

**Curso Técnico Superior Profissional em:** Tecnologias e Programação de Sistemas de Informação

**1.º Ano/ 2.º Semestre**

**Unidade Curricular:** Desenvolvimento Web - Back-End

**Docente:** David Jardim

### FICHA DE TRABALHO 3

## Exercícios:

1. Crie uma pasta para a aula 3
2. Crie um ficheiro app.js
3. Dado que as funções em Javascript são “First-Class” podemos efetuar atribuições de funções a variáveis e utilizá-las como argumentos para outras funções
  - a. Simule um processo de Download com a criação de 3 funções: **started, update, completed**
  - b. Efetue o código necessário para chamar a seguinte função

```
performDownload(started, update, completed);
```

- c. O output deverá ser o seguinte:

```
Started Download
0% of Download
1% of Download
2% of Download
3% of Download
4% of Download
5% of Download
```

4. Crie um módulo local com o nome **ArrayUtils.js** e implemente as seguintes funções:
  - a. boolean isEmpty(array) – Verifica se o array está vazio
  - b. number max(array) – Devolve o máximo do array
  - c. number min(array) – Devolve o mínimo do array
  - d. number average(array) – Devolve a media do array
  - e. number indexOf(array, value) – Devolve o índice de um determinado elemento no array
  - f. Array subArray(array, startIndex, endIndex) – Devolve um sub-array do array original
  - g. boolean isSameLength(a1, a2) – Compara o tamanho de dois arrays
  - h. Array reverse(array) – Inverte a ordem de um array
  - i. Array swap(array, index1, index2) – Troca dois elementos de um array
  - j. boolean contains(array, value) – Indica se um determinado valor existe no array
    - i. Já fez algum método que pode ajudar nesta operação?
  - k. Array concatenate(a1, a2) – Junta dois arrays num novo
5. Exporte um objeto que contenha as funções anteriores
6. Importe o módulo para o app.js utilizando a função **require**

Cofinanciado por:

7. Efetue testes para todas as funções e as funções tem que conseguir lidar com situações de erro, por ex. arrays vazios.

Cofinanciado por:

