
Ano Letivo 2019-20

Unidade Curricular COMPLEMENTOS DE PROGRAMAÇÃO

Cursos PROGRAMAÇÃO DE DISPOSITIVOS PARA A INTERNET
Tronco comum

Unidade Orgânica Instituto Superior de Engenharia

Código da Unidade Curricular 18411010

Área Científica FORMAÇÃO TÉCNICA, CIÊNCIAS INFORMÁTICAS

Sigla FT

Línguas de Aprendizagem Português

Modalidade de ensino Presencial

Docente Responsável Pedro Jorge Sequeira Cardoso

DOCENTE	TIPO DE AULA	TURMAS	TOTAL HORAS DE CONTACTO (*)
Pedro Jorge Sequeira Cardoso	PL; TP	TP1; PL1	15TP; 45PL

* Para turmas lecionadas conjuntamente, apenas é contabilizada a carga horária de uma delas.

ANO	PERÍODO DE FUNCIONAMENTO*	HORAS DE CONTACTO	HORAS TOTAIS DE TRABALHO	ECTS
2º	S1	15TP; 45PL	125	5

* A-Anual;S-Semestral;Q-Quadrimestral;T-Trimestral

Precedências

Sem precedências

Conhecimentos Prévios recomendados

Recomenda-se, pelo menos, a frequência prévia das U.C.: INTRODUÇÃO À PROGRAMAÇÃO E RESOLUÇÃO DE PROBLEMAS e PROGRAMAÇÃO

Objetivos de aprendizagem (conhecimentos, aptidões e competências)

A UC, lecionado numa vertente eminentemente prática, tem como objetivo que o aluno aprovado seja capaz de:

- Utilizar ferramentas para controlar versões no desenvolvimento de software, conceber e usar wikis, e conceber e usar documentação de código fonte;
- Conhecer e usar os conceitos definidos na programação orientada a objetos (POO) na implementação de soluções informáticas;
- Conceber, programar e usar sistemas de gestão de bases de dados (SGBD) relacionais.

Conteúdos programáticos

1. Ferramentas colaborativas
 1. Controladores de versões
 2. Wikis
 3. Documentação de código fonte
2. Programação Orientada a Objetos (POO)
 1. Motivações para Orientação a Objetos
 2. Classes e objetos
 3. Construtores e destrutores
 4. Sobrecarga de métodos
 5. Atributos de classe e métodos de classe
 6. Herança
 7. Sobreposição
 8. Encapsulamento e os métodos de acesso
 9. Classes abstratas e polimorfismo
3. Bases de dados relacionais
 1. Modelo relacional, modelagem de entidades e normalização
 2. Programação SQL (operações CRUD)
4. Aplicações

Demonstração da coerência dos conteúdos programáticos com os objetivos de aprendizagem da unidade curricular

No sentido de satisfazer os objetivos definidos para a UC, os conteúdos programáticos foram divididos em 3 capítulos principais, a saber:

- O capítulo 1 dos conteúdos programáticos estuda ferramentas para controlar versões no desenvolvimento de software, ferramentas para conceber wikis (e o seu uso) e ferramentas para conceber documentação de código fonte (e o seu uso).
- O capítulo 2 desenvolve as competências necessárias para implementar software recorrendo ao paradigma da programação orientada a objetos (POO).
- O capítulo 3 desenvolve os conceitos fundamentais para o desenvolvimento de sistema de gestão de bases de dados relacionais, nomeadamente a modelação, normalização e programação SQL.

Metodologias de ensino (avaliação incluída)

Funcionamento das aulas

- Aulas teóricas-práticas ζ apresentação e discussão de conceitos teórico-práticos.
- Aulas práticas-laboratoriais ζ implementação de exemplos/projetos sobre os conteúdos do curso.

Avaliação

A avaliação da UC tem 2 componentes:

- PP - Projeto de programação
- PE - Teste/exame

A classificação final será a resultante da média das duas componentes, i.e.,

$$\text{Classificação final} = 0.7 \times \text{PP} + 0.3 \times \text{PE},$$

sendo que

- o aluno deve ter um mínimo de sete (7) valores em cada uma das componentes.
- O aluno fica aprovado se a classificação final for superior a 9,5 valores.

De acordo com o n.º 3 do artigo 6.º do despacho reitoral RT 59/2015, de 28 de julho, a inclusão do cumprimento do dever de assiduidade nos métodos de avaliação é obrigatória, sendo que se considera que um estudante cumpre o dever de assiduidade a uma UC, quando não exceda o número limite de faltas correspondente a 25% das horas de contato previstas.

Demonstração da coerência das metodologias de ensino com os objetivos de aprendizagem da unidade curricular

No contexto dos objetivos definidos para a UC, o ensino será eminentemente prático onde se vão introduzindo os conceitos necessários recorrendo, sempre que possível, a casos práticos. A avaliação está desenhada de modo a que os alunos por um lado captem os conceitos mais teórico-práticos (teste/exame) e por outro lado pratiquem a implementação do aprendido através do desenvolvimento de um projeto.

Bibliografia principal

- Summerfield, M. (2008), Programming in Python 3: A Complete Introduction to the Python Language. Addison-Wesley Professional.
- Borges, L (2010). Python para desenvolvedores. Edição do autor
- Sumathi, S., Esakkirajan, S. (2007). Fundamentals of Relational Database Management Systems. Springer., 2007
- Gouveia, F.(2014). Fundamentos de Bases de Dados, FCA,
- Damas, L. (2007). SQL, 6ª edição, FCA.
- Scott Chacon, Pro Git (Expert's Voice in Software Development). Apress, 2009

Academic Year 2019-20

Course unit COMPLEMENTS OF PROGRAMMING

Courses PROGRAMMING OF INTERNET DEVICES
Tronco comum

Faculty / School INSTITUTE OF ENGINEERING

Main Scientific Area FORMAÇÃO TÉCNICA, CIÊNCIAS INFORMÁTICAS

Acronym FT

Language of instruction Portuguese

Teaching/Learning modality -

Coordinating teacher Pedro Jorge Sequeira Cardoso

Teaching staff	Type	Classes	Hours (*)
Pedro Jorge Sequeira Cardoso	PL; TP	TP1; PL1	15TP; 45PL

* For classes taught jointly, it is only accounted the workload of one.

Contact hours

T	TP	PL	TC	S	E	OT	O	Total
0	15	45	0	0	0	0	0	125

T - Theoretical; TP - Theoretical and practical ; PL - Practical and laboratorial; TC - Field Work; S - Seminar; E - Training; OT - Tutorial; O - Other

Pre-requisites

no pre-requisites

Prior knowledge and skills

-

The students intended learning outcomes (knowledge, skills and competences)

-

Syllabus

-

Demonstration of the syllabus coherence with the curricular unit's learning objectives

-

Teaching methodologies (including evaluation)

-

Demonstration of the coherence between the teaching methodologies and the learning outcomes

-

Main Bibliography

-