



Universidade Federal de Uberlândia  
Faculdade de Computação

13º Trabalho de Programação para Internet – Prof. Daniel A. Furtado  
Trabalho Individual – API Fetch com async/await e Sessões

## Instruções Gerais

---

- Esta atividade deve ser realizada individualmente;
- Tecnologias permitidas: HTML5, CSS, JavaScript, Bootstrap, PHP, MySQL e API Fetch **com** async/await (o emprego do método **then** do objeto promise anulará o trabalho). Não é permitido usar o XMLHttpRequest.
- 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**.

Leia os slides de aula disponibilizados nos endereços a seguir e resolva os exercícios seguintes.

<http://furtado.prof.ufu.br/site/teaching/PPI/PPI-Modulo8-Ajax.pdf>

<http://furtado.prof.ufu.br/site/teaching/PPI/PPI-Modulo9-Sessoes.pdf>

## Exercício 1

---

Abra o arquivo <http://www.furtado.prof.ufu.br/site/teaching/PPI/Exemplos-Ajax-Fetch.zip> e coloque o exemplo **Ex3-cep-await** online na pasta **trabalho13/ex1**. Em seguida:

- a) Teste o exemplo no navegador;
- b) Analise o código JavaScript comparando-o com o código do exemplo **Ex1-cep-fetch**;
- c) Execute a função **buscaEndereco** do exemplo **Ex3-cep-await** linha a linha, utilizando o ambiente de depuração do navegador (F12). Monitore os valores das variáveis **response** e **endereco**. Registre um *print* da tela e adicione a imagem no final do arquivo **index.html** do exemplo;

d) Acrescente comentários no arquivo **index.html** explicando o código JavaScript.

## Exercício 2

---

Faça uma cópia do exemplo **Ex3-cep-await** e altere o código JavaScript para que a requisição Ajax passe a buscar os dados do endereço utilizando a API ViaCEP, com retorno dos dados em JSON (veja detalhes em [viacep.com.br](http://viacep.com.br)). Disponibilize o exemplo modificado em **trabalho13/ex2**.

## Exercício 3

---

Coloque o exemplo **Ex4-encadeamento-await** online, na pasta **trabalho13/ex3**. Em seguida:

- Analise o código JavaScript comparando-o com o código do exemplo **Ex2-encadeamento-fetch**;
- Execute a função **buscaClimaLocal** passo a passo utilizando o modo de depuração do navegador (F12). Monitore os valores das variáveis retornadas pelas funções assíncronas passando o ponteiro do mouse sobre as mesmas durante a execução. Registre um *print* da tela e adicione a imagem no final do arquivo **index.html** do exemplo.

## Exercício 4

---

Coloque o exemplo **Ex5-infinite-scroll** online, na pasta **trabalho13/ex4**. Em seguida:

- Abra a página principal do exemplo e role a página de produtos até o final. Observe os novos produtos sendo adicionados dinamicamente. Abra o modo de desenvolvimento do navegador (F12), selecione **Network** → **Fetch/XHR** 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;
- Adicione comentários no código JavaScript descrevendo as operações. Repita o procedimento para o código PHP.

## Exercício 5

---

Este exercício utiliza a tabela **Produto**, criada anteriormente no **Trabalho10/Exercício3**, com os campos **nome do produto**, **marca** e **descrição**.

- Crie um script PHP que receba, pela URL, uma marca de produto, e retorne, no formato JSON, um array de objetos de produtos da marca indicada, onde cada objeto deve ter o **nome do produto** e a **descrição** do produto. O script deve buscar pelos dados na tabela **Produto** do banco de dados;
- Crie uma página HTML que contenha um campo de formulário do tipo **select** para que o usuário possa selecionar uma **marca** de produto de seu interesse. Insira de forma estática no **select** algumas opções de marcas (que estejam presentes na tabela do banco de dados). Assim que o usuário selecionar uma marca nesse campo, deverá aparecer, logo abaixo, todos os nomes e descrições dos produtos da marca selecionada existentes no banco de dados. Utilize uma requisição Ajax com Fetch e `async/await` para buscar os dados no servidor e inserí-los dinamicamente na página. Utilize o script PHP criado no item a).

## Exercício 6

---

Abra o arquivo <http://www.furtado.prof.ufu.br/site/teaching/PPI/Exemplos-Sessao-login.zip> e coloque o exemplo **Ex02-session-login** online, na pasta **trabalho13/ex6**.

O exemplo faz a validação de login utilizando a tabela **cliente** solicitada em trabalhos anteriores (pode ser necessário utilizar os formulários e scripts de trabalhos anteriores para cadastrar um novo cliente com o objetivo de testar o login). Faça as devidas alterações no arquivo **conexaoMysql.php**.

Após efetuar login com sucesso, observe a página **home.php** e posteriormente clique no link **SAIR** dessa página. Em seguida tente acessar a página **home.php** digitando o endereço diretamente na barra de endereços do navegador. Analise o código dos arquivos e explique o funcionamento geral do exemplo, incluindo a parte de autenticação e o uso de sessões. Acrescente a explicação no final da página **index.html** do exemplo.

## Disponibilização Online

---

As páginas dos exercícios devem ser disponibilizadas online utilizando o subdomínio gratuito registrado anteriormente, porém em pasta própria (isto é, seusubdominio.com/trabalhoX/ex1, seusubdominio.com/trabalhoX/ex2 etc.). Não altere ou exclua as pastas dos trabalhos anteriores.

Acrescente um arquivo de nome **index.html** na pasta raiz do trabalho contendo links para as páginas dos exercícios.

## Entrega

---

Além da disponibilização online, a pasta raiz contendo as subpastas dos exercícios 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**.