

# Sistema de Moeda Estudantil

Desenvolvimento Web — Release 1

## **Autores:**

- Diogo Henrique Moreira da Silva
- Gustavo Ignácio Moreira da Silva

# Arquitetura Escolhida

O sistema foi desenvolvido utilizando a arquitetura em camadas (MVC – Model, View, Controller), amplamente adotada em aplicações web modernas.

Essa arquitetura separa as responsabilidades em três camadas principais:

### **Model**

Representa os dados e as regras de negócio (Entidades como Aluno, Empresa Parceira, Professor).

### **Controller**

Processa as requisições e faz a ponte entre o front-end e o back-end.

### **View**

Responsável pela interface visual e interação com o usuário.

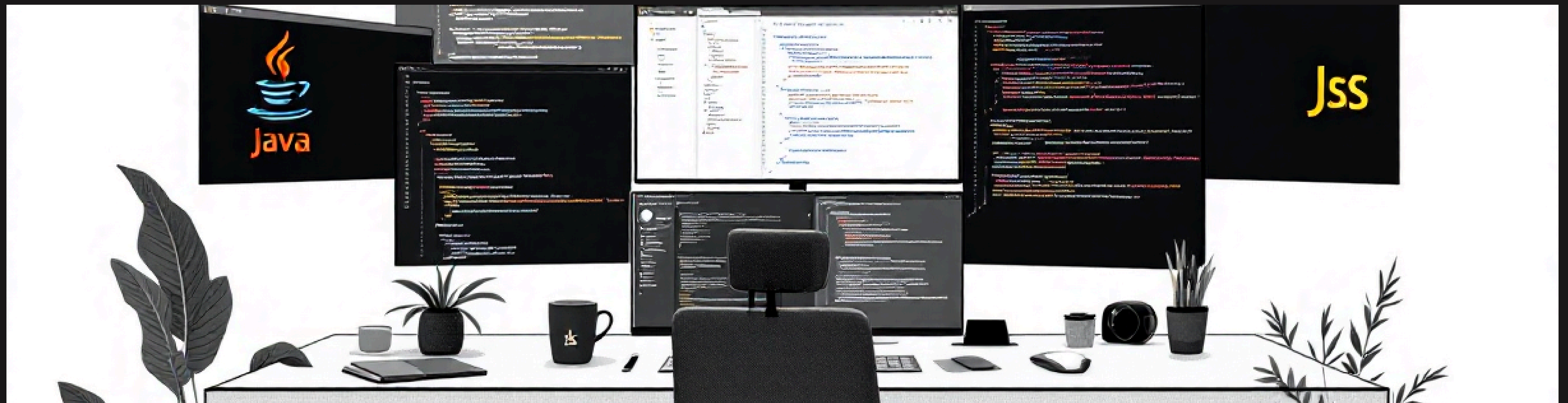
# Tecnologias Utilizadas

## Front-end:

- HTML5, CSS3 e JavaScript: para construir uma interface responsiva, leve e de fácil manutenção.
- Fetch API: usada para conectar o front-end ao back-end via requisições HTTP.

## Back-end:

- Java 17 + Spring Boot 3: fornece robustez, segurança e agilidade no desenvolvimento de APIs REST.
- Maven: gerencia dependências e facilita o build e execução do projeto.





# Banco de Dados escolhido

O sistema utiliza o PostgreSQL como sistema gerenciador de banco de dados.

- É um banco de dados relacional open-source altamente confiável e seguro.
- Suporta integração nativa com o Spring Boot via JPA/Hibernate, permitindo mapear entidades automaticamente.
- Oferece excelente performance e escalabilidade, ideal para aplicações acadêmicas e empresariais.



# Implementação e Funcionalidades

## CRUD Completo

**CRUD de Aluno e Empresa Parceira**, com operações de cadastro, login e listagem.

## API REST

Comunicação entre Front e Back-end via API REST.

## Validação de Dados

Validação de dados no back-end e mensagens de retorno no front-end.

## Persistência

Persistência de dados no banco PostgreSQL através do ORM JPA/Hibernate.

# Evoluções Planejadas



## Módulo de Professores

- Adicionar o módulo de Professores e sistema de aprovação de moedas.



## Autenticação JWT

- Implementar autenticação com JWT para maior segurança.



## Painel Administrativo

- Desenvolver o painel administrativo para gestão de usuários e transações.



## Relatórios

- Criar relatórios e dashboards para acompanhamento de desempenho e saldo estudantil.

# OBRIGADO!

