PROGRAMAÇÃO WEB AVANÇADA

Mestrado em Engenharia Informática

Código: 26205

Área Cientifica Predominante: Ciências da Computação

Docente: Paulo Rafael da Costa e Sousa

Idioma de Instrução: Português

Regime: S2

Carga Letiva: 60h Carga Trabalho: 108h

ECTS: 6,0

Objetivos

O desenvolvimento de aplicacoes para a Web e uma atividade que combina, em igual proporcao, o design, a usabilidade e a tecnologia. Para ser proficiente no desenvolvimento de aplicacoes Web, e fundamental dominar aspetos tao variados como protocolos de comunicacao, engenharia de software, gestao de redes, design de interfaces, bases de dados, linguagens de anotacao, scripting, seguranca, legislacao, entre outros.

Esta unidade curricular visa fornecer aos alunos os conhecimentos avancados para o desenvolvimento de aplicacoes Web modernas, que sejam simultaneamente seguras e eficazes (do ponto de vista da usabilidade e desempenho), tanto numa perspetiva de Front-end e Back-end, como de DevOps.

Os resultados serão alcançados gradualmente ao longo do semestre, através de exercícios práticos desenvolvidos em grupo, onde os estudantes são encorajados a pesquisar e desenvolver estratégias para implementar as tecnologias estudadas nas aulas.

Resultados da Aprendizagem

No final da unidade curricular os alunos deverao ser capazes de:

Compreender o protocolo HTTP
Compreender as diferencas entre os varios metodos e codigos de erro HTTP
Conhecer as linguagens fundamentais de programacao para a Web
Construir um sitio Web a partir de um mockup de design
Conhecer e utilizar eficazmenteframeworksCSS e Javascript
Desenvolver aplicacoesserver-side,nomeadamente utilizando a framework NestJS
Conhecer Webservicesde acesso publico e integrar os mesmos nas suas aplicacoes
Desenvolver Webservicessegundo a metodologia RESTful
Desenvolver aplicacoes Webcliente-siderecorrendo aframeworksde Javascript
Reconhecer quando e como utilizar bases de dados NoSQL
Conhecer estrategias dedeploymentautomatico
Conhecer servicos Cloud paradeploymentde aplicacoes Web

Conteúdos Programáticos

01 – Introducao a Web e ao protocolo HTTP

Standards W3C

Protocolo HTTP

Componentes da Web

Pedidos HTTP

Status codes

Formato de mensagens/MIME types

02 – Javascript e introdução ao NodeJS

Tipos de dados

Operadores: atribuicao, aritmeticos, binarios, logicos, precedencia, etc.

Expressoes, estruturas de decisao, estruturas de repeticao

Estruturas de dados (Objetos e Arrays)

Funcoes nativas

ECMA2015/ES6

Classes e funcoes

Browser objects

Introducao ao NodeJS

Ecossistema do NodeJS

Node Package Manager (npm)

Anatomia de uma aplicacao NodeJS

03 – Tópicos Avançados de Javascript

Modulos

Callbacks vs Promises vs Async

04 - Programação server-side: NestJS

O que é?

Estrutura em layers: Controllers, Services, Models

Providers

Modules

Scaffolding de aplicações NestJS

Geração de resources CRUD

Controllers

05 – Arquiteturas orientadas ao servico

Introducao as Arquiteturas Orientadas ao Servico (SOA)

SOAP vs REST

RESTful vs RESTless

Operacoes CRUD

Boas praticas de desenvolvimento de servicos RESTful

Tratamento de erros

06 - Aula de acompanhamento e apoio ao desenvolvimento do Trabalho Pratico

Espaco para reflexao e discussao e desenvolvimento de solucoes no ambito da UC

Modelo de domínio

07 – Sistemas de indexacao escalaveis e bases de dados NoSQL

Introducao ao paradigma NoSQL

Analise comparativa de bases de dados NoSQL

Bases de dados orientadas ao documento (MongoDB)

Operacoes CRUD em MongoDB

Principais operadores

Tipos de indices

Agregacoes

Comunicacao NodeJS/Mongo (Mongoose)

08 – Programação client-side: XHTML, HTML5 e CSS 3

Anatomia de um documento HTML

XHTML

Elementos introduzidos pelo HTML 5

Seletores CSS

Prioridade das regras CSS

Box model

Formatacao de texto

Posicionamento de elementos

Seletores avancados

Pseudo classes

CSS 3

Design responsivo

Frameworks de CSS

09 – Programacao client-side: React

Introducao ao React

Scaffolding de uma aplicacao React

Anatomia de uma aplicacao React

JSX

Componentes React

Estado do componente

Passagem de parametros

Tipagem de parametros

Eventos

Depuracao de aplicacoes React

10 – Programacao client-side (continuacao)

Ciclo de vida de um componente React

Tratamento de formularios

Roteamento

Hooks

11 - Aula de acompanhamento e apoio ao desenvolvimento do Trabalho Pratico

Espaco para reflexao e discussao e desenvolvimento de solucoes no ambito da UC

Push notifications

12 – DevOps

Introducao ao conceito de DevOps

Ferramentas de apoio a atividade de DevOps

Construcao de containers Docker

Composicao de containers Docker

Deployment de containers Docker

13 - Aula de acompanhamento e apoio ao desenvolvimento do Trabalho Pratico

Espaco para reflexao e discussao e desenvolvimento de solucoes no ambito da UC

Interceptors axios

14 - Aula de acompanhamento e apoio ao desenvolvimento do Trabalho Pratico

Espaco para reflexao e discussao e desenvolvimento de solucoes no ambito da UC

15 – Exame

Exame final

Bibliografia Recomendada

ReactJS by Example – Building Modern Web Applications with React. Vipul A. M., Prathamesh Sonpatki. Packt Publishing, 2016.
Full Stack Javascript, Azat Mardan, APRESS
Enterprise Web Development - Building HTML5 Applications: From Desktop to Mobile
Gourley, D., Totty, B. (2002). HTTP: The definitive guide; [understanding web internals]. Beijing, u.a.: O'Reilly.
Crockford, D. (2008). JavaScript: The good parts. Beijing: O'Reilly.

Métodos de Ensino e de Aprendizagem

As metodologias de ensino e de aprendizagem empregues nesta UC promovem a assimilacao dos conhecimentos

lecionados atraves da realizacao de exercicios praticos e estimulam as capacidades de investigacao e de auto-aprendizagem dos alunos atraves da aplicacao de desafios para resolucao individual e/ou em grupo apoiada pelo docente.

Métodos de Avaliação

Avaliacao em epoca normal

A avaliacao desta unidade curricular e constituida por 2 momentos de avaliacao: 1 trabalho pratico realizado em grupos de 3 pessoas e fora do horario letivo, e um momento de avaliacao individual que assumira a forma de um exame com nota mínima de 8 valores:

TP – Trabalho de grupo
E – Exame individual (nota mínima: 8 valores em 20)

O trabalho pratico sera avaliado segundo diversos criterios definidos no enunciado do proprio trabalho, no entanto, pelo menos as seguintes componentes farao parte dos criterios de avaliacao:

completude e qualidade da solucao apresentada,
qualidade do relatorio escrito, e
qualidade da apresentacao/defesa do trabalho (avaliação individual).
A nota final atribuida a cada aluno corresponde a seguinte formulacao:

Nota final = TP * 0.4 + E * 0.6

A detecao de fraude em algum dos trabalhos implica a reprovacao a unidade curricular.

Época de recurso

A avaliação E pode ser repetida na época de recurso, onde se mantêm as mesmas regras e fórmula de cálculo da nota final.
Nota: a avaliaçãoTPé feita,exclusivamente, durante aépoca normal.