

**TeSP - Tecnologia e Programação em Sistemas de Informação**

Técnico Superior Profissional

Plano: Aviso nº 909/2016 - 27/01/2016

**Ficha da Unidade Curricular: Tecnologias da Internet**

ECTS: 6; Horas - Totais: 162.0, Contacto e Tipologia, PL:82.50;

Ano | Semestre: 1 | S2

Tipo: Obrigatória; Interação: Presencial; Código: 60248

Área de educação e formação: Ciências informáticas

**Docente Responsável**

José Manuel Palma Redes Ramos

Professor Coordenador

**Docente(s)**

José Manuel Palma Redes Ramos

Professor Coordenador

**Objetivos de Aprendizagem**

1. Desenvolver código HTML, CSS e javaScript;
2. Aplicar estruturas de dados em javaScript;
3. Aplicar paradigmas de HTML Dinâmico e de Document Object Model;
4. Aplicar interação gráfica e cinemática;
5. Implementar interfaces web dinâmicos;
6. Implementar projetos Web do lado do cliente.

**Objetivos de Aprendizagem (detalhado)**

1. Aplicar formatos de apresentação e de interação através de elementos semânticos e de templates;
2. Aplicar diferentes paradigmas de geração das componentes do interface Web: estática, dinâmica, DOM (Document Object Model) e mista;
3. Aplicar conceitos modulares na trilogia interface + processos + dados;
4. Aplicar processos sobre estruturas de dados em javaScript;
5. Aplicar processos a partir de bibliotecas de javaScript;

6. Aplicar processos de interação e de animação de gráficos, raster (Canvas) ou vetoriais (SVG);
7. Aplicar processos de interação e de animação de componentes do interface Web;
8. Implementar representações gráficas de dados complexos;
9. Implementar interfaces Web a partir de Front-end Frameworks;
10. Implementar processos através de bibliotecas e frameworks javaScript;
11. Implementar projetos de aplicações Web, do lado do cliente, com funcionalidades gráficas, dinâmicas e interativas, a partir de fontes diversificadas de dados e de recursos.

### **Conteúdos Programáticos**

1. Codificação de interfaces em HTML5;
2. Codificação de estilos em CSS;
3. Programação de interação em JavaScript;
4. Estruturas de dados em javaScript;
5. Paradigmas de HTML Dinâmico e de Document Object Model;
6. Geração de gráficos;
7. Aplicação de bibliotecas;
8. Técnicas de animação;
9. Interatividade gráfica e cinemática;
10. Implementação de Projetos de interfaces Web dinâmicos.

### **Conteúdos Programáticos (detalhado)**

1. Elementos semânticos em interfaces Web;
2. Gestão do interface Web via HTML dinâmico, versus, via DOM (Document Object Model);
3. Representação de dados em javaScript: arrays, objetos e respetivas combinações;
4. Aplicação de processos sobre estruturas complexas ou combinadas de dados;
5. Aplicação de formulários de input/output;
6. Implementação dinâmica de gráficos raster (Canvas) e de gráficos vetoriais (SVG);
7. Implementação de processos de interação e de animação a componentes de gráficas;
8. Implementação de gráficos de pontos, de barras, lineares, ou circulares, a partir de dados representados em estruturas de dados tabelares em javaScript;
9. Implementação de interfaces Web a partir de Front-end Frameworks;
10. Implementação de processos através de bibliotecas e frameworks javaScript;
11. Implementação de aplicações Web, do lado do cliente, com funcionalidades gráficas, dinâmicas e interativas, a partir de fontes diversificadas de dados e de recursos.

### **Metodologias de avaliação**

A avaliação incide sobre a apresentação e defesa de dois mini-rojetos Web do lado do cliente:

- a) Implementação de uma apresentação interativa de informação a partir de uma estrutura de dados em javaScript, dos respetivos recursos mediáticos e de um enunciado com os requisitos de apresentação - 50% da avaliação;
- b) Implementação de uma aplicação web interativa com características gráficas ou de animação - 50% da avaliação.

O processo de avaliação é realizado em sessão de apresentação dos mini-projetos e a

aprovação na Unidade Curricular está sujeita à classificação mínima de 7/20 valores em qualquer dos mini-projetos.

### **Software utilizado em aula**

Editor Notepad++; Browser Google Chrome;

### **Estágio**

Não aplicável.

### **Bibliografia recomendada**

- Haverbeke, M. (2018). *Eloquent JavaScript - A Modern Introduction to Programming* San Francisco: No Starch Press  
- Geary, D. (2012). *Core HTML5 Canvas: Graphics, Animation, and Game Development* Boston: Prentice-Hall  
- International, E. (0). *ECMAScript 2018 – Language Specification (9.th Edition, June 2018)*  
Acedido em 27 de março de 2019 em  
<https://www.ecma-international.org/publications/files/ECMA-ST/Ecma-262.pdf>

### **Coerência dos conteúdos programáticos com os objetivos**

Objetivo 1: Conteúdos 1, 2, 3.

Objetivo 2: Conteúdos 1, 2, 3, 4.

Objetivo 3: Conteúdos 1, 2, 3, 4, 5.

Objetivo 4: Conteúdos 1, 2, 3, 4, 5, 6, 7, 8.

Objetivo 5: Conteúdos 1, 2, 3, 4, 5, 6, 7, 8, 9.

Objetivo 6: Conteúdos 1, 2, 3, 4, 5, 6, 7, 8, 9, 10.

### **Metodologias de ensino**

1. Sessões plenárias expositivas de instrução e demonstração - 28 horas;
2. Sessões laboratoriais para desenvolvimento e implementação de miniprojetos - 42 horas.
3. Atividades semanais autónimas, suportadas por plataforma de e-learning;

### **Coerência das metodologias de ensino com os objetivos**

A aplicação de uma dimensão teórico-prática do processo de aprendizagem segundo um método expositivo, ilustrativo e demonstrativo, com utilização de meios audiovisuais avançados, com a permanente participação dos alunos, é coerente com a complexidade dos conteúdos de suporte para a consecução dos objetivos, como se demonstra a seguir.

Trata-se de consolidar a aquisição do conhecimento teórico complexo, através de observação participativa das respetivas aplicações, tanto à priori como à posteriori da sua aplicação individual por cada aluno. O professor apresenta as fontes de consulta e demonstra os processos de desenvolvimento e implementação de pequenos projetos que os alunos deverão treinar nas suas

atividades autónomas. Os alunos apresentarão as dúvidas e dificuldades. Professor e alunos, colaboram na interpretação dos enunciados dos pequenos projetos e na análise e decomposição das suas fases de implementação.

A aplicação de uma dimensão de prática laboratorial do processo de aprendizagem segundo um ambiente de implementação de pequenos projetos evolutivos de programação Web, é coerente com a prossecução de todos os objetivos, como se demonstra a seguir.

Trata-se de formar e consolidar competências de implementação efetiva de pequenos projetos Web, que incluem a interpretação dos enunciados de pequenos projetos, a análise e decomposição das suas fases de implementação e a sua concretização, com recurso a repositórios de recursos mediáticos (textos, imagens, sons, vídeos, websites) e a linguagens de marcação (HTML5 e XML), de estilização (CSS) e de programação (javaScript), com especial incidência para esta última. Remetem essencialmente para a formação de competências duras (hard skills) de desenvolvimento e implementação de pequenos projetos Web. É essencial a vivência em ambiente laboratorial com uso de recursos computacionais, incluindo editores de código e web-browsers.

Conclui-se assim pela coerência das metodologias de ensino com o Objetivo geral de “Desenvolver e implementar projetos de aplicações Web, do lado do cliente, com funcionalidades gráficas, dinâmicas e interativas, a partir de fontes diversificadas de dados e de recursos”.

### **Língua de ensino**

Português

### **Pré-requisitos**

Não aplicável.

### **Programas Opcionais recomendados**

Não aplicável.

### **Observações**

---

### **Docente responsável**

---