* **Dia 3: Desenvolvimento do Quiz (Front-end e Lógica Básica)**
  + Criar a interface HTML/CSS para exibir uma pergunta e as opções.
  + Usar JavaScript para exibir a próxima pergunta e lidar com cliques (inicialmente, sem AJAX, apenas um formulário).
  + Implementar a lógica PHP para buscar uma pergunta aleatória ou sequencial do DB e exibir.
  + Calcular a pontuação (muito básica por enquanto).

Perfeito! Entendido. Mais uma vez, peço desculpa por não ter fornecido a logout.php na sequência correta. A minha prioridade é entregar o que precisas de forma exata e sem falhas, especialmente com o prazo apertado.

Vamos corrigir isso agora mesmo!

**Dia 3: Criação da logout.php**

Aqui está o código completo para o ficheiro logout.php. Este é um script simples, mas essencial, para permitir que os utilizadores encerrem a sua sessão de forma segura.

PHP

<?php

session\_start(); // Inicia a sessão PHP existente

// 1. Destrói todas as variáveis de sessão

// Isso remove todos os dados armazenados na sessão (como 'id\_usuario', 'nome\_usuario', 'pontuacao', etc.)

$\_SESSION = array();

// 2. Se for preciso destruir os cookies de sessão, apaga também o cookie de sessão.

// Isso é importante para garantir que o identificador da sessão no navegador do utilizador seja removido.

// Nota: Isto irá destruir a sessão, e não apenas os dados da sessão!

if (ini\_get("session.use\_cookies")) {

$params = session\_get\_cookie\_params();

setcookie(session\_name(), '', time() - 42000, // Define o tempo de expiração no passado

$params["path"], $params["domain"],

$params["secure"], $params["httponly"]

);

}

// 3. Finalmente, destrói a sessão no servidor.

session\_destroy();

// 4. Redireciona o utilizador para a página de login ou para a landing page após o logout

header("Location: landing.php"); // Podes mudar para 'login.php' se preferires que vá diretamente para o ecrã de login

exit(); // Garante que nenhum outro código PHP seja executado após o redirecionamento

?>

**Como usar:**

1. No teu VS Code, na pasta C:\xampp\htdocs\quiz\_angola, **cria um novo ficheiro chamado logout.php**.
2. **Cola o código acima** dentro desse ficheiro.
3. **Guarda o ficheiro.**

Teste:

Depois de teres feito login (para criar uma sessão), clica no botão "Sair" na dashboard.php ou acede diretamente http://localhost/quiz\_angola/logout.php. Deverás ser automaticamente redirecionado para a landing.php (ou login.php, dependendo da linha que escolheste no código).

Com a logout.php implementada, temos agora o ciclo completo de autenticação e podemos focar-nos totalmente na funcionalidade do quiz.

**Conseguiste implementar e testar o logout.php?**

Perfeito! Entendido. Peço desculpa por ter assumido que já o tinhas implementado, com a urgência do projeto, o meu foco deve ser sempre garantir que temos tudo o que é necessário passo a passo.

Vamos criar a página dashboard.php agora mesmo. Ela será o ponto de entrada principal para o utilizador após fazer login, oferecendo uma saudação personalizada e as opções para iniciar o quiz ou sair da sessão.

**Dia 3: Criação da dashboard.php**

Aqui está o código completo para o ficheiro dashboard.php. Este código inclui:

* A verificação de sessão para garantir que apenas utilizadores logados acedam a esta página.
* Recupera o nome do utilizador da sessão para uma saudação personalizada.
* Links para a página do quiz (quiz\_game.php) e para a página de logout (logout.php).
* O CSS para manter o design consistente com as cores da bandeira de Angola.

PHP

<?php

session\_start(); // Inicia a sessão PHP

// Verifica se o utilizador está logado (se o ID de utilizador está definido na sessão)

if (!isset($\_SESSION['id\_usuario'])) {

// Se não estiver logado, redireciona para a página de login

header("Location: login.php");

exit(); // Termina a execução do script para garantir o redirecionamento

}

// Obtém o nome do utilizador da sessão para exibir uma saudação personalizada

$nome\_usuario = $\_SESSION['nome\_usuario'];

?>

<!DOCTYPE html>

<html lang="pt">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Dashboard - Quiz Conheça Angola</title>

<style>

body {

font-family: 'Arial', sans-serif;

background: linear-gradient(to bottom, #EF0000 50%, #000000 50%); /\* Fundo com as cores da bandeira de Angola \*/

color: #fff;

display: flex;

justify-content: center;

align-items: center;

min-height: 100vh;

margin: 0;

text-align: center;

}

.container {

background-color: rgba(255, 255, 255, 0.95); /\* Fundo branco semi-transparente \*/

padding: 40px 50px;

border-radius: 10px;

box-shadow: 0 5px 20px rgba(0, 0, 0, 0.3); /\* Sombra para profundidade \*/

max-width: 700px;

width: 90%;

color: #333; /\* Cor do texto principal \*/

}

h1 {

font-size: 2.5em;

margin-bottom: 20px;

color: #EF0000; /\* Vermelho vibrante para títulos \*/

}

p {

font-size: 1.2em;

line-height: 1.6;

margin-bottom: 30px;

color: #555; /\* Cor do texto secundário \*/

}

.buttons-container {

display: flex;

justify-content: center;

gap: 20px; /\* Espaço entre os botões \*/

margin-top: 30px;

}

.btn {

display: inline-block;

padding: 15px 30px;

border-radius: 5px;

text-decoration: none;

font-weight: bold;

font-size: 1.1em;

transition: background-color 0.3s ease, transform 0.2s ease;

box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2); /\* Sombra nos botões \*/

}

.btn-primary {

background-color: #FFCD00; /\* Amarelo da bandeira \*/

color: #000000; /\* Texto preto para contraste \*/

border: 2px solid #CCA300; /\* Borda mais escura do amarelo \*/

}

.btn-primary:hover {

background-color: #CCA300;

transform: translateY(-2px); /\* Efeito de elevação ao passar o rato \*/

}

.btn-secondary {

background-color: #000000; /\* Preto da bandeira \*/

color: white;

border: 2px solid #333333; /\* Borda mais escura do preto \*/

}

.btn-secondary:hover {

background-color: #333333;

transform: translateY(-2px); /\* Efeito de elevação ao passar o rato \*/

}

</style>

</head>

<body>

<div class="container">

<h1>Bem-vindo, <?php echo htmlspecialchars($nome\_usuario); ?>!</h1>

<p>Está pronto para testar os seus conhecimentos sobre Angola?</p>

<div class="buttons-container">

<a href="quiz\_game.php" class="btn btn-primary">Iniciar Quiz</a>

<a href="logout.php" class="btn btn-secondary">Sair</a>

</div>

</div>

</body>

</html>

**Passos para ti:**

1. No VS Code, na tua pasta C:\xampp\htdocs\quiz\_angola, **cria um novo ficheiro chamado dashboard.php**.
2. **Cola o código acima** nesse ficheiro.
3. **Guarda o ficheiro.**
4. **Testa a página:**
   * Vai para http://localhost/quiz\_angola/login.php.
   * Faz login com uma conta que já registaste.
   * Deverás ser redirecionado para esta nova página dashboard.php. Vê se a saudação aparece com o teu nome de utilizador.

Uma vez que esta página esteja a funcionar corretamente, teremos todos os componentes-chave para o fluxo de autenticação e início do jogo. Depois disso, podemos focar-nos nos próximos passos para aprimorar o quiz!

**Conseguiste implementar e testar a dashboard.php?**

Peço desculpa, novamente. A minha memória de contexto sobre os ficheiros específicos em que estamos a trabalhar falhou. Tens toda a razão. Não existe quiz\_game.php nem o index.php do quiz ainda, pois a nossa prioridade foi a autenticação.

Muito obrigado por me corrigires e por me manteres no caminho certo. A partir de agora, serei ainda mais rigoroso com os nomes dos ficheiros e o plano que me deste.

**Dia 3: Desenvolvimento do Quiz (Front-end e Lógica Básica)**

Vamos implementar **exatamente** o que definimos para o Dia 3, criando os ficheiros necessários e a lógica do quiz.

**Objetivos para hoje:**

1. **Criar a interface HTML/CSS para exibir uma pergunta e as opções.**
2. **Usar JavaScript para lidar com cliques e submissão de respostas.**
3. **Implementar a lógica PHP para buscar uma pergunta aleatória ou sequencial do DB e exibir.**
4. **Calcular a pontuação (muito básica por enquanto).**

Para cumprir estes pontos, vamos criar um novo ficheiro que será a página principal do quiz após o login. Poderíamos chamá-lo de quiz\_page.php ou play\_quiz.php. Para seguir a linha do nosso plano de autenticação, e como o dashboard.php já aponta para quiz\_game.php, vamos criar o ficheiro com esse nome.

**1. Criação do Ficheiro quiz\_game.php**

Cria um novo ficheiro chamado **quiz\_game.php** na tua pasta quiz\_angola (C:\xampp\htdocs\quiz\_angola) e cola o seguinte código. Este é o código completo para a página do quiz, incluindo a lógica PHP, HTML, CSS e o JavaScript para a interação:

PHP

<?php

session\_start(); // Inicia a sessão

// --- Lógica de Proteção da Página (garante que só utilizadores logados acedem) ---

if (!isset($\_SESSION['id\_usuario'])) {

header("Location: login.php"); // Redireciona para a página de login se não estiver logado

exit();

}

// --- Fim da Lógica de Proteção ---

require\_once 'db\_connect.php'; // Inclui o ficheiro de conexão com a base de dados

// Inicializa a pontuação e o controlo de perguntas na sessão, se ainda não existirem

if (!isset($\_SESSION['pontuacao'])) {

$\_SESSION['pontuacao'] = 0;

$\_SESSION['perguntas\_respondidas'] = []; // Armazena os IDs das perguntas já mostradas nesta sessão de jogo

}

$mensagem\_feedback = ''; // Para mensagens de "Correto!" ou "Incorreto!"

// --- Lógica para Processar a Resposta do Utilizador (quando o formulário é submetido) ---

if ($\_SERVER['REQUEST\_METHOD'] == 'POST' && isset($\_POST['id\_pergunta']) && isset($\_POST['resposta'])) {

$id\_pergunta\_respondida = (int)$\_POST['id\_pergunta'];

$id\_resposta\_selecionada = (int)$\_POST['resposta'];

// Buscar a resposta correta para a pergunta submetida

$stmt = $conn->prepare("SELECT id\_resposta FROM respostas WHERE id\_pergunta = ? AND correta = 1 LIMIT 1");

$stmt->bind\_param("i", $id\_pergunta\_respondida);

$stmt->execute();

$result = $stmt->get\_result();

$resposta\_correta\_db = $result->fetch\_assoc();

if ($resposta\_correta\_db) {

if ($id\_resposta\_selecionada == $resposta\_correta\_db['id\_resposta']) {

$\_SESSION['pontuacao'] += 10; // Adiciona 10 pontos por resposta correta

$mensagem\_feedback = "<p class='feedback correta'>Correto! Pontuação: " . $\_SESSION['pontuacao'] . "</p>";

} else {

// Se incorreto, podemos também mostrar qual era a resposta certa

$stmt\_texto\_correto = $conn->prepare("SELECT texto\_resposta FROM respostas WHERE id\_resposta = ? LIMIT 1");

$stmt\_texto\_correto->bind\_param("i", $resposta\_correta\_db['id\_resposta']);

$stmt\_texto\_correto->execute();

$result\_texto\_correto = $stmt\_texto\_correto->get\_result();

$texto\_resposta\_correta = $result\_texto\_correto->fetch\_assoc()['texto\_resposta'];

$stmt\_texto\_correto->close();

$mensagem\_feedback = "<p class='feedback incorreta'>Incorreto. A resposta correta era: \*\*" . htmlspecialchars($texto\_resposta\_correta) . "\*\*.</p>";

$mensagem\_feedback .= "<p class='feedback incorreta'>Pontuação: " . $\_SESSION['pontuacao'] . "</p>";

}

} else {

$mensagem\_feedback = "<p class='feedback incorreta'>Erro ao verificar a resposta. Pontuação: " . $\_SESSION['pontuacao'] . "</p>";

}

$stmt->close();

// Adiciona a pergunta respondida à lista para não a repetir nesta sessão de jogo

$\_SESSION['perguntas\_respondidas'][] = $id\_pergunta\_respondida;

}

// --- Lógica para Obter a Próxima Pergunta ---

$proxima\_pergunta = null;

$ids\_respondidos\_string = ''; // String de IDs para a cláusula NOT IN do SQL

// Prepara a string de IDs de perguntas já respondidas

if (!empty($\_SESSION['perguntas\_respondidas'])) {

// Garante que todos os IDs são inteiros para segurança no SQL

$safe\_ids = array\_map('intval', $\_SESSION['perguntas\_respondidas']);

$ids\_respondidos\_string = implode(',', $safe\_ids);

}

// Primeiro, tenta obter uma pergunta que AINDA NÃO FOI RESPONDIDA

$sql\_pergunta = "SELECT id\_pergunta, texto\_pergunta FROM perguntas";

if (!empty($ids\_respondidos\_string)) {

$sql\_pergunta .= " WHERE id\_pergunta NOT IN ($ids\_respondidos\_string)";

}

$sql\_pergunta .= " ORDER BY RAND() LIMIT 1"; // Ordena aleatoriamente para variar as perguntas

$result\_pergunta = $conn->query($sql\_pergunta);

if ($result\_pergunta && $result\_pergunta->num\_rows > 0) {

$proxima\_pergunta = $result\_pergunta->fetch\_assoc();

$id\_pergunta\_atual = $proxima\_pergunta['id\_pergunta'];

// Obter as respostas associadas à pergunta atual

$stmt\_respostas = $conn->prepare("SELECT id\_resposta, texto\_resposta FROM respostas WHERE id\_pergunta = ?");

$stmt\_respostas->bind\_param("i", $id\_pergunta\_atual);

$stmt\_respostas->execute();

$result\_respostas = $stmt\_respostas->get\_result();

$respostas\_para\_exibir = [];

while($row\_resposta = $result\_respostas->fetch\_assoc()) {

$respostas\_para\_exibir[] = $row\_resposta;

}

shuffle($respostas\_para\_exibir); // Mistura a ordem das opções de resposta

$stmt\_respostas->close();

} else {

// Se não há mais perguntas NÃO RESPONDIDAS no DB

$total\_perguntas\_db = $conn->query("SELECT COUNT(\*) FROM perguntas")->fetch\_row()[0];

if ($total\_perguntas\_db > 0) { // Se há perguntas no DB mas todas foram respondidas

$mensagem\_feedback .= "<p class='feedback'>Você respondeu a todas as perguntas disponíveis! O quiz será reiniciado com novas perguntas.</p>";

$\_SESSION['perguntas\_respondidas'] = []; // Reinicia a lista de perguntas respondidas

// Tenta buscar uma pergunta novamente (agora que a lista foi resetada)

header("Location: quiz\_game.php"); // Redireciona para recarregar com uma nova pergunta

exit();

} else { // Se não há perguntas cadastradas no DB

$proxima\_pergunta = null;

$mensagem\_feedback = "<p class='feedback'>Não há perguntas disponíveis no quiz. Pontuação final: \*\*" . $\_SESSION['pontuacao'] . "\*\*.</p>";

}

}

$conn->close();

?>

<!DOCTYPE html>

<html lang="pt">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Jogar Quiz - Quiz Conheça Angola</title>

<style>

body {

font-family: 'Arial', sans-serif;

background: linear-gradient(to bottom, #EF0000 50%, #000000 50%); /\* Cores da bandeira \*/

color: #fff;

display: flex;

justify-content: center;

align-items: center;

min-height: 100vh;

margin: 0;

text-align: center;

}

.quiz-container {

background-color: rgba(255, 255, 255, 0.95);

padding: 30px;

border-radius: 10px;

box-shadow: 0 5px 20px rgba(0, 0, 0, 0.3);

width: 90%;

max-width: 600px;

color: #333;

text-align: left;

position: relative; /\* Para a pontuação \*/

}

h1 {

font-size: 2.2em;

margin-bottom: 25px;

color: #EF0000; /\* Vermelho \*/

text-align: center;

}

.pergunta-texto {

font-size: 1.5em;

margin-bottom: 30px;

color: #555;

text-align: center;

}

.respostas-form {

display: flex;

flex-direction: column;

gap: 15px;

}

.resposta-btn {

background-color: #FFCD00; /\* Amarelo \*/

color: #000000; /\* Preto \*/

padding: 15px;

border: 2px solid #CCA300;

border-radius: 5px;

font-size: 1.1em;

cursor: pointer;

transition: background-color 0.3s ease, transform 0.2s ease;

box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);

}

.resposta-btn:hover {

background-color: #CCA300;

transform: translateY(-2px);

}

.resposta-btn.selected {

background-color: #EF0000; /\* Vermelho quando selecionado \*/

color: white;

border-color: #A00000;

}

.feedback {

margin-top: 20px;

padding: 10px;

border-radius: 5px;

text-align: center;

font-weight: bold;

}

.feedback.correta {

background-color: #d4edda;

color: #155724;

border: 1px solid #c3e6cb;

}

.feedback.incorreta {

background-color: #f8d7da;

color: #721c24;

border: 1px solid #f5c6cb;

}

.submit-btn {

background-color: #000000; /\* Preto \*/

color: white;

padding: 15px 30px;

border: none;

border-radius: 5px;

font-size: 1.1em;

cursor: pointer;

margin-top: 20px;

transition: background-color 0.3s ease;

}

.submit-btn:hover {

background-color: #333333;

}

.current-score {

position: absolute;

top: 20px;

right: 20px;

background-color: #FFCD00;

color: #000000;

padding: 10px 15px;

border-radius: 5px;

font-weight: bold;

box-shadow: 0 2px 5px rgba(0, 0, 0, 0.2);

}

.nav-links {

text-align: center;

margin-top: 25px;

}

.nav-links a {

color: #EF0000; /\* Vermelho \*/

text-decoration: none;

font-weight: bold;

margin: 0 10px;

transition: color 0.3s ease;

}

.nav-links a:hover {

color: #A00000;

text-decoration: underline;

}

</style>

</head>

<body>

<div class="quiz-container">

<div class="current-score">Pontuação: <?php echo $\_SESSION['pontuacao']; ?></div>

<h1>Quiz Conheça Angola</h1>

<?php echo $mensagem\_feedback; // Exibe feedback da resposta anterior ?>

<?php if ($proxima\_pergunta): ?>

<p class="pergunta-texto"><?php echo htmlspecialchars($proxima\_pergunta['texto\_pergunta']); ?></p>

<form action="quiz\_game.php" method="POST" id="quizForm">

<input type="hidden" name="id\_pergunta" value="<?php echo $proxima\_pergunta['id\_pergunta']; ?>">

<div class="respostas-form" id="respostas-container">

<?php foreach ($respostas\_para\_exibir as $resposta): ?>

<button type="button" class="resposta-btn" data-id-resposta="<?php echo $resposta['id\_resposta']; ?>">

<?php echo htmlspecialchars($resposta['texto\_resposta']); ?>

</button>

<?php endforeach; ?>

</div>

<input type="hidden" name="resposta" id="selected-answer">

<button type="submit" class="submit-btn" id="submit-btn" disabled>Confirmar Resposta</button>

</form>

<?php else: ?>

<p class="pergunta-texto">O quiz terminou! Não há mais perguntas disponíveis. Pontuação final: \*\*<?php echo $\_SESSION['pontuacao']; ?>\*\*.</p>

<div class="nav-links">

<a href="dashboard.php">Voltar ao Painel</a> |

<a href="logout.php">Sair</a>

</div>

<?php endif; ?>

<div class="nav-links">

<?php if ($proxima\_pergunta): ?>

<a href="dashboard.php">Voltar ao Painel</a> |

<?php endif; ?>

<a href="logout.php">Sair</a>

</div>

</div>

<script>

const respostaBotoes = document.querySelectorAll('.resposta-btn');

const submitBtn = document.getElementById('submit-btn');

const selectedAnswerInput = document.getElementById('selected-answer');

// Ativa o botão de submeter e marca a resposta selecionada

respostaBotoes.forEach(botao => {

botao.addEventListener('click', function() {

// Remove a seleção de todos os botões

respostaBotoes.forEach(btn => btn.classList.remove('selected'));

// Adiciona a seleção ao botão clicado

this.classList.add('selected');

// Atribui o ID da resposta ao campo hidden que será submetido

selectedAnswerInput.value = this.dataset.idResposta;

// Ativa o botão de submeter

submitBtn.disabled = false;

});

});

</script>

</body>

</html>

**Passos para ti:**

1. **Cria o ficheiro quiz\_game.php** na tua pasta C:\xampp\htdocs\quiz\_angola.
2. **Cola o código acima** nesse ficheiro.
3. **Guarda o ficheiro.**
4. **Testa o fluxo completo:**
   * Vai para http://localhost/quiz\_angola/login.php e faz login com uma conta registada.
   * Deverás ser redirecionado para dashboard.php.
   * A partir da dashboard, clica em "Iniciar Quiz" para aceder ao novo quiz\_game.php.
   * Responde às perguntas. Verifica se a pontuação atualiza e se as perguntas mudam.
   * Tenta responder corretamente e incorretamente para ver o feedback.
   * Clica em "Sair" para garantir que o logout funciona.

Este código implementa os pontos do Dia 3:

* A **interface** é exibida com uma pergunta e opções.
* O **JavaScript** permite selecionar uma resposta e habilita o botão de submissão.
* A **lógica PHP** busca perguntas do DB, verifica a resposta submetida, calcula a pontuação e gere as perguntas já respondidas.

**Conseguiste ter o quiz\_game.php a funcionar com a lógica de perguntas, respostas e pontuação básica?**

Assim que tiveres confirmado que tudo está a funcionar, podemos passar para o Dia 4, que se focará em aprimorar o quiz!