

Relatório Técnico – Mini-sistema da Padaria Pão e Confia

1. Objetivo do Projeto

O projeto teve como objetivo evoluir um esqueleto inicial em PHP, que continha operações básicas de CRUD (Create, Read, Update, Delete) e um script SQL simples, transformando-o em um mini-sistema funcional para gestão da padaria Pão e Confia. O sistema gerencia categorias, produtos, clientes e pedidos, aplicando boas práticas de programação, segurança e usabilidade.

2. Modelagem de Dados

A modelagem partiu do levantamento dos requisitos e foi representada por um modelo entidade-relacionamento com as seguintes tabelas e relacionamentos:

Categorias (1) —< Produtos

Clientes (1) —< Pedidos

Pedidos (1) —< Itens do Pedido

Tabelas principais:

categorias: id, nome, descrição, data de criação

produtos: id, categoria_id, nome, SKU, preço, estoque, status, data de criação

clientes: id, nome, e-mail, telefone, documento, endereço, data de criação

pedidos: id, cliente_id, status, total, observações, datas de criação/atualização

itens_pedido: id, pedido_id, produto_id, quantidade, preço unitário, subtotal calculado

Chaves estrangeiras foram definidas com integridade referencial, e índices foram adicionados para melhorar o desempenho em consultas.

3. Decisões de Implementação

Organização do Código:

Separação em camadas: *configuração*, *conexão com banco*, *models*, *controllers* e *views*.
Reutilização de código via funções e classes para operações CRUD.

Padrão de URLs: CRUDs organizados em rotas claras.

Front-end: HTML5 e CSS3 com Bootstrap para responsividade e usabilidade.

Back-end: PHP procedural com funções organizadas, e mysqli/PDO com prepared statements.

Banco de Dados: MySQL/MariaDB, script inicial atualizado com constraints, índices e *checks*.

4. Pontos de Validação e Segurança

Validação de dados no front-end e back-end:

Campos obrigatórios (nome, preço, e-mail etc.).

Tipos e formatos (preço numérico, e-mail válido, estoque não negativo).

Prevenção contra ataques:

Uso de *prepared statements* para evitar SQL Injection.

Escape de saídas em HTML para evitar XSS.

Proteção contra CSRF em formulários (tokens).

Autenticação e Autorização:

Módulo opcional de login com *hash* seguro de senha.

Restrições por perfil de usuário (admin/atendente).

Controle de Estoque:

Atualização automática ao registrar pedidos.

Bloqueio de venda com estoque insuficiente.

5. Usabilidade

Interface intuitiva com menus claros e botões de ação visíveis.

Feedback ao usuário (mensagens de sucesso ou erro).

Listagens com paginação para melhor desempenho.

Filtros e busca em listagens (por nome, categoria, cliente).

6. Testes

Testes manuais em cada módulo CRUD.

Testes de fluxo: cadastro de categoria → produto → cliente → pedido.

Verificação de validações e mensagens de erro.