



Universidade Federal de Uberlândia
Faculdade de Computação – Sistemas de Informação – Prof. Daniel A. Furtado
Trabalho de Desenvolvimento Web II

Instruções Gerais

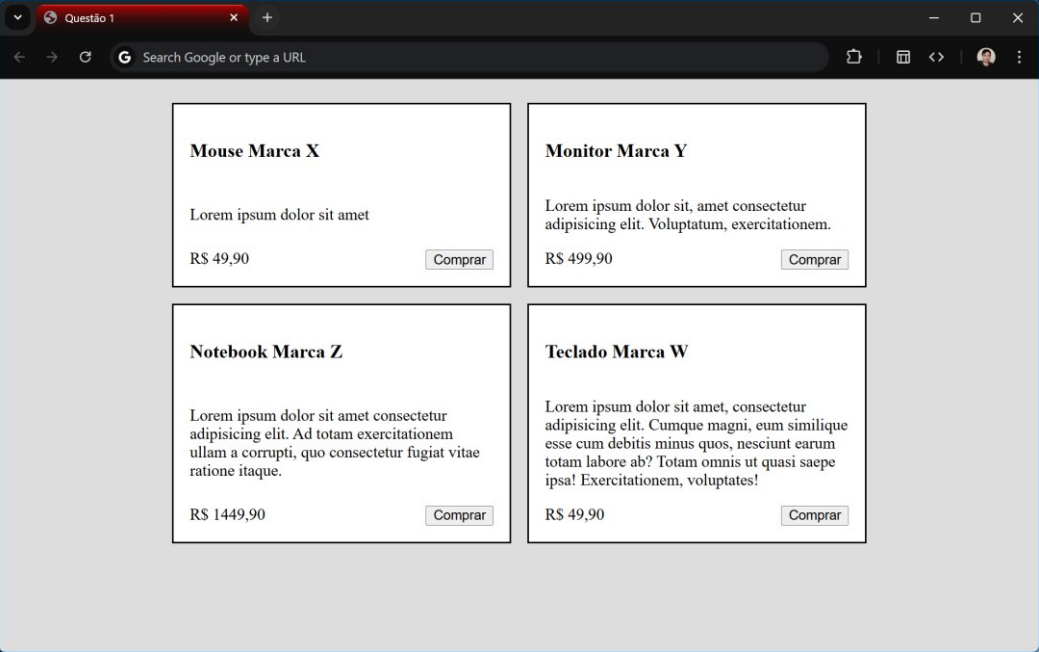
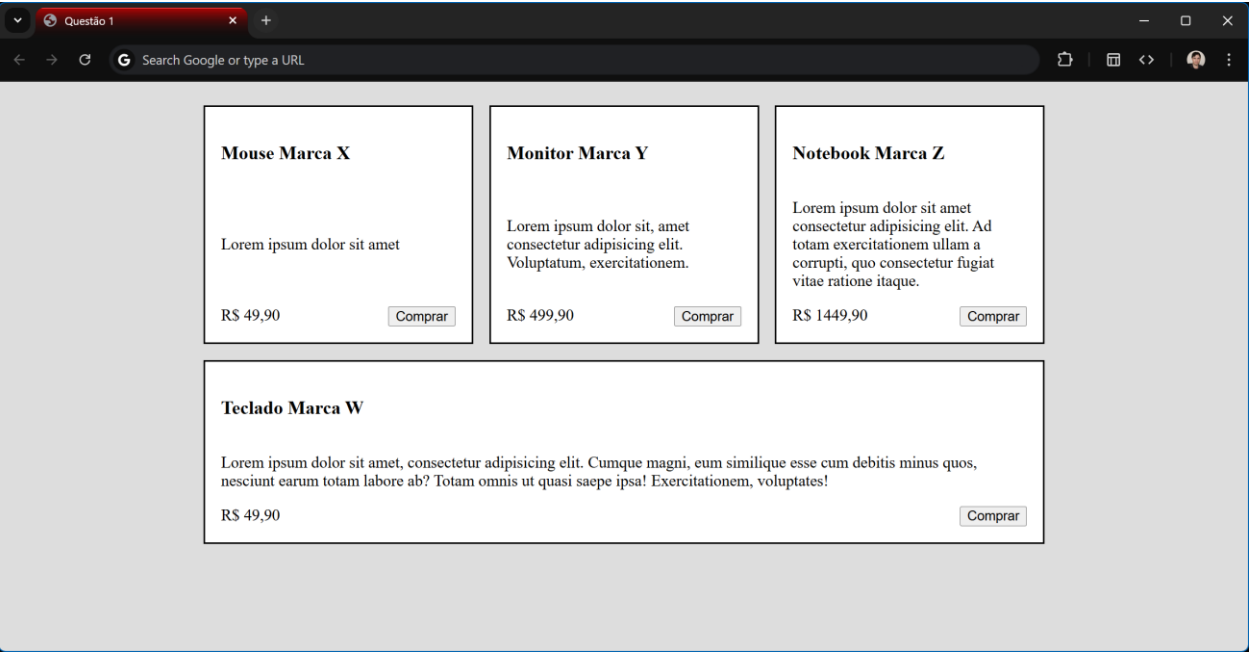
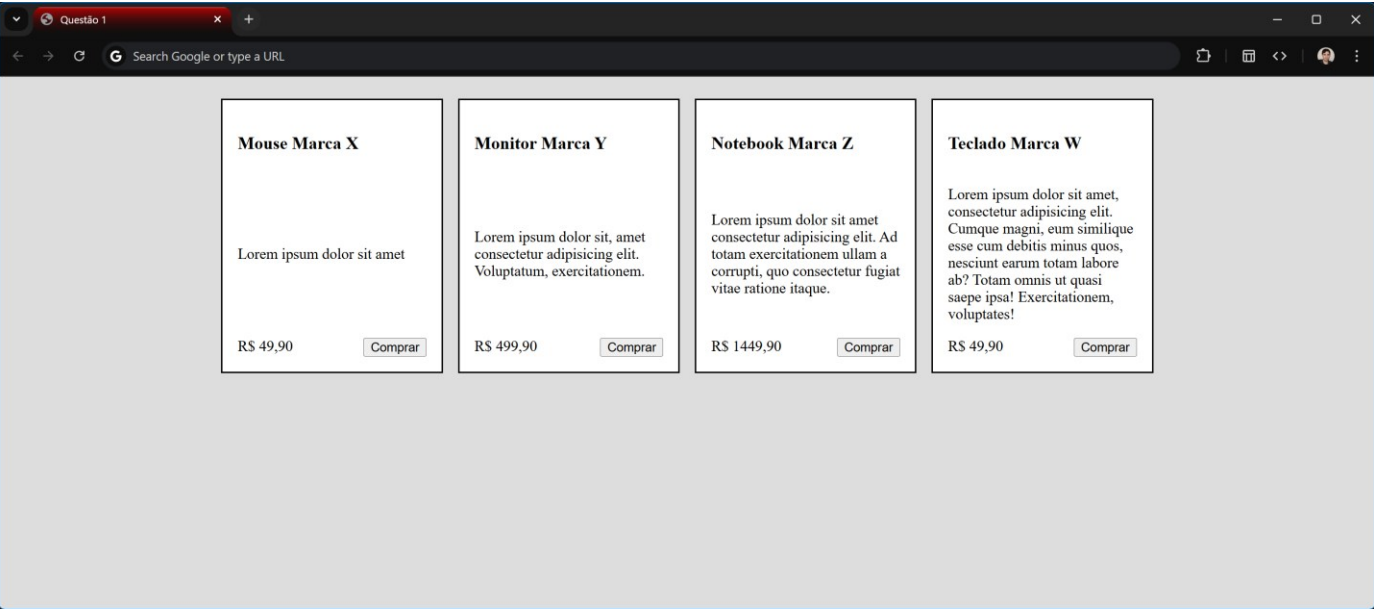
- Leia atentamente as especificações associadas às questões antes de resolvê-las;
- As questões com codificação devem ser resolvidas utilizando apenas HTML, CSS, JavaScript e PHP, com os devidos tratamentos de erros e da forma mais eficiente possível. A utilização de recursos não apresentados em aula anulará a questão;
- Questões resolvidas de formas diferentes daquelas solicitadas não serão consideradas;
- Todo código deve estar 100% indentando. Código sem a devida indentação será considerado ilegível, de difícil manutenção, e resultará em penalização de 30% sobre o valor da questão.
- A comunicação com o banco de dados deve ser feita utilizando o PHP Data Objects (PDO);
- Aspectos de segurança devem ser considerados para prevenir ataques como XSS e SQL Injection;
- Alunos envolvidos em **qualquer tipo de plágio**, total ou parcial, terão seus trabalhos **anulados** e receberão **nota zero**.

Questão 1 – 10 pontos

Considere o código HTML a seguir. Escreva o respectivo código CSS para que a página se apresente como nas figuras à medida que a largura da viewport é reduzida. Utilize no máximo **4 regras** CSS e no máximo **18 declarações** (excluindo media queries). Quando acessada de um dispositivo com largura de **viewport** inferior a **500 pixels CSS**, a região principal deverá expandir e ocupar 100% da largura. **Não altere o código HTML**.

- Largura ocupada pela região principal: **70%**.
- Tamanhos dos paddings: **1rem**.
- Largura principal inicial dos cards: **200px**.
- Cor de fundo da página: **#ddd**; cor de fundo dos cards: **white**.

```
<body>
  <main>
    <div>
      <h3>Mouse Marca X</h3>
      <p>Lorem ipsum dolor sit amet</p>
      <div><span>R$ 49,90</span><button>Comprar</button></div>
    </div>
    <div>
      <h3>Monitor Marca Y</h3>
      <p>Lorem ipsum com 10 palavras</p>
      <div><span>R$ 499,90</span><button>Comprar</button></div>
    </div>
    <div>
      <h3>Notebook Marca Z</h3>
      <p>Lorem ipsum com 20 palavras</p>
      <div><span>R$ 1449,90</span><button>Comprar</button></div>
    </div>
    <div>
      <h3>Teclado Marca W</h3>
      <p>Lorem ipsum com 30 palavras</p>
      <div><span>R$ 49,90</span><button>Comprar</button></div>
    </div>
  </main>
</body>
```



Questão 2 – 10 pontos

Considere o formulário HTML a seguir:

```
<form action="cadastra-fornecedor.php" method="POST" name="form1">
  <input type="text" id="A" name="nome">
  <input type="text" id="B" name="cnpj">
  <input type="text" id="C" name="emailContato">
  <input type="text" id="D" name="telefoneContato">
  <button>Cadastrar</button>
</form>
```

Escreva o código do script **cadastra-fornecedor.php**, que receba o formulário acima e faça a inserção dos dados nas tabelas correlacionadas **Fornecedor** e **ContatoFornecedor**, conforme código SQL a seguir. Considere a existência de um arquivo **conexaoMysql.php** contendo a função **mysqlConnection()** para conexão com o MySQL utilizando PDO. Essa função deve ser utilizada no código. Recursos adequados devem ser empregados para que o banco de dados permaneça sempre consistente.

```
CREATE TABLE Fornecedor
(
  id int PRIMARY KEY auto_increment,
  nome varchar(50),
  cnpj varchar(50)
)

CREATE TABLE ContatoFornecedor
(
  emailContato varchar(50),
  telefoneContato varchar(50),
  idFornecedor int not null,
  FOREIGN KEY(idFornecedor) REFERENCES Fornecedor(id)
)
```

Questão 3 – 10 pontos

Considere o código HTML a seguir:

```
<form name="form2">
  Código do Fornecedor: <input type="text" id="codigo" name="codigo">
  Nome: <input type="text" id="nome" name="nome">
  CNPf: <input type="text" id="cnpj" name="cnpj">
</form>
```

Quando o usuário informar um código de fornecedor campo **código** do formulário, os dados do respectivo fornecedor devem aparecer “instantaneamente” nos campos **nome** e **cnpj**. Forneça o código JavaScript completo, **sem** utilizar o método `.then`, que implemente essa funcionalidade utilizando a técnica Ajax com a API **Fetch** e **async/await**. Suponha que exista um script PHP no servidor chamado **busca-fornecedor.php** que receba um parâmetro **cod** pela URL e retorne um objeto em **JSON** contendo o **nome** e o **cnpj** do respectivo fornecedor. Em caso de erros, as seguintes mensagens devem ser apresentadas ao usuário:

- Se o servidor estiver *offline* e não responder à requisição → mensagem “Requisição não completada”;
- Se o servidor retornar um código de status de erro → mensagem “status inválido”;
- Se o servidor retornar um código de sucesso, mas não retornar um JSON → mensagem “JSON inválido”.

Questão 4 – 10 pontos

a) Considere que a requisição HTTP a seguir seja realizada do front-end de um site A para um site B:

POST /cadastra-cliente.php HTTP/1.1

Host: https://siteB.com

Origin: https://siteA.com

Content-Type: application/json

Content-Length: 17

{"nome":"Fulano"}

Considere também que o arquivo **cadastra-cliente.php**, no servidor do site B, tenha o seguinte conteúdo:

```
<?php

if ($_SERVER['REQUEST_METHOD'] == 'OPTIONS')
    exit(0);

$origem = $_SERVER['HTTP_ORIGIN'] ?? '';

if ($origem == 'https://siteA.com') {
    header("Access-Control-Allow-Origin: *");
    header("Access-Control-Allow-Methods: GET, POST, OPTIONS");
    header("Access-Control-Allow-Headers: Content-Type");
}

// Considere que aqui tenha o código para cadastro do cliente

echo "Cadastro realizado com sucesso.";
```

Explique detalhadamente como a requisição será tratada pelo navegador, pelo servidor, e qual o resultado.

b) Considere que a requisição HTTP a seguir tenha sido gerada a partir de uma requisição Ajax, partindo do front-end do site A, com a intenção de buscar o nome de um cliente no site B, para apresentação na página utilizando a técnica Ajax.

GET /get-cliente.php HTTP/1.1

Host: https://siteB.com

Origin: https://siteA.com

Considere também que o arquivo **get-cliente.php**, no servidor do site B, tenha o seguinte conteúdo:

```
<?php

if ($_SERVER['REQUEST_METHOD'] == 'OPTIONS')
    exit(0);

if ($_SERVER['HTTP_ORIGIN'] == 'https://siteA.com') {
    header('Content-Type: application/json; charset=utf-8');
    echo json_encode(['nome' => 'Fulano']);
}
else {
    echo "Operação não permitida";
}
```

Explique detalhadamente como a requisição será tratada pelo navegador, pelo servidor, e qual o resultado.