

#### Caracterização da Unidade Curricular / Characterization of the Curricular Unit

**Designação da Unidade Curricular (UC) / Title of Curricular Unit (CU):** Aplicações e Tecnologias Web / Web Applications and Technologies

**Área científica da UC / CU Scientific Area:** Informática / Computer Science

**Semestre / Semester:** 4º

**Número de créditos ECTS / Number of ECTS credits:** 6

**Carga horária por tipologia de horas / Workload by type of hours:** TP: 22,5; PL: 45; OT: 9; O: 13,5

**Carga letiva semanal / Weekly letive charge:** 4,5h

#### Objetivos de aprendizagem (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

No final da unidade curricular os alunos deverão desenvolver de forma autónoma sítios e aplicações para a WEB baseados no tratamento e processamento de conteúdos de forma dinâmica, bem como assegurar o processo de alojamento e de implementação na WEB.

#### Intended learning outcomes (knowledge, skills and competences to be developed by the students):

At the end of the curricular unit, students should be autonomous in the development of websites and applications for the WEB based on interactive and dynamic content, as well, ensuring the process of hosting and implementation on the WEB.

#### Conteúdos programáticos:

##### 1. Conceitos básicos

- 1.1. Código executado vs. código processado
- 1.2. Hiperligações, hipertexto, hipermedia
- 1.3. Navegador web: interface ou aplicação
- 1.4. Serviços partilhados vs. dedicados
- 1.5. Domínio e serviço

##### 2. PHP

- 2.1. Variáveis e tipo de dados
- 2.2. Controle do fluxo de um programa
- 2.3. Funções nativas vs. funções definidas pelo utilizador
- 2.4. Ficheiros: leitura e escrita
- 2.5. Formulários: métodos POST e GET
- 2.6. PHP e MySQL

2.7. Conceito de sessão

### **3. HTML e CSS**

- 3.1. Estrutura de um documento HTML
- 3.2. Editores
- 3.3. Marcas, elementos, propriedades, valores
- 3.4. Validação em HTML e CSS
- 3.5. Frameworks: bootstrap

### **4. Node.js**

- 4.1. Estrutura Node.js e princípios operacionais
- 4.2. A função de node.js em estruturas javascript

### **5. Angular**

- 5.1. Estrutura de aplicações angulares
- 5.2. Modularização, controladores, rotas e visualizações
- 5.3. Requisições e serviços
- 5.4. Filtros, componentes e testes
- 5.5. Formulários

### **6. React**

- 6.1. Estrutura das aplicações React
- 6.2. Modularização, dados, componentes
- 6.3. Requisições e serviços

### **Syllabus:**

#### **1. Basic concepts**

- 1.1. Code executed vs. code processed
- 1.2. Hyperlinks, Hypertext, Hypermedia
- 1.3. Web browser: interface or application
- 1.4. Shared vs. dedicated services
- 1.5. Domain and service concepts

#### **2. PHP**

- 2.1. Variables and data type
- 2.2. Control of the Flow of a PHP program
- 2.3. PHP native functions vs. user defined functions

Sem Validade  
Administrativa

- 2.4. Files: reading and writing
- 2.5. Forms: POST and GET methods
- 2.6. PHP and MySQL
- 2.7. Concept of session

### **3. HTML and CSS**

- 3.1. Basic structure of a document
- 3.2. Editing tools
- 3.3. Tags, elements, properties, values
- 3.4. Validation of HTML and CSS documents
- 3.5. Frameworks: Twiter bootstrap

### **4. Node.js**

- 4.1. Node.js structure and operating principles
- 4.2. The role of node.js in javascript frameworks

### **5. Angular**

- 5.1. Structure of angular applications
- 5.2. Modularization, controllers, routes, and views
- 5.3. Requisitions and services
- 5.4. Filters, components and tests
- 5.5. Forms

### **6. React**

- 6.1. Structure of react applications
- 6.2. Modularization, data, components
- 6.3. Requisitions and services

Sem Validade  
Administrativa