# Sistema de Login e Cadastro - PHP 8 PDO MariaDB

Sistema completo de autenticação de usuários desenvolvido com PHP 8, PDO (PHP Data Objects) e MariaDB/MySQL. Implementa as melhores práticas de segurança, incluindo criptografia de senhas com bcrypt, prepared statements, proteção contra força bruta, controle de sessões e código totalmente comentado em português.

## 🎯 Características Principais

### Funcionalidades

* **Cadastro de novos usuários** com validação completa de dados
* **Login seguro** com autenticação de credenciais
* **Dashboard personalizado** após autenticação
* **Edição de perfil** com alteração de dados e senha
* **Listagem de usuários** cadastrados no sistema
* **Logout seguro** com limpeza de sessão e cookies
* **Proteção contra força bruta** com limite de tentativas
* **Sessões com timeout** por inatividade
* **Keep-alive automático** para manter sessão ativa
* **Opção "Lembrar-me"** com cookies seguros

### Segurança

* **Criptografia de senhas** usando password\_hash() com bcrypt
* **Prepared statements** em todas as queries SQL (proteção contra SQL Injection)
* **Proteção XSS** com htmlspecialchars() em todas as saídas
* **Validação server-side** de todos os dados
* **Regeneração de ID de sessão** para prevenir session fixation
* **Cookies HttpOnly** para prevenir acesso via JavaScript
* **Limite de tentativas de login** com bloqueio temporário
* **Registro de tentativas** de login para auditoria
* **Timeout de sessão** configurável
* **Sanitização de entradas** do usuário

## 📁 Estrutura de Arquivos

sistema-login/

│

├── config.php # Configurações gerais e do banco de dados

├── conexao.php # Classe de conexão PDO (padrão Singleton)

├── index.php # Página inicial (redireciona conforme autenticação)

├── cadastro.php # Página de cadastro de novos usuários

├── login.php # Página de login

├── dashboard.php # Dashboard (área restrita)

├── perfil.php # Edição de perfil do usuário

├── usuarios.php # Listagem de usuários cadastrados

├── logout.php # Processo de logout

├── keep-alive.php # Endpoint para manter sessão ativa

├── style.css # Estilos CSS responsivos

├── database.sql # Script SQL para criar banco e tabelas

├── README.md # Este arquivo de documentação

└── INSTALACAO.md # Guia detalhado de instalação

## 🗄️ Estrutura do Banco de Dados

### Tabela: usuarios

| Campo | Tipo | Descrição |
| --- | --- | --- |
| id | INT | Chave primária auto-incrementável |
| nome | VARCHAR(100) | Nome completo do usuário |
| email | VARCHAR(100) | E-mail único para login |
| senha | VARCHAR(255) | Hash da senha (bcrypt) |
| foto\_perfil | VARCHAR(255) | Caminho da foto de perfil (opcional) |
| ativo | TINYINT(1) | Status da conta (1=ativa, 0=inativa) |
| criado\_em | TIMESTAMP | Data/hora de criação da conta |
| atualizado\_em | TIMESTAMP | Data/hora da última atualização |
| ultimo\_login | TIMESTAMP | Data/hora do último login |

### Tabela: tentativas\_login

| Campo | Tipo | Descrição |
| --- | --- | --- |
| id | INT | Chave primária auto-incrementável |
| email | VARCHAR(100) | E-mail da tentativa |
| ip\_address | VARCHAR(45) | Endereço IP da tentativa |
| sucesso | TINYINT(1) | 1=sucesso, 0=falha |
| data\_tentativa | TIMESTAMP | Data/hora da tentativa |

## 🚀 Instalação Rápida

### Pré-requisitos

* PHP 8.0 ou superior
* MariaDB 10.3+ ou MySQL 5.7+
* Servidor web (Apache, Nginx)
* Extensões PHP: PDO, pdo\_mysql, mbstring, json

### Passo 1: Criar o Banco de Dados

Execute o arquivo database.sql no seu servidor MariaDB/MySQL:

mysql -u root -p < database.sql

Ou importe via phpMyAdmin.

### Passo 2: Configurar a Conexão

Edite o arquivo config.php com suas credenciais:

define('DB\_HOST', 'localhost');

define('DB\_NAME', 'sistema\_login');

define('DB\_USER', 'root');

define('DB\_PASS', '');

### Passo 3: Configurar o Servidor

**Apache**: Coloque os arquivos no diretório htdocs ou www.

**PHP Built-in Server** (desenvolvimento):

cd sistema-login

php -S localhost:8000

### Passo 4: Acessar o Sistema

Abra o navegador e acesse:

http://localhost/sistema-login/

## 📖 Como Usar

### Cadastrar Novo Usuário

1. Acesse a página de cadastro (cadastro.php)
2. Preencha nome, e-mail e senha (mínimo 6 caracteres)
3. Confirme a senha
4. Clique em "Criar Conta"
5. Será redirecionado automaticamente para o login

### Fazer Login

1. Acesse a página de login (login.php)
2. Digite seu e-mail e senha
3. Marque "Lembrar-me" se desejar (opcional)
4. Clique em "Entrar"
5. Será redirecionado para o dashboard

### Usuários de Teste

O sistema vem com 3 usuários pré-cadastrados para teste:

| E-mail | Senha | Nome |
| --- | --- | --- |
| [admin@sistema.com](mailto:admin@sistema.com) | 123456 | Administrador |
| [joao@email.com](mailto:joao@email.com) | 123456 | João Silva |
| [maria@email.com](mailto:maria@email.com) | 123456 | Maria Santos |

**⚠️ IMPORTANTE**: Altere essas senhas em produção!

## 🔒 Recursos de Segurança Implementados

### 1. Criptografia de Senhas

As senhas são criptografadas usando password\_hash() com o algoritmo bcrypt:

$senha\_hash = password\_hash($senha, PASSWORD\_DEFAULT);

A verificação é feita com password\_verify():

if (password\_verify($senha\_digitada, $senha\_hash\_banco)) {

// Login válido

}

### 2. Prepared Statements

Todas as queries SQL usam prepared statements para prevenir SQL Injection:

$sql = "SELECT \* FROM usuarios WHERE email = ?";

$stmt = $pdo->prepare($sql);

$stmt->execute([$email]);

### 3. Proteção XSS

Todas as saídas são filtradas com htmlspecialchars():

echo htmlspecialchars($usuario['nome']);

### 4. Proteção Contra Força Bruta

O sistema limita tentativas de login e bloqueia temporariamente após exceder o limite:

define('MAX\_LOGIN\_ATTEMPTS', 5); // Máximo de tentativas

define('LOCKOUT\_TIME', 900); // Bloqueio por 15 minutos

### 5. Controle de Sessão

Sessões têm timeout configurável e regeneração periódica de ID:

define('SESSION\_TIMEOUT', 1800); // 30 minutos de inatividade

### 6. Cookies Seguros

Cookies são configurados com flags de segurança:

ini\_set('session.cookie\_httponly', 1); // Não acessível via JavaScript

ini\_set('session.cookie\_samesite', 'Strict'); // Proteção CSRF

## 🎨 Personalização

### Alterar Cores do Tema

Edite o arquivo style.css e modifique o gradiente:

background: linear-gradient(135deg, **#667eea** 0%, **#764ba2** 100%);

### Alterar Timeout de Sessão

Edite o arquivo config.php:

define('SESSION\_TIMEOUT', 3600); // 1 hora

### Alterar Limite de Tentativas de Login

Edite o arquivo config.php:

define('MAX\_LOGIN\_ATTEMPTS', 3); // 3 tentativas

define('LOCKOUT\_TIME', 1800); // 30 minutos

### Adicionar Campos ao Perfil

1. Adicione o campo na tabela usuarios
2. Adicione o campo no formulário de cadastro
3. Adicione o campo no formulário de perfil
4. Atualize as queries SQL correspondentes

## 📊 Fluxo de Autenticação

### Cadastro

Usuário preenche formulário

↓

Validação dos dados (server-side)

↓

Verifica se email já existe

↓

Criptografa senha com bcrypt

↓

Insere no banco de dados

↓

Redireciona para login

### Login

Usuário digita credenciais

↓

Verifica tentativas de login (proteção força bruta)

↓

Busca usuário por email

↓

Verifica se conta está ativa

↓

Valida senha com password\_verify()

↓

Cria sessão com dados do usuário

↓

Atualiza último login no banco

↓

Redireciona para dashboard

### Proteção de Páginas

Usuário acessa página restrita

↓

Verifica se está logado (sessão existe)

↓

Verifica timeout de inatividade

↓

Se válido: permite acesso

↓

Se inválido: redireciona para login

## 🔧 Configurações Avançadas

### Habilitar HTTPS

Para usar HTTPS, edite config.php:

ini\_set('session.cookie\_secure', 1); // Cookies apenas via HTTPS

define('BASE\_URL', 'https://seudominio.com/sistema-login/');

### Modo de Produção

Desabilite o modo de desenvolvimento em config.php:

define('DEV\_MODE', false);

Isso irá:

* Ocultar mensagens de erro detalhadas
* Registrar erros em arquivo de log
* Melhorar a segurança geral

### Logs de Sistema

Os logs são gravados automaticamente quando DEV\_MODE está ativo. Para logs em produção, configure:

ini\_set('log\_errors', 1);

ini\_set('error\_log', ROOT\_PATH . '/logs/php-errors.log');

Crie o diretório de logs:

mkdir logs

chmod 755 logs

## 📱 Responsividade

O sistema é totalmente responsivo e se adapta a:

* **Desktop**: Layout completo com todas as funcionalidades
* **Tablet**: Layout ajustado para telas médias
* **Mobile**: Layout otimizado para smartphones

Media queries implementadas:

@media (max-width: 768px) {

/\* Estilos para mobile \*/

}

## 🎓 Conceitos Demonstrados

Este projeto demonstra:

* **PDO (PHP Data Objects)**: Interface moderna para bancos de dados
* **Prepared Statements**: Prevenção de SQL Injection
* **Password Hashing**: Criptografia segura de senhas
* **Session Management**: Controle de sessões de usuário
* **Cookie Security**: Uso seguro de cookies
* **Input Validation**: Validação de dados do usuário
* **Error Handling**: Tratamento de erros com try-catch
* **MVC Pattern** (simplificado): Separação de lógica
* **Responsive Design**: Interface adaptável
* **Security Best Practices**: Melhores práticas de segurança

## 🐛 Solução de Problemas

### Erro: "Access denied for user"

**Solução**: Verifique as credenciais em config.php.

### Erro: "Could not find driver"

**Solução**: Habilite a extensão PDO no php.ini:

extension=pdo\_mysql

### Sessão não persiste

**Solução**: Verifique se o diretório de sessões tem permissão de escrita:

chmod 755 /var/lib/php/sessions

### CSS não carrega

**Solução**: Verifique o caminho do arquivo CSS e limpe o cache do navegador.

## 📚 Documentação Adicional

* [Guia de Instalação Detalhado](INSTALACAO.md)
* [Documentação do PDO](https://www.php.net/manual/pt_BR/book.pdo.php)
* [Password Hashing](https://www.php.net/manual/pt_BR/function.password-hash.php)
* [Session Security](https://www.php.net/manual/pt_BR/session.security.php)

## 🤝 Contribuições

Este é um projeto educacional. Sinta-se livre para:

* Adicionar novas funcionalidades
* Melhorar a segurança
* Otimizar o código
* Corrigir bugs
* Melhorar a documentação

## 📄 Licença

Este projeto é de código aberto e está disponível para uso educacional e comercial.

## ✨ Funcionalidades Futuras

Possíveis melhorias para implementar:

* Recuperação de senha por e-mail
* Autenticação de dois fatores (2FA)
* Login social (Google, Facebook)
* Upload de foto de perfil
* Histórico de logins
* Notificações por e-mail
* API RESTful
* Testes automatizados
* Docker para deploy
* Internacionalização (i18n)

## 📞 Suporte

Para dúvidas ou problemas:

1. Verifique a documentação
2. Consulte os logs de erro
3. Teste a conexão com o banco
4. Verifique as permissões dos arquivos

**Desenvolvido com ❤️ para fins educacionais**

**Versão:** 1.0.0  
**Data:** 2025  
**Linguagem:** PHP 8+  
**Banco de Dados:** MariaDB/MySQL  
**Código:** 100% comentado em português