



Universidade Federal de Uberlândia  
Faculdade de Computação – Prof. Daniel A. Furtado  
5º Trabalho de Desenvolvimento Web II  
Técnica Ajax – Requisições HTTP Assíncronas com a API Fetch

## Instruções Gerais

---

- Esta atividade deve ser realizada **individualmente**;
- Tecnologias permitidas: HTML5, CSS, JavaScript, Bootstrap, PHP, MySQL, API Fetch sem `async/await`. O objeto XMLHttpRequest **não é permitido** neste trabalho;
- Aspectos de segurança devem ser considerados para evitar ataques XSS e SQL Injection;
- Sintaxe da XHTML como `<img/>` ou `<br/>` não é permitida (anulará o trabalho);
- O website deve ser hospedado e disponibilizado online, conforme orientações disponíveis no final deste documento;
- Ao construir o website, utilize dados fictícios. **Jamais utilize** dados pessoais como seu nome, CPF, endereço, e-mail etc.;
- Esteja atento às **observações sobre plágio** apresentadas no final deste documento;
- Trabalhos com implementações utilizando trechos de códigos retirados de sites da Internet ou de trabalhos de semestres anteriores serão anulados;
- As páginas web não devem conter qualquer conteúdo de caráter imoral, desrespeitoso, pornográfico, discurso de ódio, desacato etc.;
- O website deve ser validado utilizando as ferramentas disponíveis nos endereços **validator.w3.org** e **jigsaw.w3.org/css-validator** (não deve conter nenhum erro ou *warning*);
- O trabalho deve ser entregue até a data/hora definida pelo professor. Não deixe para enviar o trabalho nos últimos instantes, pois eventuais problemas relacionados à eventos adversos como instabilidade de conexão, congestionamento de rede etc., não serão aceitos como motivos para entrega da atividade por outras formas ou em outras datas;
- Este trabalho deve ser feito **mantendo os trabalhos anteriores intactos**, ou seja, os trabalhos anteriores devem permanecer online conforme foram entregues;
- Trabalhos enviados por e-mail ou pelo MS Teams **não serão considerados**.

## Material de Apoio

<https://furtado.prof.ufu.br/site/teaching/DW2/DW2-Modulo4-HTTP-Ajax.pdf>

<https://furtado.prof.ufu.br/site/teaching/PPI/PPI-Modulo4-JavaScript.pdf>

## Exercício 1

---

Abra o arquivo <https://furtado.prof.ufu.br/site/teaching/DW2/Exemplos-Trab5.zip> e coloque os exemplos online. Abra a página **index.html** (menu de opções), teste o primeiro exemplo (**ex1-cep-fetch**) e siga os passos a seguir:

1. Analise o código JavaScript e HTML no **VS Code**. Leia atentamente os comentários;
2. Utilize o arquivo SQL fornecido para criar e povoar a tabela **endereço** no banco de dados do infinityfree;

3. Altere o arquivo **busca-endereco.php** para que os dados sejam buscados na tabela do banco de dados. Utilize como referência o exemplo do slide 15. Teste o exemplo modificado utilizando o CEP 38400-300;
4. Faça uma cópia do arquivo **index.html** do exemplo e renomeie a cópia para **index2.html**. Altere o código JavaScript do novo arquivo para que a requisição Ajax passe a buscar os dados do endereço utilizando a API ViaCEP, com retorno em JSON, utilizando a URL <https://viacep.com.br/ws/CEP-DE-BUSCA/json/>.

## Exercício 2

---

Teste o segundo exemplo (**ex2-encadeamento-fetch**) no navegador acessando a página sem utilizar HTTPS (pois a API utilizada não suporta HTTPS). Analise o código JavaScript no **VS Code** e insira comentários explicando a função **buscaClimaLocal**.

## Exercício 3

---

Este exercício tem como objetivo criar uma página de cadastro de veículo que realiza a submissão dos dados do formulário utilizando uma requisição Ajax com a API Fetch. Procedimentos:

- a) Crie uma página HTML contendo um formulário de cadastro com os campos marca, modelo, ano, cor e quilometragem;
- b) Crie um script PHP que receba os dados do veículo por POST e os insira na tabela **veiculo** do banco de dados (conforme código SQL do trabalho anterior);
- c) Teste o cadastro enviando o formulário da forma tradicional;
- d) Acrescente o código JavaScript para permitir que o formulário seja enviado em segundo plano, com Ajax, utilizando a API Fetch. Depois de submeter o formulário, o código JavaScript deve limpar os campos para facilitar a realização de novos cadastros. Veja o código dos slides 50 e 82. Não deve ocorrer nenhum tipo de redirecionamento na página de cadastro.

## Disponibilização Online

---

O trabalho deve entregar pelo sistema SAAT e disponibilizado online utilizando o subdomínio gratuito registrado em site de hospedagem. Como este trabalho consiste em modificações dos arquivos de exemplo, não é necessário criar subpastas para cada exercício. Ao acessar o endereço a seguir, deverá abrir a página principal contendo o **menu de opções** modificado:

[seusubdominio.com/trabalho5](http://seusubdominio.com/trabalho5)

## Entrega

---

Além da disponibilização online, a pasta raiz deve ser compactada no formato **zip** e enviada pelo Sistema Acadêmico de Aplicação de Testes (SAAT) até a data limite indicada pelo professor em sala de aula.

Adicione também um arquivo de nome **link.txt**, na pasta raiz, contendo a URL do trabalho online (para a pasta raiz do trabalho).

## Sobre Eventuais Plágios

---

Este é um trabalho individual. Os alunos envolvidos em qualquer tipo de plágio, total ou parcial, seja entre equipes ou de trabalhos de semestres anteriores ou de materiais disponíveis na Internet (exceto os materiais de aula disponibilizados pelo professor), serão duramente penalizados (art. 196 do Regimento Geral da UFU). Todos os alunos envolvidos terão seus **trabalhos anulados** e receberão **nota zero**.