# **Documento de Arquitetura de Software: MyMesa**

## **1. Introdução**

Este documento descreve a arquitetura de software para um aplicativo de gerenciamento de mesa, que permite aos clientes fazerem pedidos, aos garçons receberem e enviarem pedidos à cozinha, e à cozinha preparar os pedidos. O sistema também inclui um sistema de login para acesso seguro.

## **2. Visão Geral da Arquitetura**

O aplicativo de gerenciamento de mesa será desenvolvido utilizando uma arquitetura de cliente-servidor, com os seguintes componentes principais:

* Cliente: Interface de usuário para clientes e garçons.
* Servidor de Aplicação: Lógica de negócios e gerenciamento de pedidos.
* Servidor de Banco de Dados: Armazenamento de dados do sistema.
* Módulo de Autenticação: Sistema de login para autenticação de usuários.

## **3. Componentes do Sistema**

### **3.1 Cliente**

* Interface do Cliente: Interface de usuário para clientes acessarem o cardápio e fazerem pedidos.
* Interface do Garçom: Interface separada para garçons receberem pedidos e interagirem com o sistema/cozinha.

### **3.2 Servidor de Aplicação**

* Lógica de Negócios: Gestão de pedidos, processamento de pagamentos, e gestão de mesas.
* API REST: Interface de programação de aplicação para comunicação com clientes e garçons.
* Gestor de Pedidos: Roteamento de pedidos entre garçons e cozinha.
* Notificações em Tempo Real: Mecanismo para notificar garçons e clientes sobre atualizações de pedidos.

### **3.3 Servidor de Banco de Dados**

* Base de Dados: Armazenamento de informações sobre cardápio, pedidos, clientes e mesas.
* Banco de Dados: MySQL WorkBench

### **3.4 Módulo de Autenticação**

* Servidor de Autenticação: Lida com a autenticação de usuários e geração de tokens de acesso.
* Armazenamento de Credenciais: Banco de dados para armazenamento seguro de credenciais de usuários.

## **4. Tecnologias Utilizadas**

* Linguagem de Programação: JavaScript, HTML, CSS.
* Banco de Dados: MySQL WorkBench

## **5. Considerações de Segurança**

* Criptografia de Dados: Criptografia para o não vazamento dos dados dos usuários.
* Autenticação e Autorização: Implementação robusta de autenticação e autorização para garantir acesso seguro aos dados.

## **6. Considerações de Manutenção e Extensibilidade**

* Documentação Adequada: Manter documentação atualizada para facilitar o entendimento e a manutenção do sistema por parte dos desenvolvedores.
* Testes Automatizados: Implementação de testes automatizados para garantir a estabilidade do sistema durante alterações e adições de funcionalidades.

## **7. Diagrama de Arquitetura**

BPMN.IO - (No Github)

## **8. Padrão de Codificação**

* Comentários utilizando “/\* \*/”;
* Indentação utilizando ‘TAB’;
* Variáveis bem definidas em português;
* Usar o padrão camelCase;
* Quebra de linha com 80 caracteres

## **9. Infraestrutura**

Servidor Web, Banco de dados.

## **10. Conclusão**

Este documento fornece uma visão geral da arquitetura de software para o aplicativo de gerenciamento de mesa. Detalhes adicionais sobre a implementação e funcionamento do sistema serão discutidos durante o desenvolvimento.