



Universidade Federal de Uberlândia

Faculdade de Computação

7º Trabalho de Programação para Internet – Prof. Daniel A. Furtado

Trabalho Individual – Requisições Ajax com JavaScript e XMLHttpRequest

INTRUÇÕES GERAIS

- Esta atividade deve ser realizada individualmente;
- Utilize apenas as tecnologias HTML5, CSS, JavaScript, Bootstrap 5, PHP, MySQL e o objeto XMLHttpRequest;
- **Não é permitido o uso** de tecnologias adicionais como jQuery, React, API Fetch, etc. Trabalhos utilizando tais tecnologias serão anulados;
- A comunicação com o MySQL deve ser feita utilizando o *PHP Data Objects* (PDO);
- Os recursos adequados das linguagens devem ser utilizados;
- O website deve ser hospedado e disponibilizado online, conforme orientações disponíveis no final deste documento;
- 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 bem estruturado utilizando uma pasta raiz e subpastas de acordo com as instruções das videoaulas sobre HTML;
- O website deve ser validado utilizando a ferramenta disponível no endereço validator.w3.org. Não deve conter nenhum erro ou *warning*;
- O trabalho deve ser entregue até a data definida pelo professor em aula síncrona;
- Trabalhos enviados por e-mail **não serão considerados** (veja instruções no final);
- Este trabalho deve ser feito **mantendo os trabalhos anteriores intactos**, ou seja, os trabalhos anteriores devem permanecer online conforme foram entregues.

Estudar as páginas 1-38 do material disponibilizado no endereço:

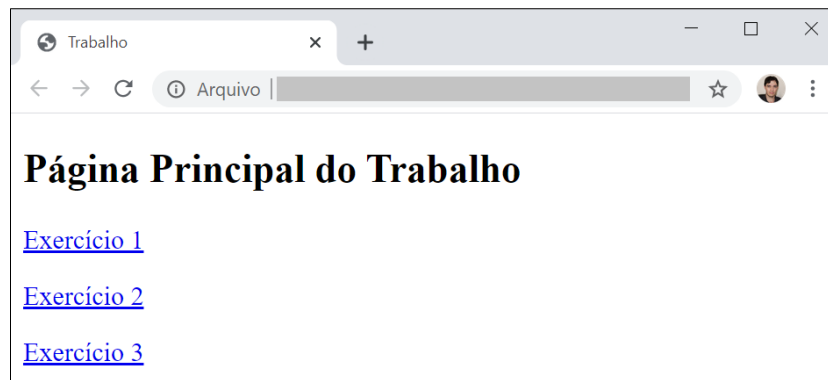
<http://www.daniel.prof.ufu.br/teaching/PPI/PPI-Modulo7-AJAX-VA.pdf>

Em seguida, faça os exercícios descritos a seguir. As páginas dos exercícios devem ser disponibilizadas nos endereços:

www.seusubdominio.dominio.com/trabalho7/ex1,

www.seusubdominio.dominio.com/trabalho7/ex2, etc.

A página principal desta atividade deverá conter links para as páginas dos exercícios, conforme exemplo a seguir:



Exercício 1. Faça o download dos arquivos de exemplo disponibilizados pelo professor no endereço <http://www.daniel.prof.ufu.br/teaching/PPI/PPI-Modulo7-Ajax-Exemplos.zip>.

- Após descompactar, coloque o exemplo da pasta **Ex1-cep-xhr** online utilizando sua conta no **awardspace.com** e faça testes digitando os CEPs 38400-100, 38400-200 e 38400-201. A pasta deve ser renomeada para **ex1**
- Estude detalhadamente os arquivos do exemplo, com foco no código JavaScript responsável por efetuar as requisições Ajax utilizando o XMLHttpRequest. Estude também o respectivo script PHP.

Exercício 2. Faça uma cópia da pasta do exercício anterior e renomeie a cópia para **ex2**. Em seguida, altere o script PHP para que a busca pelo endereço seja realizada utilizando uma tabela do MySQL. Crie uma tabela contendo os campos **cep**, **rua**, **bairro** e **cidade** e insira manualmente alguns registros para efetuar os testes (um dos registros deve conter o cep 38400-000).

O arquivo HTML deve ser mantido igual ao do exercício anterior. O script PHP deve continuar utilizando a definição da classe **Endereco**.

Aspectos de segurança devem ser considerados para evitar ataques de injeção de SQL.

Exercício 3. No exercício 3 do trabalho anterior foi solicitada a implementação de cadastro de usuário, listagem dos usuários cadastrados e teste de login. Faça uma cópia dessa atividade e realize as adaptações necessárias para que os dados do **formulário de login** sejam submetidos ao script PHP utilizando uma requisição Ajax. Utilize como base o exemplo do **slide 38** (requisição Ajax com objeto **FormData**) do material indicado no início deste roteiro.

O script PHP deve retornar um objeto utilizando a classe a seguir:

```

class RequestResponse
{
    public $success;
    public $destination;

    function __construct($success, $destination)
    {
        $this->success = $success;
        $this->destination = $destination;
    }
}

```

O atributo **success** deve ter um valor lógico indicando o resultado da validação do usuário. O atributo **destination** deve indicar a página de destino que o navegador deve redirecionar o usuário em caso de sucesso no login (o redirecionamento deve ser feito do lado do cliente, utilizando JavaScript, após receber a resposta do servidor – veja o código de exemplo a seguir).

```

if (response.success)
    window.location = response.destination;
else
    document.querySelector("#loginFailMsg").style.display = 'block';

```

Exercício 4. Crie uma página para seleção de especialidade médica seguindo os passos a seguir:

- A página deve disponibilizar um campo de formulário do tipo **select** contendo os nomes das seguintes especialidades: *cardiologia*, *dermatologia* e *oftalmologia*;
- Crie um segundo campo do tipo **select**, inicialmente vazio;
- Crie uma única tabela no MySQL contendo os campos **nome_medico**, **telefone_medico**, **especialidade_medico**;
- Inseria 9 registros na tabela contendo dados fictícios de médicos (três registros para cada uma das especialidades listadas no item a);
- Utilize JavaScript para implementar uma requisição Ajax para buscar dinamicamente nessa tabela os dados de todos os médicos da especialidade selecionada no **select** do item a) e inseri-los no segundo **select** conforme detalhado a seguir. Não utilize botões adicionais para disparar a requisição (utilize o evento *change* do select).

O servidor deve retornar, por meio de um script PHP, um vetor de objetos contendo os dados dos médicos da especialidade indicada. Ao receber a resposta da requisição, o código JavaScript deve utilizar as propriedades **nome_medico** e **telefone_medico** para produzir uma *string* única como “Fulano da Silva – 9876-9999” e então inseri-la dinamicamente no segundo **select** (veja o código de exemplo a seguir e o trecho da videoaula: <https://youtu.be/MuZnSjqMKDw?t=190>)

```

var campoSelect = document.getElementById("idDoCampoSelect");
var option = document.createElement("option");
option.text = "Texto da nova opção";
option.value = "Value da nova opção";
campoSelect.add(option);

```

Disponibilização Online

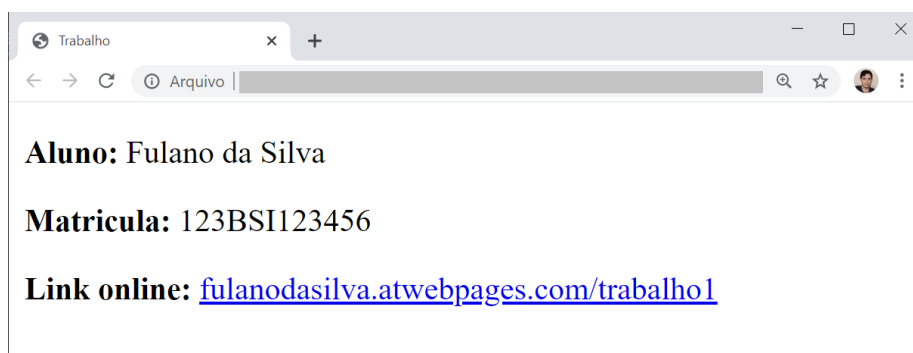
O website deve ser colocado online utilizando um serviço de hospedagem gratuito, conforme apresentado no final da primeira videoaula sobre HTML. Recomenda-se a utilização do awardspace.com, conforme detalhado na videoaula.

A página principal do trabalho deve ser disponibilizada por meio de um endereço no formato: www.subdominiocadastrado.dominioescolhido.com/trabalho7, **sem a indicação de um nome de arquivo em particular** como principal.html (veja a parte da videoaula de HTML sobre o arquivo index.html).

O trabalho deve ser disponibilizado online de forma independente dos trabalhos anteriores. Não deve substituir ou referenciar os arquivos de trabalhos anteriores.

Entrega

Primeiramente o trabalho deve ser colocado online até a data limite de entrega indicada pelo professor. Em seguida, envie pelo Sistema de Aplicação de Testes (SAAT), até a data limite, apenas um arquivo HTML simples (arquivo de texto não será aceito) contendo seus dados e um **link que dê acesso imediato ao trabalho online**. Não envie arquivos compactados, arquivos de texto ou arquivos do trabalho pelo sistema SAAT ou por e-mail. Veja a seguir um exemplo de arquivo HTML a ser enviado pelo SAAT:



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**.