Atividades pedagógicas para realização em domicílio

|  |  |
| --- | --- |
| **Disciplina: PROGRAMAÇÃO WEB** | **Carga horária: 2** |
| **Professor: ALAN MATEUS** | |
| **Aluno:** | **Série: 2 ANO** |
| **Semana de 12 a 19/05** | |

6 Atividades Teórico-Reflexivas Individuais (Entrega via GitHub Opcional)

**Objetivo Geral:** Compreender os conceitos de efeitos visuais e validação de formulários com JavaScript, e demonstrar entendimento da arquitetura cliente-servidor e do padrão MVC através de análise, descrição e pesquisa.

**Instruções:**

1. **Crie uma conta no GitHub** (opcional).
2. **Crie um novo repositório público** com o nome teoria-dinamica-organizacao-web (opcional).
3. **Para cada atividade abaixo, crie um arquivo de texto (Markdown** .md **recomendado)** dentro do repositório (opcional) ou em um documento de sua preferência.
4. **Responda cada atividade de forma clara e completa**, baseando-se em suas pesquisas e compreensão dos conceitos.
5. **Cite as fontes** que você utilizar para cada resposta.
6. **(Opcional)** Faça commits regulares com mensagens claras descrevendo suas alterações e envie (push) para o seu repositório no GitHub.

**Atividades:**

**Atividade 1: Descrevendo Efeitos Visuais com JavaScript**

* **Tarefa:** Pesquise e descreva detalhadamente como o JavaScript é utilizado para criar efeitos visuais em páginas web. Explique como o JavaScript interage com o CSS para modificar estilos dinamicamente e como animações simples podem ser implementadas diretamente no navegador. Discuta também como as bibliotecas de JavaScript facilitam a criação de efeitos mais complexos, fornecendo exemplos de funcionalidades comuns dessas bibliotecas.

**Atividade 2: Analisando a Validação de Formulários com JavaScript**

* **Tarefa:** Explique o processo de validação de formulários utilizando JavaScript no lado do cliente. Descreva como o JavaScript captura os dados inseridos nos campos de um formulário e como ele pode ser usado para verificar se esses dados atendem a critérios específicos antes de serem enviados ao servidor. Introduza o conceito de expressões regulares e explique como elas podem ser aplicadas para realizar validações de formato mais avançadas (forneça um exemplo de uma expressão regular para validar um e-mail e explique seu padrão).

**Atividade 3: Detalhando a Arquitetura Cliente-Servidor na Web**

* **Tarefa:** Descreva em detalhes como funciona a comunicação entre um navegador web (o cliente) e um servidor na internet ao acessar uma página web. Explique o papel de cada um (cliente e servidor) e detalhe o fluxo de eventos desde a solicitação inicial até o recebimento da página pelo navegador. Inclua na sua explicação os conceitos de requisição HTTP, resposta HTTP e, brevemente, o papel do DNS.

**Atividade 4: Explorando o Padrão de Arquitetura MVC**

* **Tarefa:** Explique o padrão de arquitetura MVC (Model-View-Controller) utilizado no desenvolvimento de aplicações web. Descreva a responsabilidade de cada um dos seus três componentes (Model, View, Controller) e ilustre, através de um cenário hipotético de uma pequena aplicação web (por exemplo, exibir e interagir com uma lista de livros), como cada componente interage com os outros para realizar as funcionalidades da aplicação.

**Atividade 5: Discutindo a Importância da Interatividade e Validação no Desenvolvimento Web**

* **Tarefa:** Elabore uma discussão sobre a importância da interatividade (através de efeitos visuais com JavaScript) e da validação de formulários no lado do cliente para a experiência do usuário e para a qualidade de uma aplicação web. Apresente argumentos que justifiquem o uso dessas técnicas, considerando aspectos como engajamento do usuário, feedback imediato, prevenção de erros e otimização do processamento no servidor.

**Atividade 6: Refletindo sobre a Organização do Código com o Padrão MVC**

* **Tarefa:** Apresente uma reflexão sobre os benefícios de adotar um padrão de arquitetura como o MVC para organizar o código de aplicações web, especialmente aquelas de maior complexidade. Discuta como a separação de responsabilidades promovida pelo MVC contribui para a manutenibilidade, escalabilidade, reutilização de código e colaboração entre desenvolvedores em um projeto web.

**Formato de Entrega:**

* Um documento de texto (formato .txt, .doc, .pdf ou Markdown .md).
* **(Opcional)** O link para o seu repositório público no GitHub (https://github.com/<seu-usuario>/teoria-dinamica-organizacao-web) contendo os arquivos de texto com suas respostas.

**Critérios de Avaliação:**

* Clareza e correção das respostas para cada atividade.
* Demonstração de compreensão dos conceitos teóricos.
* Qualidade da pesquisa e das informações apresentadas.
* Formatação clara e organização do texto.
* Citação das fontes utilizadas para cada resposta.
* **(Se aplicável)** Organização dos arquivos no repositório do GitHub.