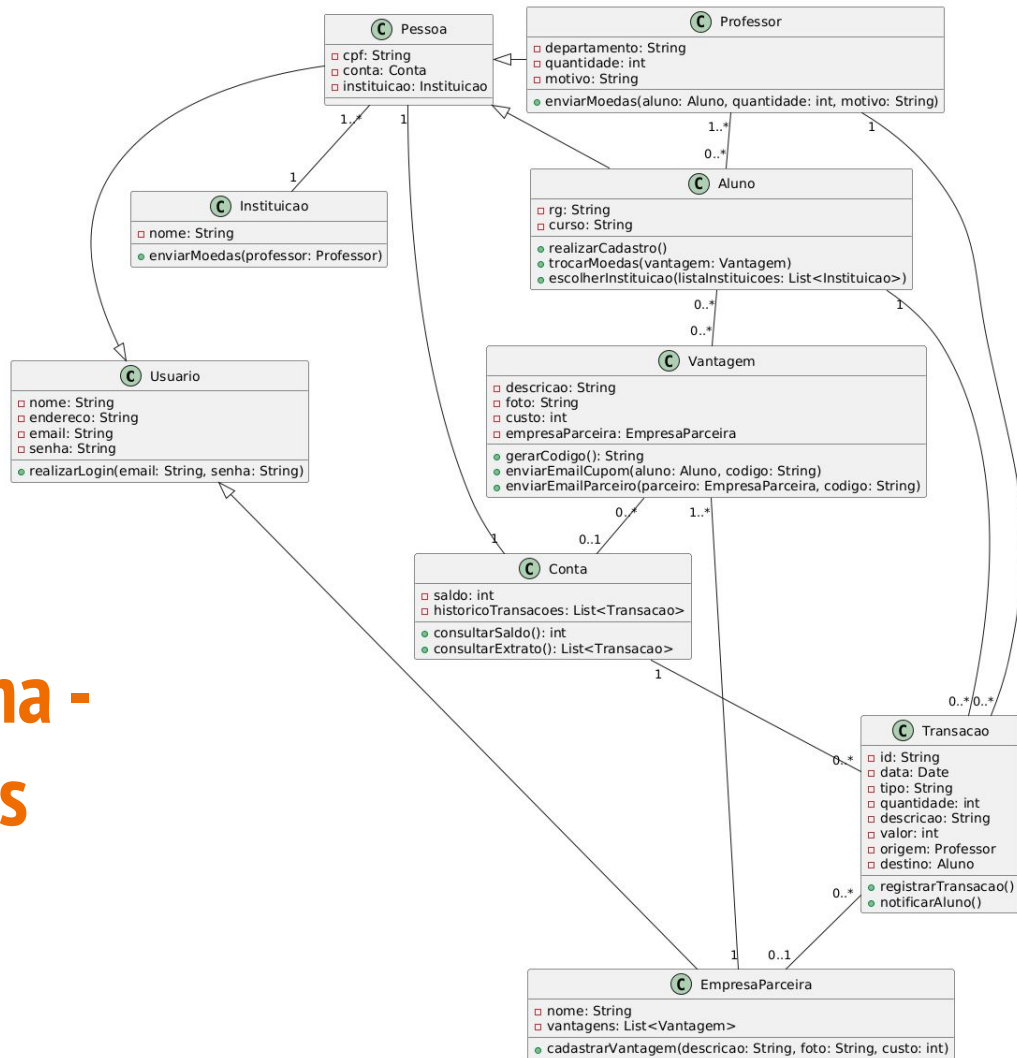

Sistema de Moeda Estudantil

— Maisa, Maria Eduarda e Miguel —

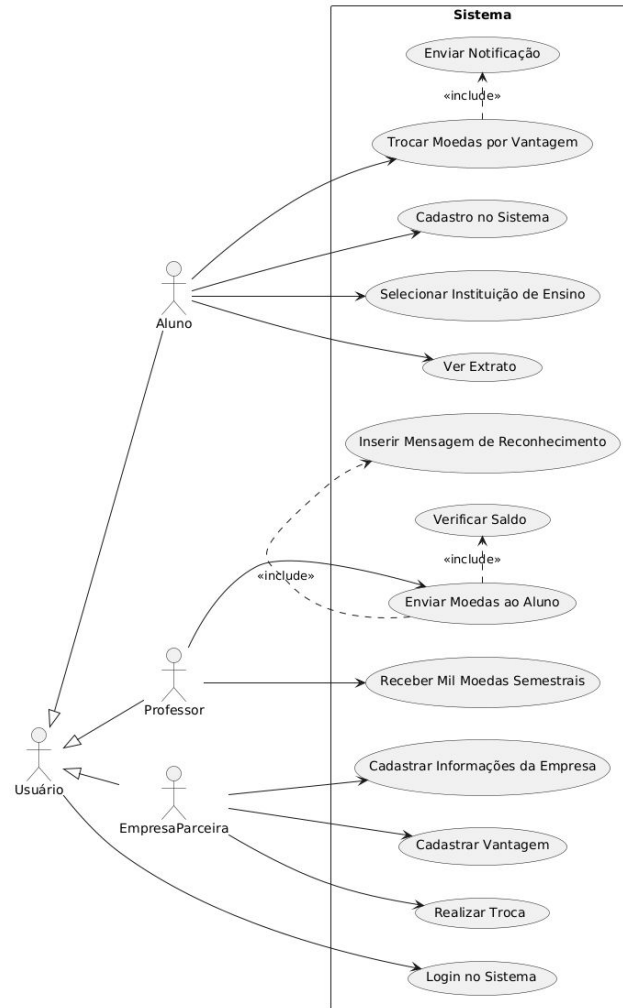
Visão Geral do Sistema

- ❑ É um sistema de gerenciamento de créditos estudantis
- ❑ Permite que estudantes administrem e usem moedas virtuais
- ❑ É uma aplicação voltada para atividades escolares
- ❑ O objetivo principal é auxiliar no controle e na distribuição de créditos

Visão Geral do Sistema - diagrama de classes



Visão Geral do Sistema - diagrama de caso de uso



Principais Ferramentas e Frameworks

- ❑ **Spring Boot:** framework para criação de aplicações web
- ❑ **Spring Security:** autentica e controla o acesso seguro
- ❑ **Spring Data JPA:** facilita a persistência de dados com o banco de dados

Autenticação e Segurança

- ❑ **Spring Security;** configurado para autenticação de usuários
- ❑ **JWT Tokens:** para controle de sessão e autenticação
- ❑ Controle de acesso implementado em endpoints

Fluxo de Dados

- ❑ O fluxo começa na interface do usuário
- ❑ Os dados são enviados ao backend
 - ❑ para validar e processar as transações
- ❑ Os dados são armazenados no banco de dados
 - ❑ para garantir o registro completo de cada operação
 - ❑ para facilita a recuperação e a consulta dos mesmos

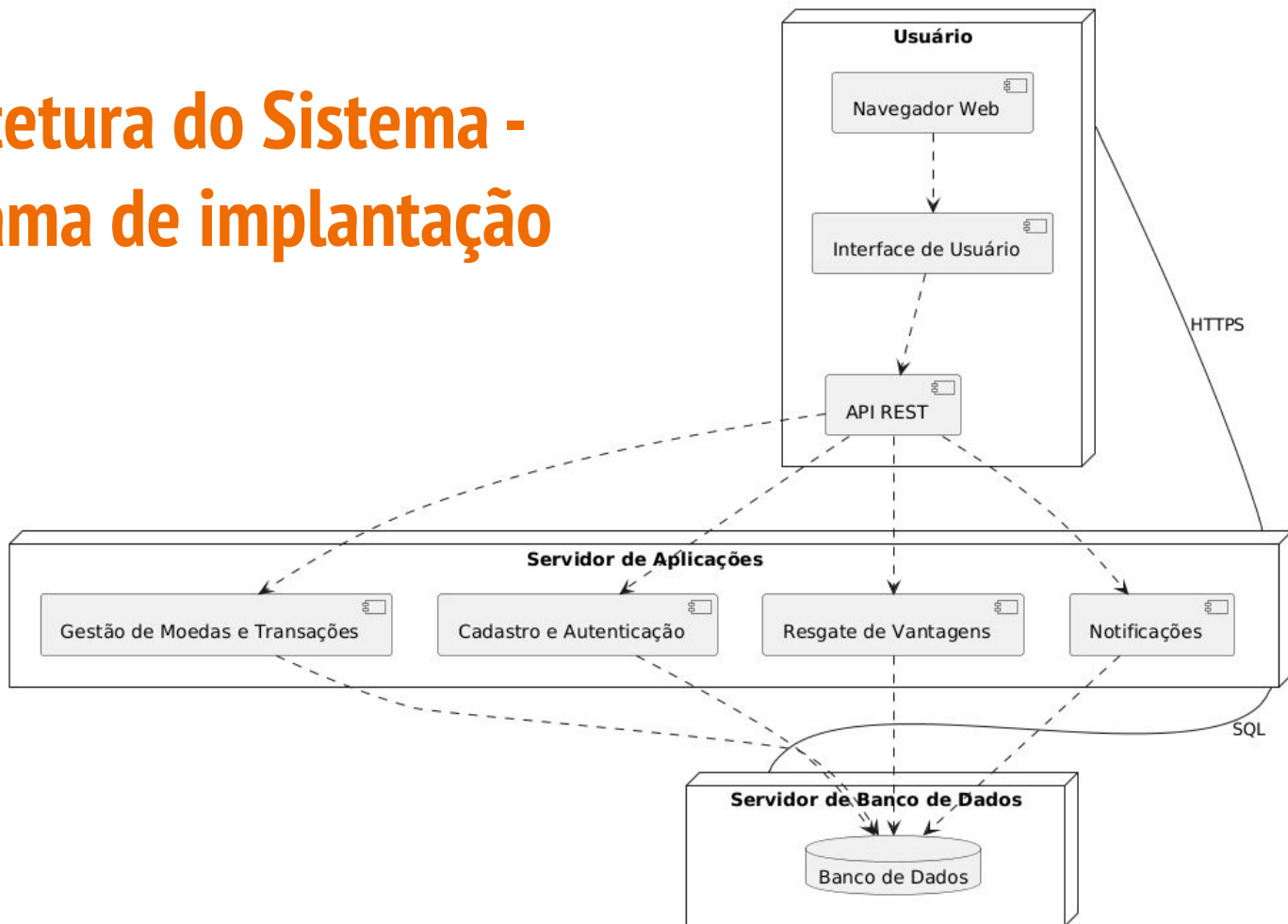
Arquitetura do Sistema

- ❑ A arquitetura do sistema é dividida em camadas:
 - ❑ interface do usuário
 - ❑ backend
 - ❑ banco de dados
- ❑ Cada camada possui responsabilidades específicas
- ❑ A comunicação entre camadas é realizada por meio de interfaces bem definidas
- ❑ **Camada de Controladores:** faz a interação com as rotas da API
- ❑ **Camada de Persistência:** faz o gerenciamento de dados com JPA e Hibernate

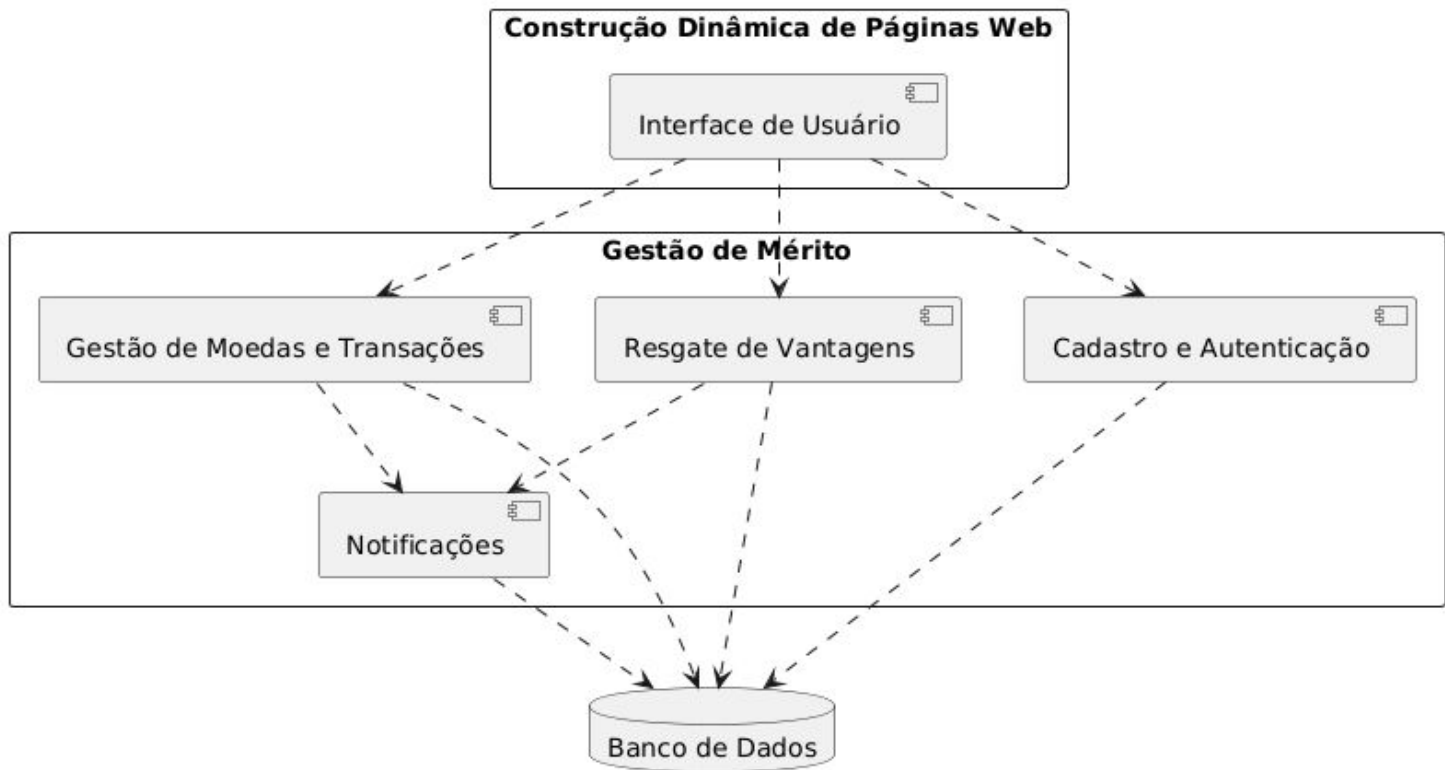
Arquitetura do Sistema - Padrão MVC

- ❑ **Model:** representa a lógica de dados e operações do sistema
- ❑ **View:** interface de interação com o usuário
- ❑ **Controller:** define os endpoints para processar e retornar as requisições

Arquitetura do Sistema - diagrama de implantação



Arquitetura do Sistema - diagrama de componentes



Frontend

- ❑ **HTML:** é a linguagem padrão para estruturar conteúdo na web, com o auxílio de tags para definir diferentes elementos de uma página e, assim, informar ao navegador como renderizar o conteúdo da página
- ❑ **CSS:** é uma linguagem de estilo usada para controlar a aparência de elementos HTML por meio de seletores
- ❑ **JavaScript :** é uma linguagem de programação usada para adicionar interatividade a páginas web

Camada de Persistência

- ❑ Ela é composta por um banco de dados relacional, inclui classes Java para operações CRUD e possui classes DAO que implementam o acesso aos dados das tabelas principais
- ❑ É a camada responsável pela interação com o banco de dados por meio do uso de repositórios
 - ❑ **Repositórios:** interfaces para manipulação de entidades
 - ❑ **Entidades:** classes que representam as tabelas
 - ❑ **JPA e Hibernate:** utilizados para o mapeamento objeto-relacional

DAO e EntityManager

- ❑ Juntos, eles foram responsáveis pela interação direta com o banco de dados para gerenciar entidades e trabalharam juntos para separar a lógica de persistência da lógica de negócio
 - ❑ **DAO:** padrão de projeto para encapsular operações de acesso a dados
 - ❑ **EntityManager:** gerencia a persistência e busca de entidades do banco

DAO

- ❑ é uma estrutura organizacional operada por regras de programação em blockchain
- ❑ as classes DAO têm o objetivo de lidar com a comunicação direta com a base de dados e são responsáveis por operações CRUD para os objetos específicos
- ❑ no projeto, as classes DAO encapsulam as interações com a base de dados, promovendo uma separação clara entre a lógica de negócios e o acesso aos dados

EntityManager

- ❑ é um serviço responsável por gerenciar as entidades
- ❑ ele provê APIs para criar consultas, buscando objetos, sincronizando objetos, e inserindo objetos no banco de dados
- ❑ no projeto, ele é utilizado nas classes DAO para gerenciar as operações de persistência e a recuperação de dados no banco

Diagrama Entidade Relacionamento

Pessoa (id_pessoa, cpf, nome, endereco, email, senha, conta_id, instituicao_id)

Aluno (id_aluno, rg, curso, id_pessoa)

Professor (id_professor, departamento, quantidade, motivo, id_pessoa)

Instituicao (id_instituicao, nome)

Empresa Parceira (id_empresa, nome)

Vantagem (id_vantagem, descricao, foto, custo, empresa_id)

Transacao (id_transacao, data, tipo, quantidade, descricao, valor, origem_id, destino_id, empresa_id)

Conta (id_conta, saldo)

Banco de Dados - MySQL

- ❑ é um sistema de gerenciamento de banco de dados relacional, que armazena dados em tabelas relacionais
- ❑ ele permite realizar o CRUD de registros em tabelas
- ❑ armazena informações de forma persistente no sistema
- ❑ se integra com JPA e Hibernate
- ❑ a conexão é gerenciada pelo Spring Data JPA e configurada no arquivo de propriedades
- ❑ as chaves estrangeiras garantem a integridade referencial entre as tabelas
- ❑ no projeto, o arquivo transactions.sql define uma tabela chamada transactions, que é usada para registrar transações feitas pelos usuários no sistema

Obrigado
