UNISUAM (Centro Universitário Augusto Motta)

DOCUMENTAÇÃO DA ARQUITETURA E FUNCIONAMENTO DO SITE

PERIGO TECH

Participantes: Breno Morais, Diogo Bello, Italo Romulo, Luiz Dias, Yan Oliveira

Campo Grande/RJ, 24 de Setembro de 2025

1. INTRODUÇÃO

Este documento detalha a arquitetura e o funcionamento de um sistema web composto por um portal de e-commerce e um módulo de gerenciamento de usuários. O objetivo é descrever cada componente, desde a estrutura dos bancos de dados até a lógica de programação dos arquivos PHP, JavaScript e SQL, oferecendo uma visão completa dos fluxos operacionais, funcionalidades e considerações de segurança da aplicação.

2. ARQUITETURA GERAL DO PROJETO

O sistema foi projetado de forma modular, com uma separação clara entre a gestão de usuários e a vitrine de produtos. Essa abordagem utiliza dois bancos de dados distintos para garantir a organização e a segurança dos dados.

2.1 Sistema de Cadastro e Autenticação de Usuários

Este módulo é responsável por todas as operações relacionadas aos usuários. Utiliza o banco de dados cadastro-tech e seus principais arquivos são cadastro.php, login.php, testeLogin.php e o conjunto de arquivos do painel administrativo. A conexão é gerenciada pelo arquivo config.php.

2.2 Sistema da Loja Virtual (E-commerce)

Este módulo corresponde à interface pública da loja. Utiliza o banco de dados perigotech para armazenar e exibir informações de produtos. Seus arquivos principais são loja.php, prod.php e carrinho.php. A configuração de conexão está contida diretamente nos próprios arquivos que a utilizam.

2.3 Estrutura do Banco de Dados de Usuários (cadastro_tech.sql)

O arquivo cadastro_tech.sql define a estrutura da tabela principal para o sistema de usuários. A tabela, denominada cadastro_tech, é projetada para armazenar todas as informações relevantes dos usuários registrados na plataforma. As colunas incluem dados pessoais (nome, e-mail, CPF, data de nascimento), informações de contato (telefone, endereço, CEP), credenciais de acesso (login, senha) e dados para recuperação de conta (nome da mãe). O campo idusuarios serve como chave primária auto-incrementável, garantindo um identificador único para cada registro.

3. FLUXO DE CADASTRO E AUTENTICAÇÃO DO USUÁRIO

O fluxo de autenticação descreve o percurso completo do usuário, desde o primeiro acesso e registro até a validação de segurança para acesso a áreas restritas.

3.1 Cadastro de Novo Usuário (cadastro.php)

A página cadastro.php serve como ponto de entrada para novos usuários. O front-end apresenta um formulário para coleta de dados pessoais, enquanto o back-end em PHP processa essas informações. O script PHP utiliza Prepared Statements para inserir os dados no banco de forma segura, prevenindo ataques de SQL Injection. Em caso de sucesso, o usuário é redirecionado para a página de login.

3.2 Configuração do Banco de Dados (config.php)

O arquivo config.php centraliza as credenciais de conexão com o banco de dados cadastro-tech, estabelecendo uma instância de conexão mysqli que é reutilizada em todos os scripts que manipulam dados de usuários.

3.3 Tela de Login (login.php)

Apresenta um formulário para que o usuário insira suas credenciais (login e senha). Os dados são submetidos via método POST ao arquivo testeLogin.php para validação.

3.4 Processamento de Login (testeLogin.php)

Este script valida as credenciais do usuário. Ele recebe os dados, conecta-se ao banco cadastro-tech e executa uma consulta para verificar a existência do usuário. Em caso de sucesso, armazena os dados do usuário na sessão ($_SESSION) e o redireciona para a área apropriada (neste caso, sistema.php). Caso contrário, o usuário é redirecionado para a loja.

3.5 Autenticação de Dois Fatores – 2FA (2fa.php e back.php)

Após o login, o arquivo 2fa.php implementa uma camada adicional de segurança. Ele exibe uma pergunta de segurança selecionada aleatoriamente (ex.: "Qual o nome da sua mãe?"). A resposta do usuário é enviada de forma assíncrona ao back.php, que valida a informação contra os dados armazenados no banco. Após três tentativas malsucedidas, o usuário é desconectado.

4. A LOJA VIRTUAL

Esta seção descreve os componentes que formam a interface de e-commerce da aplicação.

4.1 Estrutura do Banco de Dados da Loja (BD_tela_inicial.sql)

O arquivo BD_tela_inicial.sql contém os comandos para criar o banco de dados perigotech e a tabela produtos. Ele também inclui registros INSERT para popular a loja com um catálogo inicial de produtos, definindo a base de dados sobre a qual a loja opera.

4.2 Página da Loja (loja.php)

A página loja.php atua como a vitrine principal da loja virtual. O back-end em PHP

conecta-se ao banco perigotech e busca os produtos, organizando-os por categorias em carrosséis gerados dinamicamente.

Para padronizar a exibição de cada item, foi criada a função renderProduto(), que garante consistência visual e facilita a manutenção do código.

4.3 Página de Detalhes do Produto (prod.php)

Ao clicar em um produto na loja, o usuário é direcionado para a página prod.php.

O script captura o ID do produto via URL ($_GET['id']), busca suas informações detalhadas no banco de dados e as exibe de forma organizada. Além disso, a lógica em PHP calcula e apresenta diferentes opções de parcelamento do preço, garantindo clareza para o usuário.

4.4 Carrinho de Compras (carrinho.php)

Atualmente, o carrinho.php apresenta uma interface estática do carrinho de compras.

Os produtos exibidos são fixos (hardcoded).

As interações, como adicionar ou remover itens, são simuladas por alertas em JavaScript.

Esta abordagem foi adotada como protótipo, sendo prevista a implementação futura de funcionalidades dinâmicas, mantendo a consistência visual da página.

5. PAINEL ADMINISTRATIVO

Esta seção é de acesso restrito e destina-se ao gerenciamento de usuários cadastrados.

5.1 Dashboard do Sistema (sistema.php)

O sistema.php é o painel central da administração. O acesso é protegido por uma verificação que garante que apenas usuários logados e com permissão de administrador (presentes no array $admin_users) possam visualizá-lo. A página exibe uma tabela com todos os usuários cadastrados e oferece opções para editar ou excluir cada registro.

5.2 Edição de Cliente (edit.php)

Este arquivo apresenta um formulário pré-preenchido com os dados de um usuário específico, selecionado a partir do dashboard. A função htmlspecialchars() é utilizada para exibir os dados de forma segura, prevenindo ataques de Cross-Site Scripting (XSS). As alterações são submetidas ao atualizar_cliente.php.

5.3 Atualização do Cliente (atualizar_cliente.php)

Recebe os dados do formulário de edição e executa uma consulta UPDATE no banco de dados. Para prevenir SQL Injection, utiliza a função mysqli_real_escape_string() para tratar os dados recebidos antes de construir a consulta. Após a atualização, o administrador é redirecionado de volta ao dashboard.

5.4 Exclusão de Cliente (delete.php)

Este script é responsável por remover um usuário do banco de dados. Ele recebe o ID do usuário, verifica sua existência e, se confirmado, executa um comando DELETE. Uma confirmação via JavaScript é solicitada no front-end para prevenir exclusões acidentais.

6. CONSIDERAÇÕES DE SEGURANÇA

Durante a análise do projeto, foram identificados pontos críticos de segurança no script testeLogin.php. A construção de consultas SQL por concatenação direta de variáveis torna a aplicação vulnerável a SQL Injection. Além disso, a comparação direta de senhas indica que elas estão armazenadas em texto claro no banco de dados. Recomenda-se enfaticamente a adoção de Prepared Statements também para o login e a implementação de hashing de senhas, utilizando funções como password_hash() e password_verify().

7. CONCLUSÃO

O projeto apresenta uma arquitetura bem definida, com uma clara separação entre a lógica de negócio do e-commerce e o gerenciamento de usuários. As funcionalidades

cobrem o ciclo completo de vida do usuário e a exibição de produtos. Embora funcional, a aplicação requer melhorias de segurança, especialmente no processo de autenticação, para garantir a proteção dos dados e a integridade do sistema.