

## Desenvolvimento Web 2

### Exercício 2 – Lista de tarefas a fazer melhorada

#### Part 1 – Ajax e JSON

Neste exercício, você deve melhorar a lista de tarefas a fazer aplicando AJAX para armazenar a lista em format JSON no servidor, permitindo que o usuário elimine uma tarefa da lista.

Para isso, você deve instalar um servidor Web (Apache 2, por exemplo) em seu computador, onde deve colocar o arquivo `tarefas.json` e um programa lado-servidor como o `tarefas-jason.php`

Você pode tomar como base para o seu código os exemplos disponibilizados no SIGAA e adapta-los. Os códigos estão reproduzidos aqui também.

#### Parte 2 – CSS e W3.CSS

Utilize CSS diretamente ou o framework W3.CSS para trazer uma formatação de estilo visual com *responsividade*, aplicando o conceito de grid.

Você pode utilizar um template. Colocamos aqui o exemplo apresentado em aula, que se encontra também no SIGAA.

Exemplo usando AJAX para exibir a lista a partir de um arquivo JSON:

```
<!DOCTYPE HTML>
<html lang="pt-br">
<head>
<meta charset="UTF-8">

<title>Lista de Tarefas - Exemplo com JS 5 - JSON</title>
</head>
<body>
<h1>Lista de Tarefas</h1>
<p>por Jair C Leite, 2018</p>
<h2 id="afazer">A fazer</h2>
  <button id="btn" onclick="ajaxRequest()">Carregar tarefas</button>
<div id="lista">
</div>

<script type="text/javascript" src="todo-app-json-render.js"></script>
```

Arquivo todo-app-json-render.js

```
</body>
</html>
function ajaxRequest() {
  var taskRequest = new XMLHttpRequest();
  taskRequest.open('GET', 'tarefas.json')
  taskRequest.onload = function() {
    if (taskRequest.status >= 200 && taskRequest.status < 400) {
      var taskList = JSON.parse(taskRequest.responseText);
      renderHTML(taskList);
    } else {
      console.log("Servidor ativo, mas ocorreu um erro!");
    }
  };
  taskRequest.onerror = function() {
    console.log("Erro de conexão");
  }
  taskRequest.send();
}

function renderHTML(data) {
  var taskContainer = document.getElementById("lista"),
      htmlString = "<ul>";
```

```
for (i=0; i < data.length; i++) {  
    htmlString += "<li>" + data[i].tarefa + ", " + data[i].responsavel +  
    "</li>";  
}  
htmlString += "</ul>";  
taskContainer.insertAdjacentHTML('beforeend', htmlString);  
}
```

Arquivo tarefas.json

```
[  
  {  
    "tarefa": "Projeto WS",  
    "responsavel": "Joao"  
  },  
  {  
    "tarefa": "Prestação de Contas",  
    "responsavel": "Maria"  
  },  
  {  
    "tarefa": "Planilha Financeira",  
    "responsavel": "Carlos"  
  }  
]
```

Exemplo usando AJAX adicionar uma tarefa na lista usando um arquivo JSON:

```
<!DOCTYPE HTML>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Lista de Tarefas - Exemplo com JS 6 - JSON</title>

  </head>
  <body>
    <h1>Lista de Tarefas</h1>
    <p>por Jair C Leite, 2018</p>
    <div class="container">
      <p>
        <label>Tarefa: </label>
        <input id="tarefa" type="text">
        <label>Responsável: </label>
        <input id="responsavel" type="text">
        <button id="addbtn">+</button>
      </p>

      <h2>A fazer</h2>
      <ul id="lista">
      </ul>

    </div>

    <script type="text/javascript" src="todo-app-json-add.js"></script>

  </body>
</html>
```

Arquivo todo-app-json-add.js

```
function ajaxLoadRequest() {
  var taskRequest = new XMLHttpRequest();
  taskRequest.open('GET', 'tarefas.json');
  taskRequest.onload = function () {
    if (taskRequest.status >= 200 && taskRequest.status < 400) {
      var taskListJson = taskRequest.responseText;
      localStorage.setItem('taskList', taskListJson);
    } else {
      console.log("Servidor ativo, mas ocorreu um erro!");
    }
  }
}
```

```

    }
};
taskRequest.onerror = function() {
    console.log("Erro de conexão");
}
taskRequest.send();
}

var ajaxAddRequest = function () {
    console.log("entrando add...");
    var novaTarefa = document.getElementById("tarefa").value,
        novoResponsavel = document.getElementById("responsavel").value,
        tasks = localStorage.getItem('taskList'),
        taskListObj = JSON.parse(tasks),
        toAdd = {"tarefa":novaTarefa , "responsavel":novoResponsavel};
    taskListObj.push(toAdd);
    //grava nova lista de tarefas com Ajax
    var taskRequest = new XMLHttpRequest();
    taskRequest.open('POST', 'tarefas-json.php');
    taskRequest.onload = function() {
        if (taskRequest.status >= 200 && taskRequest.status < 400) {
            console.log("Sucesso em ajaxAddRequest\n" +
taskRequest.responseText);
        } else {
            console.log("Servidor ativo, mas ocorreu um erro!");
        }
    };
    taskRequest.onerror = function() {
        console.log("Erro de conexão");
    }
    taskRequest.setRequestHeader("Content-type", "application/x-www-form-
urlencoded");
    var queryString = "tarefas=" + JSON.stringify(taskListObj);
    taskRequest.send(queryString);
    loadRender();
}

var ajaxDelRequest = function () {
    /* implementar o código aqui como exercício */
    console.log("ajaxDelRequest");
}

```

```

function renderHTML(data) {
    var taskContainer = document.getElementById("lista"),
        htmlString = "";
    for (i=0; i < data.length; i++) {
        htmlString += "<li>" + data[i].tarefa + ", " + data[i].responsavel +
        "</li>";
    }
    taskContainer.innerHTML = htmlString;
}

function loadRender() {
    ajaxLoadRequest();
    var tasks = localStorage.getItem('taskList');
    var taskListObj = JSON.parse(tasks);
    renderHTML(taskListObj);
}

// PRINCIPAL
var addbtn = document.getElementById("addbtn");
addbtn.addEventListener("click", ajaxAddRequest);
loadRender();

```

Arquivo tarefas-json.php

```

<?php
$tarefas = null;

echo "PHP OK1\n";

if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    echo "PHP OK2\n";
    if(!empty($_POST['tarefas'])) {
        $tarefas = $_POST['tarefas'];
        echo "PHP OK3 tarefas=". $tarefas. "\n";
        $users = fopen("tarefas.json", "w+") or die("unable to open file!");
        echo "PHP OK4\n";
        fwrite($users, $tarefas);
        fclose($users);
    } else
        echo "Valores a adicionar estão vazios. Nada a inserir!";
}
?>

```