

# **Especificação 4+1 - CineFácil**

**Felipe Rodrigues Santana**

**Gabriel de Souza Teles**

**Matheus Lima da Cruz**

**Nelson Ramon Timóteo dos Santos**

# Sumário

<b>1. Introdução</b>	<b>3</b>
2. Casos de Uso (Visão de Cenário)	4
<b>3. Visão Lógica</b>	<b>8</b>
<b>4. Visão de Processo (Dinâmica)</b>	<b>12</b>
Escolher Filme	15
Escolher Sessão de Filme	15
Escolher Poltrona	15
Pagamento	16
<b>5. Visão de Desenvolvimento (Visão de Implementação)</b>	<b>17</b>
Frontend	17
Backend	18
Banco de Dados	18
<b>6. Mapeamento</b>	<b>19</b>
Entidades e Relacionamentos	19
<b>7. Esquema do Banco de Dados</b>	<b>20</b>
<b>8. Referências</b>	<b>23</b>

# 1. Introdução

## 1.1 Objetivo

Este documento tem como objetivo descrever a arquitetura do sistema CineFácil, uma plataforma de venda de ingressos para cinemas, utilizando o modelo 4+1 visões arquiteturais. A arquitetura do sistema será apresentada sob diferentes perspectivas para fornecer uma visão clara e detalhada de sua estrutura, funcionamento e organização. O foco principal é garantir que o sistema seja escalável, eficiente e fácil de manter, atendendo aos requisitos funcionais e não funcionais definidos.

## 1.2 Escopo

O CineFácil é um sistema de gerenciamento de vendas de ingressos de cinema que permite aos usuários:

Cadastrar-se e realizar login no sistema;

Visualizar filmes em exibição e escolher uma sessão;

Selecionar poltronas e itens adicionais;

Finalizar a compra e realizar pagamentos;

Administrar filmes, sessões e itens adicionais para usuários com perfil de administrador.

O escopo do documento abrange a descrição das quatro principais visões arquiteturais do sistema: lógica, processo, desenvolvimento, e casos de uso.

## 1.3 Visão Geral

O sistema será descrito com base nas seguintes visões:

**Casos de Uso:** Detalha as funcionalidades do sistema através dos casos de uso mais relevantes.

**Visão Lógica:** Descreve a estrutura dos principais componentes e suas interações.

**Visão de Processo:** Foca no comportamento dinâmico e nas interações entre os componentes durante a execução do sistema.

**Visão de Desenvolvimento:** Mostra como o sistema está organizado em termos de módulos de código e suas dependências.

## 2. Casos de Uso (Visão de Cenário)

### 2.1 Objetivo

A Visão de Cenário tem como objetivo descrever as principais funcionalidades do sistema CineFácil por meio de casos de uso que ilustram as interações entre os usuários (atores) e o sistema. Os casos de uso identificam os principais serviços oferecidos pela aplicação, ajudando a compreender o comportamento do sistema sob diferentes perspectivas de uso.

### 2.2 Lista de Casos de Uso

Abaixo estão listados os principais casos de uso que definem as funcionalidades do CineFácil:

**Cadastro de Usuário:** O usuário se cadastra no sistema fornecendo informações pessoais.

**Login de Usuário:** O usuário realiza login no sistema para acessar as funcionalidades.

**Visualizar Filmes em Exibição:** O usuário pode visualizar a lista de filmes atualmente em exibição no cinema.

**Escolher Filme:** O usuário seleciona um filme para visualizar as sessões disponíveis.

**Escolher Sessão do Filme:** O usuário escolhe a sessão de um filme para a qual deseja comprar ingressos.

**Escolher Poltrona:** O usuário seleciona uma poltrona disponível na sessão escolhida.

**Escolher Adicionais:** O usuário pode escolher itens adicionais, como pipoca ou bebidas, para adicionar ao pedido.

**Revisar Escolhas:** O usuário revisa todas as suas escolhas (filme, sessão, poltrona, e adicionais) antes de finalizar o pedido.

**Editar Escolhas:** O usuário pode editar as suas escolhas (filme, sessão, poltrona, ou adicionais) antes da finalização.

**Finalizar Pedido:** O usuário finaliza a compra de ingressos e adicionais.

**Escolher o Método de Pagamento:** O usuário seleciona a forma de pagamento para concluir o pedido.

**Cancelar Pedido:** O usuário pode cancelar um pedido antes da confirmação final.

**Consultar Histórico de Pedidos:** O usuário visualiza o histórico de compras realizadas no sistema.

**Adicionar Novo Filme (Admin):** O administrador adiciona um novo filme ao catálogo do cinema.

**Remover um Filme (Admin):** O administrador remove um filme do catálogo do cinema.

**Adicionar uma Sessão (Admin):** O administrador adiciona uma nova sessão de filme.

**Remover uma Sessão (Admin):** O administrador remove uma sessão existente.

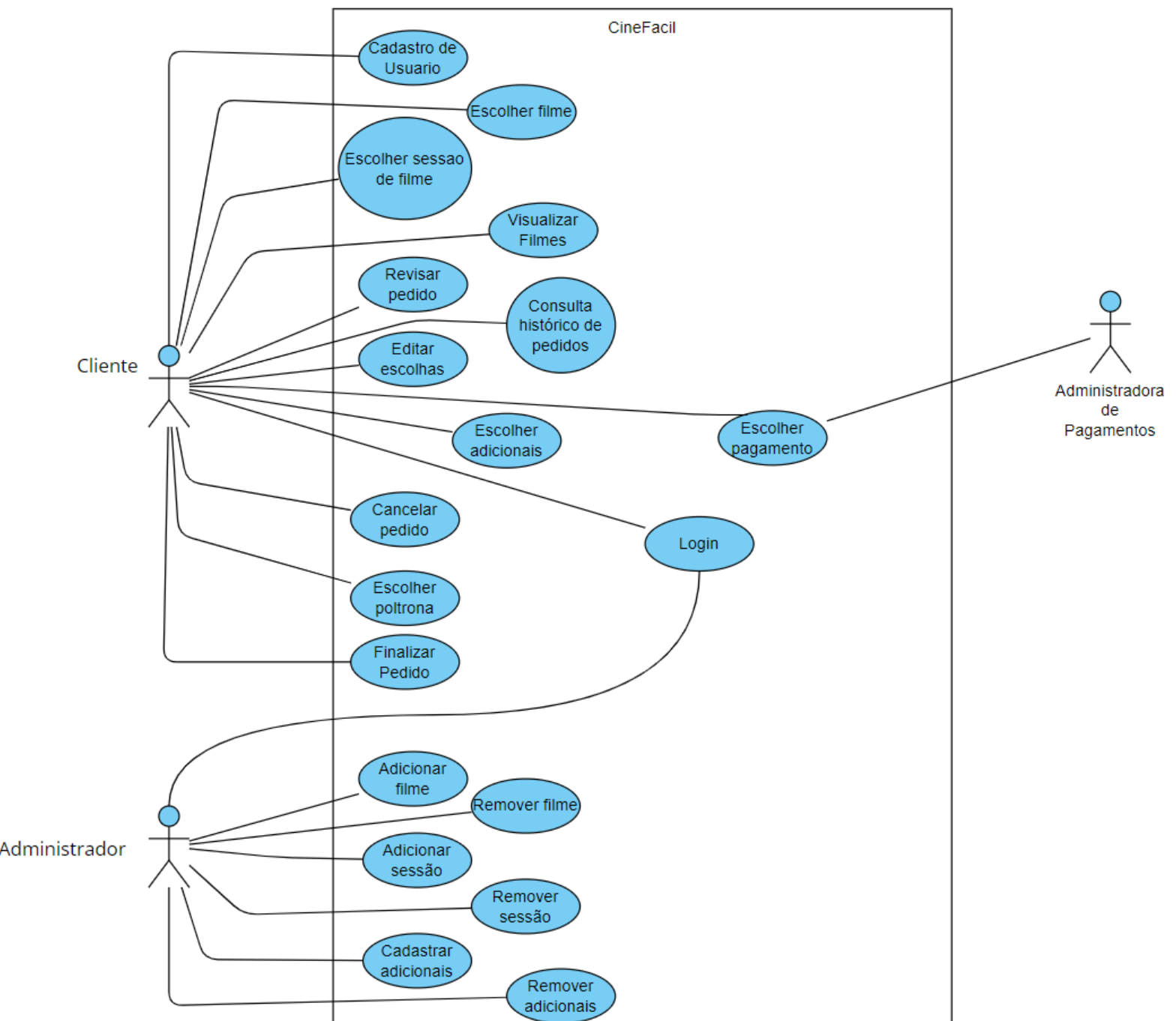
**Cadastrar Itens Adicionais (Admin):** O administrador adiciona novos itens adicionais ao sistema (ex.: pipoca, bebidas).

**Remover Itens Adicionais (Admin):** O administrador remove itens adicionais do sistema.

## 2.3 Diagramas de Casos de Uso

Abaixo estão os diagramas de casos de uso que ilustram a interação entre os atores e o sistema CineFácil:

Diagrama de Casos de Uso - Usuário Regular e do Administrador: Mostra as interações de um usuário comum com o sistema, como cadastro, login, escolha de filme, compra de ingressos, consulta de histórico e adicionar/remover filmes, sessões e itens adicionais.



## 2.4 Atores

Os principais atores do sistema CineFácil são:

**Usuário Regular:** Um cliente que usa o sistema para comprar ingressos e itens adicionais.

**Administrador:** Um usuário com privilégios administrativos que gerencia filmes, sessões, e itens adicionais.

## 3. Visão Lógica

### 3.1 Objetivo

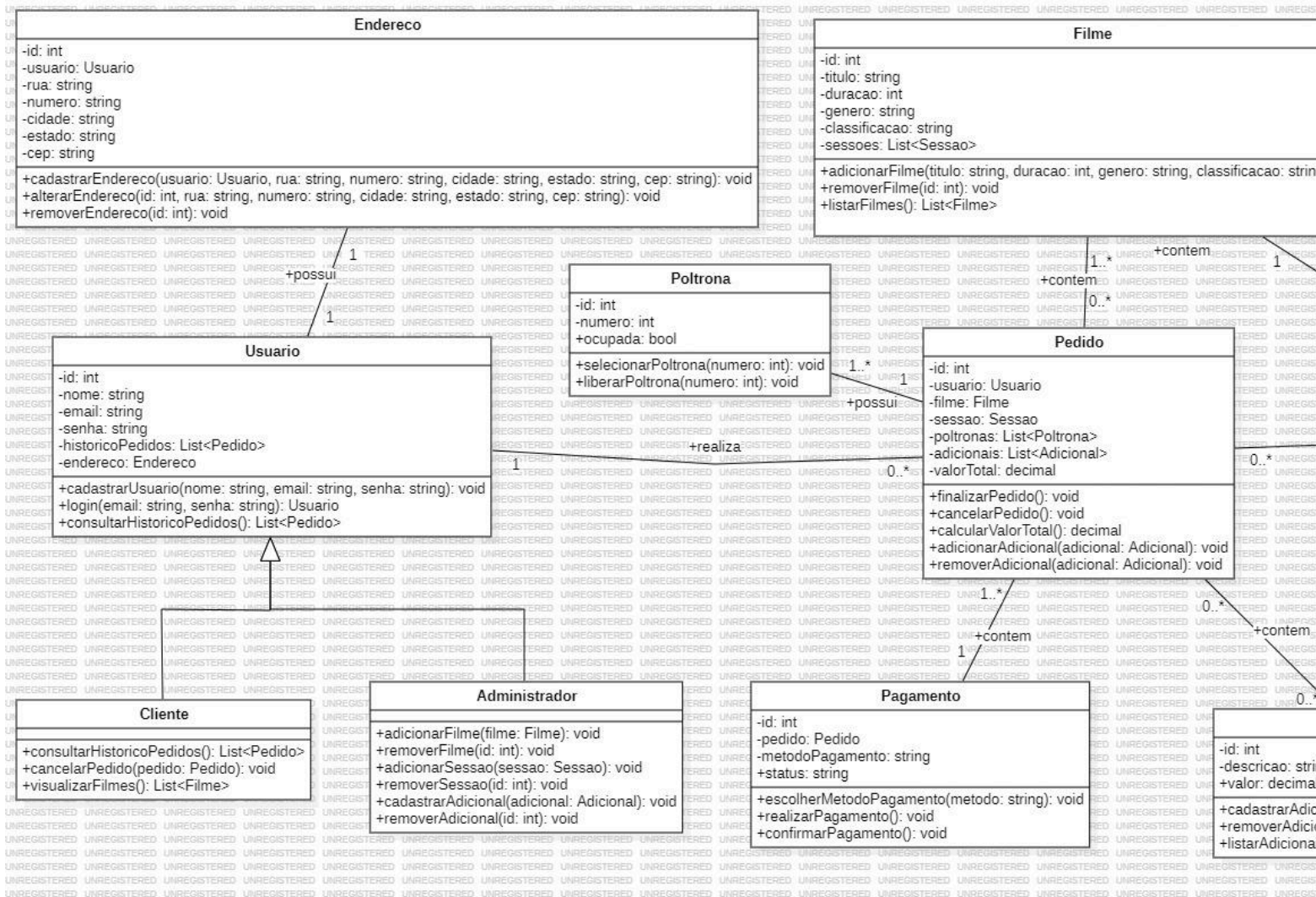
A Visão Lógica tem como objetivo descrever a estrutura dos principais componentes do sistema CineFácil e as interações entre eles. Ela foca nos elementos essenciais do sistema, como as classes, suas responsabilidades e como elas se relacionam, fornecendo uma visão clara de como o sistema é organizado do ponto de vista lógico.

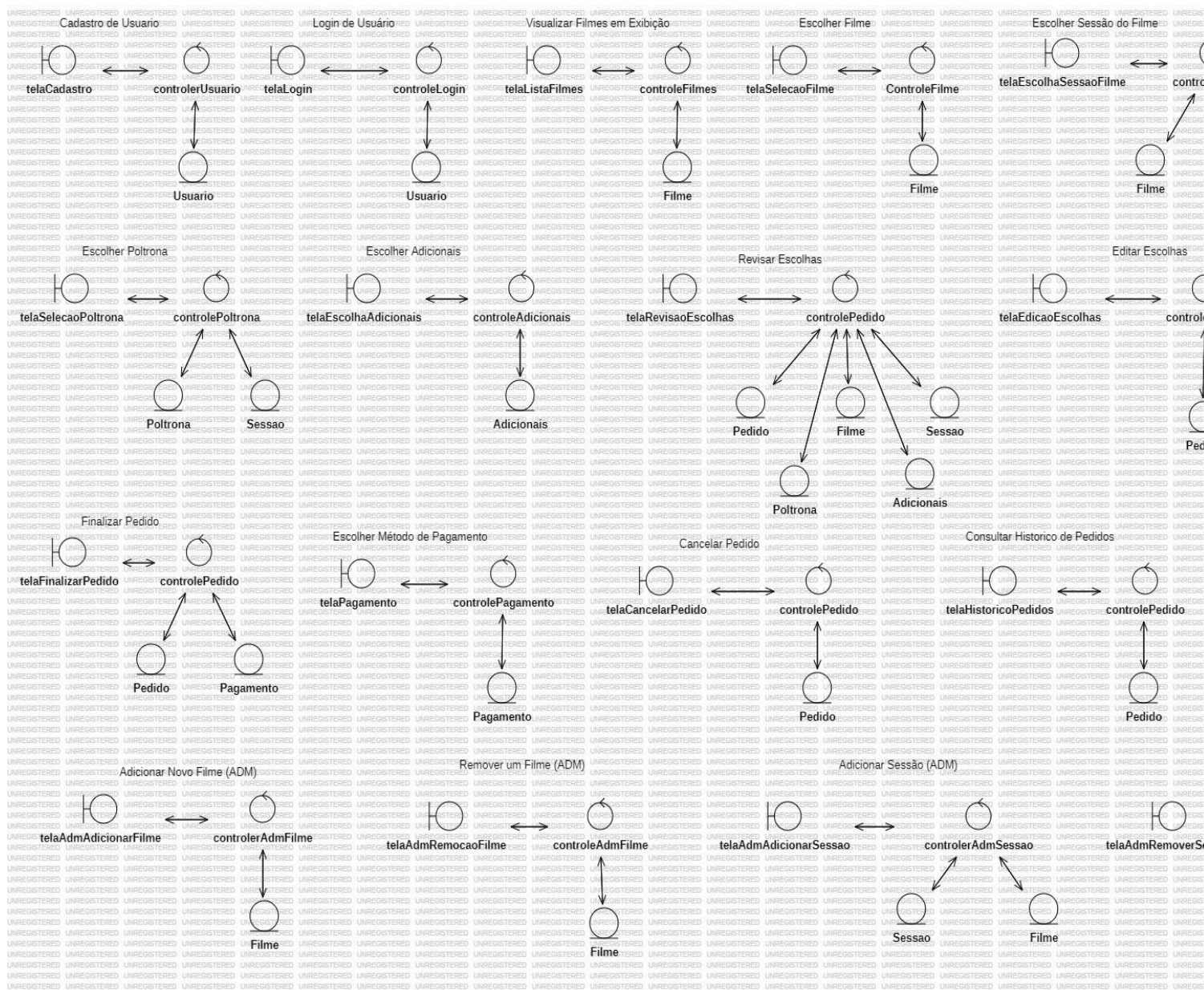
### 3.2 Diagrama de Classes (UML)

O Diagrama de Classes mostra a estrutura estática do sistema, detalhando as classes e seus relacionamentos. O diagrama será dividido em camadas de acordo com as responsabilidades de cada componente:

- **Camada de Controle:**
  - Contém as classes responsáveis pela lógica de negócios e gerenciamento das interações do sistema.
  - Exemplo: *UsuarioController*, *FilmeController*, *SessaoController*, *PagamentoController*.
- **Camada de Entidades:**
  - Representa os objetos de domínio do sistema, como *Usuario*, *Filme*, *Sessao*, *Poltrona*, *ItemAdicional*, *Pedido*.
  - Cada classe de entidade terá atributos que representam o estado do objeto no sistema.
- **Camada de Interface (Fronteira):**
  - Classes que interagem diretamente com os usuários e administradores.
  - Exemplo: *TelaCadastroUsuario*, *TelaEscolhaFilme*, *TelaPagamento*, *TelaGerenciamentoAdmin*.







### 3.3 Descrição dos Principais Componentes

Cada componente do sistema é descrito com suas responsabilidades e interações:

#### **UsuarioController:**

Responsável pelo controle de cadastro e login de usuários.

Interage com a entidade Usuario e a interface TelaCadastroUsuario para gerenciar essas operações.

#### **FilmeController:**

Gerencia a exibição e escolha de filmes.

Interage com as entidades Filme e Sessao, e com a interface TelaEscolhaFilme.

#### **CarrinhoDeComprasController:**

Gerencia as seleções temporárias do usuário antes da finalização do pedido.

Interage com Poltrona, ItemAdicional e Pedido.

#### **PagamentoController:**

Responsável pela lógica de pagamento e interação com as opções de pagamento disponíveis.

Interage com o carrinho de compras e as entidades Pedido e Pagamento.

#### **AdminController:**

Gerencia o acesso e as operações de administração, como adicionar e remover filmes, sessões e itens adicionais.

Interage com as interfaces de administração e as entidades Filme, Sessao, e ItemAdicional.

### 3.4 Relacionamentos entre Componentes

Os componentes do sistema interagem da seguinte forma:

- Camada de Interface (Fronteira) faz a comunicação entre o usuário e o sistema, capturando as entradas e mostrando as saídas.
- Camada de Controle contém a lógica de negócio e validação, gerenciando o fluxo de dados entre a interface e as entidades.
- Camada de Entidades armazena os dados de domínio do sistema, representando o estado dos objetos persistentes (filmes, usuários, sessões, etc.).

## 4. Visão de Processo (Dinâmica)

### 4.1 Objetivo

A Visão de Processo descreve a **dinâmica** e o comportamento do sistema **CineFácil** em tempo de execução, enfatizando como os componentes interagem entre si para realizar as principais funcionalidades. O foco é nos processos concorrentes, nas interações e nas trocas de mensagens entre os diferentes componentes do sistema, além de destacar os fluxos principais de execução.

### 4.2 Processos Principais

Os principais processos do CineFácil que envolvem a interação entre componentes durante a execução do sistema são:

#### **Processo de Autenticação:**

O usuário envia suas credenciais para o sistema.

O sistema valida as credenciais com o banco de dados e cria uma sessão.

Caso as credenciais sejam inválidas, o sistema retorna um erro e solicita uma nova tentativa.

#### **Processo de Seleção de Filme e Sessão:**

O usuário escolhe um filme a partir da lista de exibição.

O sistema recupera as sessões disponíveis para o filme escolhido.

O usuário seleciona uma sessão específica e o sistema exibe as poltronas disponíveis.

#### **Processo de Seleção de Poltronas e Itens Adicionais:**

O usuário seleciona as poltronas desejadas.

O sistema verifica a disponibilidade em tempo real para garantir que as poltronas não tenham sido reservadas por outro usuário.

O usuário pode selecionar itens adicionais, como pipoca ou bebidas.

#### **Processo de Revisão e Finalização de Pedido:**

O usuário revisa suas escolhas (filme, sessão, poltronas, itens adicionais).

O sistema mostra o resumo do pedido, incluindo o valor total.

O usuário confirma a finalização do pedido e escolhe o método de pagamento.

### **Processo de Pagamento:**

O usuário escolhe o método de pagamento.

O sistema interage com gateways de pagamento para processar a transação.

Após a confirmação do pagamento, o sistema gera o pedido final e disponibiliza os ingressos.

### **Processo de Administração de Filmes e Sessões (Admin):**

O administrador acessa o painel de administração.

O sistema permite ao administrador adicionar, editar ou remover filmes e sessões.

As alterações são refletidas em tempo real no catálogo de filmes e sessões disponíveis para os usuários.

## **4.3 Diagramas de Sequência**

Os diagramas de sequência ilustram as interações entre os componentes do sistema ao longo do tempo, detalhando o fluxo de mensagens entre objetos e atores em um determinado cenário.

- Diagrama de Sequência - Processo de Compra de Ingressos:

Mostra o fluxo desde a escolha do filme até a finalização do pedido, incluindo as interações entre o usuário, os controladores (ex.: FilmeController, SessaoController, CarrinhoDeComprasController) e as entidades (ex.: Filme, Sessao, Poltrona).

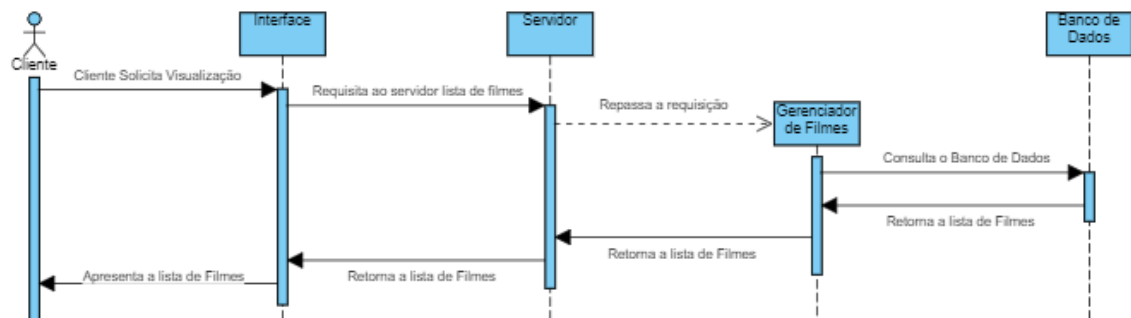
- Diagrama de Sequência - Processo de Pagamento:

Mostra as interações entre o usuário, o sistema de pagamento e os componentes que gerenciam o pedido e a transação financeira.

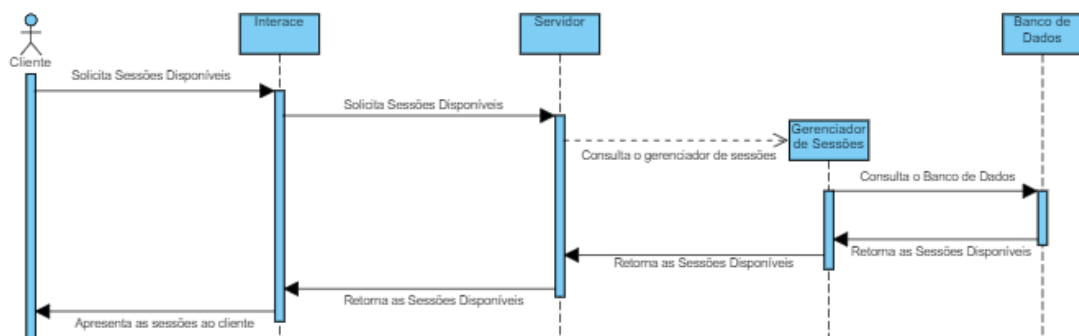
- Diagrama de Sequência - Administração de Sessões:

Mostra como o administrador adiciona ou remove sessões, interagindo com o banco de dados e atualizando as informações para os usuários em tempo real.

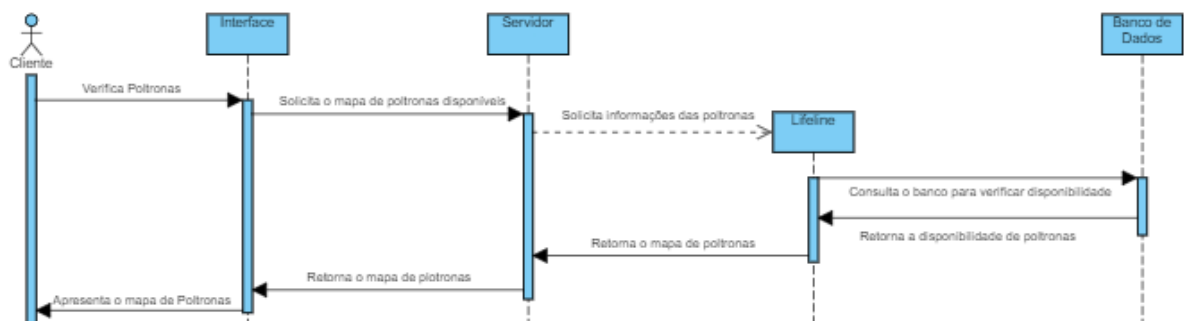
## Escolher Filme



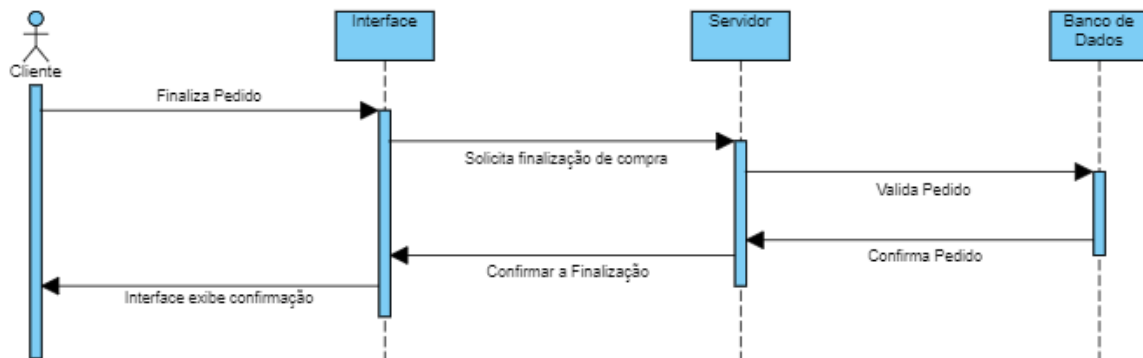
## Escolher Sessão de Filme



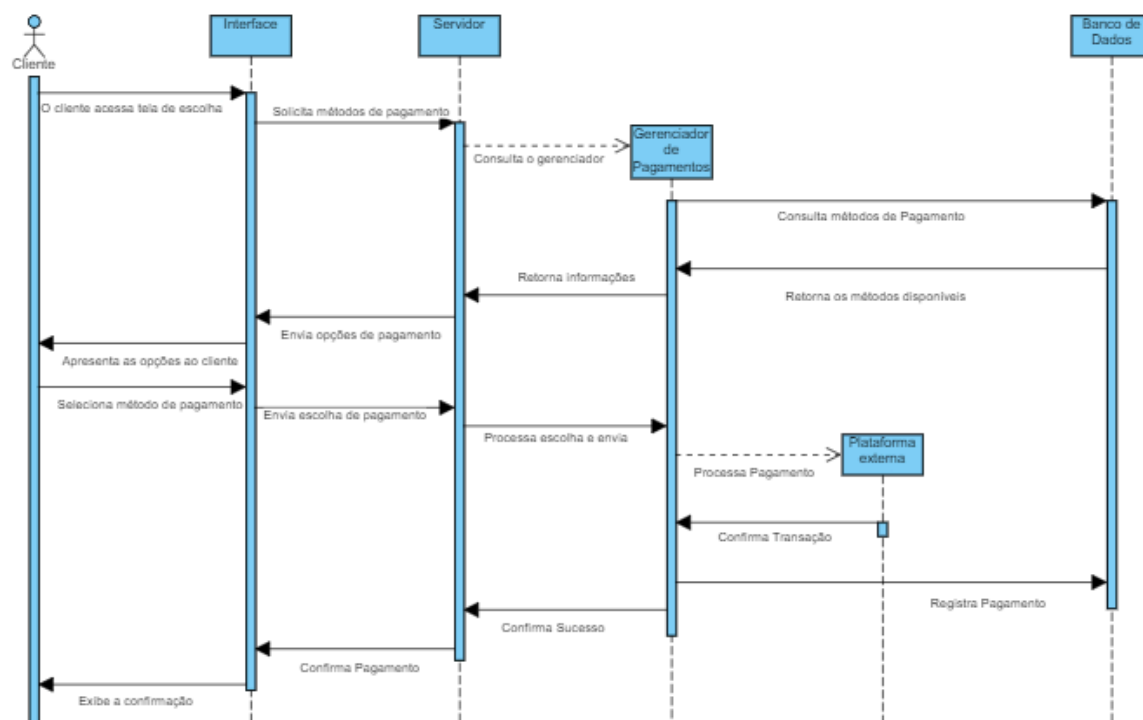
## Escolher Poltrona



## Finalizar Pedido



## Pagamento



## 5. Visão de Desenvolvimento (Visão de Implementação)

### 5.1 Objetivo



A Visão de Desenvolvimento descreve a estrutura física e modular do sistema **CineFácil**, mostrando como os componentes do software são organizados em módulos, pacotes e subsistemas para facilitar a implementação e a manutenção. Essa visão também detalha as tecnologias e frameworks utilizados no desenvolvimento, além de boas práticas adotadas para garantir escalabilidade e manutenção. A arquitetura do sistema CineFácil segue o padrão **Model-View-Controller (MVC)**, visando a separação clara das responsabilidades entre as camadas do sistema. O uso do modelo MVC permite uma melhor organização do código, facilitando a manutenção e escalabilidade do sistema.

## 5.1 Visão Geral da Arquitetura

- **Model:** Responsável pela lógica de negócios e pelo gerenciamento dos dados da aplicação. No caso do CineFácil, o C# é utilizado no backend para processar as regras de negócio e interagir com o banco de dados relacional PostgreSQL. O modelo contém as definições das entidades e gerencia as operações de leitura e escrita no banco de dados.
- **View:** responsável pela visualização, implementada com Vue.js, é responsável pela interação com o usuário final. Ela apresenta as informações ao usuário e captura as entradas, enviando-as para o controlador. A utilização de Vue.js permite uma interface dinâmica, altamente interativa, e separa as preocupações da interface do usuário (UI) da lógica de negócios.
- **Controller:** O controller, implementado em C#, é responsável por receber as solicitações do usuário através da camada de visualização, processar as entradas, e atualizar o modelo e a visão de acordo. Ele atua como intermediário entre a visão e o modelo, garantindo que as regras de negócios sejam aplicadas corretamente.

## 5.2 Detalhamento das Camadas

### Frontend

- **Tecnologia:** Vue.js
- **Função:** Gerenciamento da interface do usuário, exibindo as informações provenientes do backend e enviando as requisições do usuário para o controlador.
- **Comunicação:** As interações entre o usuário e o sistema são gerenciadas por meio de chamadas HTTP (GET, POST, PUT, DELETE), enviadas para a camada de controlador via API RESTful.

### Backend

- **Tecnologia:** C# com ASP.NET Core
- **Função:** Processar as solicitações recebidas do frontend, aplicar as regras de negócio e interagir com o banco de dados PostgreSQL. O controlador

organiza a lógica de fluxo entre o usuário e os dados, enquanto o modelo gerencia o acesso e a integridade dos dados.

- **Comunicação:** O backend expõe uma **API RESTful** que lida com as requisições do frontend e realiza operações de consulta, inserção, atualização e exclusão no banco de dados.

## Banco de Dados

- **SGBD: PostgreSQL**
- **Função:** Armazenamento de dados persistentes relacionados às sessões de cinema, usuários, pedidos e pagamentos. O banco de dados é acessado através da camada de modelo no backend, utilizando consultas SQL parametrizadas.

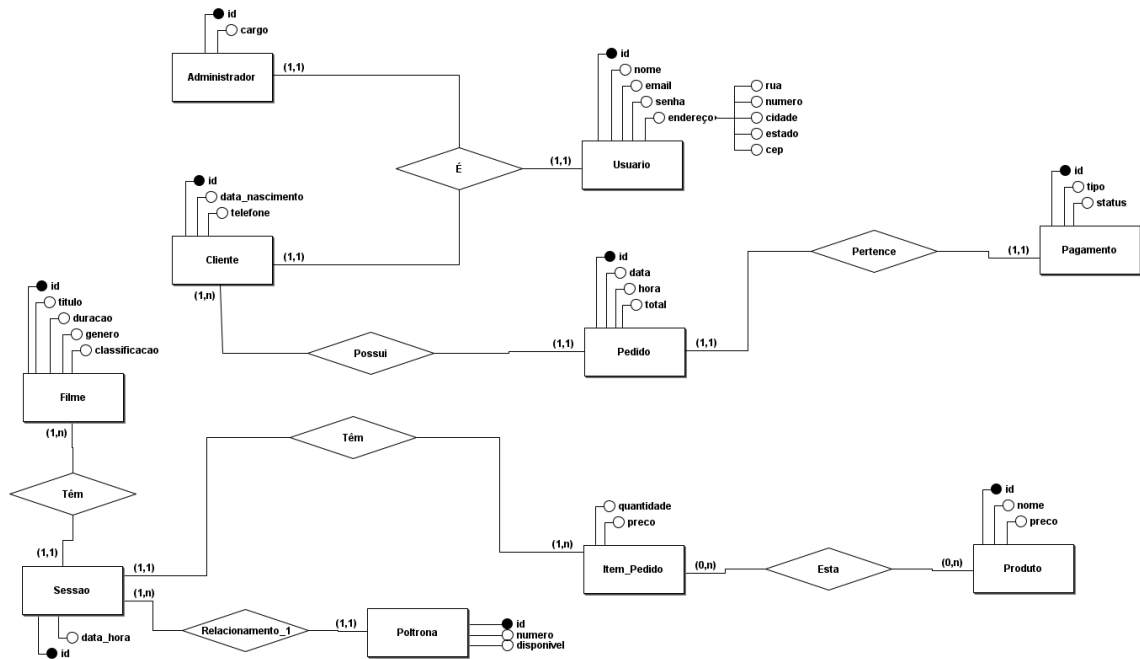
## 5.3 Padrões de Comunicação

- **API RESTful:** O sistema adota o padrão **REST** para a comunicação entre o frontend e o backend. A API gerada pelo backend utiliza chamadas HTTP (GET, POST, PUT, DELETE) para realizar as operações solicitadas pelo usuário e retornar os dados no formato **JSON**.
- **Autenticação e Autorização:** Para garantir a segurança, a arquitetura implementa mecanismos de autenticação, controlando o acesso às diferentes funcionalidades do sistema.

# 6. Mapeamento

Seguindo o diagrama de classes, mapeamos as entidades (classes) e seus atributos para a estrutura relacional, que será documentada utilizando os artefatos de arquitetura descritos na ISO/IEC/IEEE 42010. Isso envolve a descrição dos componentes do sistema e suas interações.

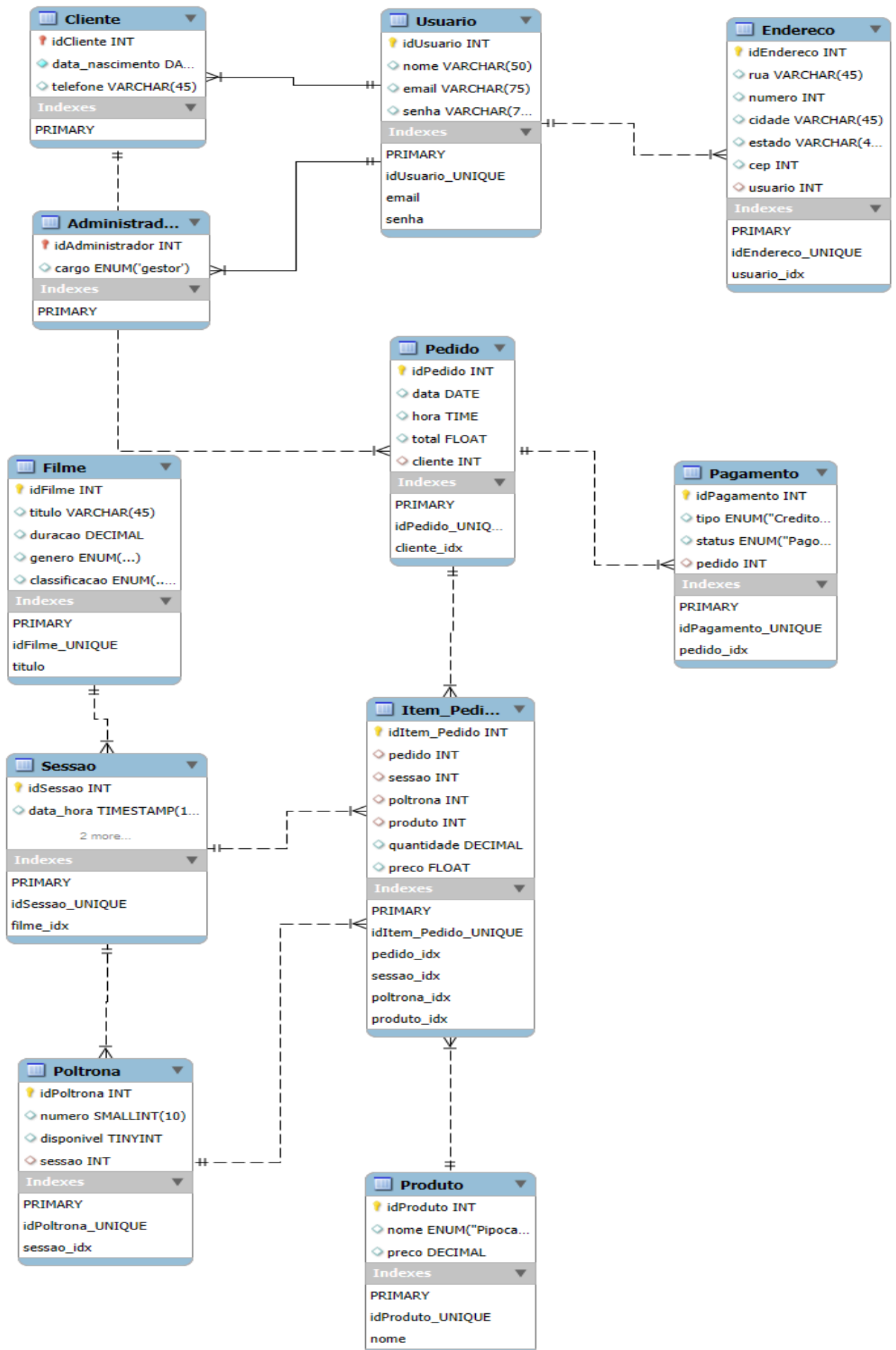
## Entidades e Relacionamentos



## 7. Esquema do Banco de Dados

Conforme as recomendações das normas ISO/IEC/IEEE 42010, o esquema do banco de dados define a estrutura lógica do banco de dados relacional. Ele inclui as tabelas e as respectivas chaves primárias, estrangeiras, e restrições de integridade.

- **Objetivo:** Definir o esquema do banco de dados que será utilizado para armazenar e organizar as informações do sistema CineFácil.



## 8. Referências

**IEEE.** *IEEE Recommended Practice for Software Requirements Specifications* (IEEE Std 830-1998). IEEE, 1998. DOI: 10.1109/IEEESTD.1998.88286.

**Pressman, Roger S., Maxim, Bruce R.** *Engenharia de Software*. 9ª ed., McGraw Hill Brasil, 2021.

**Sommerville, Ian.** *Engenharia de Software*. 9ª ed., Pearson Prentice Hall, 2011.

**Bezerra, Eduardo.** *Princípios de Análise e Projeto de Sistemas com UML*. Elsevier, 2007.