Luiz Gustavo de Melo Silva

DOCUMENTAÇÃO DE BANCO DE DADOS

Documentação elaborada para composição da nota da Disciplina de Projeto de Banco de Dados, orientado pelo profª Tacyana Batista.

Sumário

Minimundo

O EasyOrder é um sistema de cardápio digital integrado para restaurantes, com foco em otimizar o atendimento e proporcionar uma experiência moderna aos clientes.

No restaurante, cada mesa possui um QR Code exclusivo. Quando o cliente escaneia esse código com seu smartphone, é automaticamente direcionado ao sistema, que reconhece de qual mesa o pedido será feito.

Antes de realizar o pedido, o cliente precisa se cadastrar com informações básicas (nome, e-mail ou telefone, senha e data de nascimento). Esse cadastro permite que o sistema registre seu histórico de pedidos, atribua pontos de fidelidade e personalize a experiência.

O cardápio digital é organizado em categorias (entradas, pratos principais, sobremesas, bebidas, etc.), e cada item do cardápio possui descrição, preço, custo, tempo de preparo estimado, imagem e possibilidade de personalização (como adicionar/remover ingredientes).

Durante a realização do pedido, o cliente pode:

- Selecionar múltiplos itens;
- Adicionar observações ou personalizações;
- Revisar e confirmar o pedido;
- Realizar o pagamento pelo próprio aplicativo (cartão, Pix, carteiras digitais) ou deixar para pagar na mesa.

Assim que o pedido é confirmado, ele é enviado automaticamente para o sistema da cozinha, que organiza e prioriza os pedidos. O cliente recebe notificações em tempo real sobre o andamento (pedido recebido, em preparo, pronto e entregue).

O sistema também coleta feedbacks e avaliações após a refeição, registrando comentários e notas de 1 a 5, para que o restaurante possa melhorar seu atendimento e cardápio.

No lado administrativo, os gestores do restaurante têm acesso a:

- Gerenciamento completo do cardápio;
- Controle de estoque, com movimentações de entrada, saída e ajustes;
- Registro de promoções e programas de fidelidade;
- Relatórios de vendas, consumo por categoria e preferências de clientes;
- Registro de logs de atividades feitas tanto por clientes quanto por funcionários.

Benefícios para o restaurante:

- Redução de erros e tempo de atendimento;
- Melhor controle de estoque e pedidos;
- Relatórios estratégicos para tomada de decisão;
- Sistema de fidelidade que aumenta o engajamento dos clientes.

Benefícios para o cliente:

- Autonomia e agilidade no pedido;
- Facilidade no pagamento digital;
- Transparência com notificações de status;
- Participação em promoções e programas de fidelidade.

MER (Modelo de Entidade e Relacionamento)

O Modelo de Entidade e Relacionamento (MER) é o projeto conceitual do banco de dados. Ele é apresentado em duas partes: a descrição detalhada das entidades e seus atributos, e um resumo dos relacionamentos e suas cardinalidades.

Entidades e Atributos

- **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: datetime default current_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: datetime default current_timestamp
  - loyaltyPoints: int default 0

- **desks**
  - idDesk: primary key autoincrement
  - deskNumber: varchar(10) unique not null
  - qrcodeUld: 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: datetime default current_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: datetime default current_timestamp
- **orders**
  - idOrder: primary key autoincrement
  - idCliente: foreign key clientes(idCliente)
  - idDesk: foreign key → desks(idDesk)
  - idPromocao: foreign key → promocoes (idPromocao)
  - timeData: datetime default current_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(idltem)
  - amount: int default 1 not null
  - unityPrice: decimal(10,2) not null
  - custom: json
  - observacao: text
- **feedbacks**
  - idFeedback: primary key autoincrement
  - idOrder: foreign key orders(idOrder)
  - idCliente: foreign key clientes(idCliente)
  - avaliacao: int (1 a 5)
  - comments: text
  - timeData: datetime default current_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: datetime default current_timestamp

Relacionamentos e Cardinalidades

- categories (1) --- (N) menultens: Uma categoria pode ter vários itens, mas um item pertence a uma única categoria.
- clientes (1) --- (N) orders: Um cliente pode fazer vários pedidos, mas um pedido pertence a um único cliente.
- desks (1) --- (N) orders: Uma mesa pode ter vários pedidos ao longo do tempo, mas um pedido é feito de uma única mesa.
- orders (N) --- (M) menultens: Um pedido pode conter vários itens, e um item pode estar em vários pedidos. Este relacionamento é implementado pela entidade associativa itemOrder.
- menultens (1) --- (1) stock: Cada item do cardápio possui um único registro de estoque correspondente.
- stock (1) --- (N) stockMovement: Um item em estoque pode ter várias movimentações (entradas, saídas, etc.).
- orders (1) --- (1) feedbacks: Um pedido pode ter no máximo um feedback.
- clientes (1) --- (N) feedbacks: Um cliente pode registrar vários feedbacks.
- promocoes (N) --- (M) orders: Uma promoção pode ser aplicada a vários pedidos, e um pedido pode ter várias promoções. Requer uma tabela associativa para implementação.
- usersAdmin (1) --- (N) logs: Um usuário administrador pode gerar vários registros de log.
- clientes (1) --- (N) logs: Um cliente pode gerar vários registros de log.

DER (Diagrama de Entidade e Relacionamento)

Esta seção descreve textualmente o diagrama, detalhando o significado de cada relacionamento e como ele é implementado através das chaves primárias (PK) e estrangeiras (FK).

- **Relacionamento Categoria-Item**:
    - **Regra:** Uma categories agrupa N menuItens.
    - **Descrição:** Garante que todo item do cardápio esteja classificado, como "Entrada" ou "Prato Principal".
    - **Implementação:** A chave primária categories.idCategory (PK) é usada como chave estrangeira menuItens.idCategory (FK).
- **Relacionamento Cliente-Pedido**:
    - **Regra:** Um clientes realiza N orders.
    - **Descrição:** Associa um pedido a quem o fez, permitindo histórico de pedidos e programas de fidelidade.
    - **Implementação:** clientes.idCliente (PK) é usada como orders.idCliente (FK).
- **Relacionamento Mesa-Pedido**:
    - **Regra:** Uma desks sedia N orders.
    - **Descrição:** Identifica a localização física de um pedido dentro do restaurante.
    - **Implementação:** desks.idDesk (PK) é usada como orders.idDesk (FK).
- **Relacionamento Pedido-Item (N:M)**:
    - **Regra:** Um orders contém M menuItens.
    - **Descrição:** É o relacionamento central que forma o pedido. A entidade associativa itemOrder detalha quais itens, em qual quantidade (amount) e com qual preço (unityPrice) compõem cada pedido.
    - **Implementação:** itemOrder possui duas chaves estrangeiras: idOrder (de orders) e idItem (de menuItens).

- **Relacionamento Item-Estoque (1:1)**:
    - **Regra:** Um menuItens possui um stock.
    - **Descrição:** Controla a quantidade disponível de cada item do cardápio.
    - **Implementação:** menuItens.idItem (PK) é usada como stock.idItem (FK), que também deve ser definida como UNIQUE para garantir a cardinalidade 1:1.
- **Relacionamento Estoque-Movimentação**:
    - **Regra:** Um stock sofre N stockMovement.
    - **Descrição:** Rastreia todas as entradas, saídas e ajustes no estoque de um item.
    - **Implementação:** stock.idStock (PK) é usada como stockMovement.idStock (FK).
- **Relacionamento Pedido-Feedback (1:1)**:
    - **Regra:** Um orders recebe um feedbacks.
    - **Descrição:** Permite que a avaliação do cliente seja diretamente ligada ao pedido que a originou.
    - **Implementação:** orders.idOrder (PK) é usada como feedbacks.idOrder (FK), que deve ser UNIQUE.
- **Relacionamento Cliente-Feedback**:
    - **Regra:** Um clientes fornece N feedbacks.
    - **Descrição:** Cria um histórico de todas as avaliações feitas por um cliente.
    - **Implementação:** clientes.idCliente (PK) é usada como feedbacks.idCliente (FK).
- **Relacionamento Promoção-Pedido (N:M)**:
    - **Regra:** Uma promocoes é aplicada em M orders.
    - **Descrição:** Permite que cupons ou ofertas sejam aplicados a um pedido. Como um pedido pode ter múltiplos cupons e um cupom pode ser usado várias vezes, é necessário uma tabela de ligação.
    - **Implementação:** Requer uma tabela associativa (ex: Order_Promocoes) com as chaves estrangeiras idOrder e idPromocao.

Scripts DDL (Data Definition Language)

Scripts DML (Data Manipulation Language)

Scripts DQL (Data Query Language)

Conclusão

Anexos


usersAdmin
idUser INT
name VARCHAR(255)
email VARCHAR(255)
password VARCHAR(100)
type ENUM(...)
activo TINYINT
dateRegister DATETIME
lastLogin DATETIME
Indexes
clientes
idCliente INT
name VARCHAR(255)
email VARCHAR(255)
telefone VARCHAR(20)
password VARCHAR(255)
birthDate DATETIME
dateRegister DATETIME
loyaltyPoints INT
Indexes
logs
idLog INT
idUser INT
idClientes INT
acao VARCHAR(100)
descricao TEXT(255)
ipAddress VARCHAR(50)
timeData DATETIME
Indexes
categories
idCategory INT
name VARCHAR(50)
descricao TEXT(255)
order INT
activo TINYINT
Indexes
feedbacks
idFeedback INT
idOrder INT
idCliente INT
avaliacao INT
comments TEXT(255)
timeData DATETIME
Indexes
desks
idDesk INT
deskNumber INT
qrcodeUid VARCHAR(50)
capacity INT
condition ENUM(...)
Indexes
itemOrder
idItemOrder INT
idOrder INT
idItem INT
amount INT
unityPrice DECIMAL(10,2)
custom JSON
observacao TEXT(255)
Indexes
menuItens
idItem INT
idCategory INT
name VARCHAR(100)
descricao TEXT(255)
price DECIMAL(10,2)
cost DECIMAL(10,2)
timePreparo INT
emphasis TINYINT
activo TINYINT
imagePath VARCHAR(255)
options JSON
Indexes
orders
idOrder INT
idPromocoes INT
idCliente INT
idDesk INT
timeData DATETIME
status ENUM(...)
total DECIMAL(10,2)
observacao TEXT(255)
Indexes
stock
idStock INT
idItem INT
amount DECIMAL(10,2)
unity VARCHAR(20)
updateData DATETIME
Indexes
stockMovement
idMovement INT
idStock INT
type ENUM(...)
amount DECIMAL(10,2)
descricao TEXT(255)
timeDate DATETIME
Indexes
promocoes
idPromocoes INT
title VARCHAR(100)
descricao VARCHAR(45)
type ENUM(...)
valorDesconto DECIMAL(10,2)
codigo VARCHAR(50)
validoDe DATETIME
validoAte DATETIME
status TINYINT
Indexes