English version at the end of this document

Ano Letivo	2020-21
Unidade Curricular	COMPLEMENTOS DE PROGRAMAÇÃO
Cursos	PROGRAMAÇÃO DE DISPOSITIVOS PARA A INTERNET
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 e/ou à distância
Docente Responsável	Pedro Jorge Sequeira Cardoso

DOCENTE	TIPO DE AULA	TURMAS	TOTAL HORAS DE CONTACTO (*)

* 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 RESOLUCAÇÃO 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. 3. Bases de dados relacionais
    1. Modelo relacional, modelagem de entidades e normalização
    2. Programação SQL (operações CRUD)
4. Aplicações

Metodologias de ensino (avaliação incluída)

Funcionamento das aulas

- Aulas teóricas-práticas ¿ apresentação e discussão de conceitos teórico-práticos.
- Aulas práticas-laboratoriais ¿ 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.,

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

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.

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	2020-21
Course unit	COMPLEMENTS OF PROGRAMMING
Courses	PROGRAMMING OF INTERNET DEVICES
Faculty / School	INSTITUTE OF ENGINEERING
Main Scientific Area	
Acronym	
Language of instruction	Portuguese
Teaching/Learning modality	Classroom-based and/or distance learning
Coordinating teacher	Pedro Jorge Sequeira Cardoso

Teaching staff	Type	Classes	Hours (*)

* 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

It is recommended, at least, the prior frequency of the courses: INTRODUCTION TO PROGRAMMING AND PROBLEM RESOLUTION and PROGRAMMING

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

The course is taught in an eminently practical way, aiming to make the approved student able to:

- Use tools to control versions in software development, design and use of wikis, and design and use of source code documentation;
- Know and use the object-oriented programming (OOP) concepts in the implementation of computational solutions;
- Design, program and use relational database management systems (DBMS).

Syllabus

1. Collaborative tools
    1. Version controllers
    2. Wikis
    3. Source code documentation
2. Object Oriented Programming (OOP)
    1. Motivations
    2. Classes and objects
    3. Constructores and destructores
    4. Method's overload
    5. Class attributes and class methods
    6. Enheritance
    7. Encapsulation and access methods
    8. Abstract classes and polymorphism
3. Relational databases
    1. Relational model, entity modeling and normalization
    2. SQL programming (CRUD operations)
4. Applications

Teaching methodologies (including evaluation)

Class operation

- Theoretical-practical classes - presentation and discussion of theoretical and practical concepts.
- Practical-laboratory classes - implementation of examples / projects on the course content.

Evaluation

The UC evaluation has 2 components:

- PP - Programming project
- PE - Test / Exam

The final classification will be the result of the weighetd average of the two components, i.e.,

$$\text{Final classification} = 0.7 \times PP + 0.3 \times PE,$$

being that

- the student must have a minimum of seven (7) values in each of the components.
- The student is approved if the final classification is higher than 9.5.

According to paragraph 3 of article 6 of the ruling dispatch RT 59/2015, of 28 July, the inclusion of compliance with the duty of attendance in the assessment methods is mandatory, and
considers that a student fulfills the duty of attendance to a CU, when he does not exceed the limit number of absences corresponding to 25% of the anticipated contact hours.

Main Bibliography

- 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