**SENAI ETTORE ZANINI**

**TÉCNICO DESENVOLVIMENTO DE SISTEMAS**

**Ana Luiza de Assis**

**Luís Otávio Bordin Pereira**

**Maria Luísa Campos**

**Rafaela Pereira Gusmão**

**CinePop**

**Sertãozinho**  
 **2025**

Sumário

[Sumário 2](#_Toc212629858)

[1 BENCHMARKING 1](#_Toc212629859)

[1.1 Objetivo 1](#_Toc212629860)

[1.2 Dados coletados 1](#_Toc212629861)

[1.3 Resultado 3](#_Toc212629862)

[2 REQUISITOS DO SISTEMA 4](#_Toc212629863)

[2.1 Requisitos Funcionais 4](#_Toc212629864)

[2.2 Requisitos Não Funcionais 14](#_Toc212629865)

[3 MODELAGEM 20](#_Toc212629866)

[3.1 Diagrama de Classe 20](#_Toc212629867)

[3.2 Diagrama de Entidade Relacionamento (DER) 20](#_Toc212629868)

[4. Dicionário API 21](#_Toc212629869)

[4.1. Dicionário de Endpoints 21](#_Toc212629870)

[4.2 Dicionário de Dados 23](#_Toc212629871)

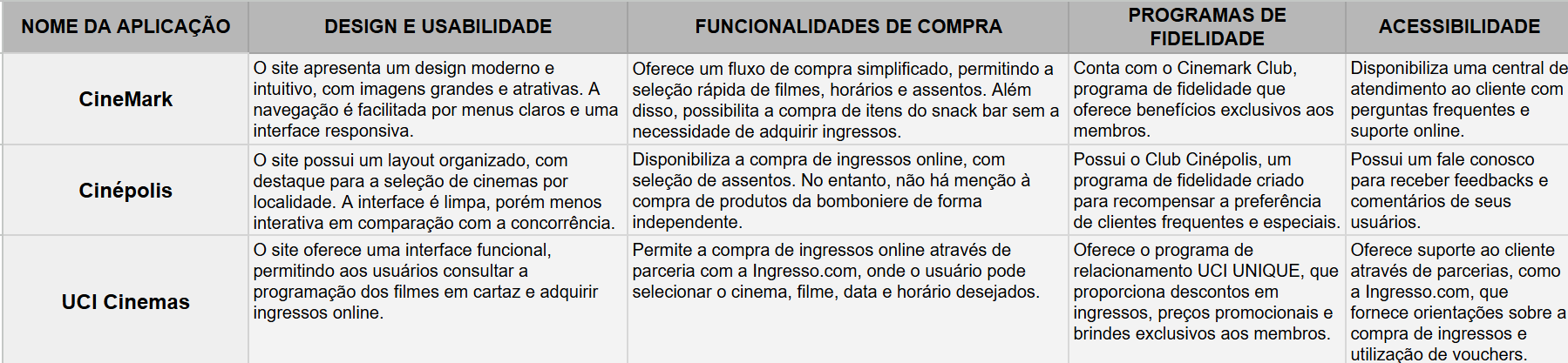
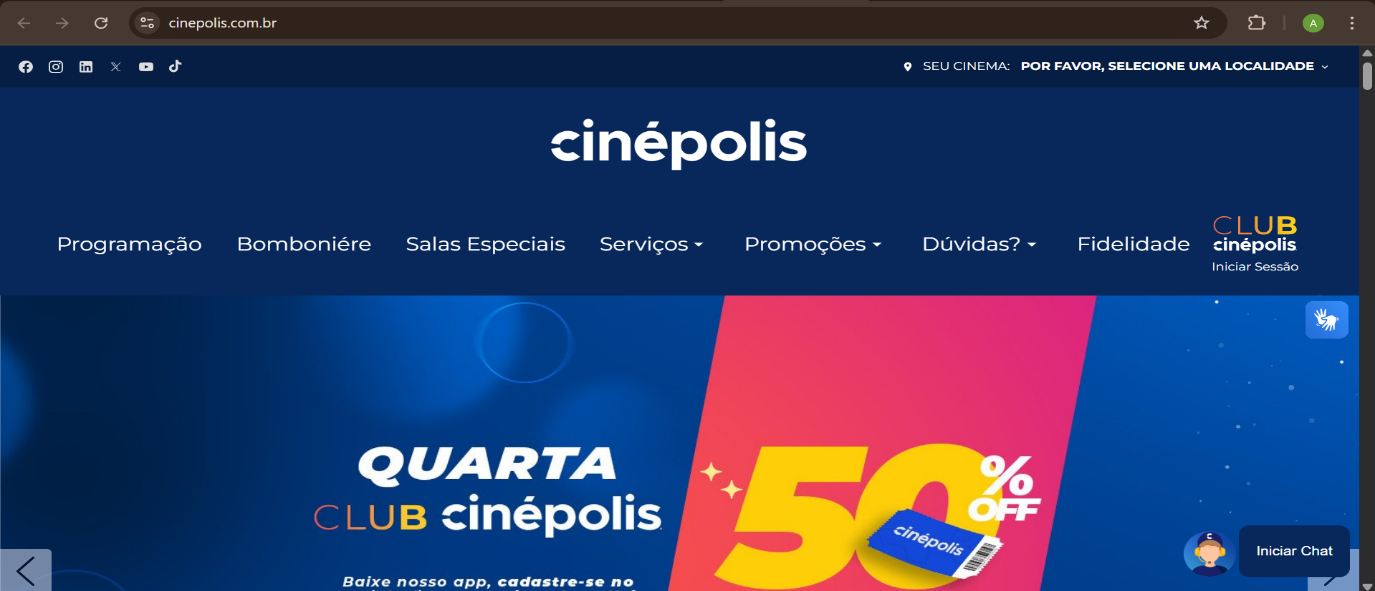
# 1 BENCHMARKING

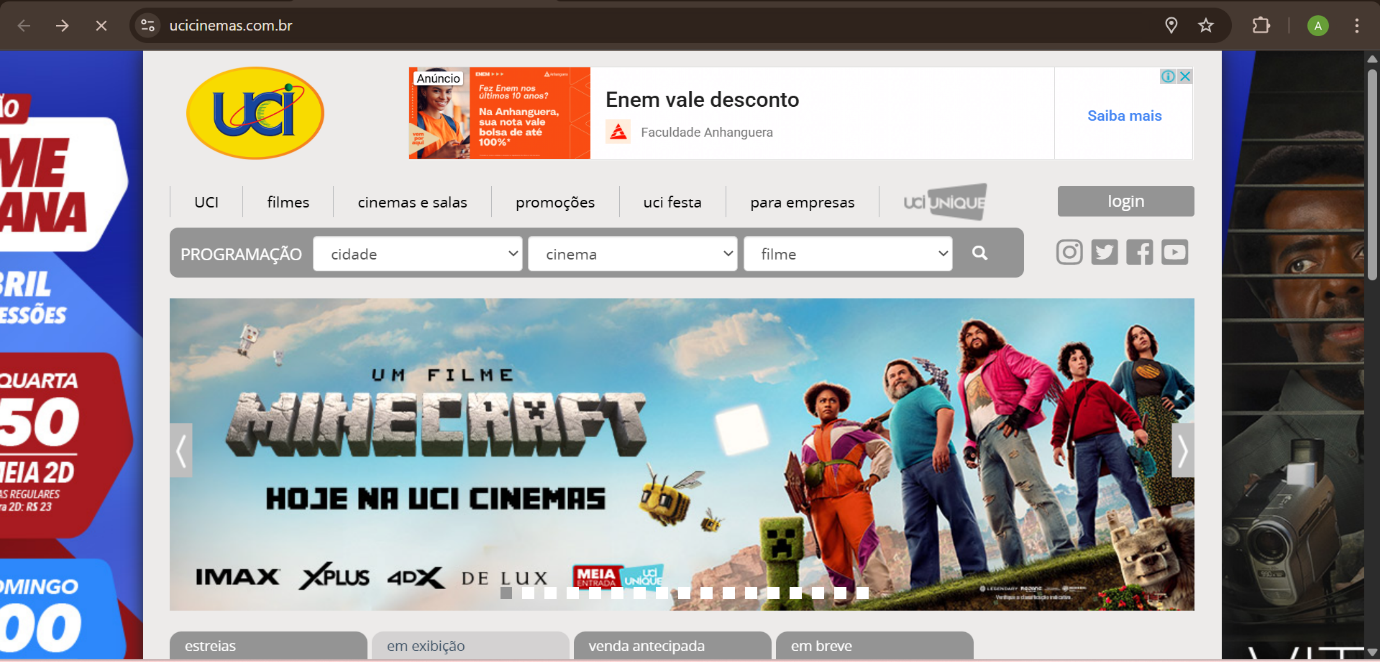
O benchmarking é uma estratégia poderosa utilizada para aumentar a competitividade e promover a melhoria contínua nas organizações. Trata-se de um processo estruturado de comparação entre os produtos, serviços, métodos e estratégias de uma empresa e aqueles adotados por líderes de mercado ou concorrentes diretos. Através dessa avaliação, torna-se possível reconhecer pontos de melhoria, incorporar boas práticas e impulsionar o desempenho da organização de forma mais eficiente.

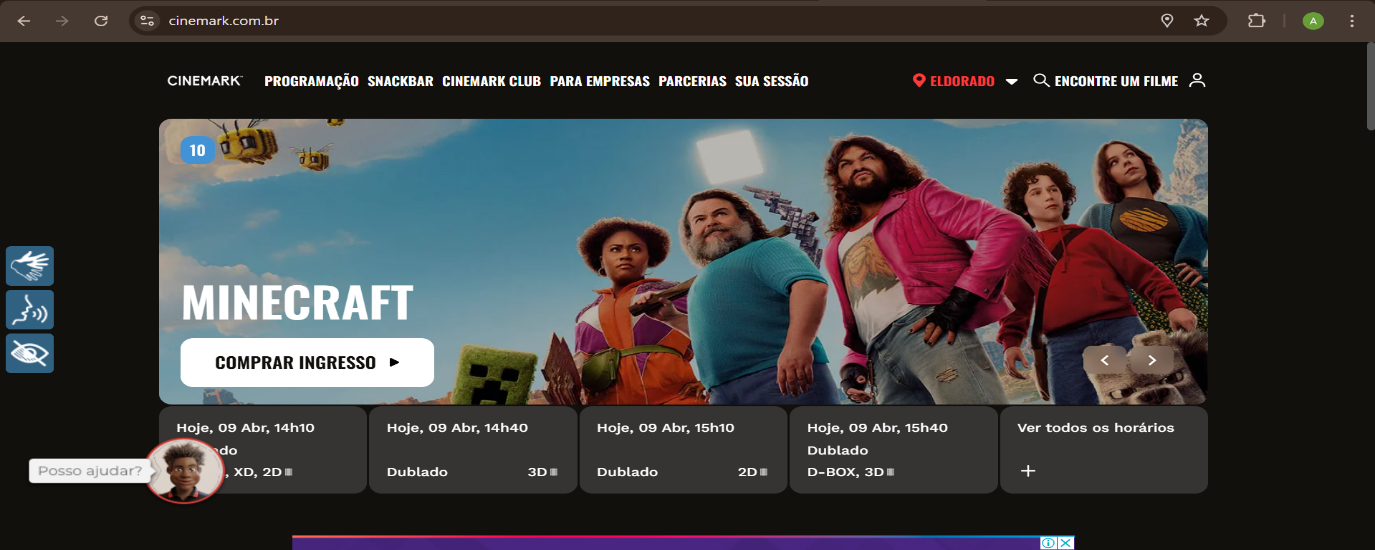
## 1.1 Objetivo

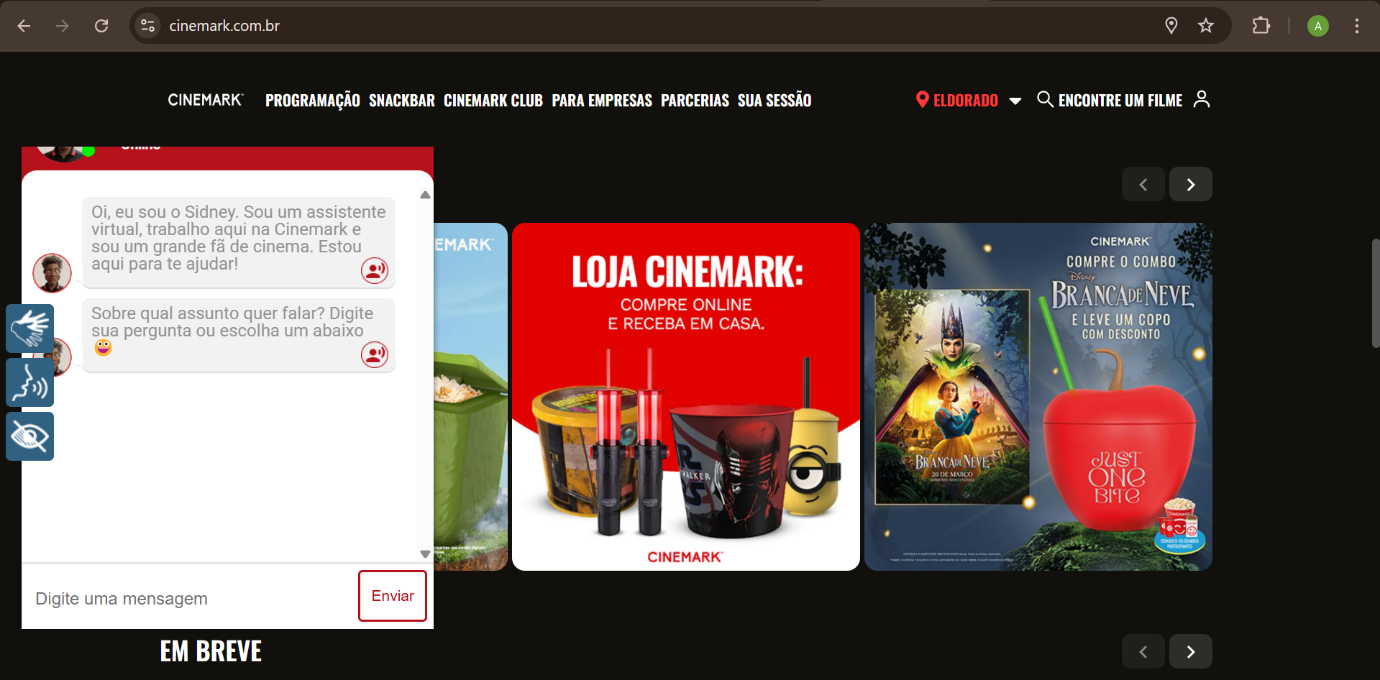
O objetivo principal do benchmarking é identificar boas práticas e implementá-las de maneira adaptada à realidade da empresa, sem simplesmente copiar processos, mas sim aprender e evoluir com base em referências bem-sucedida.

## 1.2 Dados coletados

A partir da análise dos dados obtidos na pesquisa de aplicações de tema similares, foi possível adicionar diversos elementos à aplicação do cinema.







## 1.3 Resultado

Durante o processo, analisamos diferentes redes de cinema para entender melhor como funciona a estrutura de seus sites e como apresentam as informações para o público. Escolhemos as redes UCI, Cinemark e Cinépolis por serem populares e possuírem modelos de navegação distintos. Inserimos fotos dos sites dessas três redes no trabalho para ilustrar as diferenças visuais, de layout e de organização das sessões, horários e filmes. Isso nos ajudou a definir quais elementos são importantes para incluir em nossa própria aplicação de cinema, como a clareza nas informações, o uso de imagens dos filmes e a facilidade de navegação.

Entre os modelos analisados, escolhemos como principal exemplo a aplicação do Cinemark, por ser visualmente mais atrativo e intuito para o usuário. Aplicaremos as características de imagens grandes, design moderno, layout organizado, consulta, seleção rápida de filmes, escolha de horários e assentos, com o objetivo de melhorar a experiência do usuário.

<https://www.ucicinemas.com.br/>

<https://www.cinepolis.com.br/>

<https://www.cinemark.com.br/>

# 2 REQUISITOS DO SISTEMA

Este capítulo descreve os requisitos necessários para o funcionamento adequado do sistema do Cinepop. Os requisitos são divididos em duas categorias principais: requisitos funcionais, que especificam as funcionalidades que o sistema deve prover, e requisitos não funcionais, que estabelecem critérios de qualidade, desempenho e segurança.

## 2.1 Requisitos Funcionais

Os requisitos funcionais definem as operações que o sistema Cinepop deve ser capaz de realizar. Abaixo, estão listados os principais requisitos funcionais identificados na Tabela 1, para o sistema:

|  |  |
| --- | --- |
| Requisito Funcional | |
| RF001 | Cadastrar Cliente |
| RF002 | Atualizar Cliente |
| RF003 | Remover Cliente |
| RF004 | Cadastrar Filme |
| RF005 | Atualizar Filme |
| RF006 | Remover Filme |
| RF007 | Cadastrar Sessão |
| RF008 | Atualizar Sessão |
| RF009 | Remover Sessão |
| RF010 | Cadastrar Sala |
| RF011 | Atualizar Sala |
| RF012 | Remover Sala |
| RF013 | Cadastrar Ingresso |
| RF014 | Atualizar Ingresso |
| RF015 | Remover Ingresso |
| RF016 | Login e Autenticação do Usuário |

Tabela 1 - Requisitos Funcionais

**2.1.1 RF001 – Cadastrar Cliente**

Descrição: O sistema deve permitir que o usuário cadastre novos clientes no sistema de cinema.

Prioridade: Alta

Entrada:

* Nome completo do cliente (obrigatório).
* Email (obrigatório, único).
* Senha (obrigatório).
* CPF (opcional, único).
* Celular (opcional, único).

Processamento:

* O sistema deve validar os campos obrigatórios.
* O sistema deve verificar a unicidade do email, CPF e celular, se preenchidos.

Saída:

* Mensagem de sucesso ao cadastrar o cliente.
* Mensagem de erro em caso de campos obrigatórios não preenchidos ou duplicidade de email/CPF/celular.

Pós-condição:

* O cliente cadastrado deve estar disponível para login e compra de ingressos.

**2.1.2 RF002 – Atualizar Cliente**

Descrição: O sistema deve permitir que o usuário atualize as informações de um cliente existente.

Prioridade: Média

Entrada:

* ID do cliente a ser atualizado (obrigatório).
* Campos a serem editados (nome completo, email, senha, CPF, celular).

Processamento:

* O sistema deve validar o ID do cliente.
* O sistema deve validar os campos a serem atualizados
* O sistema deve verificar a unicidade de email, CPF e celular se forem alterados.

Saída:

* Mensagem de sucesso ao atualizar o cliente.
* Mensagem de erro em caso de cliente não encontrado, campos inválidos ou duplicidade de email/CPF/celular.

Pós-condição:

* As informações do cliente devem ser atualizadas no sistema.

**2.1.3 RF003 – Remover Cliente**

Descrição: O sistema deve permitir que o usuário remova um cliente do sistema.

Prioridade: Média

Entrada:

* ID do cliente a ser removido (obrigatório).

Processamento:

* O sistema deve validar o ID do cliente.
* O sistema deve verificar se o cliente possui ingressos futuros associados que impediriam a remoção.

Saída:

* Mensagem de sucesso ao remover o cliente.
* Mensagem de erro em caso de cliente não encontrado ou impossibilidade de remoção.

Pós-condição:

* O cliente não deve mais estar disponível no sistema.

**2.1.4 RF004 – Cadastrar Filme**

Descrição: O sistema deve permitir que o usuário cadastre novos filmes no catálogo de exibição do cinema.

Prioridade: Alta

Entrada:

* Título do filme (obrigatório).
* Sinopse (obrigatório).
* Duração (obrigatório).
* Classificação etária (obrigatório).
* Gênero (obrigatório).
* Ano de lançamento (obrigatório).
* Poster do filme (obrigatório, URL ou caminho).

Processamento:

* O sistema deve validar os campos obrigatórios e os formatos especificados.

Saída:

* Mensagem de sucesso ao cadastrar o filme.
* Mensagem de erro em caso de campos obrigatórios não preenchidos, formatos inválidos ou duplicidade de cadastro.

Pós-condição:

* O filme cadastrado deve ser exibido na lista de filmes disponíveis para sessões.

**2.1.5 RF005 – Atualizar Filme**

Descrição: O sistema deve permitir que o usuário atualize as informações de um filme existente no catálogo.

Prioridade: Média

Entrada:

* ID do filme a ser atualizado (obrigatório).
* Campos a serem editados (título, sinopse, duração, classificação etária, gênero, ano de lançamento, pôster do filme).

Processamento:

* O sistema deve validar o ID do filme.
* O sistema deve validar os campos a serem atualizados.

Saída:

* Mensagem de sucesso ao atualizar o filme.
* Mensagem de erro em caso de filme não encontrado, campos inválidos ou duplicidade.

Pós-condição:

* As informações do filme devem ser atualizadas no sistema.

**2.1.6 RF006 – Remover Filme**

Descrição: O sistema deve permitir que o usuário remova um filme do catálogo.

Prioridade: Média

Entrada:

* ID do filme a ser removido (obrigatório).

Processamento:

* O sistema deve validar o ID do filme.
* O sistema deve verificar se o filme possui sessões futuras ou ativas associadas que impediriam a remoção.

Saída:

* Mensagem de sucesso ao remover o filme.
* Mensagem de erro em caso de filme não encontrado ou impossibilidade de remoção.

Pós-condição:

* O filme não deve mais estar disponível no catálogo e suas sessões associadas devem ser canceladas ou removidas.

**2.1.7 RF007 – Cadastrar Sessão**

Descrição: O sistema deve permitir que o usuário cadastre novas sessões de filmes, associando-as a um filme e uma sala.

Prioridade: Alta

Entrada:

* ID do filme (obrigatório).
* ID da sala (obrigatório).
* Data e hora de início da sessão (obrigatório).
* Data e hora de fim da sessão (opcional).

Processamento:

* O sistema deve validar os IDs do filme e da sala.
* O sistema deve verificar a disponibilidade da sala para o período da sessão.

Saída:

* Mensagem de sucesso ao cadastrar a sessão.
* Mensagem de erro em caso de IDs inválidos, horários conflitantes ou sala indisponível.

Pós-condição:

* A sessão cadastrada deve estar disponível para venda de ingressos.

**2.1.8 RF008 – Atualizar Sessão**

Descrição: O sistema deve permitir que o usuário atualize as informações de uma sessão existente.

Prioridade: Média

Entrada:

* ID da sessão a ser atualizada (obrigatório).
* Campos a serem editados (ID do filme, ID da sala, data e hora de início, data e hora de fim).

Processamento:

* O sistema deve validar o ID da sessão.
* O sistema deve validar os novos IDs de filme e sala.
* O sistema deve verificar a disponibilidade da sala para o novo período da sessão.
* O sistema deve verificar se a atualização impacta ingressos já vendidos para a sessão.

Saída:

* Mensagem de sucesso ao atualizar a sessão.
* Mensagem de erro em caso de sessão não encontrada, informações inválidas, ou sala indisponível.

Pós-condição:

* As informações da sessão devem ser atualizadas no sistema.

**2.1.9 RF009 – Remover Sessão**

Descrição: O sistema deve permitir que o usuário remova uma sessão.

Prioridade: Média

Entrada:

* ID da sessão a ser removida (obrigatório).

Processamento:

* O sistema deve validar o ID da sessão.
* O sistema deve verificar se a sessão possui ingressos vendidos.

Saída:

* Mensagem de sucesso ao remover a sessão.
* Mensagem de erro em caso de sessão não encontrada ou impossibilidade de remoção.

Pós-condição:

* A sessão não deve mais estar disponível para venda, e os ingressos associados devem ser cancelados.

**2.1.10 RF010 – Cadastrar Sala**

Descrição: O sistema deve permitir que o usuário cadastre novas salas de exibição no cinema.

Prioridade: Alta

Entrada:

* Número da sala (obrigatório, único).
* Tipo da sala (obrigatório, ex: "2D", "3D", "IMAX").
* Número de assentos (obrigatório).
* Fileira (obrigatório, se aplicável, para cálculo de assentos).
* Disponibilidade (opcional, ex: "Disponível", "Em Manutenção").

Processamento:

* O sistema deve validar os campos obrigatórios e seus formatos.
* O sistema deve verificar a unicidade do número da sala.

Saída:

* Mensagem de sucesso ao cadastrar a sala.
* Mensagem de erro em caso de campos obrigatórios não preenchidos, formatos inválidos ou duplicidade de número de sala.

Pós-condição:

* A sala cadastrada deve estar disponível para ser associada a sessões.

**2.1.11 RF011 – Atualizar Sala**

Descrição: O sistema deve permitir que o usuário atualize as informações de uma sala existente.

Prioridade: Média

Entrada:

* ID da sala a ser atualizada (obrigatório).
* Novas informações da sala (número da sala, tipo da sala, número de assentos, fileira, disponibilidade).

Processamento:

* O sistema deve validar o ID da sala.
* O sistema deve validar os campos a serem atualizados (por exemplo, unicidade do número da sala se alterado).
* O sistema deve verificar se a atualização impacta sessões futuras ou ingressos vendidos.

Saída:

* Mensagem de sucesso ao atualizar a sala.
* Mensagem de erro em caso de sala não encontrada, campos inválidos ou duplicidade.

Pós-condição:

* As informações da sala devem ser atualizadas no sistema.

**2.1.12 RF012 – Remover Sala**

Descrição: O sistema deve permitir que o usuário remova uma sala do cinema.

Prioridade: Média

Entrada:

* ID da sala a ser removida (obrigatório).

Processamento:

* O sistema deve validar o ID da sala.
* O sistema deve verificar se a sala possui sessões futuras ou ativas associadas que impediriam a remoção.

Saída:

* Mensagem de sucesso ao remover a sala.
* Mensagem de erro em caso de sala não encontrada ou impossibilidade de remoção.

Pós-condição:

* A sala não deve mais estar disponível no sistema e suas sessões associadas devem ser canceladas.

**2.1.13 RF013 – Cadastrar Ingresso**

Descrição: O sistema deve permitir que o usuário cadastre um novo ingresso para uma sessão, associando-o a um cliente.

Prioridade: Alta

Entrada:

* ID da sessão (obrigatório).
* ID do cliente (obrigatório).
* Status do ingresso (obrigatório, ex: "Reservado", "Vendido", "Disponível").
* Preço do ingresso (obrigatório).

Processamento:

* O sistema deve validar os IDs da sessão e do cliente.
* O sistema deve verificar a disponibilidade de assentos na sessão.
* O sistema deve validar o preço.

Saída:

* Mensagem de sucesso ao cadastrar o ingresso.
* Mensagem de erro em caso de IDs inválidos, sessão lotada ou preço inválido.

Pós-condição:

* O ingresso deve ser gerado e associado à sessão e, se aplicável, ao cliente.

**2.1.14 RF014 – Atualizar Ingresso**

Descrição: O sistema deve permitir que o usuário atualize as informações de um ingresso existente.

Prioridade: Média

Entrada:

* ID do ingresso a ser atualizado (obrigatório).
* Novas informações do ingresso (ID da sessão, ID do cliente, status do ingresso, preço do ingresso).

Processamento:

* O sistema deve validar o ID do ingresso.
* O sistema deve validar os campos a serem atualizados (ex: mudança de status).

Saída:

* Mensagem de sucesso ao atualizar o ingresso.
* Mensagem de erro em caso de ingresso não encontrado, campos inválidos ou inconsistências.

Pós-condição:

* As informações do ingresso devem ser atualizadas no sistema.

**2.1.15 RF015 – Remover Ingresso**

Descrição: O sistema deve permitir que o usuário remova um ingresso, geralmente em caso de cancelamento.

Prioridade: Média

Entrada:

* ID do ingresso a ser removido (obrigatório).

Processamento:

* O sistema deve validar o ID do ingresso.
* O sistema deve liberar o assento na sessão.

Saída:

* Mensagem de sucesso ao remover o ingresso.
* Mensagem de erro em caso de ingresso não encontrado ou impossibilidade de remoção.

Pós-condição:

* O ingresso não deve mais ser válido e o assento associado deve ser liberado.

**2.1.16 RF016 – Login e Autenticação do Usuário**  
Descrição: O sistema deve permitir que o usuário realize login utilizando suas credenciais, garantindo a autenticação para acesso às funcionalidades restritas.  
Prioridade: Alta

Entrada:

* Email do cliente (obrigatório).
* Senha do cliente (obrigatório).

Processamento:

* O sistema deve validar se os campos obrigatórios foram preenchidos.
* O sistema deve verificar se o email existe no cadastro de clientes.
* O sistema deve validar se a senha informada corresponde à senha cadastrada para o cliente.
* O sistema deve gerar e retornar um token de autenticação (ex: JWT) em caso de sucesso.
* O sistema deve armazenar a sessão do usuário autenticado (quando aplicável).

Saída:

* Mensagem de sucesso ao autenticar o usuário.
* Token de autenticação válido.
* Mensagem de erro em caso de email não cadastrado, senha incorreta ou campos obrigatórios não preenchidos.

Pós-condição:

* O usuário autenticado deve ter acesso às funcionalidades restritas do sistema (ex: compra de ingressos, atualização de dados).
* O token de autenticação deve ser utilizado para validar requisições posteriores do usuário durante a sessão.

## ****2.2 Requisitos Não Funcionais****

Os requisitos não funcionais definem os atributos de qualidade que o sistema Cinepop deve possuir. Abaixo, estão listados os principais requisitos não funcionais identificados na Tabela 2, para o sistema:

|  |  |
| --- | --- |
| Requisito Não Funcional | |
| RNF001 | Escolha de Assento na Compra |
| RNF002 | Seleção de Sala e Sessão |
| RNF003 | Tempo de Resposta |
| RNF004 | Capacidade de Usuários Concorrentes |
| RNF005 | Disponibilidade do Sistema |

Tabela 2 - Requisitos não funcionais

**2.2.1 RNF001 – Escolha de Assento na Compra**

Descrição: O sistema deve permitir que o usuário visualize e selecione graficamente os assentos desejados na sala de cinema de forma intuitiva e clara durante o processo de compra de ingressos.

Prioridade: Alta

Entrada:

* Interface visual da sala (mapa de assentos).
* Status dos assentos (disponível, ocupado, selecionado).
* Cliques e interações do usuário no mapa de assentos.

Processamento:

* Carregamento dinâmico do layout da sala e status dos assentos.
* Marcação visual clara dos assentos selecionados e ocupados.
* Validação de assentos selecionados (não permitir seleção de assentos já ocupados).
* Atualização em tempo real da disponibilidade dos assentos para evitar conflitos.

Saída:

* Mapa de assentos exibido com clareza e distinção entre status.
* Seleção de assentos intuitiva e sem erros de sobreposição.
* Mensagem de erro para assento já ocupado.

Pós-condição:

* Usuários conseguem escolher seus assentos de forma eficiente e sem confusão.
* Redução de frustração e erros durante o processo de compra de ingressos.

**2.2.2 RNF002 – Seleção de Sala e Sessão**

Descrição: O sistema deve permitir que o usuário selecione a sala e a sessão desejada de forma fácil e clara, com informações visíveis sobre o filme, horário, tipo de sala e assentos disponíveis.

Prioridade: Alta

Entrada:

* Lista de filmes em exibição.
* Datas de exibição disponíveis.
* Horários de sessão para cada filme/data.
* Informações de sala e disponibilidade de assentos.

Processamento:

* Exibição organizada de filmes, datas e sessões.
* Filtros e opções de busca para facilitar a localização.
* Apresentação clara de detalhes da sessão.
* Redirecionamento suave para a próxima etapa da compra após a seleção.

Saída:

* Tela de seleção de filmes/sessões de fácil navegação e compreensão.
* Informações cruciais da sessão visíveis e atualizadas.
* Navegação direta para a tela de escolha de assentos após a seleção.

Pós-condição:

* Usuários encontram e selecionam a sessão desejada rapidamente.
* Experiência de usuário fluida desde a escolha do filme até a sessão.

**2.2.3 RNF003 – Tempo de Resposta**

Descrição: O sistema deve responder às principais operações (pesquisa de filmes/sessões, carregamento de mapa de assentos, finalização de compra) em um tempo máximo de 3 segundos para 95% das requisições, mesmo em picos de acesso.

Prioridade: Alta

Entrada:

* Requisições de pesquisa de filmes/sessões.
* Requisições de carregamento de mapa de assentos.
* Requisições de finalização de compra.
* Número de usuários simultâneos no sistema.

Processamento:

* Monitoramento contínuo do tempo de resposta da aplicação e banco de dados.
* Otimização de consultas, algoritmos e uso de cache para operações frequentes.
* Testes de carga simulando cenários de alto tráfego.
* Análise e otimização de infraestrutura (servidores, rede).

Saída:

* Tempo médio de resposta para pesquisa de filmes/sessões inferior a 1 segundo.
* Tempo de carregamento de mapa de assentos inferior a 2 segundos.
* Tempo de finalização de compra inferior a 3 segundos.
* Relatórios de desempenho mostrando que 95% das requisições estão dentro do limite de 3 segundos.

Pós-condição:

* Usuários experienciam um sistema rápido e responsivo, sem atrasos perceptíveis.
* Redução de abandono de carrinho de compra devido a lentidão.

**2.2.4 RNF004 – Capacidade de Usuários Concorrentes**

Descrição: O sistema deve ser capaz de suportar 500 usuários simultâneos realizando operações de consulta (pesquisa, visualização de sessões) e 100 usuários simultâneos realizando operações de compra de ingressos, sem degradação perceptível no tempo de resposta ou funcionalidade.

Prioridade: Alta

Entrada:

* Simulações de carga com 500 usuários de consulta e 100 usuários de compra.
* Métricas de utilização de CPU, memória, I/O de disco e conexões de rede dos servidores.
* Tempo de resposta médio e percentual de erros sob carga.

Processamento:

* Execução de testes de carga e estresse com perfis de usuário concorrentes.
* Otimização da arquitetura do sistema para escalabilidade horizontal (adicionar mais servidores).
* Configuração de pools de conexão de banco de dados e balanceadores de carga.
* Otimização de transações críticas para garantir atomicidade e consistência sob concorrência (ex: seleção de assentos).

Saída:

* Tempos de resposta dentro dos limites estabelecidos (RNF003) sob a carga especificada.
* Utilização de recursos do servidor (CPU, memória) abaixo de 80% sob carga máxima.
* Taxa de erros de transação (ex: compra de assento duplo) próxima de zero.

Pós-condição:

* O sistema permanece estável e responsivo durante picos de demanda.
* A experiência de compra do usuário não é comprometida pela quantidade de acessos.

**2.2.5 RNF005 – Disponibilidade do Sistema**

Descrição: O sistema deve estar disponível para compra de ingressos e consulta de informações 99,9% do tempo, excluindo janelas de manutenção previamente agendadas e comunicadas.

Prioridade: Altíssima

Entrada:

* Logs de tempo de atividade (uptime) dos servidores de aplicação e banco de dados.
* Registros de interrupções de serviço e seus tempos de duração.
* Relatórios de falhas inesperadas do sistema.

Processamento:

* Implementação de arquitetura de alta disponibilidade (redundância, clusters, failover).
* Monitoramento proativo 24/7 da infraestrutura e aplicação.
* Execução de backups regulares e testados do banco de dados (diários, diferenciais).
* Manutenção de um plano de recuperação de desastres (DRP) testado e atualizado.
* Aplicação de atualizações e patches com estratégia de zero downtime ou downtime mínimo.

Saída:

* Uptime mensal registrado de no mínimo 99,9%.
* Tempo máximo de inatividade não planejado de X minutos por mês.
* Notificação automática da equipe de TI em caso de falha.

Pós-condição:

* O sistema está consistentemente acessível aos usuários para compra e consulta.
* A perda de receita devido à indisponibilidade do sistema é minimizada.
* A confiança do cliente no serviço online do cinema é mantida.

# 3 MODELAGEM

A modelagem de software é a prática usada por desenvolvedores de sistema para criar representações de um sistema, necessário para sua criação, desenvolvimento e manutenção. Ela serve como a planta de uma construção, guiando todas as etapas de forma alinhada.

## 3.1 Diagrama de Classe

Os diagramas de classes representado na Figura 1, são fundamentais pois representam visualmente a estrutura do sistema, mostrando classes, atributos, métodos e relacionamentos. Eles facilitam a compreensão do design do sistema, ajudam no planejamento e na organização do código.

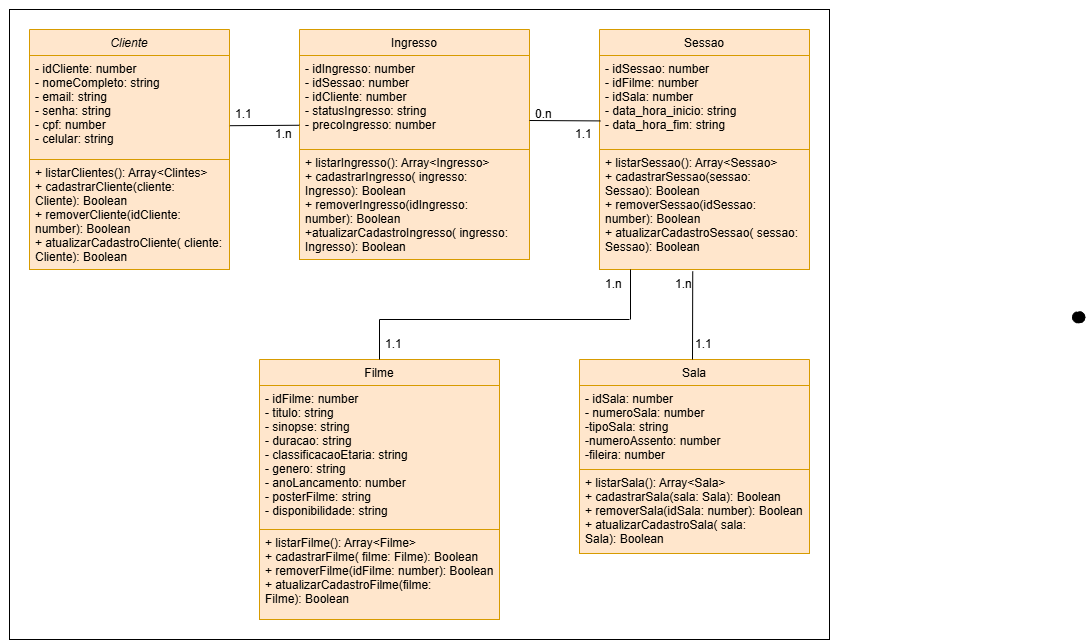


Figura - Diagrama de Classe

## 3.2 Diagrama de Entidade Relacionamento (DER)

O Diagrama Entidade-Relacionamento (DER), representado na Figura 2, é fundamental pois visualiza a estrutura do banco de dados, mostrando entidades, atributos e relacionamentos. Ele facilita a compreensão do design dos dados, ajudando no planejamento e na organização da persistência.

Enquanto o DER modela a estrutura do banco de dados (como os dados são armazenados e relacionados), o Diagrama de Classe (Capítulo 3.1) modela a estrutura do software (classes, atributos, métodos e suas interações na lógica da aplicação). Ambos são complementares para o desenvolvimento do sistema.

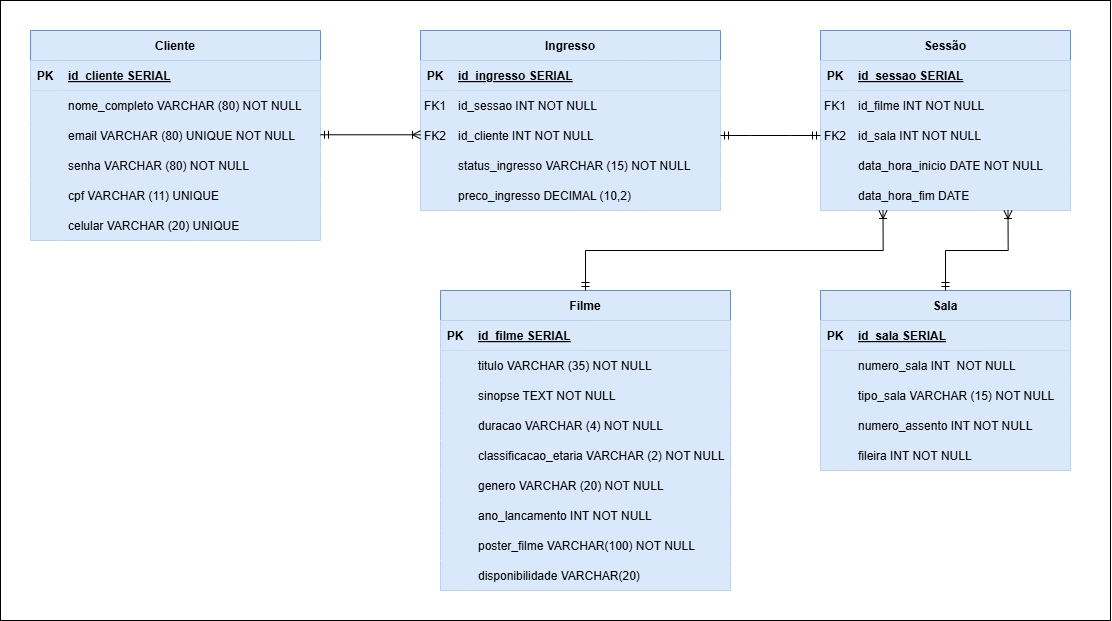


Figura - Diagrama Entidade Relacionamento (DER)

# 4. Dicionário API

O dicionário de API ele serve para mostrar e explicar como funciona a API, que mostra todos os endpoints (caminhos), os dados que ela recebe e devolve, os métodos usados ( como GET, POST, PUT, DELETE).

Serve como uma **guia de referência** para todos os desenvolvedores entenderem como se comunicar com a API. Quais são as informações podem ser enviadas, e quais respostas esperar e as regras a seguir.

## 4.1. Dicionário de Endpoints

Um dicionário de Endpoints é um documento que pode listar e descrever todas as rotas (endpoints) de uma aplicação ou API.

Ele mostra quais caminhos que vão estar disponíveis, para que servem, quais métodos HTTP usam (como GET, POST, PUT, DELETE) e quais informações precisam ser enviadas ou serão retornadas.

|  |  |  |
| --- | --- | --- |
| Endpoint | Tipo de Requisição | Parâmetros |
| / | GET | Nenhum |
| /api-docs | GET | Nenhum |
| /filmes | GET | Nenhum |
| /filmes | POST | nome, descricao, duracao, classificacao |
| /filmes | DELETE | idFilme |
| /filmes | PUT | idFilme, nome, descricao, duracao, classificacao |
| /salas | GET | Nenhum |
| /salas | POST | numeroSala, tipoSala, numeroAssento, fileira |
| /salas | DELETE | idSala |
| /salas | PUT | idSala, numeroSala, tipoSala, numeroAssento, fileira |
| /sessoes | GET | Nenhum |
| /sessoes | POST | idFilme, idSala, dataHora, idioma, tipoExibicao |
| /sessoes | DELETE | idSessao |
| /sessoes | PUT | idSessao, idFilme, idSala, dataHora, idioma, tipoExibicao |
| /clientes | GET | Nenhum |
| /clientes | POST | nome, email, senha |
| /clientes | DELETE | idCliente |
| /clientes | PUT | idCliente, nome, email, senha |
| /login | POST | email, senha |
| /rota-protegida | GET | Token JWT válido |
| /ingressos | GET | Nenhum |
| /ingressos | POST | idSessao, idCliente, statusIngresso, precoIngresso |
| /ingressos | DELETE | idIngresso |
| /ingressos | PUT | idIngresso, idSessao, idCliente, statusIngresso, precoIngresso |

## 4.2 Dicionário de Dados

Um dicionário de dados é um documento ou tabela que reúne e descreve todas as informações usadas em um Sistema ou banco de dados.

Ele serve para organizer e pradonizar os dados , explicando o que cada campo significa, qual seu tipo (texto, número, data, etc.), tamanho, formato e onde ele é usado.

|  |  |  |  |
| --- | --- | --- | --- |
| Nome do Campo na API | Tipo | Descrição | Regra |
| idFilme | int | Identificador único do filme | Obrigatório para atualização/remoção |
| nome | string | Nome do filme | Obrigatório |
| descricao | string | Descrição do filme | Opcional |
| duracao | int | Duração em minutos | Obrigatório |
| classificacao | string | Classificação indicativa | Obrigatório |
| idSala | int | Identificador único da sala | Obrigatório para atualização/remoção |
| numeroSala | int | Número da sala | Obrigatório |
| tipoSala | string | Tipo da sala (2D, 3D, IMAX) | Obrigatório |
| numeroAssento | int | Quantidade de assentos | Obrigatório |
| fileira | string | Quantidade de fileiras | Obrigatório |
| idSessao | int | Identificador da sessão | Obrigatório para atualização/remoção |
| idFilme | int | Filme exibido | Obrigatório |
| idSala | int | Sala da exibição | Obrigatório |
| dataHora | datetime | Data e hora da sessão | Obrigatório |
| idioma | string | Idioma exibido | Obrigatório |
| tipoExibicao | string | Tipo (legendado, dublado) | Obrigatório |
| idCliente | int | Identificador do cliente | Obrigatório para atualização/remoção |
| nome | string | Nome do cliente | Obrigatório |
| email | string | Email do cliente | Obrigatório e único |
| senha | string | Senha do cliente | Obrigatório |
| idIngresso | int | Identificador do ingresso | Obrigatório para atualização/remoção |
| idSessao | int | Sessão escolhida | Obrigatório |
| idCliente | int | Cliente comprador | Obrigatório |
| statusIngresso | string | Status (ativo, cancelado) | Obrigatório |
| precoIngresso | decimal | Preço do ingresso | Obrigatório |