DATABASE = seguro

SCRIPT TABELA:

```
CREATE TABLE usuarios (
   id INT AUTO_INCREMENT PRIMARY KEY,
   username VARCHAR(50) NOT NULL,
   email VARCHAR(100) NOT NULL,
   senha VARCHAR(255) NOT NULL,
   autenticacao_habilitada TINYINT(1) DEFAULT 0,
   codigo_autenticacao INT,
   criado_em TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

Função verificarSenha: A função JavaScript verifica se a senha atende aos critérios de segurança:
Pelo menos 8 caracteres.
Pelo menos uma letra maiúscula.
Pelo menos uma letra minúscula.
Pelo menos um número.
Pelo menos um caractere especial.
A função exibe uma mensagem indicando se a senha é forte ou não.

Vamos fazer um resumo do aplicativo, incluindo seu funcionamento, funções, bibliotecas utilizadas, variáveis importantes e fluxo de operação.

Resumo do Aplicativo de Registro e Autenticação em Duas Etapas

Funcionalidades Principais

1. Registro de Usuário:
- Formulário de registro com campos para nome de usuário, e-mail, senha e confirmação de senha.
- Verificação da força da senha e se ambas as senhas coincidem.
- Verificação no servidor se o nome de usuário ou e-mail já estão registrados.
- Habilitação opcional de autenticação em duas etapas (2FA).

2. Autenticação em Duas Etapas (2FA):
- Geração de um código de autenticação para usuários que habilitarem o 2FA.
- Simulação da verificação do código de autenticação via modal.

3. Login de Usuário:
- Formulário de login com verificação de credenciais.
- Verificação do código de autenticação para usuários com 2FA habilitado.

Estrutura do Projeto

- Arquivos PHP:
- `db.php`: Conexão com o banco de dados MySQL.
- `register.php`: Lida com o registro de novos usuários.
- `login.php`: Lida com a autenticação de usuários.
- `autenticacao.php`: Página de verificação do código de autenticação em duas etapas.
- `dashboard.php`: Página de dashboard, acessível apenas para usuários autenticados.

- `verificar_codigo.php`: Verifica o código de autenticação em duas etapas inserido pelo usuário.

Resumo da Funcionalidade de **verificar_codigo.php**:
Este código verifica o código de autenticação em duas etapas inserido pelo usuário:

Inicia a sessão e verifica se o usuário está logado.
Conecta ao banco de dados e busca o código de autenticação armazenado para o usuário.
Compara o código inserido pelo usuário com o código armazenado no banco de dados.
Se o código for correto, conclui a autenticação em duas etapas e redireciona o usuário para o dashboard.
Exibe mensagens de erro ou sucesso conforme apropriado.

- Banco de Dados:
- Tabela `usuarios`:
  - `id`: ID do usuário (INT, AUTO_INCREMENT, PRIMARY KEY).
  - `username`: Nome de usuário (VARCHAR).
  - `email`: E-mail do usuário (VARCHAR).
  - `senha`: Hash da senha do usuário (VARCHAR).
  - `autenticacao_habilitada`: Indicador de autenticação em duas etapas habilitada (TINYINT).
  - `codigo_autenticacao`: Código de autenticação gerado para 2FA (VARCHAR).

Bibliotecas e Dependências

- PHP: Linguagem principal para o backend.
- MySQL: Sistema de gerenciamento de banco de dados.
- XAMPP: Plataforma de desenvolvimento que inclui Apache, MySQL, PHP e Perl.
- HTML e CSS: Para criar o frontend dos formulários.

Fluxo de Operação

1. Registro de Usuário:
- O usuário acessa `register.php`.
- Insere nome de usuário, e-mail, senha e confirmação de senha.
- Opcionalmente, habilita a autenticação em duas etapas.
- O sistema verifica se as senhas coincidem e se a senha é forte.
- No backend (`register.php`), o sistema verifica se o usuário ou e-mail já estão registrados.
- Se todas as verificações passarem, o usuário é registrado no banco de dados e redirecionado para a página de autenticação (`autenticacao.php`) ou login (`login.php`).

2. Autenticação em Duas Etapas:
- Se o usuário habilitou 2FA, um código é gerado e salvo no banco de dados.
- O usuário é redirecionado para `autenticacao.php` onde insere o código de autenticação.
- O código é verificado e, se correto, o usuário é autenticado e redirecionado para o dashboard.

3. Login de Usuário:
- O usuário acessa `login.php` e insere suas credenciais.
- O sistema verifica as credenciais no banco de dados.
- Se o usuário tem 2FA habilitado, é redirecionado para `autenticacao.php` para inserir o código de autenticação.
- Após a verificação do código, o usuário é autenticado e redirecionado para o dashboard.

Códigos Completos dos Arquivos

`db.php`

```php
<?php
$host = 'localhost';
$user = 'root';
$password = '';
$dbname = 'seginf';
$port = 7306; // Use a porta configurada no XAMPP

$mysqli = new mysqli($host, $user, $password, $dbname, $port);

if ($mysqli->connect_error) {
    die("Erro de conexão: " . $mysqli->connect_error);
}
?>
```

`register.php`

```php
<?php
session_start();

require_once 'db.php';

if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    $username = $_POST['username'];
    $email = $_POST['email'];
    $senha = $_POST['senha'];
    $confirm_senha = $_POST['confirm_senha'];

    if ($senha !== $confirm_senha) {
        $_SESSION['error'] = "As senhas não coincidem. Por favor, tente novamente.";
        header('Location: register.php');
        exit();
    }

    $sql_check_user = "SELECT * FROM usuarios WHERE username='$username' OR email='$email'";
    $result_check_user = $mysqli->query($sql_check_user);

    if ($result_check_user->num_rows > 0) {
        $_SESSION['error'] = "Usuário ou e-mail já registrado. Por favor, escolha outro.";
        header('Location: register.php');
        exit();
    }

    $senha_hash = password_hash($senha, PASSWORD_DEFAULT);
```

```
    $sql = "INSERT INTO usuarios (username, email, senha) VALUES ('$username', '$email', '$senha_hash')";
    if ($mysqli->query($sql) === TRUE) {
        $_SESSION['success'] = "Usuário registrado com sucesso!";

        if (isset($_POST['autenticacao_duas_etapas']) && $_POST['autenticacao_duas_etapas'] == 1) {
            $userid = $mysqli->insert_id;
            $codigo_autenticacao = rand(100000, 999999);

            $sql_update = "UPDATE usuarios SET autenticacao_habilitada=1, codigo_autenticacao='$codigo_autenticacao' WHERE id=$userid";
            $mysqli->query($sql_update);

            $_SESSION['message'] = "Autenticação em duas etapas habilitada. Um código de autenticação foi enviado para você.";
            header('Location: autenticacao.php');
            exit();
        } else {
            header('Location: login.php');
            exit();
        }
    } else {
        $_SESSION['error'] = "Erro ao registrar o usuário: " . $mysqli->error;
    }
}

$mysqli->close();
?>

<!DOCTYPE html>
<html lang="pt-BR">
<head>
    <meta charset="UTF-8">
    <title>Registro de Usuário</title>
    <script>
        function verificarSenha() {
            var senha = document.getElementById('senha').value;
            var confirmSenha = document.getElementById('confirm_senha').value;
            var mensagem = document.getElementById('mensagem-senha');
            var forte = /^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[@$!%*?&])[A-Za-z\d@$!%*?&]{8,}$/;

            if (senha !== confirmSenha) {
                mensagem.style.color = 'red';
                mensagem.textContent = 'As senhas não coincidem.';
                return false;
            }

            if (forte.test(senha)) {
                mensagem.style.color = 'green';
                mensagem.textContent = 'Senha forte.';
                return true;
```

```
        } else {
            mensagem.style.color = 'red';
            mensagem.textContent = 'A senha deve ter pelo menos 8 caracteres, incluindo letras maiúsculas, minúsculas, números e caracteres especiais.';
            return false;
        }
    }
  </script>
</head>
<body>
  <h2>Registro de Usuário</h2>
  <?php if (isset($_SESSION['error'])): ?>
    <p style="color: red;"><?php echo $_SESSION['error']; ?></p>
    <?php unset($_SESSION['error']); ?>
  <?php endif; ?>
  <?php if (isset($_SESSION['success'])): ?>
    <p style="color: green;"><?php echo $_SESSION['success']; ?></p>
    <?php unset($_SESSION['success']); ?>
  <?php endif; ?>
  <form action="register.php" method="post" onsubmit="return verificarSenha();">
    <label for="username">Nome de Usuário:</label><br>
    <input type="text" id="username" name="username" required><br><br>
    <label for="email">E-mail:</label><br>
    <input type="email" id="email" name="email" required><br><br>
    <label for="senha">Senha:</label><br>
    <input type="password" id="senha" name="senha" required oninput="verificarSenha();"><br><br>
    <label for="confirm_senha">Confirme a Senha:</label><br>
    <input type="password" id="confirm_senha" name="confirm_senha" required oninput="verificarSenha();"><br>
    <span id="mensagem-senha"></span><br><br>
    <label>
      <input type="checkbox" name="autenticacao_duas_etapas" value="1"> Habilitar Autenticação em Duas Etapas
    </label><br><br>
    <input type="submit" value="Registrar">
  </form>
</body>
</html>
```

`login.php`

```php
<?php


session_start();

require_once 'db.php';
```

```
if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    $username = $_POST['username'];
    $senha = $_POST['senha'];

    $sql = "SELECT * FROM usuarios WHERE username='$username'";
    $result = $mysqli->query($sql);

    if ($result->num_rows == 1) {
        $user = $result->fetch_assoc();
        if (password_verify($senha, $user['senha'])) {
            $_SESSION['userid'] = $user['id'];
            $_SESSION['username'] = $user['username'];

            if ($user['autenticacao_habilitada'] == 1) {
                $_SESSION['codigo_autenticacao'] = $user['codigo_autenticacao'];
                header('Location: autenticacao.php');
                exit();
            } else {
                header('Location: dashboard.php');
                exit();
            }
        } else {
            $_SESSION['error'] = "Senha incorreta.";
        }
    } else {
        $_SESSION['error'] = "Usuário não encontrado.";
    }
}

$mysqli->close();
?>

<!DOCTYPE html>
<html lang="pt-BR">
<head>
    <meta charset="UTF-8">
    <title>Login</title>
</head>
<body>
    <h2>Login</h2>
    <?php if (isset($_SESSION['error'])): ?>
        <p style="color: red;"><?php echo $_SESSION['error']; ?></p>
        <?php unset($_SESSION['error']); ?>
    <?php endif; ?>
    <form action="login.php" method="post">
        <label for="username">Nome de Usuário:</label><br>
        <input type="text" id="username" name="username" required><br><br>
        <label for="senha">Senha:</label><br>
        <input type="password" id="senha" name="senha" required><br><br>
        <input type="submit" value="Login">
    </form>
```

```
</body>
</html>
```

`autenticacao.php`

```php
<?php
session_start();

if (!isset($_SESSION['codigo_autenticacao'])) {
    header('Location: login.php');
    exit();
}

if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    $codigo = $_POST['codigo'];

    if ($codigo == $_SESSION['codigo_autenticacao']) {
        unset($_SESSION['codigo_autenticacao']);
        header('Location: dashboard.php');
        exit();
    } else {
        $_SESSION['error'] = "Código de autenticação incorreto.";
    }
}
?>

<!DOCTYPE html>
<html lang="pt-BR">
<head>
    <meta charset="UTF-8">
    <title>Autenticação em Duas Etapas</title>
    <script>
        function mostrarModal() {
            document.getElementById('modal').style.display = 'block';
        }

        function fecharModal() {
            document.getElementById('modal').style.display = 'none';
        }
    </script>
</head>
<body>
    <h2>Autenticação em Duas Etapas</h2>
    <?php if (isset($_SESSION['error'])): ?>
        <p style="color: red;"><?php echo $_SESSION['error']; ?></p>
        <?php unset($_SESSION['error']); ?>
    <?php endif; ?>
    <p>Por favor, insira o código de autenticação enviado para você.</p>
    <form action="autenticacao.php" method="post">
```

```
      <input type="text" name="codigo" required><br><br>
      <input type="submit" value="Verificar Código">
    </form>
    <button onclick="mostrarModal()">Ver Código de Autenticação</button>
    <div id="modal" style="display: none;">
      <div>
        <p>Código de Autenticação: <?php echo $_SESSION['codigo_autenticacao']; ?></p>
        <button onclick="fecharModal()">Fechar</button>
      </div>
    </div>
</body>
</html>
```

`dashboard.php`

```php
<?php
session_start();

if (!isset($_SESSION['userid'])) {
    header('Location: login.php');
    exit();
}
?>

<!DOCTYPE html>
<html lang="pt-BR">
<head>
    <meta charset="UTF-8">
    <title>Dashboard</title>
</head>
<body>
    <h2>Bem-vindo, <?php echo $_SESSION['username']; ?></h2>
    <p>Você está autenticado com sucesso!</p>
    <a href="logout.php">Sair</a>
</body>
</html>
```

`logout.php`

```php
<?php
session_start();
session_unset();
session_destroy();
header('Location: login.php');
?>
```

Script SQL para a Tabela de Usuários

```sql
CREATE TABLE usuarios (
    id INT AUTO_INCREMENT PRIMARY KEY,
    username VARCHAR(50) NOT NULL UNIQUE,
    email VARCHAR(100) NOT NULL UNIQUE,
    senha VARCHAR(255) NOT NULL,
    autenticacao_habilitada TINYINT(1) DEFAULT 0,
    codigo_autenticacao VARCHAR(6) DEFAULT NULL
);
```

Fluxo de Testes

1. Configuração Inicial:
- Certifique-se de que o servidor Apache e o MySQL estão rodando no XAMPP.
- Crie o banco de dados `seginf` e a tabela `usuarios` usando o script SQL fornecido.
- Coloque os arquivos PHP na pasta `htdocs` do XAMPP.

2. Registro de Usuário:
- Acesse `register.php` no navegador.
- Preencha os campos do formulário e registre um novo usuário.
- Verifique se a senha é forte e se as senhas coincidem.

3. Autenticação em Duas Etapas:
- Se habilitado, após o registro, o sistema redirecionará para `autenticacao.php`.
- Insira o código de autenticação exibido no modal para completar a autenticação.

4. Login de Usuário:
- Acesse `login.php` e faça login com as credenciais registradas.
- Se o 2FA estiver habilitado, o sistema pedirá o código de autenticação.

5. Dashboard:
- Após o login e a autenticação (se aplicável), o usuário será redirecionado para `dashboard.php`.

6. Logout:
- Na página de dashboard, clique no link "Sair" para fazer logout e retornar à página de login.

Conclusão

O aplicativo agora está completo com uma interface de registro robusta, que inclui verificação de senha forte, confirmação de senha, prevenção de registro duplicado e autenticação em duas etapas opcional. O fluxo de teste garante que todas as funcionalidades principais sejam verificadas.

Aqui estão as funções principais do aplicativo, junto com uma breve descrição de cada uma:

Funções do Aplicativo

Registro de Usuário
- Função: `verificarSenha()`

- Descrição: Verifica se a senha e a confirmação de senha coincidem e se a senha é forte (contendo letras maiúsculas, minúsculas, números e caracteres especiais).
- Local: `register.php`
- Uso: Chamada no evento `oninput` dos campos de senha e confirmação de senha e no evento `onsubmit` do formulário de registro.

Login de Usuário
- Função: `header('Location: autenticacao.php')`
- Descrição: Redireciona o usuário para a página de autenticação em duas etapas se a autenticação estiver habilitada.
- Local: `login.php`
- Uso: Após a verificação de credenciais e se a autenticação em duas etapas estiver habilitada para o usuário.

Autenticação em Duas Etapas (2FA)
- Função: `mostrarModal()`
- Descrição: Exibe o modal contendo o código de autenticação.
- Local: `autenticacao.php`
- Uso: Chamada no evento `onclick` do botão para mostrar o modal com o código de autenticação.

- Função: `fecharModal()`
- Descrição: Fecha o modal exibindo o código de autenticação.
- Local: `autenticacao.php`
- Uso: Chamada no evento `onclick` do botão para fechar o modal.

Sessões e Segurança
- Função: `session_start()`
- Descrição: Inicia uma nova sessão ou resume a sessão existente.
- Local: Em todos os arquivos PHP onde sessões são manipuladas (`register.php`, `login.php`, `autenticacao.php`, `dashboard.php`, `logout.php`).
- Uso: Para gerenciar a sessão do usuário, armazenando informações como `userid`, `username`, `codigo_autenticacao`.

- Função: `session_unset()` e `session_destroy()`
- Descrição: Limpa e destrói todas as variáveis de sessão.
- Local: `logout.php`
- Uso: Utilizado para finalizar a sessão do usuário ao fazer logout.

Banco de Dados
- Função: `new mysqli()`
- Descrição: Cria uma nova conexão com o banco de dados MySQL.
- Local: `db.php`
- Uso: Para estabelecer a conexão com o banco de dados em todos os arquivos PHP.

- Função: `$mysqli->query()`
- Descrição: Executa uma consulta SQL no banco de dados.
- Local: Em todos os arquivos PHP onde consultas ao banco de dados são feitas (`register.php`, `login.php`).
- Uso: Para verificar usuários existentes, inserir novos registros, atualizar informações de autenticação, etc.

- Função: `$mysqli->close()`
  - Descrição: Fecha a conexão com o banco de dados.
  - Local: No final de arquivos PHP que interagem com o banco de dados (`register.php`, `login.php`).
  - Uso: Para liberar recursos após a execução de operações no banco de dados.

Fluxo Resumido do Aplicativo

1. Registro de Usuário
  - Verifica a força e a correspondência da senha.
  - Verifica duplicidade de nome de usuário e e-mail.
  - Insere novo usuário no banco de dados.
  - Gera código de autenticação se 2FA estiver habilitado.
  - Redireciona para a página de autenticação ou login.

2. Login de Usuário
  - Verifica credenciais.
  - Redireciona para a página de autenticação se 2FA estiver habilitado.
  - Redireciona para o dashboard se a autenticação for bem-sucedida.

3. Autenticação em Duas Etapas
  - Exibe modal com o código de autenticação.
  - Verifica o código de autenticação.
  - Redireciona para o dashboard se o código estiver correto.

4. Dashboard
  - Exibe informações do usuário autenticado.
  - Permite logout, encerrando a sessão.

5. Logout
  - Limpa e encerra a sessão do usuário.

Bibliotecas Utilizadas

- PHP: Linguagem de programação para o backend.
- MySQL: Sistema de gerenciamento de banco de dados.
- XAMPP: Plataforma de desenvolvimento que inclui Apache, MySQL, PHP e Perl.
- HTML e CSS: Para a interface do usuário.

Variáveis Importantes

- `$_SESSION`: Usada para armazenar dados de sessão, como `userid`, `username`, `codigo_autenticacao`.
- `$mysqli`: Objeto de conexão com o banco de dados MySQL.
- `$username`, `$email`, `$senha`, `$confirm_senha`: Variáveis para capturar dados do formulário de registro.
- `$codigo`: Variável para capturar o código de autenticação inserido pelo usuário.

Este resumo abrange todas as principais funcionalidades, funções, bibliotecas, e variáveis utilizadas no aplicativo. Ele deve ajudar a entender o fluxo e a estrutura do aplicativo, bem como os mecanismos de segurança implementados.