
Ano Letivo 2021-22

Unidade Curricular PROGRAMAÇÃO ORIENTADA A OBJETOS

Cursos TECNOLOGIAS INFORMÁTICAS

Unidade Orgânica Instituto Superior de Engenharia

Código da Unidade Curricular 19321001

Área Científica CIÊNCIAS INFORMÁTICAS

Sigla

Código CNAEF (3 dígitos) 481

Contributo para os Objetivos de Desenvolvimento Sustentável - ODS (Indicar até 3 objetivos) 4, 8, 9

Línguas de Aprendizagem Português

Modalidade de ensino

Presencial

Docente Responsável

Pedro Jorge Sequeira Cardoso

DOCENTE	TIPO DE AULA	TURMAS	TOTAL HORAS DE CONTACTO (*)
Cristiano Lourenço Cabrita	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
1º	S1	15TP; 45PL	156	6

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

Precedências

Sem precedências

Conhecimentos Prévios recomendados

programação

Objetivos de aprendizagem (conhecimentos, aptidões e competências)

Proporcionar uma visão geral sobre os conceitos definidos em programação orientada a objetos (POO). Capacitar o aluno a oferecer soluções algorítmicas para problemas computacionais. Utilizar uma linguagem de POO para implementação de soluções.

Conteúdos programáticos

Introdução ao paradigma da Programação Orientada a Objetos

1. Perspetiva histórica
 2. Características da programação orientada a objetos (encapsulamento, partilha de comportamento, evolução)
 3. Objetos (noção de Objeto e identidade; protocolo de utilização de um Objeto; mensagens e métodos; noção de estado e comportamento.
 4. Linguagens baseadas em classes; instanciação de Objetos.
 5. Hierarquias de Objetos: Mecanismos de herança; Reutilização de código; Sistemas reflexivos; Polimorfismo.
 6. Relações entre Objetos: Extensão, generalização e especialização; Associação, agregação e composição; Coesão e acoplamento.
 7. Aplicações (GUI, ligações a bases de dados, aplicações móveis, serviços web)
-

Metodologias de ensino (avaliação incluída)

Metodologias de ensino

Aulas Teórico-Práticas: Método expositivo, orientado de acordo com um ensino baseado em problemas, com projeção e explicação dos objetivos e conteúdos correspondentes a cada aula.

Aulas Práticas: Prática laboratorial em computador. Resolução e codificação de problemas tipificados, selecionados em conformidade com o conteúdo teórico semanal.

Avaliação

Componentes da avaliação classificadas de 0-20 valores:

- Trabalho(s) prático(s) (TP)
- Prova escrita (PE): Teste / Exame

Classificação mínima em cada uma das componentes 7 valores.

Nota final ($\geq 9,5$): $0.3 \text{ PE} + 0.7 \text{ TP}$

De acordo com o n.º 3 do artigo 6.º do despacho reitoral RT 59/2015, de 28 de julho, nos cursos técnicos superiores profissionais, a inclusão do cumprimento do dever de assiduidade nos métodos de avaliação é obrigatória, nos seguintes termos: a) Considera-se 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 contacto previstas

Bibliografia principal

- * José Braga de Vasconcelos (2015). Python - Algoritmia e Programação Web. FCA.
- * Ernesto Costa (2015). Programação em Python - Fundamentos e Resolução de Problemas. FCA
- * Luiz Eduardo Borges (2014). Python para Desenvolvedores. Novatec
- * -(2021). Python 3.9.5 documentation. <https://www.python.org/>

Academic Year 2021-22

Course unit

Courses COMPUTER TECHNOLOGIES

Faculty / School INSTITUTE OF ENGINEERING

Main Scientific Area

Acronym

CNAEF code (3 digits) 481

Contribution to Sustainable Development Goals - SGD (Designate up to 3 objectives) 4, 8, 9

Language of instruction Portuguese

Teaching/Learning modality Presential

Coordinating teacher Pedro Jorge Sequeira Cardoso

Teaching staff	Type	Classes	Hours (*)
Cristiano Lourenço Cabrita	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	156

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

Programming

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

Provide an overview of concepts defined in object-oriented programming (OOP). Enable the student to offer algorithmic solutions to computational problems. Use an OOP language to implement solutions.

Syllabus

Introduction to the Object-Oriented Programming paradigm

1. Historical perspective
2. Characteristics of object-oriented programming (encapsulation, behavior sharing, evolution)
3. Objects (notion of Object and identity; protocol for using an Object; messages and methods; notion of state and behavior).
4. Languages based on classes; object instantiation.
5. Object Hierarchies: Inheritance mechanisms; Code reuse; Reflective systems; Polymorphism.
6. Relations between Objects: Extension, generalization and specialization; Association, aggregation, and composition; Cohesion and coupling.
7. Applications (GUI, database connections, mobile applications, web services)

Teaching methodologies (including evaluation)

Teaching methodologies

Theoretical-Practical Classes: Lecture method, guided according to problem-based learning, with projection and explanation of the objectives and corresponding content each class.

Practical classes: Computer laboratory practice. Resolution and codification of typified problems, selected in accordance with the weekly theoretical content.

Assessment

Assessment components rated from 0-20 values:

- Practical assignment(s) (TP)
- Written test (PE): Test / Exam

Minimum classification in each of the components 7 values.

Final grade (≥ 9.5): $0.3 \text{ PE} + 0.7 \text{ TP}$

According to paragraph 3 of article 6 of the rectoral order RT 59/2015, of 28 July, in professional higher technical courses, the inclusion of compliance with the duty of attendance in the assessment methods is mandatory, in the following terms : a) It is considered that a student fulfills the duty of attendance to a UC, when it does not exceed the limit number of absences corresponding to 25% of the foreseen contact hours

Main Bibliography

- * José Braga de Vasconcelos (2015). Python - Algoritmia e Programação Web. FCA.
- * Ernesto Costa (2015). Programação em Python - Fundamentos e Resolução de Problemas. FCA
- * Luiz Eduardo Borges (2014). Python para Desenvolvedores. Novatec
- * -(2021). Python 3.9.5 documentation. <https://www.python.org/>