

CURSO:	Análise e Desenvolvimento de Sistemas Back-End	Avaliação Formadora (Competência 1)
Analisar, projetar e programar soluções Web em arquitetura Client-Server com PHP, considerando UX, para criar aplicações funcionais e interativas, reforçando lógica, raciocínio estruturado e implementação de fluxos dinâmicos no servidor.		

Objetivo

Desenvolver uma aplicação web simples utilizando PHP (sem banco de dados), aplicando os conhecimentos de lógica de programação, estruturas condicionais, estruturas de repetição e manipulação de dados em arrays. O projeto consistirá em um pequeno quiz de perguntas e respostas processado no backend, com respostas enviadas via formulário HTML e interpretadas pelo PHP.

Tipo de Atividade

- Trabalho em grupo, máximo 4 alunos
- **Entrega na Plataforma AVA (<https://graduacao.unisuam.edu.br>)**
- Valor total: 5,0 pontos

Descrição do Trabalho

Você deve criar uma aplicação chamada "Mini Quiz Backend", com as seguintes funcionalidades:

- Apresentar perguntas com três opções de resposta, armazenadas em um array PHP;
- Permitir ao usuário escolher uma opção e enviar via formulário;
- Verificar no backend (PHP) se a resposta está certa ou errada;
- Controlar a pontuação ao longo do quiz usando variáveis de sessão (\$_SESSION);
- Mostrar a pontuação final ao término do quiz;
- Permitir reiniciar o quiz.

Requisitos Técnicos

- HTML com estrutura semântica adequada (página inicial e páginas de perguntas);
- CSS básico para organização visual (arquivo externo style.css);
- Lógica principal em PHP (arquivo quiz.php ou index.php);
- Uso de arrays para armazenar perguntas, alternativas e respostas corretas;
- Utilização de condicionais (if, else) e estruturas de repetição (foreach, for ou while);
- Uso de sessões PHP para controlar a pontuação e o progresso;
- Organização do código e versionamento com Git/GitHub.

Fluxo Esperado

1. Tela Inicial: Botão para iniciar o quiz.
2. Tela de Pergunta: Mostra a pergunta atual, três opções e botão de envio.
3. Feedback da Resposta: Indica se o usuário acertou ou errou e botão "Próxima".
4. Tela Final: Mostra a pontuação e botão para reiniciar o quiz.

Estrutura Esperada no GitHub

```
/quiz-backend
├── index.php      # Página inicial e controle do quiz
├── style.css      # Estilo CSS
├── perguntas.php  # Arquivo com array de perguntas e respostas
└── README.md     # Explicação do projeto (opcional)
```

Observações para o Desenvolvimento

- Não usar banco de dados — todas as perguntas e respostas devem estar em arrays PHP.
- Controlar o número da questão atual com `$_SESSION['indice']`.
- Usar `$_SESSION['pontuacao']` para armazenar a pontuação.
- O aluno deve pensar na lógica de “ir para a próxima questão” e “finalizar o quiz”.
- Pode usar IA para ideias, mas precisa adaptar e entender o código.

Vídeo de Apresentação

Gravar vídeo (até 5 minutos) mostrando:

- O funcionamento do quiz no navegador;
- O código-fonte no PHP;
- A lógica usada para verificar respostas e calcular pontuação.

Sugestão de ferramenta: Loom ou gravação de tela padrão.

Publicar o vídeo no YouTube e enviar o link no AVA junto com o link do GitHub.

Entrega e Envio no AVA

O aluno deve enviar pelo AVA:

1. Link do repositório no GitHub contendo todos os arquivos do projeto.
2. Link do vídeo no YouTube com a apresentação do projeto (mostrando funcionamento e explicação do código).

Atenção: Não é necessário enviar arquivos no AVA, apenas os links.

Critérios de Avaliação (5,0 pontos)

Item	Valor
Funcionalidade geral do quiz	2,0
Uso correto de estruturas de controle	1,0
Estrutura HTML organizada	0,5
CSS aplicado de forma clara e legível	0,5
Uso correto de Git/GitHub	0,5
Apresentação em vídeo clara e objetiva	0,5