 <small>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</small>	Programação em Ambiente Web Docentes – FAS, JRMR, NJR Ficha Prática 3
--	---

#### Documentação de Apoio

- Slides disponíveis na plataforma moodle da Unidade Curricular
- Cheat sheet para HTML - <https://websitesetup.org/html5-cheat-sheet/>
- Cheat sheet para CSS - <https://websitesetup.org/css3-cheat-sheet/>
- Browser Web APIs - <https://developer.mozilla.org/en-US/docs/Web/API>

## 1 Revisão aula teórica

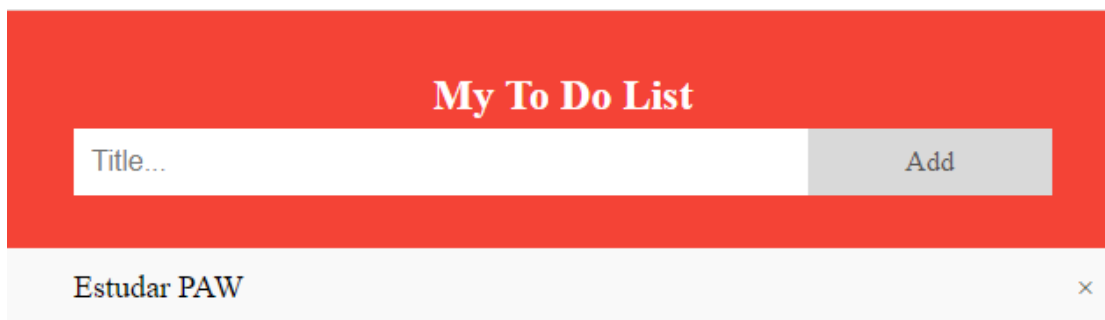
Usando os slides da aula teórica 3, deve criar ficheiros com extensão html e reproduzir os exemplos pedidos nesta secção. Deve implementar todo o comportamento dos exemplos com recurso a apenas um ficheiro.

Os projetos que deve reproduzir são:

- Exemplo slide 13;
- Exemplo slide 20;
- Exemplo slide 27;
- Exemplo slide 31;
- Exemplo slides 51,52,53;

## 2 Lista de tarefas simples

Crie uma simples aplicação do estilo ToDo, Figura 1, com um formulário que capture notas composto por uma caixa de texto e um botão para adicionar a uma lista de notas já existente.



**Figura 1 - Exemplo de uma simples aplicação ToDo**

1. Crie o código-fonte html, css e javascript necessário para resolver este problema. O aspeto da página dever ser semelhante ao idealizado no mockup acima demonstrado. Pode utilizar frameworks para auxílio ao design como a framework bootstrap;
2. Atente que a aplicação deve conseguir listar os diferentes ToDos criados e ter uma opção para os remover caso não sejam necessários. A remoção deve ser individual;
3. Qual o melhor método para guardar dados, tendo em conta os conhecimentos atuais? Justifique a sua resposta.
4. Inclua persistência de dados usando a persistência HTML5 Storage para que a aplicação recorde as suas notas entre sessões do browser.
5. Crie um pequeno form para capturar o nome do utilizador e o guardar numa cookie. Adicione uma função que deve ser executada após o evento “DOMContentLoaded” para verificar a existência do cookie com o nome do utilizador, caso exista o form para pedir esta informação deve ser escondido. Sempre que for reconhecido o nome do utilizador no mockup deve ser substituída a palavra “My” pelo nome do utilizador.

### 3 Weather Web App

Utilizando o código disponível no repositório git em <https://github.com/fasIPP/weatherInterface> e proceda ao clone do projeto para uma pasta local. Deve criar um repositório git local e registar todas as alterações a respostas que incluam desenvolvimento.

O uso deste projeto implica o registo no website [openweathermap.org](https://openweathermap.org) para obter uma chave de acesso à api fornecida por este website. Para o efeito registe-se no site: <https://openweathermap.org>. Para ajuda com este processo pode seguir o conjunto de indicações em: <https://openweathermap.org/guide> até ao passo I-3. Iremos apenas usar serviços free deste serviço.

Após criada a conta deve obter a sua API\_Key em [https://home.openweathermap.org/api\\_keys](https://home.openweathermap.org/api_keys) e copiar o código para o projeto substituindo o placeholder “YOUR\_KEY\_HERE” no ficheiro main.html.

Em seguida dê respostas às seguintes questões:

1. Teste a aplicação e verifique o seu correto funcionamento.
2. O projeto no repositório usa que versão do javascript?
3. O projeto faz uso de alguma(s) frameworks para desenvolvimento web já estudadas?
4. Separe o conteúdo do projeto organizando-o em ficheiros “.html”, “.css” e “.js”.
5. Assegure-se que não existe javascript fora do ficheiro javascript, css fora do ficheiro css e html fora do ficheiro html. Use as propriedades DOM para atingir este efeito.
6. Identifique funções onde o jQuery, carregado na página poderia ser utilizado e reescreva as funções para tomar partido desta framework.
7. Altere novamente o projeto para com recurso à API de geolocalização do HTML5 pré-preencher os campos do formulário fornecido com a posição actual do utilizador da aplicação.
8. Considere o uso de uma biblioteca de fonts e images, fontawesome, (<https://fontawesome.com/icons?d=gallery>) para contextualizar o estado do tempo com uma imagem. Deve importar o ficheiro css da biblioteca para o seu projeto com:
 

```
<link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.7.1/css/all.css" integrity="sha384-fnmOCqbTlWIlj8LyTjo7mOUStjsKC4pOpQbqyi7RrhN7udi9RwhKkMHpvLbHG9Sr" crossorigin="anonymous">
```
9. Estude os serviços disponibilizados pela API openweathermap e implemente novos pedidos à API no seu projeto para obter o weather forecast dos próximos dias para a localização escolhida também.

### 4 Lista de tarefas avançada

Fazendo uso de um projeto moderno sobre ToDos, usando a framework bootstrap para formatação da pagina, complete a aplicação, implementando as funções em falta para que esta funcione corretamente.

Para este exercício deverá fazer uso da ferramenta de versões git e clonar o repositório publicamente acessível em <https://github.com/fasIPP/AdvancedToDo>. Deve também fazer commit de cada resposta no seu git local.

1. Faça clone do repositório git para o seu computador local (git clone <https://github.com/fasIPP/AdvancedToDo>).
2. Verifique que o estado atual do projeto está conforme a Figura 2.
3. Estude o código-fonte em html, css e javascript de forma a se aperceber do formato e estrutura da aplicação. A aplicação deve registar novos ToDos a partir do formulário definido e adicionar à lista já existente. Quando a checkbox é premida, o ToDo deve

migrar para a secção “Already Done”. Finalmente, após clique no botão na secção “Already Done” o ToDo deve desaparecer da sua página.

- Complete o código presente no ficheiro “js/main.js” com comentário “(To Complete)” para que a aplicação funcione de acordo com o esperado. (Nota: em eventos as funções que contêm a palavra this, refere-se ao elemento html que invocou o evento)
- Adicione persistência entre sessões do browser com a ajuda de HTML5 Storage.

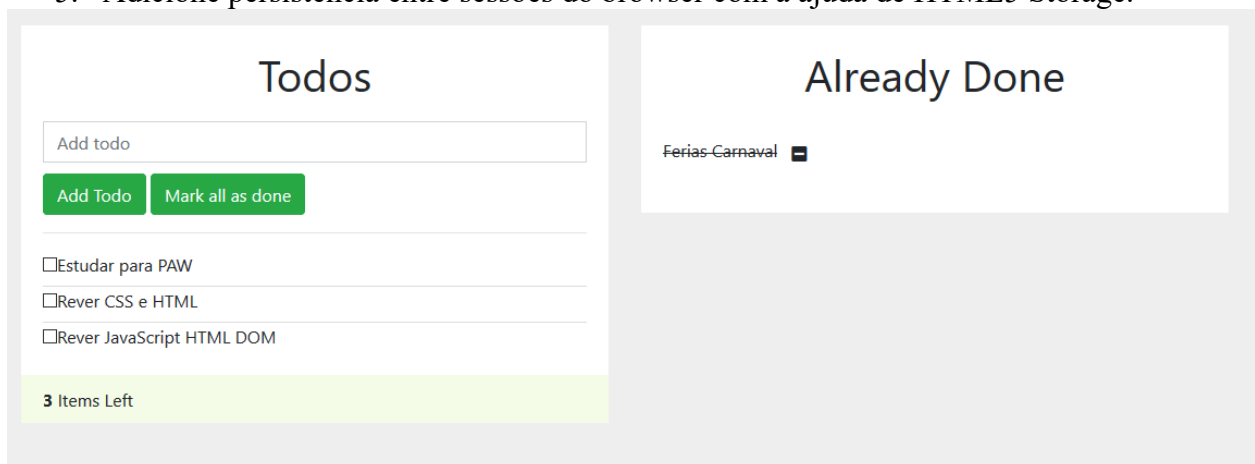


Figura 2 - Aspeto final da aplicação ToDo Avançado

## 5 Interface para API OpenChargeMaps

Considerando o exemplo já implementado nas aulas teóricas, iremos fazer uso da API OpenChargeMap para apresentar numa lista/grelha os pontos de abastecimento elétricos mais próximos de uma determinada localização. O uso desta API não requer registo no website e pode ser usada diretamente nos projetos.

### 5.1 Comportamento básico

Deve ser implementado um formulário para novamente capturar a latitude e longitude do local a pesquisar, assim como o número máximo de resultados a serem devolvidos pela API. Deve ser usado o seguinte endpoint para pedidos GET em AJAX  
`https://api.openchargemap.io/v3/poi/?output=json&latitude=\${latitude}&longitude=\${longitude}&maxresults=\${maxResults}`

Deve ser criada uma grelha com recurso a bootstrap que após a submissão do formulário deve apresentar uma lista com informação relevante sobre cada posto de carregamento de forma agradável para o utilizador.

### 5.2 Uso da biblioteca de mapas em javascript (Opcional)

Deve ser criado um mapa com marcadores indicando a posição de cada carregador. Para recurso a esta funcionalidade irá ser usada a biblioteca leaflet disponível em <https://leafletjs.com/>. Pode encontrar um simples tutorial de como configurar o seu mapa em: <https://leafletjs.com/examples/quick-start/>.

Sumariamente deve:

Deve adicionar ao header do seu projeto as tags:

```
<link rel="stylesheet" href="https://unpkg.com/leaflet@1.4.0/dist/leaflet.css"
      integrity="sha512-puBpdR07980ZvTTbP4A8Ix/1+A4dHDD0DGqYW6RQ+9jxkRFclaxxQb/SJAWZfWAKuyeQUyt07+7N4QKrDh+drA==">
```

```
crossorigin=""/>  
<script src="https://unpkg.com/leaflet@1.4.0/dist/leaflet.js"  
  integrity="sha512-  
QVftwZFqvtRNi0ZyCtsznlKSW0StnDORoeFr1enyq5mVL4tmKB3S/EnC3rRJcxCPavG10IcrVGSmPh6Qw  
5lwrg=="  
  crossorigin=""></script>
```

Após este facto deve criar um elemento HTML div com o id “myMap”  
Configurar via CSS o tamanho div com id “myMap” para apresentar o mapa  
Inicializar o mapa numa posição lat, lon com um zoom definido:

```
var mymap = L.map('mapid').setView([lat, lon], zoom);
```

Criar marcadores e adiciona-los ao mapa criado. Considere o exemplo de adicionar um marcador ao mapa.

```
var marker = L.marker([51.5, -0.09])  
marker.bindPopup("<b>Hello world!</b><br>I am a popup.")  
marker.addTo(mymap);
```