



INSTITUTO SUPERIOR DO LITORAL DO PARANÁ

Prof. Luiz Efigênio

Exame – 2º Bimestre

Disciplina: Administração de Banco de Dados II

Professor: Luiz Carlos Efigênio

Avaliação: Exame 2º Bimestre

MINI-PROJETO DEADB — Sistema de Quiz com PHP + MySQL

Objetivo:

Criar um sistema simples de cadastro/login, criação de perguntas, criação de quiz e resposta de quiz.

Os alunos **não vão programar backend completo**, apenas **encher o HTML que você dará com chamadas PHP + SQL** usando *mysqli* ou *PDO*.

Toda a regra de negócio pesada será resolvida com SQL, Stored Procedures e Trigger.

Arquitetura Simplificada

- **PHP (frontend + chamadas SQL)**

- login.php
- registrar.php
- criar_pergunta.php
- criar_quiz.php
- responder_quiz.php
- resultado.php

- **MySQL**

- tabelas
- trigger
- stored procedure



- consultas SELECT/INSERT/UPDATE de apoio

BÁSICO DO PHP (para explicar aos alunos antes do mini-projeto)

1. Sintaxe geral do PHP

```
<?php  
echo "Olá, mundo!";  
?>
```

2. Variáveis

```
<?php  
$nome = "Luiz";  
$idade = 26;  
?>
```

3. Arrays

```
$frutas = ["maçã", "banana", "uva"];
```

4. Condições

```
if ($idade >= 18) {  
    echo "Maior de idade";  
} else {  
    echo "Menor de idade";  
}
```



5. Loops

```
for ($i = 0; $i < 5; $i++) {  
    echo $i;  
}
```

6. Funções

```
function somar($a, $b) {  
    return $a + $b;  
}
```

7. Trabalhando com formulários HTML

```
<form method="POST" action="pagina.php">  
    <input type="text" name="nome">  
    <button type="submit">Enviar</button>  
</form>
```

8. Recebendo dados no PHP

```
$nome = $_POST["nome"];  
$email = $_POST["email"];
```

9. Sessions (para login)

```
session_start();  
$_SESSION["user_id"] = 10;
```

10. Conexão com MySQL (mysqli)

```
$conn = new mysqli("localhost", "root", "", "deadb_quiz");

if ($conn->connect_error) {
    die("Erro: " . $conn->connect_error);
}
```

FRONTEND + PHP BASE (com queries em aberto para os alunos preencherem)

Esses são os **arquivos que você entregará prontos**, com **lacunas SQL** para os alunos completarem durante o mini-projeto.

conexao.php

```
<?php

$host = "localhost";

$user = "root";

$pass = "";

$db = "deadb_quiz";

$conn = new mysqli($host, $user, $pass, $db);
```



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

registrar.php

```
<?php  
session_start();  
include "conexao.php";  
  
if ($_SERVER["REQUEST_METHOD"] === "POST") {  
    $nome = $_POST["nome"];  
    $email = $_POST["email"];  
    $senha = $_POST["senha"];  
  
    $hash = password_hash($senha, PASSWORD_DEFAULT);  
  
    // TODO: Query SQL de cadastro  
  
    // INSERT INTO users (name, email, password_hash) VALUES  
    (?, ?, ?);  
  
    $stmt = $conn->prepare("A QUERY VAI AQUI");
```



```
$stmt->bind_param("sss", $nome, $email, $hash);

$stmt->execute();

echo "Cadastro realizado!";

}

?>

<form method="POST">

    <input type="text" name="nome" placeholder="Nome"
required><br>

    <input type="email" name="email" placeholder="Email"
required><br>

    <input type="password" name="senha" placeholder="Senha"
required><br>

    <button type="submit">Registrar</button>

</form>
```

login.php

```
<?php

session_start();

include "conexao.php";

if ($_SERVER["REQUEST_METHOD"] === "POST") {
```



```
$email = $_POST["email"];
$senha = $_POST["senha"];

// TODO: Query SQL do login

// SELECT id, name, email, password_hash FROM users WHERE
email = ?;

$stmt = $conn->prepare("A QUERY VAI AQUI");
$stmt->bind_param("s", $email);
$stmt->execute();
$result = $stmt->get_result();
$user = $result->fetch_assoc();

if ($user && password_verify($senha,
$user["password_hash"])) {
    $_SESSION["user_id"] = $user["id"];
    header("Location: criar_pergunta.php");
    exit;
} else {
    echo "Login inválido";
}
}
```



?>

```
<form method="POST">

    <input type="email" name="email" placeholder="Email"><br>

    <input type="password" name="senha"
placeholder="Senha"><br>

    <button type="submit">Entrar</button>

</form>
```

criar_pergunta.php

```
<?php

session_start();

include "conexao.php";

if (!isset($_SESSION["user_id"])) {

    die("Acesso negado.");

}

if ($_SERVER["REQUEST_METHOD"] === "POST") {

    $texto = $_POST["texto"];
```




```
// TODO: Query SQL criar pergunta
// INSERT INTO questions (text) VALUES (?);

$stmt = $conn->prepare("A QUERY VAI AQUI");
$stmt->bind_param("s", $texto);
$stmt->execute();

$question_id = $conn->insert_id;

// Cadastrar alternativas

for ($i = 1; $i <= 4; $i++) {
    $opt = $_POST["opt$i"];
    $correta = ($_POST["correta"] == $i) ? 1 : 0;

    // TODO: Query SQL criar alternativa
    // INSERT INTO options (question_id, text, is_correct)
VALUES (?, ?, ?);

    $stmt2 = $conn->prepare("A QUERY VAI AQUI");
    $stmt2->bind_param("isi", $question_id, $opt,
$correta);
```



```
$stmt2->execute();

}

echo "Pergunta criada com sucesso!";

}

?>

<form method="POST">

    <input type="text" name="texto" placeholder="Texto da
pergunta" required><br>

    <label>Alternativas:</label><br>

    <input type="text" name="opt1" required> (1)<br>
    <input type="text" name="opt2" required> (2)<br>
    <input type="text" name="opt3" required> (3)<br>
    <input type="text" name="opt4" required> (4)<br>

    <label>Correta:</label>

    <select name="correta">

        <option value="1">1</option>

        <option value="2">2</option>

        <option value="3">3</option>
```



```
<option value="4">4</option>
</select>

<button type="submit">Salvar</button>
</form>
```

criar_quiz.php

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

if ($_SERVER["REQUEST_METHOD"] === "POST") {

    $title = $_POST["title"];

    // TODO: Query SQL criar quiz
    // INSERT INTO quizzes (title) VALUES (?);

    $stmt = $conn->prepare("A QUERY VAI AQUI");
    $stmt->bind_param("s", $title);
    $stmt->execute();
```



```
$quiz_id = $conn->insert_id;

foreach ($_POST["questions"] as $qid) {

    // TODO: Query SQL vincular pergunta ao quiz

    // INSERT INTO quiz_questions (quiz_id, question_id)
VALUES (?, ?);

    $stmt2 = $conn->prepare("A QUERY VAI AQUI");
    $stmt2->bind_param("ii", $quiz_id, $qid);
    $stmt2->execute();
}

echo "Quiz criado!";
}

// Carregar perguntas
$perguntas = $conn->query("SELECT * FROM questions");
?>

<form method="POST">

    <input type="text" name="title" placeholder="Título do
INSTITUTO SUPERIOR DO LITORAL DO PARANÁ | Paranaguá
Rua João Eugênio, 534 - Costeira, Paranaguá - PR | 83203-400 – Brasil
```



quiz">

<label>Escolha as perguntas:</label>

<?php while (\$q = \$perguntas->fetch_assoc()): ?>

 <input type="checkbox" name="questions[]" value="<?=\$q['id'] ?>"> <?=\$q["text"] ?>

<?php endwhile; ?>

<button type="submit">Criar Quiz</button>

</form>

responder_quiz.php

<?php

session_start();

include "conexao.php";

\$quiz_id = \$_GET["quiz_id"];

// TODO: Query SQL para carregar as perguntas do quiz

// SELECT q.id AS question_id, q.text, o.id AS option_id,
o.text

INSTITUTO SUPERIOR DO LITORAL DO PARANÁ | Paranaguá

Rua João Eugênio, 534 - Costeira, Paranaguá - PR | 83203-400 – Brasil



```
// FROM quiz_questions qq
// INNER JOIN questions q ON q.id = qq.question_id
// INNER JOIN options o ON o.question_id = q.id
// WHERE qq.quiz_id = ?;

$stmt = $conn->prepare("A QUERY VAI AQUI");
$stmt->bind_param("i", $quiz_id);
$stmt->execute();
$result = $stmt->get_result();

// Registrar tentativa
// TODO: INSERT INTO attempts (quiz_id, user_id) VALUES (?, ?)

$stmt2 = $conn->prepare("A QUERY VAI AQUI");
$stmt2->bind_param("ii", $quiz_id, $_SESSION["user_id"]);
$stmt2->execute();
$attempt_id = $conn->insert_id;
?>

<form method="POST" action="resultado.php">
    <input type="hidden" name="attempt_id" value="<=?=$attempt_id ?>">
```



```
<?php while ($row = $result->fetch_assoc()): ?>

    <p><?= $row["text"] ?></p>

    <input type="radio" name="q<?= $row['question_id'] ?>"
value="<?= $row['option_id'] ?>"> <?= $row['option_text']
?><br>

<?php endwhile; ?>

<button type="submit">Enviar Respostas</button>

</form>
```

resultado.php

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

$attempt_id = $_POST["attempt_id"];

foreach ($_POST as $key => $value) {
    if ($key === "attempt_id") continue;

    $question_id = str_replace("q", "", $key);
    $option_id = $value;
```



```
// TODO: INSERT resposta no attempt_answers

// INSERT INTO attempt_answers (attempt_id, question_id,
option_id)

// VALUES (?, ?, ?)

$stmt = $conn->prepare("A QUERY VAI AQUI");

$stmt->bind_param("iii", $attempt_id, $question_id,
$option_id);

$stmt->execute();

}

// TODO: Calcular nota

// CALL grade_attempt(?);

$conn->query("CALL grade_attempt($attempt_id)");

$result = $conn->query("SELECT score FROM attempts WHERE id =
$attempt_id");

$row = $result->fetch_assoc();

echo "<h2>Sua nota: " . $row["score"] . "%</h2>";

?>
```




Cronograma do Mini-Projeto

Modelagem + Criação do Banco + Inserts Básicos

1. Criar banco e tabelas.
2. Inserir usuários de teste.
3. Inserir perguntas e alternativas.
4. Exercício guiado com SELECT + INNER JOIN.

PHP + Queries Essenciais

1. Criar telas de:
 - Login
 - Cadastro
 - Criar Pergunta
2. Criar as queries em PHP usando prepare().
3. Testar autenticação com sessão.
4. Criar um quiz e vincular perguntas.

Trigger + Stored Procedure + Responder Quiz

1. Criar trigger para marcar corretas automaticamente.
2. Criar procedure para calcular nota.
3. Fazer o aluno responder o quiz.
4. Mostrar nota no final.

1. CRIAÇÃO DO BANCO E TABELAS

```
CREATE DATABASE deadb_quiz;
```

```
USE deadb_quiz;
```

```
CREATE TABLE users (
```

INSTITUTO SUPERIOR DO LITORAL DO PARANÁ | Paranaguá

Rua João Eugênio, 534 - Costeira, Paranaguá - PR | 83203-400 – Brasil



```
id INT AUTO_INCREMENT PRIMARY KEY,  
name VARCHAR(120) NOT NULL,  
email VARCHAR(150) NOT NULL UNIQUE,  
password_hash VARCHAR(255) NOT NULL,  
created_at DATETIME DEFAULT CURRENT_TIMESTAMP  
);
```

```
CREATE TABLE questions (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    text VARCHAR(255) NOT NULL  
);
```

```
CREATE TABLE options (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    question_id INT NOT NULL,  
    text VARCHAR(255) NOT NULL,  
    is_correct TINYINT(1) DEFAULT 0,  
    FOREIGN KEY (question_id) REFERENCES questions(id)  
);
```



```
CREATE TABLE quizzes (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    title VARCHAR(255)  
);
```

```
CREATE TABLE quiz_questions (  
    quiz_id INT,  
    question_id INT,  
    PRIMARY KEY(quiz_id, question_id),  
    FOREIGN KEY (quiz_id) REFERENCES quizzes(id),  
    FOREIGN KEY (question_id) REFERENCES questions(id)  
);
```

```
CREATE TABLE attempts (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    quiz_id INT,  
    user_id INT,  
    score DECIMAL(5,2),  
    created_at DATETIME DEFAULT CURRENT_TIMESTAMP,  
    FOREIGN KEY (quiz_id) REFERENCES quizzes(id),  
    FOREIGN KEY (user_id) REFERENCES users(id)  
);
```



```
CREATE TABLE attempt_answers (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    attempt_id INT,  
    question_id INT,  
    option_id INT,  
    is_correct TINYINT(1),  
    FOREIGN KEY (attempt_id) REFERENCES attempts(id),  
    FOREIGN KEY (question_id) REFERENCES questions(id),  
    FOREIGN KEY (option_id) REFERENCES options(id)  
);
```

TRIGGER

Quando o aluno marca uma alternativa, o banco preenche automaticamente se está correta:

```
DELIMITER //
```

```
CREATE TRIGGER trg_mark_correct
```

```
BEFORE INSERT ON attempt_answers
```

```
FOR EACH ROW
```

```
BEGIN
```

```
    DECLARE flag TINYINT DEFAULT 0;
```

```
    SELECT is_correct INTO flag FROM options WHERE id =  
    NEW.option_id;
```



```
SET NEW.is_correct = flag;  
  
END //  
  
DELIMITER ;
```

STORED PROCEDURE

Calcula a nota final de uma tentativa:

```
DELIMITER //  
  
CREATE PROCEDURE grade_attempt(IN p_attempt INT)  
BEGIN  
  
    DECLARE total_q INT;  
  
    DECLARE total_correct INT;  
  
    DECLARE final_score DECIMAL(5,2);  
  
  
    SELECT COUNT(*) INTO total_q  
    FROM attempt_answers  
    WHERE attempt_id = p_attempt;  
  
  
    SELECT COUNT(*) INTO total_correct  
    FROM attempt_answers  
    WHERE attempt_id = p_attempt AND is_correct = 1;  
  
  
    SET final_score = (total_correct / total_q) * 100;
```



```
UPDATE attempts

    SET score = final_score

    WHERE id = p_attempt;

SELECT final_score AS score;

END //

DELIMITER ;
```

4. QUERIES SQL QUE OS ALUNOS VÃO USAR COM PHP

Cadastro

```
INSERT INTO users (name, email, password_hash) VALUES (?, ?, ?);
```

Login

```
SELECT id, name, email, password_hash FROM users WHERE email = ?;
```

Criar pergunta

```
INSERT INTO questions (text) VALUES (?);
```



Criar alternativas

```
INSERT INTO options (question_id, text, is_correct) VALUES (?, ?, ?);
```

Criar quiz

```
INSERT INTO quizzes (title) VALUES (?);
```

Adicionar perguntas ao quiz

```
INSERT INTO quiz_questions (quiz_id, question_id) VALUES (?, ?);
```

Carregar quiz para responder

```
SELECT q.id AS question_id, q.text AS question,
       o.id AS option_id, o.text AS option_text
FROM quiz_questions qq
INNER JOIN questions q ON qq.question_id = q.id
INNER JOIN options o ON o.question_id = q.id
WHERE qq.quiz_id = ?;
```

Registrar tentativa

```
INSERT INTO attempts (quiz_id, user_id) VALUES (?, ?);
```



Salvar resposta

```
INSERT INTO attempt_answers (attempt_id, question_id,  
option_id)  
  
VALUES (?, ?, ?);
```

A trigger cuidará do is_correct.

Calcular nota

```
CALL grade_attempt(?);
```

Mostrar nota

```
SELECT score FROM attempts WHERE id = ?;
```

REFERÊNCIAS BIBLIOGRÁFICAS

BARRETO, Jeanine dos Santos et al. *Fundamentos de Banco de Dados*. SAGAH.
ELMASRI, R.; NAVATHE, S. *Sistemas de Banco de Dados*. Pearson.
BEIGHLEY, Lynn. *PHP e MySQL: Desenvolvimento Web*. Alta Books.
W3Schools. *PHP Tutorial*.
PHP Manual Oficial – <https://www.php.net/manual/>