



Universidade Federal de Uberlândia  
Faculdade de Computação – Prof. Daniel A. Furtado  
12º Trabalho de Programação para Internet  
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. O objeto XMLHttpRequest **não é permitido** neste trabalho;
- 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/PPI/PPI-Modulo8-Ajax.pdf> (slides 57-110)

## Exercício 1

---

Abra o arquivo <http://www.furtado.prof.ufu.br/site/teaching/PPI/Exemplos-Ajax-Fetch.zip> e coloque os exemplos online. Abra a página **index.html** com o menu de opções no navegador e teste o primeiro exemplo (**ex1-cep-fetch**). Analise os códigos no **VS Code**. Leia atentamente os comentários. Altere o exemplo para que o CEP seja enviado ao servidor pelo método **POST**.

## Exercício 2

---

Teste o segundo exemplo (**ex2-encadeamento-fetch**) no navegador. Analise o código JavaScript no **VS Code** e insira comentários explicando a função **buscaClimaLocal**.

## Exercício 3

---

Abra o exemplo **ex3-cep-await** no **VS Code** e analise o código JavaScript comparando-o com o código do exemplo **ex1-cep-fetch**.

- Adicione comentários explicando o código JavaScript da função **buscaEndereco**;
- Execute a função **buscaEndereço** linha a linha, utilizando o ambiente de depuração do navegador (F12). Monitore o valor da variável **response** colocando o ponteiro do mouse sobre ela assim que a respectiva linha for executada. Qual é o valor da propriedade **response.status**? A propriedade **response.body** contém os dados do endereço? Por quê? Insira as respostas em comentários no próprio código JavaScript;
- Faça os devidos ajustes no exemplo para que os dados do endereço sejam buscados em uma tabela do banco de dados. Utilize o arquivo fornecido **busca-endereco-bd.php**. Crie a tabela utilizando o código SQL disponível no arquivo **busca-endereco-bd.php**;
- Faça uma cópia do arquivo **index.html** 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 (veja detalhes em [viacep.com.br](http://viacep.com.br)).

## Exercício 4

---

Abra o exemplo **ex4-encadeamento-await** no **VS Code** e analise o código JavaScript comparando-o com o código do exemplo **ex2-encadeamento-fetch**. Execute a função **buscaClimaLocal** passo a passo e monitore os valores das variáveis retornadas pelas funções assíncronas passando o ponteiro do mouse sobre elas durante a execução. Qual o valor das propriedades **latitude** e **longitude** da variável **local**? Explore as propriedades da variável **clima**. Em seguida, monitore a última requisição Ajax (que retorna os dados do clima) pela aba **Network** do ambiente de desenvolvimento. Observe o corpo da resposta HTTP (aba **Response**). Copie a string JSON e cole em um arquivo de nome **respostas.txt** na pasta do exemplo.

## Exercício 5

---

Abra o exemplo 5 (**ex5-infinite-scroll**) no navegador e role a página de produtos até o final. Observe os novos produtos sendo adicionados dinamicamente. Abra o modo de desenvolvimento do navegador e observe as requisições Ajax sendo disparadas à medida em que a página é rolada. Observe também o corpo da resposta HTTP (aba **Response**) de algumas requisições. Analise os códigos HTML, JavaScript e PHP. Adicione comentários descrevendo as operações.

## Exercício 6

---

Este exercício utiliza uma versão gratuita da API **Alpha Vantage** para obtenção de cotações de ações na bolsa de valores. A versão gratuita dessa API tem um limite de **5** requisições por minuto e **25** requisições por dia. Atenção para não ultrapassar os limites.

- Faça o registro para obtenção da chave de API gratuita utilizando o link <https://www.alphavantage.co/support/#api-key>;
- Faça um teste de acesso ao serviço utilizando o endereço a seguir. Troque o texto **SuaChaveDeApi** pela chave obtida no passo anterior. Observe que a cotação da ação deve ser informada pela própria URL, utilizando o símbolo da ação na B3 seguido por ".SA". Analise a estrutura do objeto JSON retornado.

[https://www.alphavantage.co/query?function=GLOBAL\\_QUOTE&symbol=PETR4.SA&apikey=SuaChaveDeApi](https://www.alphavantage.co/query?function=GLOBAL_QUOTE&symbol=PETR4.SA&apikey=SuaChaveDeApi)

- c) Crie uma página HTML contendo um **campo** de formulário e um **botão** para que o usuário possa digitar o código da ação. Acrescente o código JavaScript para buscar e exibir a cotação da ação assim que o usuário pressionar o botão. O código JavaScript deve utilizar a API **Fetch** em conjunto com **async/await**, conforme apresentado nos exemplos anteriores. Não utilize o método **.then**;
- d) Acrescente mais dois campos no formulário para permitir ao usuário consultar, de forma simultânea, a cotação de três ações diferentes. Modifique o código JavaScript para realizar as três requisições em paralelo. Utilize o **Promise.all** em conjunto com **async/await**. Não utilize o método **.then** em nenhum lugar do código;
- e) Disponibilize um link para acesso ao exercício pelo **menu de opções** dos exemplos.

## Exercício 7

---

Acesse o endereço <https://github.com/public-apis/public-apis>, escolha uma API de uso gratuito de seu interesse e crie uma página web que ofereça alguma funcionalidade utilizando requisições Ajax e a API escolhida. A página deve possuir uma breve descrição da funcionalidade e as instruções para uso. Disponibilize um link para a página no **meu de opções** dos exemplos.

## Disponibilização Online

---

O trabalho deve entregue pelo sistema SAAT e disponibilizado online utilizando o subdomínio gratuito registrado em site de hospedagem. Como este trabalho consiste de 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/**trabalho12**

## 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**.