

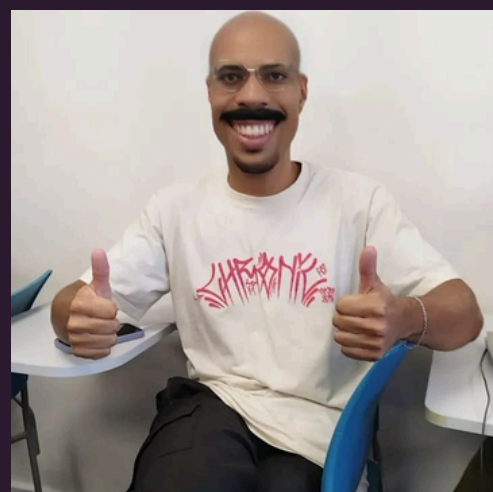
search

Laboratório de Desenvolvimento de sistemas

Arquitetura do Sistema e Camada de Persistência



QUEM ESTÁ APRESENTANDO?



Eduardo Henrique



Gabriel Lage

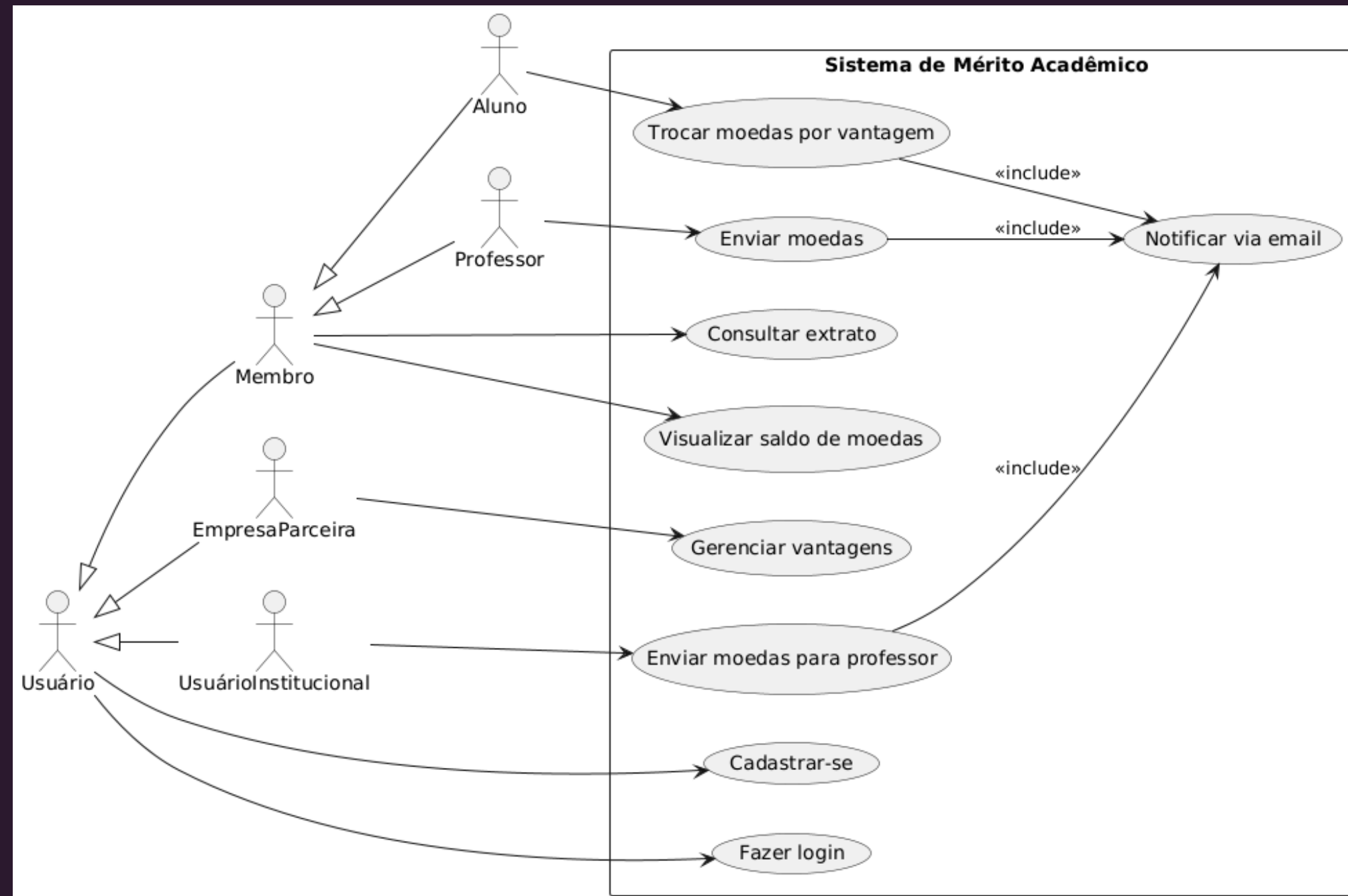


Roberta Sophia

Modelagem do projeto

- Caso de uso
- História de usuário
- Diagrama de classes
- Diagrama de componentes
- Diagrama de implantação
- Diagrama lógico

Diagrama de caso de uso



História de usuário

Usuário Geral

- Como usuário, quero me cadastrar no sistema para acessar as funcionalidades disponíveis para meu perfil.
- Como usuário, quero fazer login de forma simples para acessar o sistema de maneira segura.

Usuário Institucional

- Como usuário institucional, quero enviar moedas para professores para distribuir recursos que serão utilizados na premiação dos alunos.
- Como usuário institucional, quero que os professores sejam notificados por email ao receberem moedas para garantir que saibam que receberam novos recursos.

História de usuário

Aluno

- Como aluno, quero trocar minhas moedas por vantagens para poder aproveitar benefícios oferecidos pelas empresas parceiras.
- Como aluno, quero consultar meu extrato de transações para acompanhar o histórico de moedas recebidas e utilizadas.
- Como aluno, quero visualizar meu saldo atual de moedas para saber quantas moedas tenho disponíveis para trocar.
- Como aluno, quero receber notificações por email quando resgatar uma vantagem para ter confirmação da minha transação.

Empresa Parceira

- Como empresa parceira, quero gerenciar as vantagens oferecidas para atrair alunos e promover meus produtos/serviços.
- Como empresa parceira, quero cadastrar novas vantagens com diferentes valores de moedas para criar opções variadas para os alunos.

História de usuário

Professor

- Como professor, quero enviar moedas para os alunos para reconhecer seu bom desempenho e participação.
- Como professor, quero que os alunos sejam notificados por email quando receberem moedas para garantir que estejam cientes da premiação.
- Como professor, quero visualizar meu saldo de moedas disponíveis para distribuição para gerenciar como recompenso meus alunos.
- Como professor, quero consultar meu extrato de transações para que eu possa acompanhar o histórico de moedas enviadas aos alunos e gerenciar melhor minhas distribuições.

Diagrama de classes

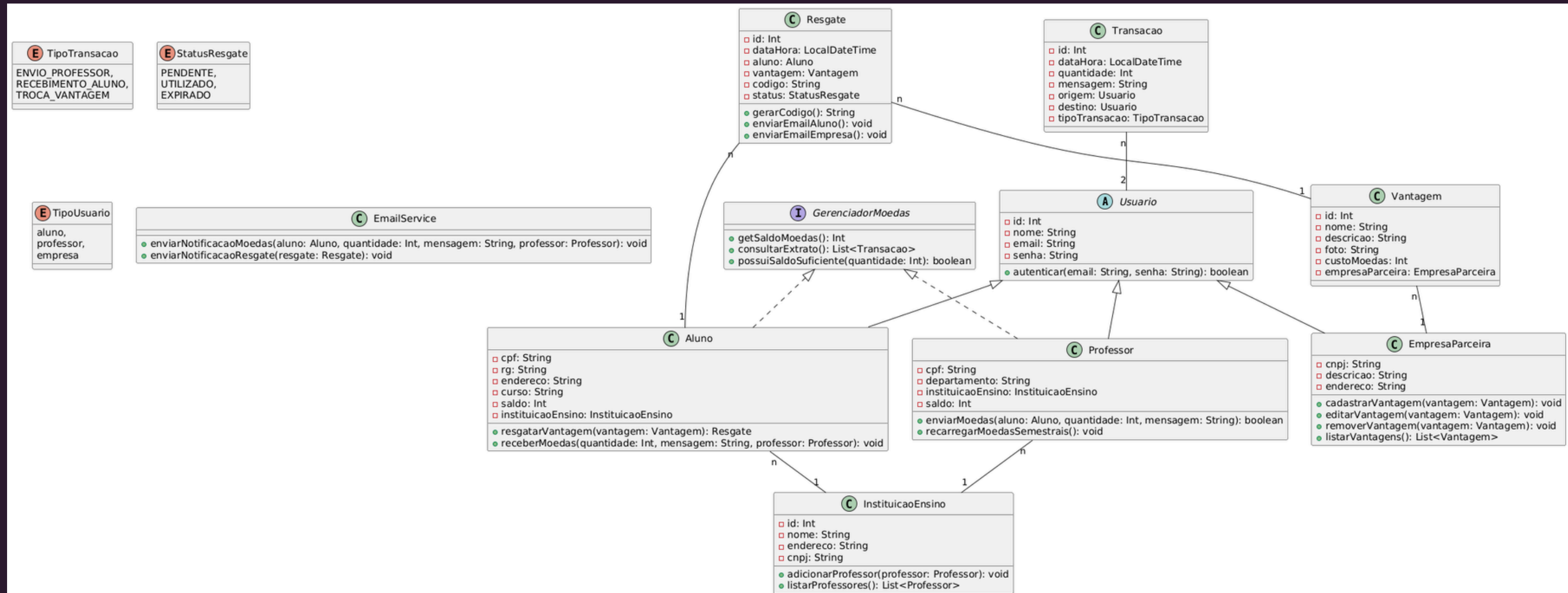


Diagrama de componentes

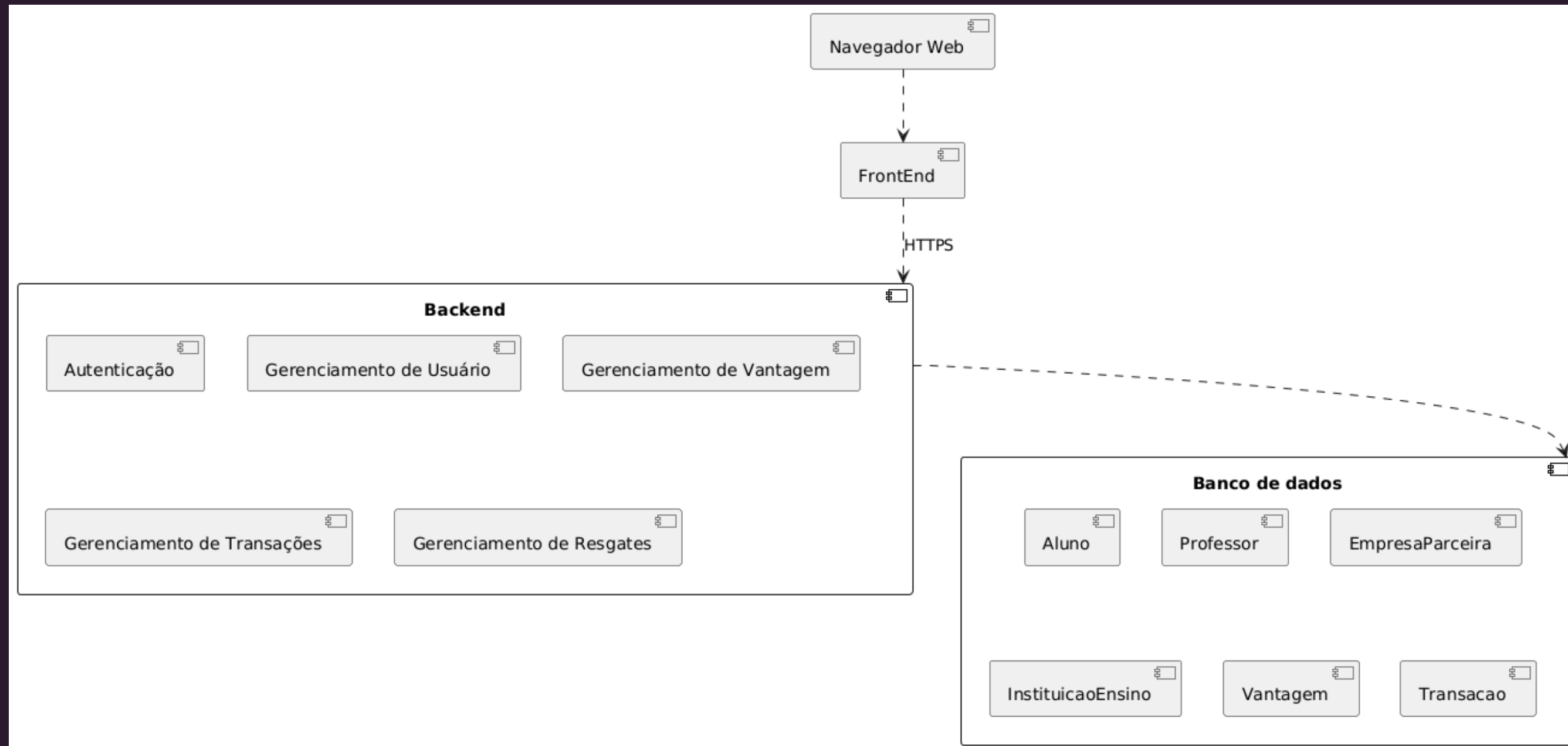


Diagrama de implantação

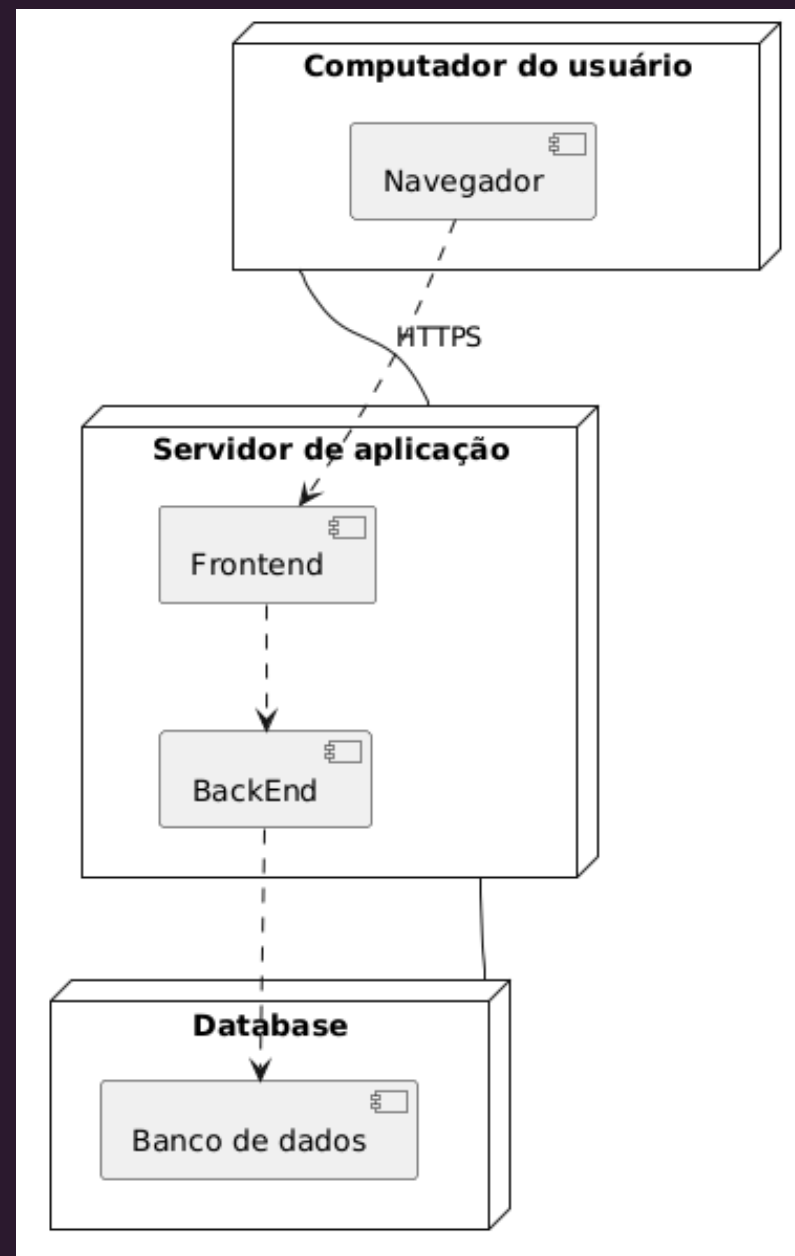


Diagrama lógico

- Usuario(id, nome, email, senha)
- Aluno(usuario_id [FK → Usuario], cpf, rg, endereco, curso, saldoMoedas, instituicaoEnsino_id [FK → InstituicaoEnsino])
- Professor(usuario_id [FK → Usuario], cpf, departamento, instituicaoEnsino_id [FK → InstituicaoEnsino], saldoMoedas)
- EmpresaParceira(usuario_id [FK → Usuario], cnpj, descricao, enderecoFisico)
- InstituicaoEnsino(id, nome, endereco, cnpj)
- Vantagem(id, nome, descricao, foto, custoMoedas, empresaParceira_cnpj [FK → EmpresaParceira])
- Transacao(id, dataHora, quantidade, mensagem, origem_id [FK → Usuario], destino_id [FK → Usuario], tipoTransacao [ENUM])
- Resgate(id, dataHora, aluno_id [FK → Aluno], vantagem_id [FK → Vantagem], codigo, status [ENUM])



Seminário de Tecnologias



Estrutura Backend

Seguindo uma estrutura MVC, com models, views e controller

Models:

Representa os dados e a lógica de persistência

Views:

Arquivos HTML/CSS consumidos diretamente pelo navegador

Controllers:

Intermediário entre a View e o Model, processa requisições, executa regras de negócio, invoca Models e retorna resposta

```
✓ src
  ✓ controllers
  > models
  > routes
  > views
  JS server.js
  {} package-lock.json
  {} package.json
```

Tecnologias



+



backend



.EJS

renderização
HTML



criptografia



banco de dados

Estratégias de acesso

Estratégia utilizada: Acesso Direto

- Utiliza comandos SQL puros
- Máximo controle sobre as consultas
- Boa performance e baixo overhead

Vantagens

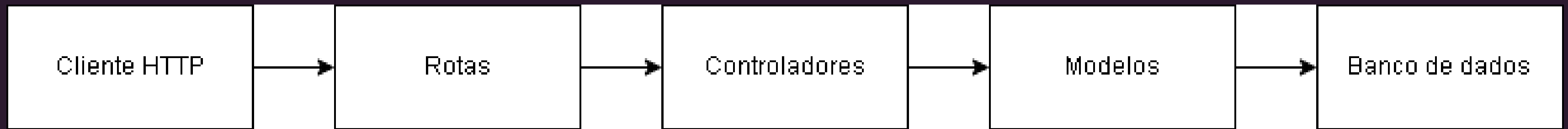
- Total controle sobre as queries
- Mais próximo do banco de dados

Considerações

- É necessário lidar com SQL manualmente
- Requer mais cuidado com injeção de SQL



Fluxo de Persistência



Rotas → Receber requisições HTTP e redirecionar para o controller

Controladores → Contém regras de negócio e validações antes de acessar dados

Modelos → Interage diretamente com o banco de dados via queries SQL

Banco de dados → Armazena os dados persistentemente

VALEU!

