

Escopo do Projeto

Desenvolver um sistema de cardápio digital para o restaurante, onde os clientes possam fazer pedidos diretamente pelo site/aplicativo sem a necessidade de um atendente físico. O processo será simplificado e eficiente, garantindo uma experiência moderna e prática para os clientes e uma operação mais ágil para o restaurante.

Funcionalidades Principais

1. Escaneamento do QrCode da Mesa:

- Cada mesa terá um QrCode exclusivo que os clientes poderão escanear usando seus smartphones.
- O QrCode direcionará o cliente para o site/aplicativo do restaurante, automaticamente associando o pedido à mesa específica.

2. Cadastro de Clientes:

- Os clientes precisarão se cadastrar antes de fazer um pedido.
- Campos de cadastro incluem: Nome, Email ou Telefone, e Data de Nascimento.
- Opção para criar uma conta para facilitar pedidos futuros e participar de programas de fidelidade.

3. Navegação e Seleção do Cardápio:

- O cardápio será exibido de forma organizada e fácil de navegar, com categorias (entradas, pratos principais, sobremesas, bebidas, etc.).
- Cada item do cardápio terá uma descrição detalhada, ingredientes, preço e opções de personalização (ex. adicionar/remover ingredientes, escolher o ponto da carne, etc.).

4. Personalização de Pratos:

- O sistema deve ser flexível para adicionar novos pratos ou atualizar os existentes.
- Opção para destacar pratos especiais ou promoções temporárias.

5. Processo de Pedido:

- Após selecionar os itens desejados, o cliente pode revisar o pedido e fazer ajustes.
- Informações essenciais do pedido incluem: Nome do Cliente, Número da Mesa, e Alterações Solicitadas.
- Confirmação do pedido com um resumo e o tempo estimado de preparo.

6. Envio Direto para a Cozinha:

- O pedido é enviado automaticamente para a cozinha assim que o cliente confirmar.
- Sistema de gerenciamento na cozinha para organizar e priorizar os pedidos recebidos, garantindo eficiência no preparo.

7. Notificações e Atualizações:

- O cliente recebe notificações sobre o status do pedido (ex. pedido recebido, em preparação, pronto para servir).
- Possibilidade de enviar notificações push ou SMS para informar o cliente sobre o andamento.

8. Pagamentos:

- Opção para pagar diretamente pelo site/aplicativo usando métodos de pagamento digital (cartão de crédito, débito, carteiras digitais).
- Possibilidade de pagar na mesa ao final da refeição, se preferido.

9. Feedback e Avaliações:

- Após a refeição, o cliente pode deixar feedback sobre os pratos e a experiência.
- Sistema de avaliação para coletar dados e melhorar o serviço.

10. Administração e Relatórios:

- Interface de administração para gerenciar o cardápio, visualizar pedidos em tempo real e gerar relatórios.
- Relatórios detalhados sobre vendas, preferências dos clientes, e desempenho do serviço.

Benefícios

- **Eficiência Operacional:** Reduz o tempo de espera e erros de comunicação entre clientes e atendentes.
- **Experiência do Cliente:** Proporciona uma experiência moderna e conveniente, com controle total sobre o pedido.
- **Flexibilidade e Atualização:** Facilita a gestão do cardápio e permite ajustes rápidos conforme necessário.
- **Engajamento e Fidelização:** Melhora a interação com o cliente através de cadastro, feedback e programas de fidelidade.

Com este sistema, o restaurante poderá oferecer um serviço mais ágil e personalizado, ao mesmo tempo em que otimiza suas operações internas.

DataBase

UsersAdmin	Stock	Despesas
Clientes	StockMovement	CaixaDiario
Desks	Orders	Feedback
Categories	ItemOrder	Promocoes
Menultens	Pagamentos	Logs

usersAdmin

- idUser: primary key autoincrement
- name: varchar(255) not null
- email: varchar(255) unique not null
- password: varchar(255) not null
- type: enum('dono', 'gerente', 'garçom', 'cozinha') not null
- activo: boolean default true
- dateRegister: timestamp
- lastLogin: datetime

clientes

- idCliente: primary key autoincrement
- name: varchar(255) not null
- email: varchar(255) unique
- telefone: varchar(20) unique not null
- password: varchar(255) not null
- birthData: date
- dateRegister: timestamp
- loyaltyPoints: int default 0

desks

- idDesk: primary key autoincrement
- deskNumber: varchar(10) unique not null
- qrcodeUId: varchar(50) unique not null
- capacity: int not null
- condition: enum('livre', 'ocupada', 'reservada', 'manutencao') default 'livre'

categories

- idCategory: primary key autoincrement
- name: varchar(50) not null
- descricao: text
- order: int default 0
- activo: boolean default true

menultens

- idItem: primary key autoincrement
- idCategory: foreign key → categories(idCategory)
- name: varchar(100) not null
- descricao: text
- price: decimal(10,2) not null
- cost: decimal(10,2)
- timePreparo: int (minutos)
- emphasis: boolean default false
- activo: boolean default true
- imagePath: varchar(255)
- options: json

stock

- idStock: primary key autoincrement
- idItem: foreign key → menultens(idItem)
- amount: decimal(10,2) not null
- unity: varchar(20) not null (*ex: kg, L, un*)
- updateData: timestamp

stockMovement

- idMovement: primary key autoincrement
- idStock: foreign key → stock(idStock)
- type: enum('entrada', 'saida', 'ajuste') not null
- amount: decimal(10,2) not null
- descricao: text
- timeDate: timestamp

orders

- idOrder: primary key autoincrement
- idCliente: foreign key → clientes(idCliente)
- idDesk: foreign key → desks(idDesk)
- timeData: timestamp
- status: enum('recebido', 'preparando', 'pronto', 'entregue', 'cancelado') default 'recebido'
- total: decimal(10,2) not null
- observacao: text

itemOrder

- idItemOrder: primary key autoincrement
- idOrder: foreign key → orders(idOrder)
- idItem: foreign key → menultens(idItem)
- amount: int default 1 not null
- unityPrice: decimal(10,2) not null
- custom: json
- observacao: text

pagamentos

- idPagamento: primary key autoincrement
- idOrder: foreign key → orders(idOrder)
- metodo: enum('credito', 'debito', 'pix', 'dinheiro', 'naConta') not null
- valor: decimal(10,2) not null
- status: enum('pendente', 'pago', 'reembolsado', 'cancelado') default 'pendente'
- timeData: timestamp
- codTransacao: varchar(100)

despesas

- idDespesa: primary key autoincrement
- titulo: varchar(100) not null
- descricao: text
- valor: decimal(10,2) not null
- type: enum('agua', 'luz', 'fornecedor', 'outros') not null
- timeData: timestamp

caixaDiario

- idCaixa: primary key autoincrement
- data: date not null
- entrada: decimal(10,2) default 0
- saida: decimal(10,2) default 0
- saldoFinal: decimal(10,2) ([calculado: entrada - saida](#))

feedbacks

- idFeedback: primary key autoincrement
- idOrder: foreign key → orders(idOrder)
- idCliente: foreign key → clientes(idCliente)
- avaliacao: int (1 a 5)
- comments: text
- timeData: timestamp
- respondido: boolean default false
- respostaAdmin: text

promocoes

- idPromocao: primary key autoincrement
- titulo: varchar(100) not null
- descricao: text
- type: enum('desconto', 'brinde', 'fidelidade') not null
- valorDesconto: decimal(10,2)
- codigo: varchar(50) unique
- validoDe: datetime
- validoAte: datetime
- status: boolean default true

logs

- idLog: primary key autoincrement
- idUser: foreign key → usersAdmin(idUser)
- idCliente: foreign key → clientes(idCliente)
- acao: varchar(100) not null
- descricao: text
- ipAddress: varchar(50)
- timeData: timestamp