

## Requisitos Gerais (para todos os sistemas)

O sistema será Web, utilizando C# ASP.NET Core no back-end.

O banco de dados será SQL Server, com acesso via Entity Framework Core e ADO.NET (em consultas mais complexas).

Haverá uma API central para comunicação entre os módulos.

O front-end será desenvolvido com Bootstrap e Ajax para simular atualizações em tempo real.

O sistema deverá permitir autenticação de usuários por perfil (garçom, gerente, caixa, etc.).

O design deve ser responsivo para rodar bem em tablet ou celular.

Logs de operação básicos (ex.: pedidos, alterações de cardápio, fechamento de conta).

## Sistema Garçom – Geração de Pedidos

O garçom deve poder abrir uma mesa nova ou vincular pedido a uma mesa existente.

Permitir adicionar itens ao pedido com observações (ex.: “sem cebola”).

Visualizar status dos pedidos (ex.: “na cozinha”, “pronto para entrega”).

Enviar pedidos para a cozinha em tempo real via API.

Possibilidade de dividir a conta por cliente/mesa.

## Sistema Cozinha – Visualização de Pedidos

Mostrar todos os pedidos enviados pelos garçons em ordem cronológica.

Permitir marcar itens como “em preparo” e depois como “pronto”.

Agrupar pedidos por mesa ou por item (ex.: “5x pizza calabresa”).

Atualização automática via Ajax.

Histórico básico dos últimos pedidos finalizados.

## Sistema Caixa – Gestão de Pedidos e Pagamentos

Visualizar todos os pedidos em aberto.

Fechar pedidos: registrar forma de pagamento (dinheiro, cartão, pix).

Emitir comprovante de pagamento (impressão ou PDF).

Possibilidade de aplicar descontos autorizados pelo gerente.

Consultar histórico de vendas (diário, semanal, mensal).

## Sistema Menu – Visualização do Cardápio

Exibir o cardápio do dia para os clientes.

Permitir consulta de valores, ingredientes e tempo médio de preparo.

Atualização em tempo real caso o gerente altere o cardápio.

Sistema público, sem necessidade de login.

## Sistema Gerente – Gestão do Cardápio e Operações

Inserir, editar e remover pratos do cardápio.

Alterar status de pratos (ex.: “indisponível”).

Definir preços e promoções.

Acompanhar relatórios de vendas e consumo.

Consultar histórico de alterações no cardápio.

## Requisitos de Log (Auditoria)

O sistema deve registrar automaticamente a criação/abertura de uma mesa.

O sistema deve registrar cada pedido feito na mesa, incluindo:

- Item(s) pedidos.
- Quantidade.
- Observações do pedido.

O sistema deve registrar alterações nos pedidos da mesa (ex.: cancelamento, troca de item).

O sistema deve registrar o fechamento da mesa (conta paga, forma de pagamento utilizada).

O sistema deve permitir a consulta de logs por mesa em um relatório detalhado.

O sistema deve armazenar os logs de forma que não possam ser editados pelos usuários comuns (apenas leitura).

O sistema deve permitir filtros básicos na consulta de logs:

Data/hora.

Número da mesa.

Tipo de ação (abertura, pedido, cancelamento, fechamento).

## Requisitos Não Funcionais

Sistema deve suportar pelo menos 20 usuários simultâneos sem perda de performance.

Comunicação entre sistemas via API REST.

Segurança básica: dados de clientes e senhas devem ser armazenados com hash e salt.

Camada de logs de erro e exceções para monitoramento.

O sistema deve ter arquitetura modular para permitir expansão futura.

Os registros de log devem ser persistidos em uma tabela própria no banco de dados (ex.: LOG\_MESA).

Cada registro deve conter no mínimo:

Data/hora do evento.

Identificação da mesa.

Ação realizada.

Usuário responsável (ex.: garçom ou caixa).

O armazenamento dos logs deve garantir integridade e não permitir exclusão sem perfil de administrador.

O sistema deve ser capaz de manter logs de pelo menos 1 ano sem perda de performance.

As consultas de log devem ser rápidas, mesmo em cenários com grande volume de registros.