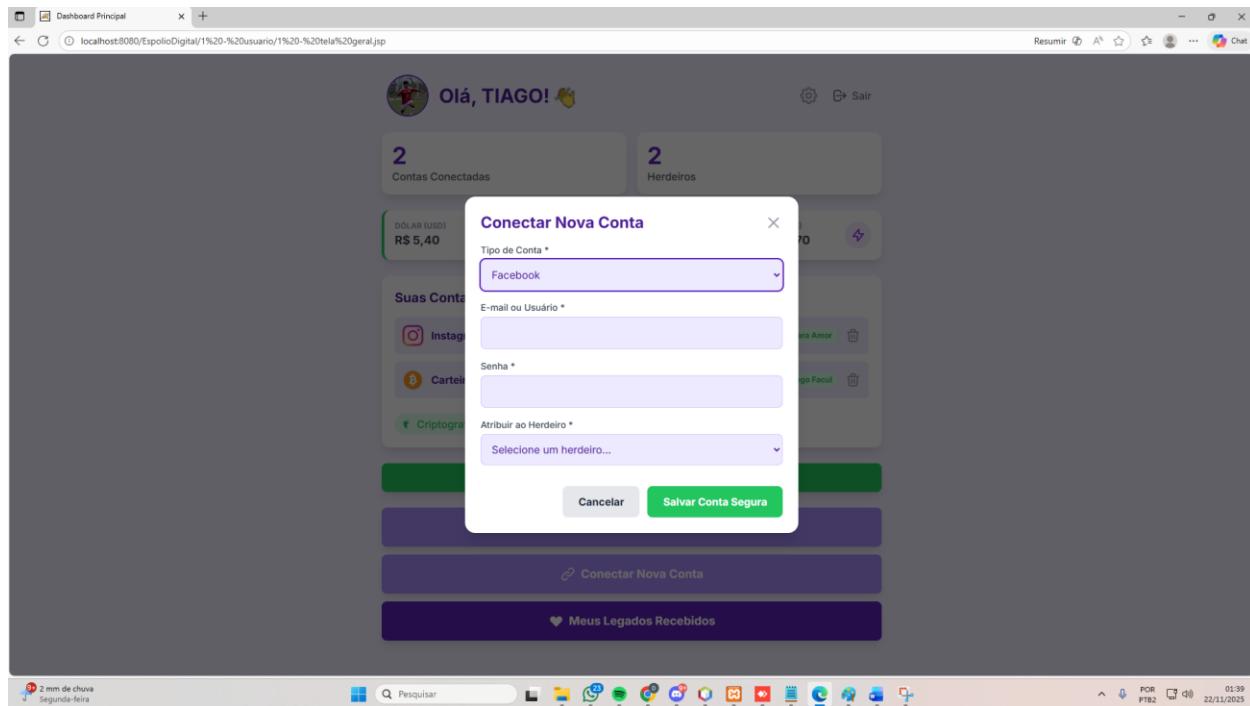


Figura 4.2: Conectar Contas

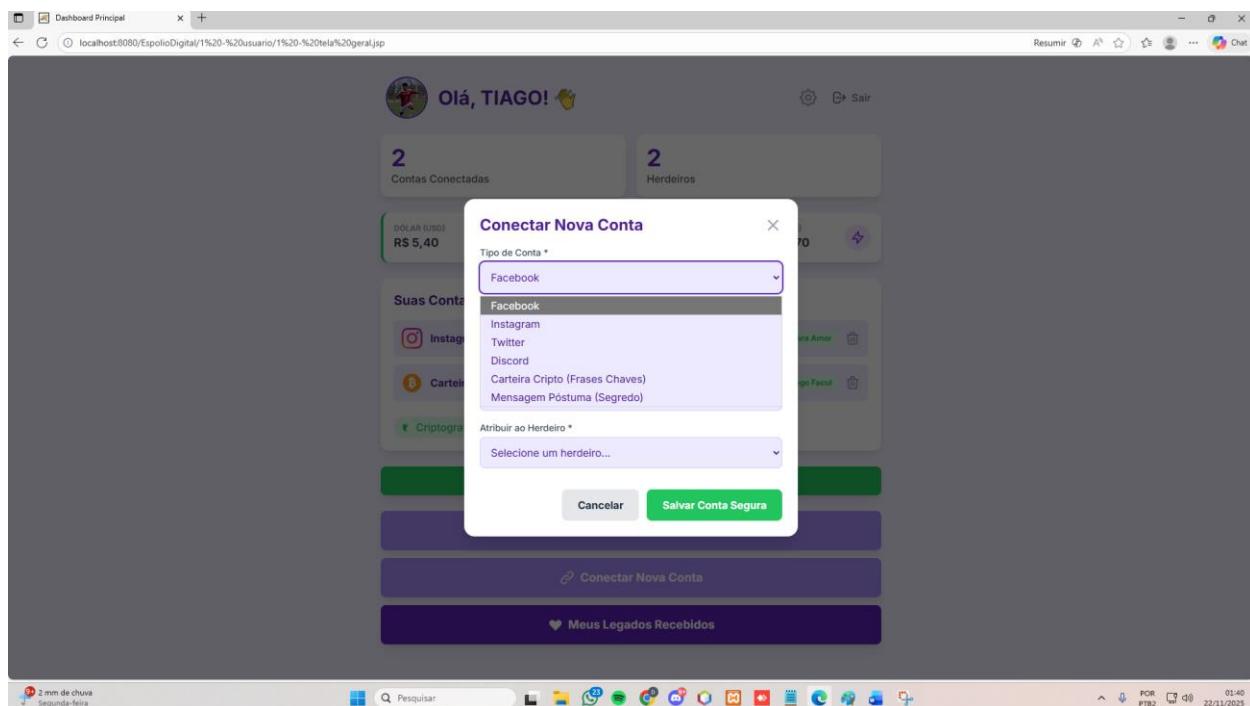
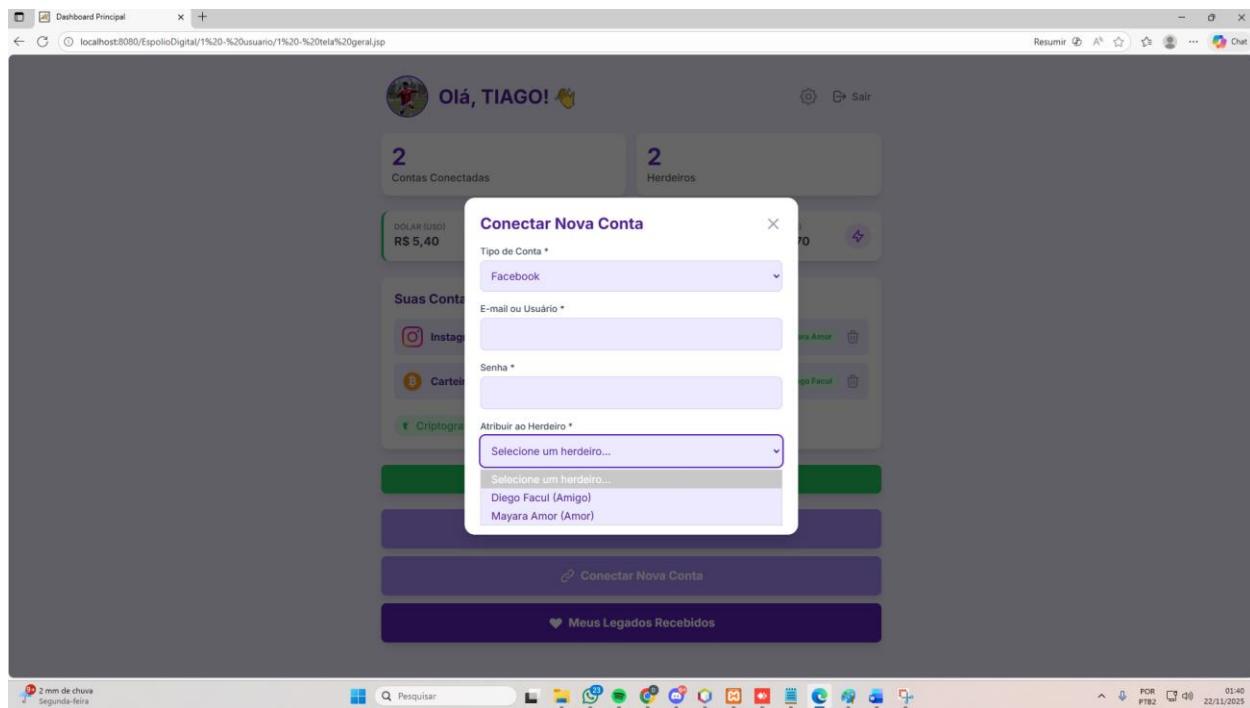


Figura 4.3: Conectar Contas



Descrição: Tela onde o usuário conecta seus ativos e transfere para um usuário herdeiro.

Figura 5.1: Meus Legados Recebidos

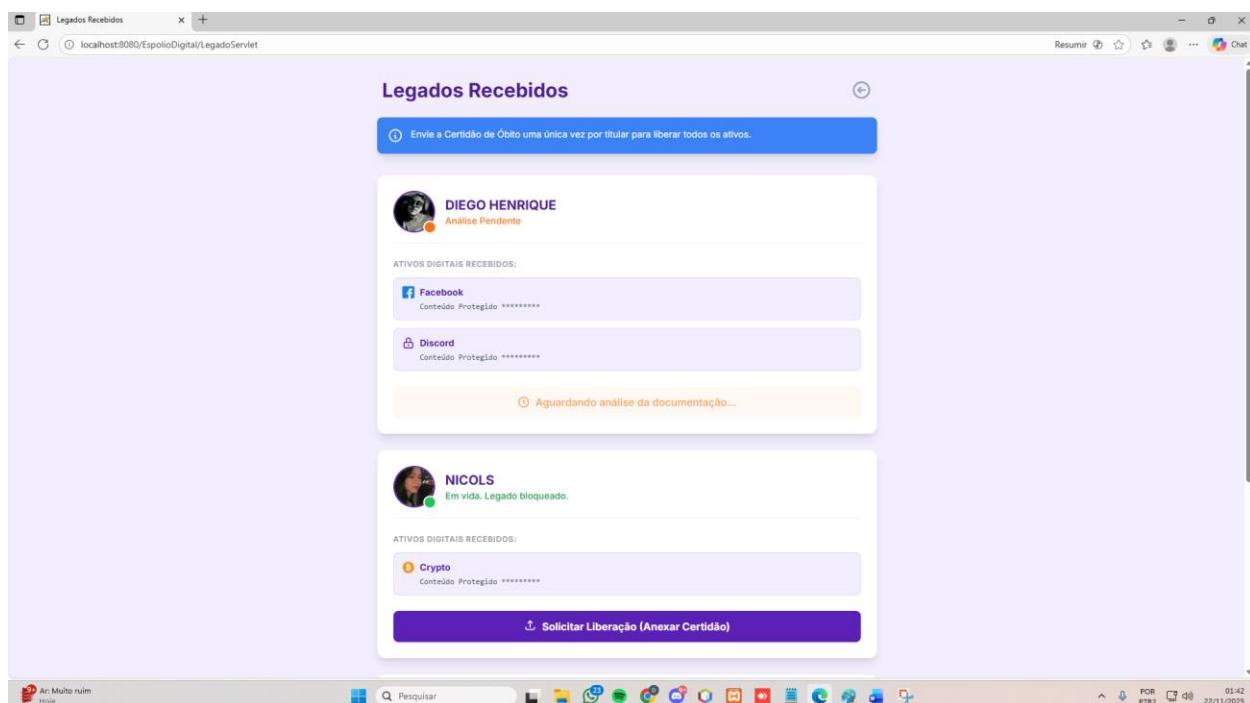


Figura 5.2: Meus Legados Recebidos

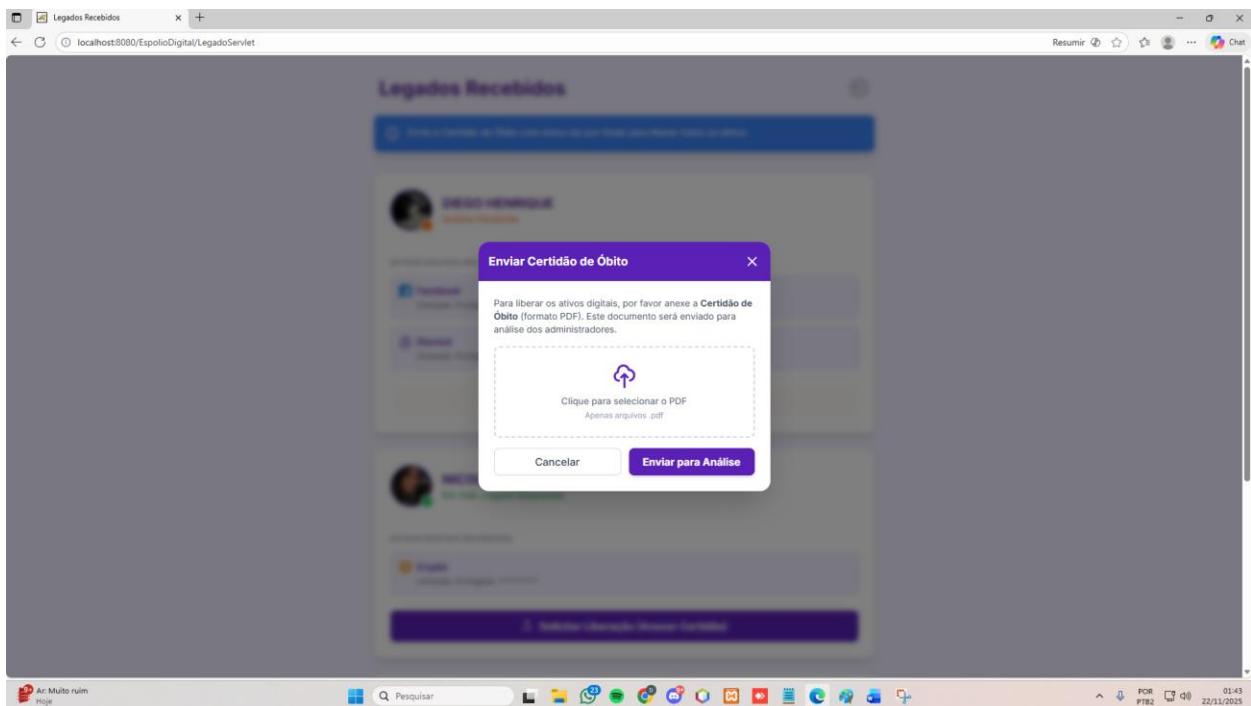
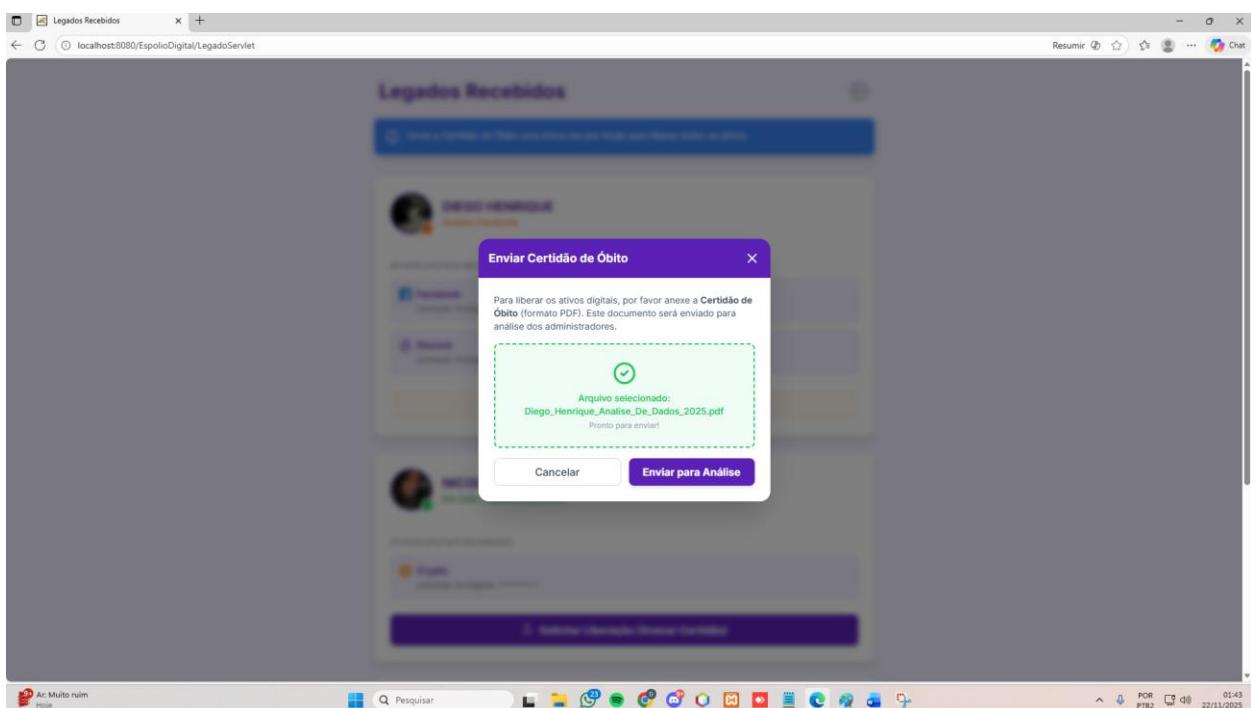


Figura 5.3: Meus Legados Recebidos



Descrição: Feedback visual após o usuário selecionar o arquivo PDF.

7. BIBLIOTECAS E AMBIENTE DE DESENVOLVIMENTO

Para o desenvolvimento e execução deste projeto, foram utilizadas as seguintes ferramentas e bibliotecas integradas ao NetBeans:

- Ambiente de Desenvolvimento (IDE): NetBeans (versão 12 ou superior).
- Linguagem de Programação: Java (JDK 17).
- Servidor de Aplicação: Apache Tomcat 9.0 (Container para Servlets e JSP).
- Banco de Dados: MySQL Server 8.0.
- Bibliotecas (JARs) adicionadas ao projeto:
 - `mysql-connector-java-8.x.x.jar`: Driver JDBC essencial para estabelecer a conexão entre a aplicação Java e o banco de dados MySQL.
 - `Java EE Web API` (Jakarta EE): Biblioteca padrão para desenvolvimento web (inclusa no Tomcat), fornecendo as classes `javax.servlet` e `javax.servlet.http` utilizadas nos Controladores.
- Front-end (Dependências Externas):
 - Tailwind CSS: Framework CSS utilitário carregado via CDN para estilização responsiva.

8. REFERÊNCIAS BIBLIOGRÁFICAS

1. **DEITEL, Paul; DEITEL, Harvey.** *Java: Como Programar*. 10. ed. São Paulo: Pearson, 2016.
2. **ELMASRI, Ramez; NAVATHE, Shamkant.** *Sistemas de Banco de Dados*. 7. ed. São Paulo: Pearson, 2018.
3. **ORACLE.** *Java Servlet Specification*. Disponível em: <https://docs.oracle.com/javaee/7/tutorial/servlets.htm>. Acesso em: 22 nov. 2025.
4. **TAILWINDCSS.** *Tailwind CSS Documentation*. Disponível em: <https://tailwindcss.com/docs>. Acesso em: 22 nov. 2025.