|  |
| --- |
| Documento de especificação dos componentes do produto |

Modelo de exemplo preenchido: Criação de um site online de venda de ingressos

1. **PROJETO DA ARQUITETURA DO SISTEMA**

1.1 Requisitos do sistema:

1. Identificar eventos disponíveis
2. Seleção de ingressos por categoria e quantidade
3. Integração com meios de pagamento online
4. Emissão de confirmações de compra
5. Sistema de gerenciamento de usuários (login) e históricos de compra

1.2 Alocação de requisitos:

1. Requisitos relacionados à interface do usuário serão alocados ao front-end
2. Requisitos de processamento de pagamento serão alocados para o desenvolvimento back-end e integração com API de pagamento

1.3 Design da arquitetura:

1.3.1 Arquitetura Front-end

1. Desenvolvimento em React.js
2. Implementação de páginas de visualização de eventos, seleção de ingresso e finalização de compra

1.3.2 Arquitetura do Back-end

1. Desenvolvimento em Node.Js utilizando Express
2. Integração com banco de dados para armazenar informações de eventos, usuários e transações

1.3.3 Arquitetura do Banco de Dados

1. Utilização do MySQL na versão 8.0.35

1.3.4 Arquitetura de Segurança

1. Utilização de HTTPS
2. Autenticação em 2 fatores para usuários que façam login de máquinas distintas
3. **CONSTRUÇÃO DO PRODUTO**

2.1 Especificação dos componentes

2.1.1 Arquitetura do Front-end

|  |  |  |  |
| --- | --- | --- | --- |
| Código do componente | Descrição | Funcionalidades | Como / Por meio de |
| [C001] – Visualização de eventos | - Exibe uma lista de shows disponíveis para compra de ingressos  - Permite filtrar evento por categoria, data e localização | - Exibe uma lista de shows disponíveis para compra de ingressos  - Permite filtrar evento por categoria, data e localização | - GUI para exibir os eventos |
| [C002] – Seleção de ingressos | - Permite aos usuários selecionar a quantidade e categoria de ingressos desejada. | - Exibe opções de ingressos disponíveis para o evento selecionado.  - Permite ao usuário escolher a quantidade e categoria de ingressos desejada. | - GUI para seleção de ingressos |

2.1.2 Arquitetura do Back-end

|  |  |  |  |
| --- | --- | --- | --- |
| Código do componente | Descrição | Funcionalidades | Como / Por meio de |
| [C003] – Sistema de exibição de eventos | - Fornece endpoints para recuperar informações sobre shows | - Lida com solicitações para recuperar lista de eventos, detalhes, etc  - Integração com o banco de dados para obter as informações atualizadas | - API RESTful com endpoints para manipulação de eventos |
| [C004] – Sistema de pagamento | - Gerencia o processamento de pagamentos para as transações de ingressos | - Integração com gateways de pagamento online.  - Validação de transações e atualização do status de compra. | - API para processamento de pagamentos com comunicação segura com provedores externos |

2.1.3 Arquitetura do Banco de Dados

|  |  |  |  |
| --- | --- | --- | --- |
| Código do componente | Descrição | Funcionalidades | Como / Por meio de |
| [C005] – Banco de dados de eventos | - Armazena informações relacionadas aos eventos, como detalhes do evento e disponibilidade de ingressos. | - Gerenciamento de tabelas para eventos e categorias de ingressos.  - Suporte a operações CRUD | - ORM (Object-Relational Mapping) para interação com o banco de dados. |
| [C006] – Banco de dados de usuário | - Armazena informações sobre os usuários registrados e seus históricos de compra | - Gerenciamento de tabelas para usuários e históricos de compra  - Suporte a operações CRUD | - ORM (Object-Relational Mapping) para interação com o banco de dados. |

2.1.4 Arquitetura de Segurança

|  |  |  |  |
| --- | --- | --- | --- |
| Código do componente | Descrição | Funcionalidades | Como / Por meio de |
| [C007] – Autenticação de usuários | - Gerencia a autenticação de usuários durante o processo de compra e login | - Verificação de credenciais de usuários.  - Geração de tokens de autenticação. | - Middleware de autenticação nas rotas relevantes |
| [C008] – Segurança da comunicação | - Garante a segurança nas comunicações entre o cliente e o servidor. | - Implementação de HTTPS para criptografar dados sensíveis.  - Proteção contra ataques comuns, como CSRF (Cross-Site Request Forgery) | - Configuração do servidor para suportar HTTPS |

2.2 Orientações de documentação para a construção dos componentes

1. Diagrama de classes e/ou diagrama de fluxo de processos para cada componente
2. Definir ciclo de vida do produto