
Documentação de Projeto

para o sistema

<IngressoFácil>

Versão 1.0

Projeto de sistema elaborado pelos alunos: João Gabriel Maia e Lucas Alves Resende
como parte da disciplina **Projeto de Software**.

<26/04/2025>

Tabela de Conteúdo

| | |
|--|----------|
| 1. Introdução | 1 |
| 2. Modelos de Usuário e Requisitos | 1 |
| 2.1 Descrição de Atores | 1 |
| 2.2 Modelo de Casos de Uso e Histórias de Usuários | 1 |
| 2.3 Diagrama de Sequência do Sistema e Contrato de Operações | 1 |
| 3. Modelos de Projeto | 1 |
| 3.1 Arquitetura | 1 |
| 3.2 Diagrama de Componentes e Implantação. | 2 |
| 3.3 Diagrama de Classes | 2 |
| 3.4 Diagramas de Sequência | 2 |
| 3.5 Diagramas de Comunicação | 2 |
| 3.6 Diagramas de Estados | 2 |
| 4. Modelos de Dados | 2 |

1. Introdução

Este documento apresenta a documentação do projeto IngressoFácil, um sistema voltado para a venda online de ingressos de cinema e eventos. O objetivo do sistema é facilitar o processo de compra de ingressos por parte dos usuários, permitindo que escolham sessões, assentos e realizem suas compras de forma rápida, segura e intuitiva.

A aplicação será composta por duas interfaces principais: uma para os clientes, onde poderão visualizar os filmes ou eventos disponíveis e comprar seus ingressos; e outra para o administrador, responsável por cadastrar os filmes/eventos, as sessões e monitorar as vendas.

Este projeto foi elaborado como parte da disciplina Projeto de Software, com o propósito de aplicar os conceitos de modelagem de requisitos, modelagem de projeto, arquitetura de software e banco de dados. O sistema servirá como exemplo prático de um software com múltiplos atores, fluxo de interação e gerenciamento de dados persistentes.

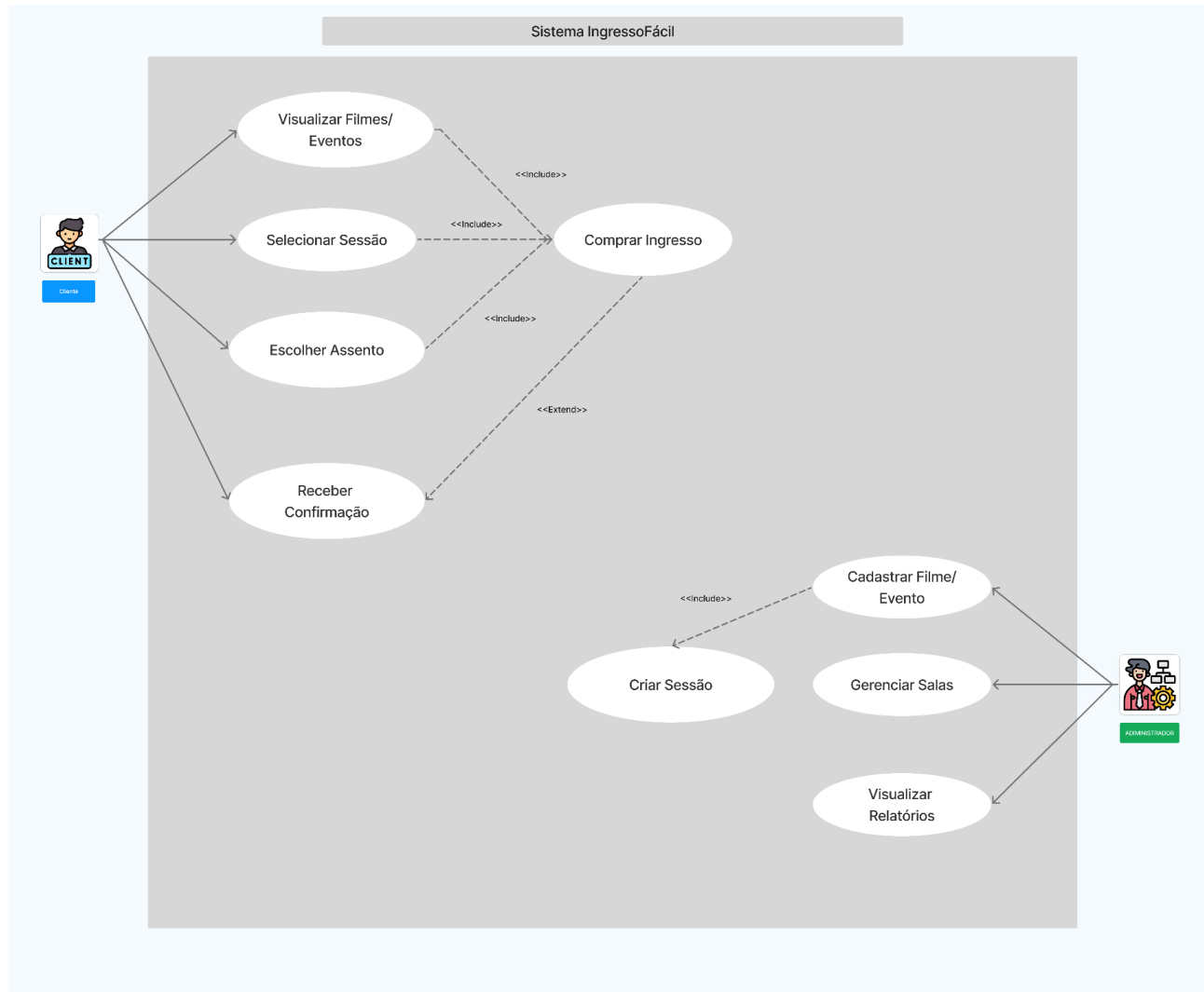
2. Modelos de Usuário e Requisitos

2.1 Descrição de Atores

Nesta seção, são apresentados os atores que interagem com o sistema IngressoFácil. Cada ator possui responsabilidades específicas de acordo com o seu papel no uso da aplicação.

| Ator | Descrição |
|---------------|--|
| Cliente | Usuário que acessa o sistema para consultar filmes ou eventos disponíveis, escolher sessões, selecionar assentos e comprar ingressos. Tem acesso à área pública do sistema. |
| Administrador | Responsável pelo cadastro e gerenciamento dos filmes/eventos, sessões e salas. Pode visualizar relatórios de vendas. Tem acesso a uma área restrita do sistema para administração. |

2.2 Modelo de Casos de Uso e Histórias de usuários



Resumo com IDs:

| ID | Nome do Caso de Uso | Atores |
|-------|---------------------------|---------|
| UC-01 | Visualizar Filmes/Eventos | Cliente |
| UC-02 | Selecionar Sessão | Cliente |
| UC-03 | Escolher Assento | Cliente |

| | | |
|-------|------------------------|---------------|
| UC-04 | Comprar Ingresso | Cliente |
| UC-05 | Receber Confirmação | Cliente |
| UC-06 | Cadastrar Filme/Evento | Administrador |
| UC-07 | Criar Sessão | Administrador |
| UC-08 | Gerenciar Salas | Administrador |
| UC-09 | Visualizar Relatórios | Administrador |

As histórias de usuário a seguir descrevem, de forma simples e objetiva, as ações que os atores podem realizar no sistema, com base em suas necessidades e objetivos.

Histórias do Cliente

- HU-01 – Visualizar Filmes/Eventos Disponíveis

Como cliente, quero visualizar os filmes e eventos disponíveis, para escolher aquele que quero assistir.

- HU-02 – Selecionar Sessão

Como cliente, quero escolher a data e o horário da sessão de um filme/evento, para encontrar o melhor horário para mim.

- HU-03 – Escolher Assento

Como cliente, quero escolher meu assento dentro da sala, para garantir um lugar confortável e disponível.

- HU-04 – Comprar Ingresso

Como cliente, quero efetuar a compra do ingresso escolhido, para confirmar minha presença no evento.

- HU-05 – Receber Confirmação da Compra

Como cliente, quero receber uma confirmação com os dados do ingresso, para poder apresentar na entrada do evento.

Histórias do Administrador

- HU-06 – Cadastrar Filmes/Eventos

Como administrador, quero cadastrar novos filmes e eventos no sistema, para que fiquem disponíveis para venda.

- HU-07 – Criar Sessões

Como administrador, quero cadastrar sessões com data, hora e sala, para organizar os horários dos eventos.

- HU-08 – Gerenciar Salas

Como administrador, quero gerenciar as salas e seus assentos, para manter o controle da capacidade de público.

- HU-09 – Visualizar Relatórios de Vendas

Como administrador, quero acessar os relatórios de vendas por sessão e evento, para acompanhar o desempenho e ocupação.

2.3 Diagrama de Sequência do Sistema e Contratos de operação

Nesta subseção é apresentado o diagrama de sequência do sistema de pelo menos, 3 Casos de Uso ou Histórias de Usuário descritos na Seção 2.3.

Contratos de operação

UC-01 – Visualizar Filmes/Eventos

- Contrato: visualizarFilmesEventos
- Operação: listar todos os filmes/eventos disponíveis no sistema
- Referências cruzadas: HU-01
- Pré-condições: O sistema deve conter pelo menos um filme/evento cadastrado
- Pós-condições: A lista de filmes/eventos é exibida para o cliente

UC-02 – Selecionar Sessão

- Contrato: selecionarSessao

- Operação: escolher uma sessão (data e hora) de um filme/evento
- Referências cruzadas: HU-02
- Pré-condições: O cliente deve ter selecionado um filme/evento disponível
- Pós-condições: A sessão escolhida é registrada para continuar a compra

UC-03 – Escolher Assento

- Contrato: escolherAssento
- Operação: selecionar um assento disponível na sala da sessão
- Referências cruzadas: HU-03
- Pré-condições: O cliente deve ter selecionado uma sessão válida
- Pós-condições: O assento escolhido é marcado como reservado temporariamente para o cliente

UC-04 – Comprar Ingresso

- Contrato: comprarIngresso
- Operação: finalizar a compra do ingresso selecionado
- Referências cruzadas: HU-04
- Pré-condições: O cliente deve ter escolhido uma sessão e um assento válido
- Pós-condições: A compra é registrada, o assento é marcado como ocupado e o ingresso é gerado

UC-05 – Receber Confirmação

- Contrato: gerarConfirmacao

- Operação: exibir e/ou enviar a confirmação da compra para o cliente
- Referências cruzadas: HU-05
- Pré-condições: A compra do ingresso deve ter sido concluída com sucesso
- Pós-condições: O cliente recebe um código ou QR code de confirmação

UC-06 – Cadastrar Filme/Evento

- Contrato: cadastrarFilmeEvento
- Operação: registrar um novo filme ou evento no sistema
- Referências cruzadas: HU-06
- Pré-condições: O administrador deve estar autenticado no sistema
- Pós-condições: O filme/evento fica disponível para criação de sessões e para os clientes visualizarem

UC-07 – Criar Sessão

- Contrato: criarSessao
- Operação: cadastrar uma nova sessão com data, hora e sala associada a um evento
- Referências cruzadas: HU-07
- Pré-condições: Deve existir pelo menos um filme/evento cadastrado e uma sala válida
- Pós-condições: A sessão fica disponível para compra de ingressos

UC-08 – Gerenciar Salas

- Contrato: gerenciarSalas
- Operação: cadastrar, editar ou remover salas do sistema
- Referências cruzadas: HU-08

- Pré-condições: O administrador deve estar autenticado
- Pós-condições: As salas são atualizadas no sistema, afetando as sessões futuras

UC-09 – Visualizar Relatórios

- Contrato: visualizarRelatorios
- Operação: acessar dados estatísticos de vendas por sessão/evento
- Referências cruzadas: HU-09
- Pré-condições: O administrador deve estar autenticado
- Pós-condições: Os dados de vendas são exibidos sem afetar o sistema

Diagrama de sequência:

- **Casos escolhidos:**

1. Selecionar Sessão (UC-02)
2. Escolher Assento (UC-03)
3. Comprar Ingresso (UC-04)
4. Cadastrar Filme/Evento (UC-06)

UC-02 – Selecionar Sessão

@startuml

actor Cliente

participant "Interface Web" as UI

participant "Controlador de Sessões" as Controller

participant "Banco de Dados" as BD

Cliente -> UI : acessar filme
UI -> Controller : solicitar sessões do filme
Controller -> BD : consultar sessões disponíveis
BD --> Controller : lista de sessões
Controller --> UI : enviar sessões para exibir
UI --> Cliente : exibir sessões
@enduml

UC-03 – Escolher Assento

@startuml
actor Cliente
participant "Interface Web" as UI
participant "Controlador de Sessões" as Controller
participant "Banco de Dados" as BD

Cliente -> UI : escolher sessão
UI -> Controller : solicitar mapa de assentos
Controller -> BD : buscar assentos da sessão
BD --> Controller : mapa com assentos e status
Controller --> UI : enviar mapa
UI --> Cliente : exibir assentos disponíveis

Cliente -> UI : selecionar assento
UI -> Controller : reservar assento
Controller -> BD : atualizar assento como reservado
BD --> Controller : confirmação
Controller --> UI : assento reservado com sucesso
UI --> Cliente : confirmação da reserva
@enduml

UC-04 – Comprar Ingresso

@startuml

actor Cliente

participant "Interface Web" as UI

participant "Controlador de Compras" as Controller

participant "Banco de Dados" as BD

participant "Serviço de Pagamento" as Pagamento

Cliente -> UI : clicar em 'Comprar'

UI -> Controller : iniciar processo de compra

Controller -> Pagamento : processar pagamento

Pagamento --> Controller : pagamento aprovado

Controller -> BD : registrar compra

Controller -> BD : atualizar assento como ocupado

BD --> Controller : confirmação

Controller --> UI : gerar ingresso

UI --> Cliente : exibir/baixar ingresso

@enduml

UC-06 – Cadastrar Filme/Evento

@startuml

actor Administrador

participant "Interface Admin" as UI

participant "Controlador de Eventos" as Controller

participant "Banco de Dados" as BD

Administrador -> UI : acessar formulário de novo evento

Administrador -> UI : preencher dados e clicar 'Salvar'

UI -> Controller : enviar dados do evento

Controller -> BD : inserir novo filme/evento

BD --> Controller : confirmação de cadastro

Controller --> UI : evento cadastrado com sucesso

UI --> **Administrador** : exibir mensagem de sucesso
@enduml

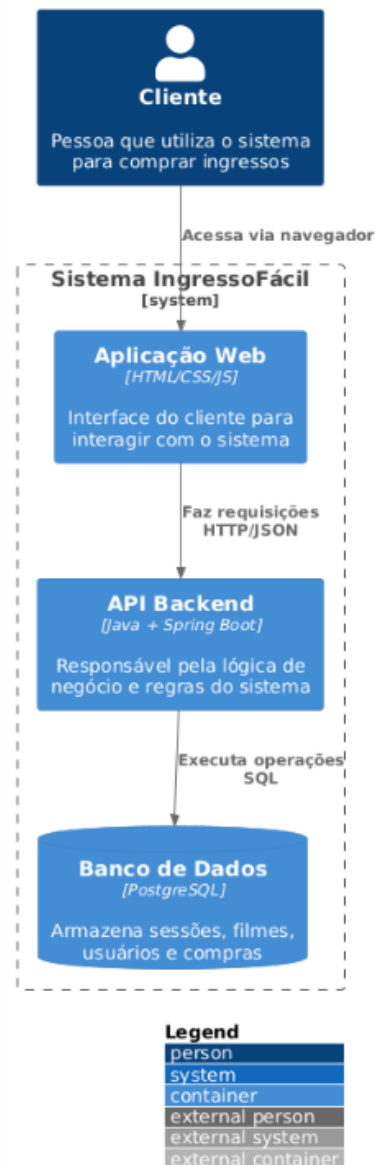
3. - Modelos de Projeto

3.1 - Arquitetura

Descrição Geral

O sistema IngressoFácil será baseado em uma arquitetura em camadas com separação de responsabilidades. Utilizaremos uma abordagem web com backend em Java (Spring Boot) e frontend em HTML/CSS/JS (ou framework web), além de um banco de dados relacional. A aplicação será dividida em 4 containers principais:

- Navegador Web (cliente)
- Aplicação Web (frontend)
- API de Negócio (backend)
- Banco de Dados (persistência)



| Camada | Tecnologia Sugerida |
|----------------|--|
| Frontend | HTML, CSS, JavaScript, Bootstrap |
| Backend/API | Java com Spring Boot |
| Banco de Dados | PostgreSQL |
| Hospedagem | Localhost (desenvolvimento) ou Heroku/Railway (produção) |
| Autenticação | Spring Security (futuro) |

Estrutura em Camadas (Lógica da API)

1. **Controller:** recebe as requisições HTTP
2. **Service:** contém a lógica de negócio
3. **Repository (DAO):** comunicação com o banco de dados
4. **Model:** representação das entidades (Filme, Sessão, Ingresso, etc)

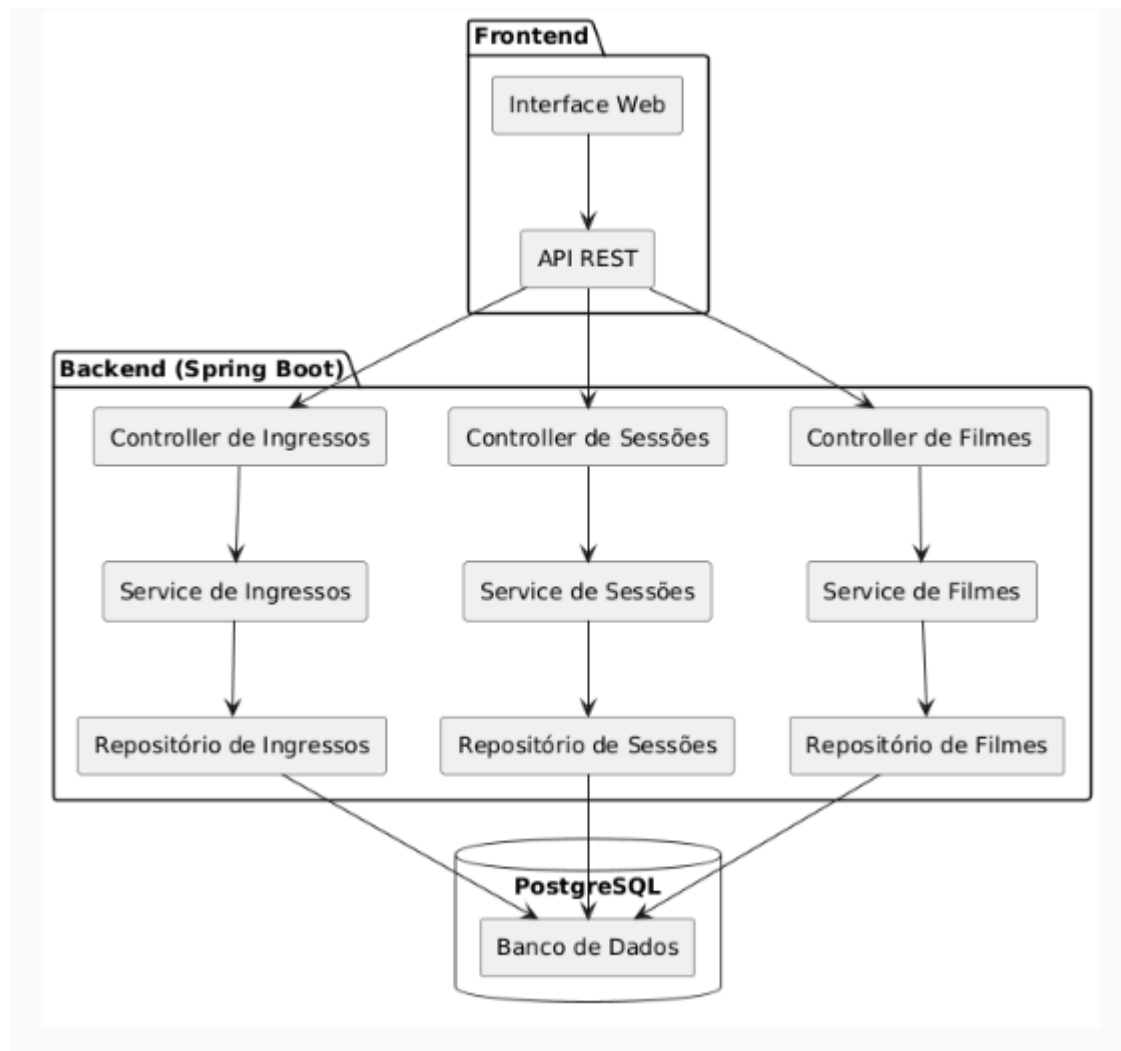
Benefícios da Arquitetura

- Separação de responsabilidades (frontend, backend, persistência)

- Facilidade de manutenção e testes
- Possibilidade de crescimento para versão mobile ou multiplataforma

3.2 - Diagrama de Componentes e Implantação

Diagrama de Componentes:

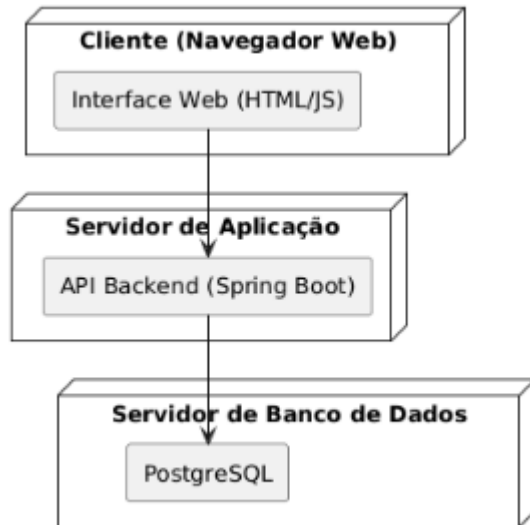


Os controllers recebem requisições da interface web (frontend).

Os services executam a lógica de negócio.

Os repositórios acessam o banco de dados.

Tudo se comunica por meio da API REST.

Diagrama de Implantação:

A interface roda no navegador do usuário.

A API (backend em Java) roda em um servidor separado.

O banco de dados roda em outro servidor (ou container).

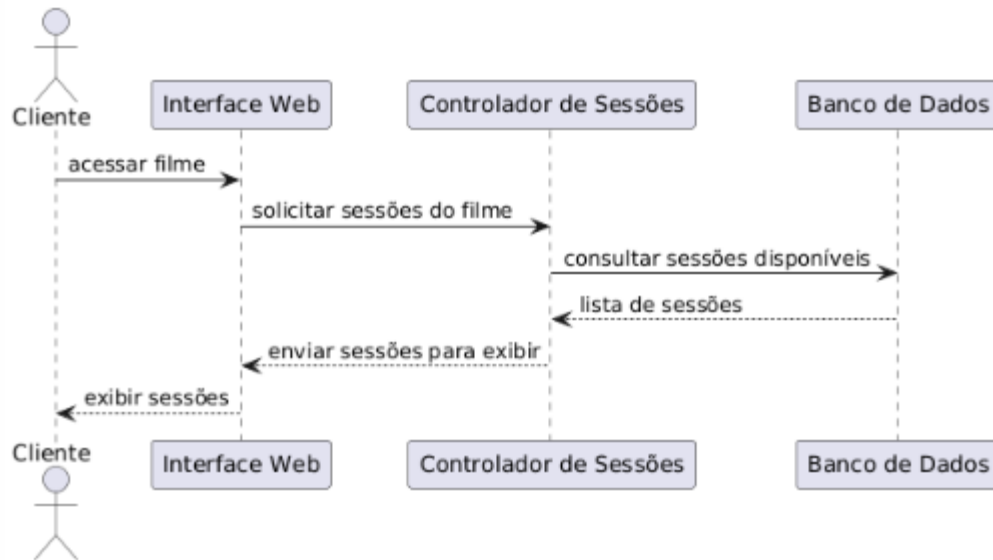
| Componente | Tecnologia Sugerida |
|-------------------------|-----------------------------|
| Cliente (Navegador Web) | HTML, CSS, JS (Bootstrap) |
| Servidor de Aplicação | Java 17, Spring Boot, Maven |
| Banco de Dados | PostgreSQL |

3.3 - Diagrama de Classes

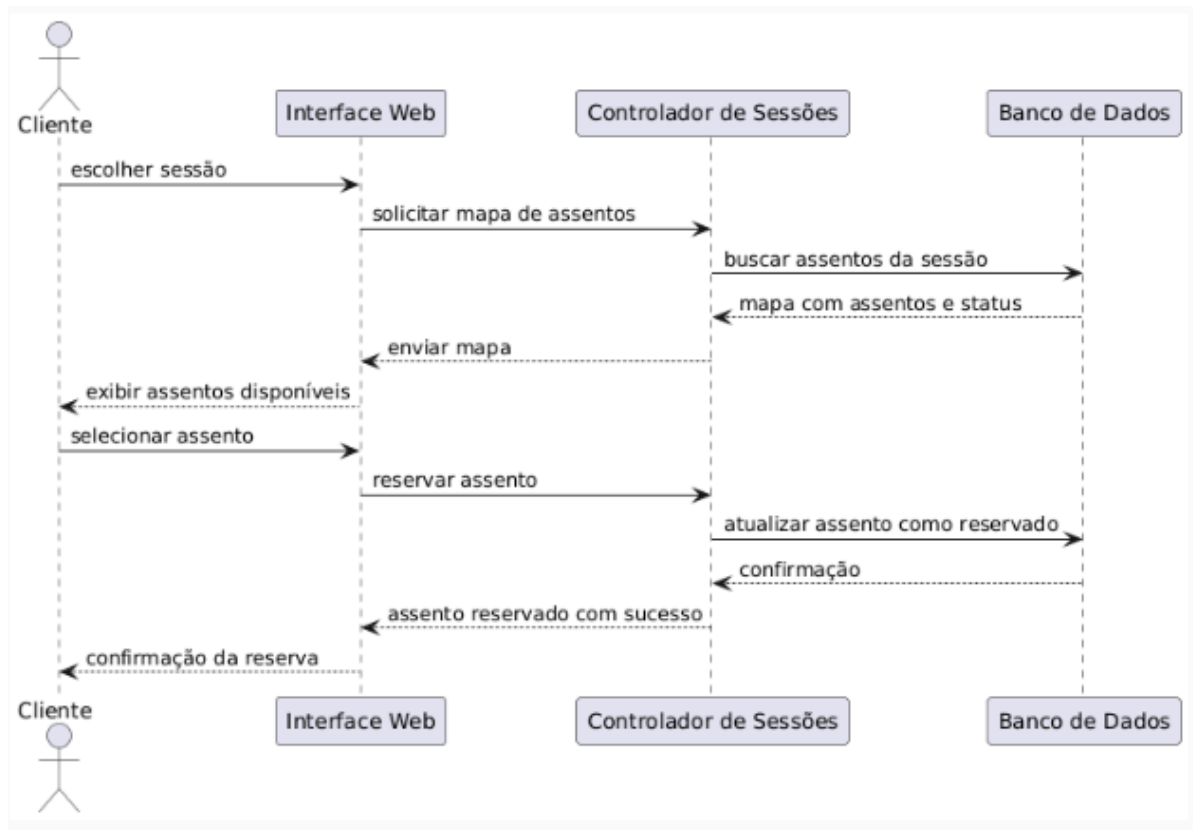


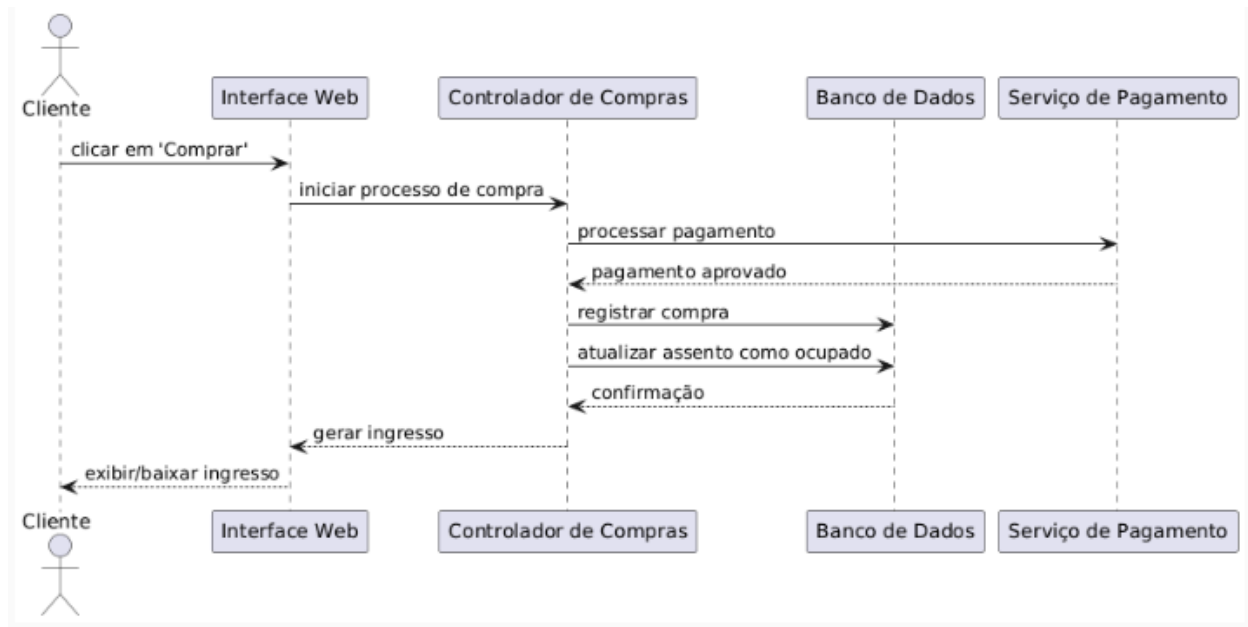
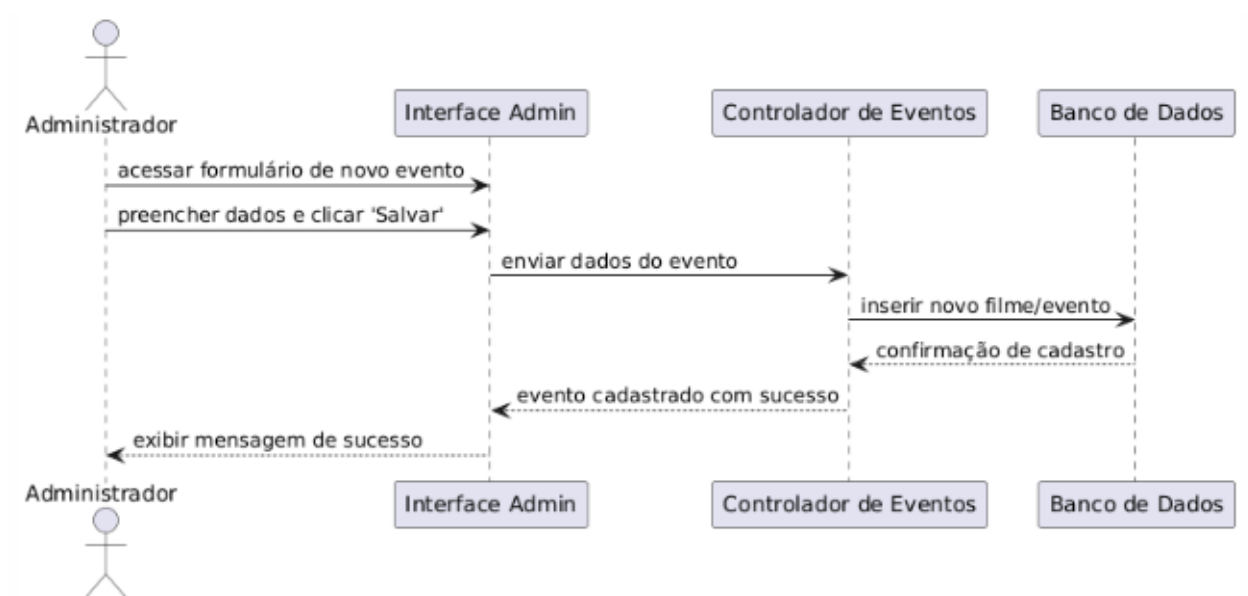
3.4 - Diagramas de Sequência

UC02: Selecionar sessão



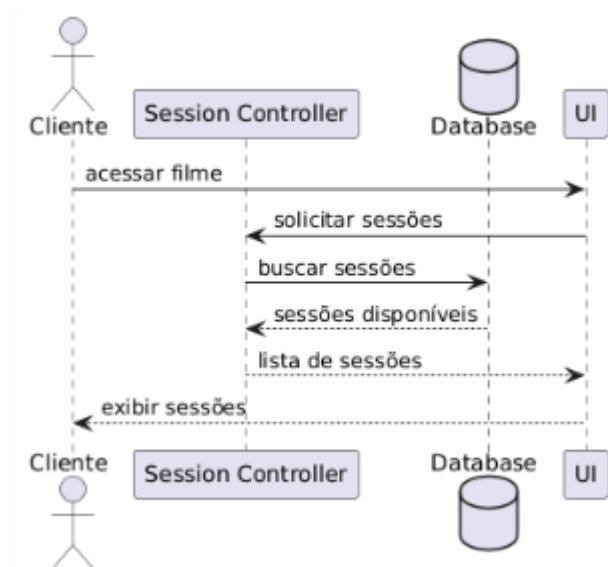
UC03: Escolher assento



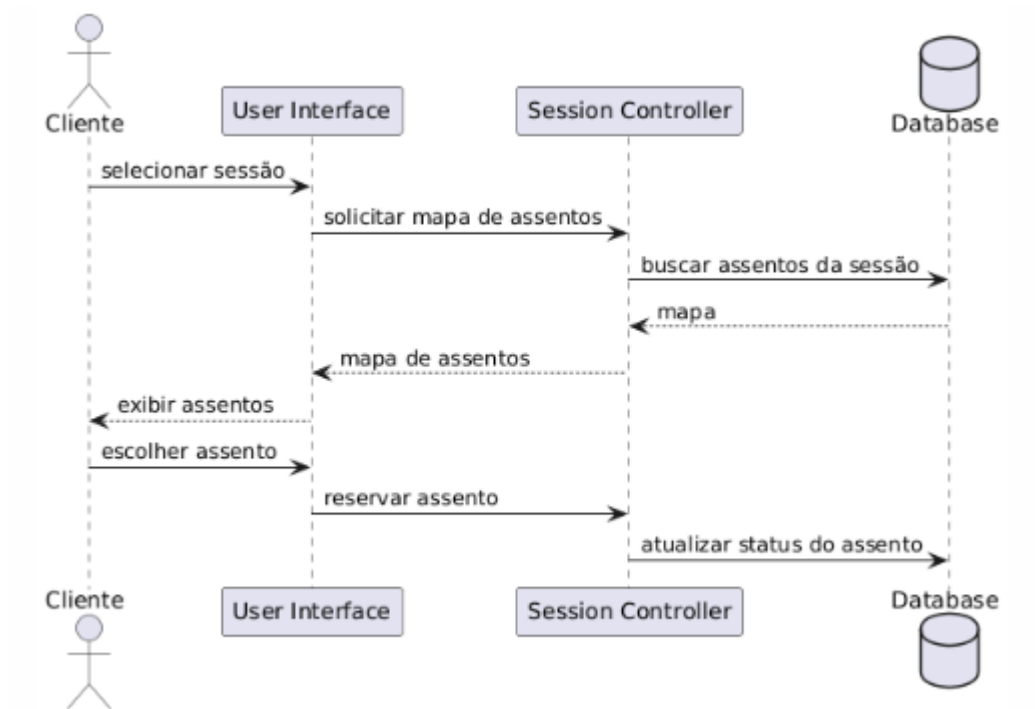
UC04: Comprar ingresso**UC05: Cadastrar filme/evento**

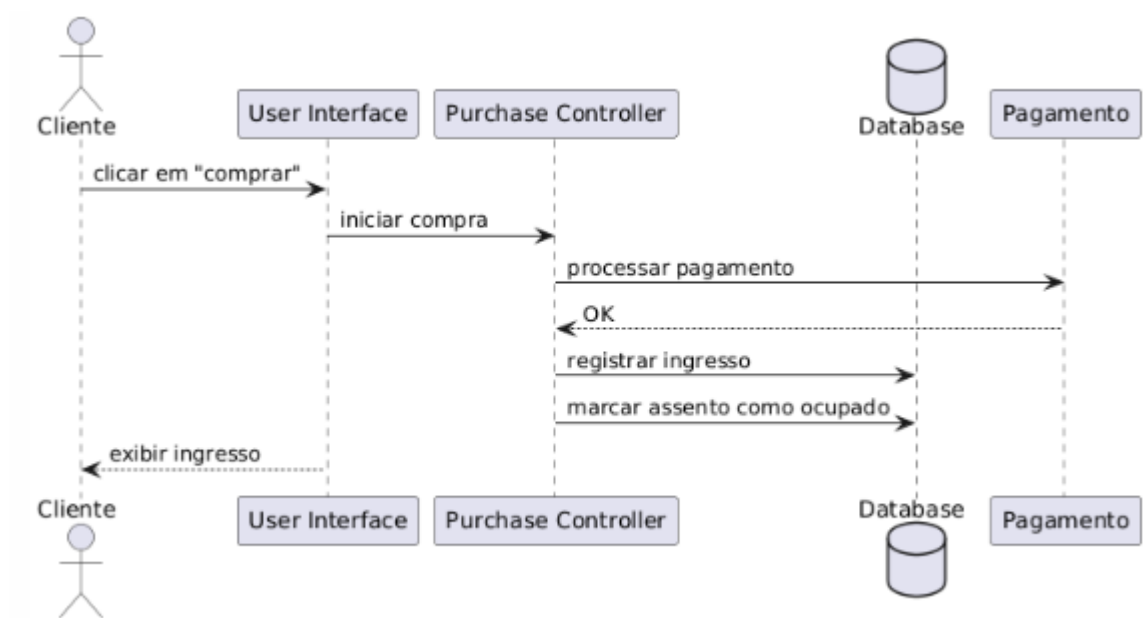
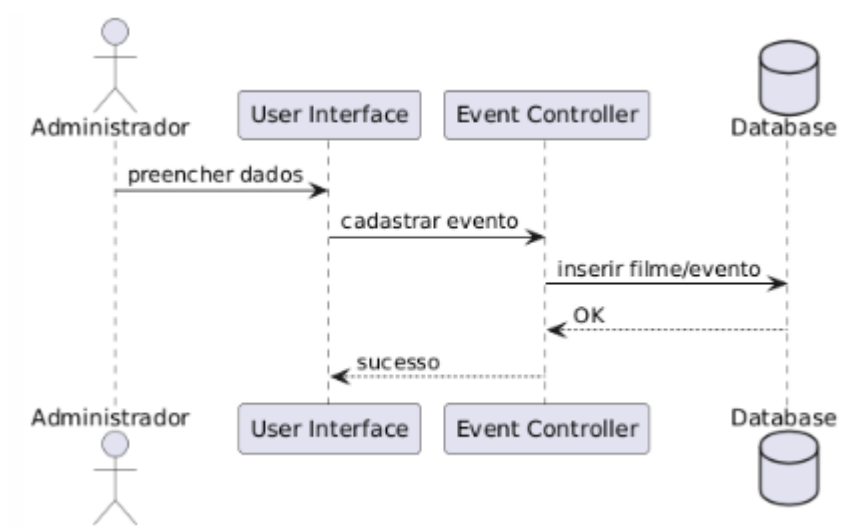
3.5 - Diagramas de Comunicação

UC02: Selecionar sessão



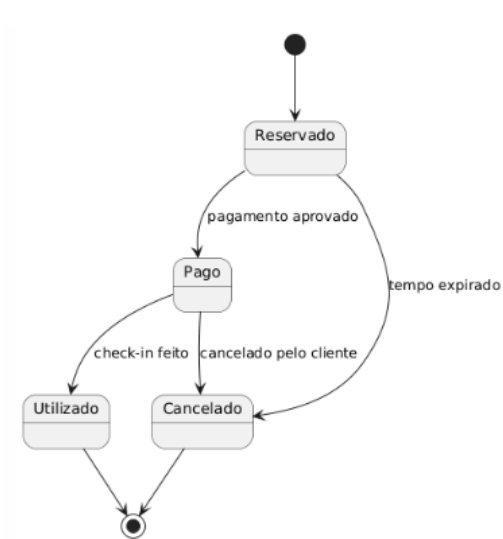
UC03: Escolher assento



UC04: Comprar ingresso**UC06: Cadastrar filme/evento**

3.6 - Diagramas de Estados

- Ciclo de vida de um ingresso:



- Estado de um assento:



- Estado de uma sessão



4. - Modelo de Dados

