

# Trabalho

## 1. Deleção em Cascata no MySQL

```
-- Criação da tabela `codigos_verificacao`
CREATE TABLE IF NOT EXISTS codigos_verificacao (
    id INT AUTO_INCREMENT PRIMARY KEY,
    usuario_id INT NOT NULL,
    codigo VARCHAR(6) NOT NULL,
    criado_em TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (usuario_id) REFERENCES usuarios(id) ON DELETE CASCADE
);

-- Criação da tabela `codigos_redefinicao`
CREATE TABLE IF NOT EXISTS codigos_redefinicao (
    usuario_id INT PRIMARY KEY,
    codigo VARCHAR(6),
    criado_em TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (usuario_id) REFERENCES usuarios(id) ON DELETE CASCADE
);
```

## 2. Hash de Senha

```
if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    $email = $_POST['email'];
    $codigo = $_POST['codigo'];
    $nova_senha = password_hash(password: $_POST['nova_senha'], algo: PASSWORD_BCRYPT);
```

## 3. Sessions

```
<?php
session_start();
include 'conexao.php';
```

## 4. Cookies

```
// Verifica o cookie de contador de visitas
if (isset($_COOKIE['login_count'])) {
    $loginCount = $_COOKIE['login_count'] + 1; // Incrementa o contador
} else {
    $loginCount = 1; // Primeira visita
}

// Atualiza o cookie com o novo valor e define validade de 30 dias
setcookie(name: 'login_count', value: $loginCount, expires_or_options: time() + (30 * 24 * 60 * 60), path: "/");
```

## 5. Functions (PHP e JavaScript)

php

```
2 references
function enviarCodigoEmail($destinatario, $codigo): void {
    $mail = new PHPMailer(exceptions: true);
```

JavaScript

```
<script>
    function verificarForcaSenha() {
        const senha = document.getElementById("senha").value;
        const forcaSenha = document.getElementById("forcaSenha");
```

6. Include e Require

```
<?php
session_start();
include 'conexao.php';
include 'mail.php'; // Arquivo PHPMailer configurado
```

7. Conexão mysqli

```
<?php
$host = 'localhost';
$usuario = 'root';
$senha = '';
$banco = 'autenticacao';
$porta = 7306;

$conexao = new mysqli(hostname: $host, username: $usuario, password: $senha, database: $banco, port: $porta);
if ($conexao->connect_error) {
    die("Erro na conexão: " . $conexao->connect_error);
}
?>
```

8. Try...catch

```

try {
    $mail->isSMTP();
    $mail->Host = 'smtp.gmail.com';
    $mail->SMTPAuth = true;

    $mail->Username = 'Seu@gmail.com'; //Preencher com e-mail para envio
    $mail->Password = '-----'; //Preencher a senha do app

    $mail->SMTPSecure = PHPMailer::ENCRYPTION_SMTPS;
    $mail->Port = 465;

    // Define o idioma e o charset
    $mail->setLanguage(langcode: 'pt_br');
    $mail->CharSet = 'UTF-8';

    $mail->setFrom(address: 'Seu@gmail.com', name: 'Sistema de Login'); //Preencher com e-mail para envio
    $mail->addAddress(address: $destinatario);

    $mail->isHTML(isHtml: true);
    $mail->Subject = 'Código de Verificação';
    $mail->Body = "Seu código de verificação é: <b>$codigo</b>";

    $mail->send();
    echo 'Código enviado com sucesso!';
} catch (Exception $e) {
    echo "Erro ao enviar código: {$mail->ErrorInfo}";
}

```

## 9. Estruturas if, elseif e else

```

<script>
    function verificarForcaSenha() {
        const senha = document.getElementById("senha").value;
        const forcaSenha = document.getElementById("forcaSenha");

        let forca = 0;
        if (senha.length >= 8) forca++;
        if (/[A-Z]/.test(senha)) forca++;
        if (/[a-z]/.test(senha)) forca++;
        if (/[0-9]/.test(senha)) forca++;
        if (/[\W]/.test(senha)) forca++;

        if (forca <= 2) {
            forcaSenha.textContent = "Fraca";
            forcaSenha.className = "strength weak";
        } else if (forca === 3 || forca === 4) {
            forcaSenha.textContent = "Média";
            forcaSenha.className = "strength medium";
        } else {
            forcaSenha.textContent = "Forte";
            forcaSenha.className = "strength strong";
        }
    }
}

```

## 10. Referências com ->

```
/**
 * Create new PHPMailer instance configured by DSN.
 *
 * @param string $dsn      DSN
 * @param bool   $exceptions Should we throw external exceptions?
 *
 * @return PHPMailer
 */
0 references | 0 overrides
public static function mailer($dsn, $exceptions = null): PHPMailer
{
    static $configurator = null;

    if (null === $configurator) {
        $configurator = new DSNConfigurator();
    }

    return $configurator->configure(mailer: new PHPMailer(exceptions: $exceptions), dsn: $dsn);
}
```

## 11. Estruturas while

```
/**
 * Check connection state.
 *
 * @return bool True if connected
 */
5 references | 0 overrides
public function connected(): bool
{
    if (is_resource(value: $this->smtp_conn)) {
        $sock_status = stream_get_meta_data(stream: $this->smtp_conn);
        if ($sock_status['eof']) {
            //The socket is valid but we are not connected
            $this->edebg(
                str: 'SMTP NOTICE: EOF caught while checking if connected',
                level: self::DEBUG_CLIENT
            );
            $this->close();

            return false;
        }

        return true; //everything looks good
    }

    return false;
}
```

## 12. Métodos POST

```

/**
 * Actually send a message via the selected mechanism.
 *
 * @throws Exception
 *
 * @return bool
 */
1 reference | 0 overrides
public function postSend(): mixed
{

```

## 13. CRUD Completo

```

// Adicionar nova questão
if ($_SERVER['REQUEST_METHOD'] === 'POST' && isset($_POST['adicionar'])) {
    $assunto = $_POST['assunto'];
    $enunciado = $_POST['enunciado'];
    $alternativa1 = $_POST['alternativa1'];
    $alternativa2 = $_POST['alternativa2'];
    $alternativa3 = $_POST['alternativa3'];
    $alternativa4 = $_POST['alternativa4'];
    $alternativa5 = $_POST['alternativa5'];
    $alternativa_correta = $_POST['alternativa_correta'];

    $sql = "INSERT INTO questoes (assunto, enunciado, alternativa1, alternativa2, alternativa3, alternativa4, alternativa5, alternativa_correta)
VALUES ('$assunto', '$enunciado', '$alternativa1', '$alternativa2', '$alternativa3', '$alternativa4', '$alternativa5', '$alternativa_correta')";

    if ($conexao->query(query: $sql) === TRUE) {
        echo "Questão adicionada com sucesso!";
    } else {
        echo "Erro: " . $sql . "<br>" . $conexao->error;
    }
}

// Editar questão existente
if ($_SERVER['REQUEST_METHOD'] === 'POST' && isset($_POST['editar'])) {
    $id = $_POST['id'];
    $assunto = $_POST['assunto'];
    $enunciado = $_POST['enunciado'];
    $alternativa1 = $_POST['alternativa1'];
    $alternativa2 = $_POST['alternativa2'];
    $alternativa3 = $_POST['alternativa3'];
    $alternativa4 = $_POST['alternativa4'];
    $alternativa5 = $_POST['alternativa5'];

    $sql = "UPDATE questoes SET assunto='$assunto', enunciado='$enunciado', alternativa1='$alternativa1', alternativa2='$alternativa2',
alternativa3='$alternativa3', alternativa4='$alternativa4', alternativa5='$alternativa5',
alternativa_correta='$alternativa_correta' WHERE id=$id";

    if ($conexao->query(query: $sql) === TRUE) {
        echo "Questão atualizada com sucesso!";
        header(header: "Location: aplicacao.php");
        exit();
    } else {
        echo "Erro: " . $sql . "<br>" . $conexao->error;
    }
}

// Deletar questão
if ($_SERVER['REQUEST_METHOD'] === 'POST' && isset($_POST['deletar'])) {
    $id = $_POST['id'];
    $conexao->query(query: "DELETE FROM questoes WHERE id=$id");
    header(header: "Location: aplicacao.php");
    exit();
}

```

## 14. Autenticação em 2 etapas

Esta validação é responsável por verificar a validade do e-mail

## 15. Finalidade da aplicação

"O objetivo principal é proporcionar aos usuários a possibilidade de se cadastrarem de forma simples e segura, além de permitir o acesso à plataforma por meio de um processo de login eficiente e protegido