

Objetivo do Projeto

Os alunos deverão desenvolver um website dinâmico que demonstre os seus conhecimentos sobre:

- A distinção entre código server-side e client-side.
 - A construção de páginas com recurso a uma linguagem de scripting server-side (PHP).
 - A utilização de estruturas de controlo de execução (condicionais e ciclos) em PHP.
 - A manipulação de objetos comuns em PHP (exemplo: arrays, datas, strings, ficheiros, etc.).
 - A construção dinâmica de páginas em função de parâmetros recebidos.
 - A manipulação de sessões e cookies para gerir a interação do utilizador com o site.
 - O tratamento adequado de erros em páginas dinâmicas.
1. **Estrutura do Website:** O site deve conter pelo menos:
 - Página inicial.
 - Páginas dinâmicas geradas com base em parâmetros.
 - Formulário para interação com o utilizador.
 - Implementação de sessão e cookies.
 - Página de erro personalizada para tratamento adequado de falhas.
 2. **Tecnologias Utilizadas:**
 - HTML e CSS para a estrutura e estilo.
 - PHP para a manipulação dinâmica dos conteúdos.
 - Arrays **ou** Base de Dados para armazenar e apresentar informação.
 - Sessões e cookies para gestão de utilizadores.
 3. **Funcionalidade Dinâmica:**
 - Os conteúdos apresentados devem ser gerados dinamicamente com base nas interações do utilizador. Formulários com entradas diversificadas.
 - Utilização de formulários para entrada de dados e respetivo processamento.
 4. **Boas Práticas:**
 - Código organizado e comentado.
 - Separação de código PHP e HTML para maior clareza e manutenção do projeto.
-

Requisitos do Projeto:

1. **Estrutura da Página:**
 - Crie uma página web utilizando HTML para a estrutura básica e CSS para o estilo que deve ser mantido em todas as páginas do site.
 - A página deve incluir um formulário de login simples com campos para nome de utilizador e senha.

2. **Código Client-Side vs Server-Side:**
 - Explique a diferença entre código client-side (HTML, CSS, JavaScript) e server-side (PHP).
 - Inclua exemplos de ambos os tipos de código na sua página.
3. **Linguagem de Scripting Server-Side:**
 - Utilize PHP para processar o formulário de login ou o outro..
 - Verifique se o nome de utilizador e a senha correspondem a valores pré-definidos.
4. **Estruturas de Controlo de Execução:**
 - Utilize estruturas de controlo como if, else, e loops em PHP para validar os dados do formulário.
5. **Manipulação de Objetos Comuns:**
 - Manipule objetos comuns em PHP, como arrays e strings.
 - Exiba uma mensagem de boas-vindas personalizada após o login bem-sucedido. Meta o utilizador ativo.
6. **Construção Dinâmica da Página:**
 - Construa a página dinamicamente com base nos parâmetros fornecidos pelo utilizador (por exemplo, exiba diferentes mensagens de boas-vindas com base no nome de utilizador).
7. **Manipulação de Sessões e Cookies:**
 - Utilize sessões para manter o estado do login do utilizador.
 - Armazene informações do utilizador em cookies para personalizar futuras visitas.
8. **Tratamento de Erros:**
 - Implemente um tratamento adequado de erros para lidar com entradas inválidas no formulário.
 - Exiba mensagens de erro claras e amigáveis ao utilizador.

Rubricas de Avaliação

1. Estrutura e Estilo das Páginas (20 pontos)

- **HTML e CSS bem estruturados e organizados (10 pontos):**
 - Utilização correta de tags HTML.
 - Código CSS limpo e bem comentado.
- **Design e usabilidade (10 pontos):**
 - Página visualmente atraente e fácil de navegar.
 - Formulário de login e outro funcional e intuitivo.

2. Diferenciação entre Código Client-Side e Server-Side (15 pontos)

- **Explicação clara e exemplos (15 pontos): Explicar no relatório do projeto**
 - Explicação detalhada da diferença entre client-side e server-side.
 - Inclusão de exemplos de código para ambos os tipos.

3. Implementação de Linguagem de Scripting Server-Side (20 pontos) comentado

- **Processamento do formulário de login com PHP (10 pontos):**
 - Verificação correta dos dados do formulário. Diversificação dos tipos de entrada de dados no formulário.
- **Utilização de estruturas de controlo (10 pontos):**
 - Implementação de if, else, e loops para validação.

4. Manipulação de Dados (10 pontos)

- **Manipulação de arrays ou bases de dados (10 pontos):**
 - Utilização correta de arrays ou integração com uma base de dados.
 - Exibição de mensagens personalizadas com base nos dados armazenados.

5. Construção Dinâmica da Página (10 pontos)

- **Construção baseada em parâmetros (10 pontos): Explicar no relatório do projecto**
 - Página adaptada dinamicamente com base nos dados do utilizador.

6. Manipulação de Sessões e Cookies (10 pontos) Explicar no relatório do projeto

- **Utilização de sessões (5 pontos):**
 - Manutenção do estado do login do utilizador.
- **Armazenamento de cookies (5 pontos):**
 - Personalização de futuras visitas com cookies.

7. Tratamento de Erros (5 pontos)

- **Tratamento adequado de erros (5 pontos): Explicar no relatório do projecto**
 - Mensagens de erro claras e amigáveis.
 - Validação de entradas inválidas.

8. Documentação e Apresentação (10 pontos) Relatório estruturado com capa, índice automático e bem formatado

- **Documentação clara e detalhada (5 pontos):**

Enunciado do Projecto – Desenvolvimento de Páginas Web Dinâmicas

Disciplina: Redes de Comunicação
Módulo: 5 – Páginas Web Dinâmicas

Tecnologias: HTML, CSS, PHP (com arrays ou bases de dados)

- Descrição de como cada requisito foi atendido.
 - **Capturas de tela e exemplos (5 pontos):**
 - Inclusão de capturas de tela da página em funcionamento.
-

Entrega do Projeto

- O **PROJETO** deverá ser entregue em formato digital, contendo todos os ficheiros necessários para a sua execução.
 - Deve ser incluído um ficheiro README explicando a estrutura do projeto e as instruções para execução.
 - O **RELATÓRIO** do projecto deve conter uma descrição das funcionalidades e tecnologias utilizadas.
-

Critérios de Avaliação e Rubricas

Critério	Descrição
Estrutura e Organização do Código	Código limpo, organizado e comentado. Separação adequada entre HTML, CSS e PHP.
Implementação de Páginas Dinâmicas	Uso correto de PHP para gerar páginas dinamicamente com base em parâmetros. Diferenciação entrada de dados dos formulários.
Manipulação de Sessões e Cookies	Gestão adequada de sessões e cookies para interação com o utilizador.
Utilização de Estruturas de Controlo	Uso correto de condicionais e ciclos para a lógica do website.
Interação com Dados (Arrays/Bases de Dados)	Utilização correta de arrays ou bases de dados para armazenar e recuperar informações.
Tratamento de Erros e Segurança	Implementação de validação de dados, prevenção de falhas e mensagens de erro amigáveis.
Aspeto Visual e Usabilidade	Design coerente e responsivo, facilitando a navegação e a experiência do utilizador.