

Ano letivo: 2022/2023

Curso: Lic. Engenharia De Redes E Sistemas De Computadores

Unidade Curricular	Programação Web
--------------------	-----------------

Lic.	Ano do curso	2º ano	2º semestre	ECTS	
------	--------------	--------	-------------	------	--

NOME do ALUNO: Keven Patrick Almeida Garcia nº30615

Prova Escrita

Versão: B

Duração: 100 minutos

Leia atentamente toda a prova antes de iniciar.

A prova é individual, não sendo permitido consultar os seus colegas. No entanto, pode consultar os apontamentos das aulas e a Internet.

O resultado final deve ser enviado para o moodle incluindo o Word da prova e PDF da prova (gravar como PDF) e os ficheiros HTML e JS desenvolvidos. Deve ser anexado o link para Github no tópico Avaliação.

No documento de resposta deve ser incluída a versão da prova.

Durante a resolução deve ir gravando o trabalho para salvaguardar as alterações.

Parte I

(25 valores)

1. À luz do que aprendeu na UC, comente a seguinte imagem.

Cofinanciado por:





R: A imagem mostra os elementos-chave na construção de sistemas web e de aplicações móveis. No canto superior esquerdo está o ícone "HTML 5", seguido pelo "JavaScript" para interatividade. Abaixo, o ícone "WEB" representa a interface do utilizador. À esquerda, um ícone de aplicação Java sugere desenvolvimento móvel. Os ícones convergem para um "Web Service" central, conectando o front-end ao back-end via tecnologias como PHP e Node.js. Na parte inferior direita, um ícone da base de dados MySQL representa o armazenamento de dados.

2. Crie um protocolo para os alunos do IPVC para almoçar na cantina. Para que servem os protocolos e dê um exemplo

Parte II

(25 valores)

1. Considera os seguintes exemplos de objetos DOM.
 - document.getElementById(id)
 - document.getElementsByTagName(tagName)
 - document.getElementsByClassName(className)

Cofinanciado por:



Porque no primeiro caso temos getElement e nos dois seguintes getElements? Dê um exemplo de utilização para cada exemplo

R: No primeiro caso, utiliza-se "getElement" porque ele recupera apenas um elemento, e no documento só pode existir um elemento com o mesmo ID. Nos outros casos, utiliza-se "getElements" porque pode ler vários elementos de acordo com sua classe ou tag.

Exemplos estão no ficheiro exemplosDOM.html da pasta Parte 2

2. Cria uma estrutura em JSON para registar Atores e Filmes. Faz um XML para a mesma estrutura. Comenta os resultados.

R: A resposta estão nos ficheiros atoresFilmes.json e atoresFilmes.xml da pasta Parte 2

Parte III

(20 valores)

1. Qual a diferença entre <p> e <pre>
2. <p> é usada para definir um parágrafo de texto e nela múltiplos espaços em branco e quebras de linha são tratados como um único espaço, enquanto que <pre> é usada para definir texto pré-formatado, onde todos os espaços, quebras de linha e tabulações são preservados exatamente como estão no código fonte.
3. Para que server

<meta charset="utf-8">

Essa tag garante que todos os caracteres, incluindo letras de diferentes idiomas e símbolos especiais, sejam exibidos corretamente.

Parte IV

(30 valores)

1. Prepara uma página com uma tabela 2x2 com estilos CSS que permitam apresentar 4 marcas de produtos de rede. Usa cores de fundo e cores de escrita e o logotipo de cada marca.

R: A resposta está na pasta Exercícios ficheiro Pt4ex1.html

Cofinanciado por:



UNIÃO EUROPEIA
Fundo Social Europeu

Parte V

(50 valores)

1. Usando o Bootstrap, construa uma página com cards que mostre 6 monumentos e atrações turísticas do seu local de residência.
2. Cada card tem de ter um botão “ver mais” para ver mais detalhes.

R: A resposta está na pasta Exercícios ficheiro Pt5ex1.html

Parte VI

(50 valores)

Considere as imagens seguintes.

```
routes > JS products.js > ...
1  const productsRouter = require('express').Router();
2  const controller = require('../controllers/products');
3  const authMiddleware = require('../middlewares/auth/auth');
4
5
6
7
8
9
10
11
12  module.exports = productsRouter;
```

Figura 2 - Rotas

Cofinanciado por:



UNIÃO EUROPEIA
Fundo Social Europeu

```
controllers > JS products.js > ...
1  const apiResponse = require('../utils/response/apiResponse');
2  const Products = require('../data/entities/products');
3
4  > exports.getAll = async (req, res) => { ...
15 }
16
17 > exports.getById = async (req, res) => { ...
30 }
31
32 > exports.create = async (req, res) => { ...
49 }
50
51 > exports.update = async (req, res) => { ...
72 }
73
74 > exports.delete = async (req, res) => { ...
92 }
```

Figura 3 - Controller Produtos

1.1 - Complete o ficheiro de rotas dos produtos.

R: A resposta está no ficheiro `routes.js` da pasta Parte 6

1.2 - Explique cada uma das linhas do ficheiro anterior

R: A resposta está no ficheiro `routescomentado.js`

1.3 - Desenvolva um ficheiro JSON que permita guardar a informação dos produtos e escreva o código para cada um dos métodos do controller products.

R: A resposta está no ficheiro `controllers.js` da pasta Parte 6 e no ficheiro `produtos.json`.

2. O Resultado final da prova escrita deve ser colocada no github sendo partilhado o link como resposta à prova.

Cofinanciado por:





Bom trabalho!

António Lira Fernandes

Cofinanciado por:



UNIÃO EUROPEIA
Fundo Social Europeu