## **Documento de Arquitetura de Software - Sistema Web de Cinema com Seleção de Cadeiras**

### **1. Introdução**

#### **1.1 Propósito**

Este documento descreve a arquitetura do software do Sistema Web de Cinema com Seleção de Cadeiras, detalhando sua estrutura modular, camadas e tecnologias utilizadas para permitir a visualização de filmes e sessões, reserva de cadeiras online, atendimento no balcão e gestão administrativa.

#### **1.2 Escopo**

O sistema será baseado em uma arquitetura web moderna e responsiva, permitindo que clientes visualizem filmes e sessões, selecionem e reservem cadeiras, e efetuem pagamentos online. Funcionários poderão realizar reservas e registrar pagamentos no balcão, enquanto administradores poderão gerenciar filmes, sessões, cadeiras e visualizar relatórios de vendas e ocupação.

#### **1.3 Definições e Acrônimos**

* **CRUD:** Create, Read, Update, Delete
* **JWT:** JSON Web Token (para autenticação)
* **REST:** Representational State Transfer
* **API RESTful:** Interface de programação de aplicações que segue os princípios REST.
* **Cliente:** Usuário que visualiza filmes, sessões e reserva cadeiras.
* **Funcionário:** Usuário que realiza reservas e registra pagamentos no balcão.
* **Administrador:** Usuário responsável pelo gerenciamento de filmes, sessões, pagamentos e funcionários.
* **Sessão:** Horário e local específico de exibição de um filme.
* **Cadeira:** Assento individual em uma sessão.
* **Ingresso Digital:** Comprovante da reserva de uma cadeira para uma sessão.

#### **1.4 Referências**

Padrões de arquitetura RESTful, princípios SOLID, práticas ágeis de desenvolvimento e guidelines de UX/UI modernas.

### **2. Arquitetura do Sistema**

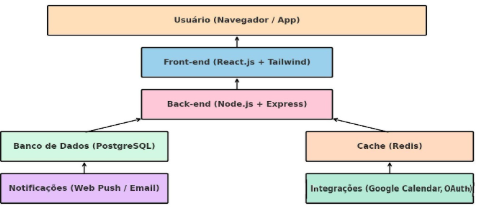
#### **2.1 Modelo Arquitetural**

O sistema será construído com base em uma arquitetura modular utilizando o padrão MVC (Model-View-Controller) e componentes desacoplados, visando alta coesão e baixo acoplamento.

#### **2.2 Camadas do Sistema**

* **Front-end:** Interface do usuário (React.js com Vite)
* **Back-end:** Lógica de negócios e APIs (Node.js com Express, autenticação JWT)
* **Banco de Dados:** MySQL para dados persistentes
* **Serviços Externos:** Integração com APIs de pagamento (Pix, operadoras de cartão).

#### **2.3 Diagrama de Arquitetura**



### **3. Módulos e Responsabilidades**

#### **3.1 Módulo de Usuários**

Gerencia a autenticação, cadastro e perfis dos diferentes tipos de usuários:

* **Cliente:** Login e gerenciamento de perfil.
* **Funcionário:** Login e acesso às funcionalidades de balcão.
* **Administrador:** Login e acesso ao painel de gerenciamento completo.

#### **3.2 Módulo de Filmes**

Responsável pelo CRUD (Create, Read, Update, Delete) de filmes. Administradores podem adicionar novos filmes, editar informações existentes e remover filmes do catálogo.

#### **3.3 Módulo de Sessões e Cadeiras**

Gerencia a criação, edição e exclusão de sessões de filmes. Inclui a funcionalidade de geração automática de cadeiras por sessão e o controle de status de cada cadeira (disponível, reservada, ocupada).

#### **3.4 Módulo de Reservas e Ingressos**

Permite que clientes selecionem cadeiras e realizem reservas, gerando um ingresso digital único para cada compra. Funcionários podem realizar reservas diretamente no balcão.

#### **3.5 Módulo de Pagamentos**

Gerencia o processamento de pagamentos. Suporta pagamentos online (Pix e cartão) para clientes e registro de pagamentos no balcão (dinheiro, cartão e Pix) para funcionários.

#### **3.6 Módulo de Notificações**

Envia alertas automáticos para clientes sobre a confirmação de reserva e detalhes da sessão.

#### **3.7 Módulo de Relatórios**

Gera métricas e estatísticas sobre o sistema, incluindo:

* Taxa de ocupação por sessão
* Receita por filme
* Tipos de pagamento utilizados

### **4. Tecnologias Utilizadas**

#### **4.1 Front-end**

* **Framework:** React.js (com Vite para build tool)
* **Estilização:** CSS (podendo utilizar um framework como Bootstrap, se necessário)

#### **4.2 Back-end**

* **Runtime:** Node.js
* **Framework:** Express
* **Autenticação:** JWT (JSON Web Token)

#### **4.3 Banco de Dados**

* **SGBD:** MySQL

#### **4.4 Integrações**

* **APIs de Pagamento:** Integração com plataformas de Pix e operadoras de cartão para processamento de transações.

### **5. Fluxo de Comunicação**

#### **5.1 Comunicação entre Módulos**

A comunicação entre o frontend e o backend, e entre os diferentes módulos lógicos do backend, ocorre via API RESTful, utilizando JSON sobre HTTP para troca de dados.

#### **5.2 Comunicação com Serviços Externos**

As integrações com serviços externos, como as plataformas de pagamento, serão realizadas através de APIs específicas e seguras, utilizando métodos de autenticação apropriados.

### **6. Considerações Finais**

#### **6.1 Riscos e Desafios**

* **Gerenciamento de concorrência:** Garantir a integridade da seleção de cadeiras em acessos simultâneos.
* **Sincronização em tempo real:** Atualização imediata do status das cadeiras para evitar overbooking.
* **Segurança de dados de usuários e pagamentos:** Proteção contra acessos não autorizados e conformidade com a LGPD.
* **Escalabilidade:** Suportar um grande volume de usuários e sessões sem perda de desempenho.

#### **6.2 Expansões Futuras**

* Implementação de um dashboard com gráficos interativos para administradores.
* Funcionalidade de cancelamento de reservas com política de reembolso.
* Integração com sistemas de fidelidade ou programas de pontos do cinema.
* Disponibilização de aplicativo móvel nativo.