
Ano Letivo 2023-24

Unidade Curricular COMPLEMENTOS DE PROGRAMAÇÃO

Cursos SISTEMAS E TECNOLOGIAS DE INFORMAÇÃO

Unidade Orgânica Instituto Superior de Engenharia

Código da Unidade Curricular 18121006

Área Científica CIÊNCIAS INFORMÁTICAS, FORMAÇÃO TÉCNICA

Sigla FT

Código CNAEF (3 dígitos) 481

Contributo para os Objetivos de
Desenvolvimento Sustentável - 4, 8, 10
ODS (Indicar até 3 objetivos)

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; TP2; PL1; PL2	28TP; 84PL

* 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	S2	14TP; 42PL	140	5

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

Precedências

Sem precedências

Conhecimentos Prévios recomendados

Conceitos de programação procedimental. Frequência da UC de 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 das 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. Interfaces: Separação entre interface e implementação de interfaces.
7. Relações entre Objetos: Extensão, generalização e especialização; Associação, agregação e composição; Coesão e acoplamento.
8. Aplicações (e.g., GUI e ligações a bases de dados)

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 semana.

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 prático ou Exercícios (TP)
- Prova escrita (PE): Teste / Exame

Classificação mínima em cada uma das componentes 7 valores.

Nota final: 0.5 PE + 0.5 TP

Nota: Regulamento de Avaliação da UAlg, ponto 3 do art.º 6, a assiduidade é obrigatória, não podendo o aluno exceder o número limite de faltas, correspondente a 25% das horas de contacto totais

Bibliografia principal

- Luiz Eduardo Borges (2014). Python para Desenvolvedores. Novatec
- Goldwasser, M. H., & Letscher, D. (2008). *Object-oriented programming in Python*. Pearson Prentice Hall.
- Lukaszewski A. (2010). MySQL for Python. Packt Publishing.
- Lutz, M. (2013). *Learning python: Powerful object-oriented programming*. " O'Reilly Media, Inc."
- Reges, S., Stepp, M., & Obourn, A. (2018). *Building Python Programs*. Pearson.
- Romano, F. (2015). Learning Python. Packt Publishing.
- Summerfield, M. (2008), Programming in Python 3: A Complete Introduction to the Python Language. Addison-Wesley Professional.

Academic Year 2023-24

Course unit PROGRAMMING II

Courses Information Systems and 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, 10

Language of instruction Portuguese

Teaching/Learning modality

Classroom-based learning

Coordinating teacher

Pedro Jorge Sequeira Cardoso

Teaching staff	Type	Classes	Hours (*)
Pedro Jorge Sequeira Cardoso	PL; TP	TP1; TP2; PL1; PL2	28TP; 84PL

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

Contact hours

T	TP	PL	TC	S	E	OT	O	Total
0	14	42	0	0	0	0	0	140

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

Concepts of procedural programming. Frequency of Programming CU.

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

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

Use an OOP language to implement programming 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, protocols of object's usage, messages and methods, notion of state and behavior).
4. Class-based languages; instantiation of objects.
5. Object Hierarchies: Inheritance mechanisms; Reuse of code; Reflection; Polymorphism.
6. Interfaces.
7. Relations between Objects: Extension, generalization and specialization; Association, aggregation and composition; Cohesion and coupling.
8. Applications (GUI, database connections)

Teaching methodologies (including evaluation)

Teaching methodologies

Theoretical-Practical classes: Expositive method, oriented according to a problem-based teaching, with projection and explanation of the objectives and contents corresponding to each week.

Practical classes: Laboratory practice in computer. Resolution and codification of typified problems, selected according to the theoretical weekly content.

Evaluation

Valuation components classified as 0-20 values:

- Practical assignments (TP)
- Written exam (PE): Test / Exam

Minimum classification in each of the components: 7 values.

Final grade: 0.5 PE + 0.5 TP

Note: Under the terms of UAlg Assessment Regulations, item 3 of article 6, attendance is mandatory and students cannot exceed the maximum number of absences, i.e. 25% of the total contact hours.

Main Bibliography

- Luiz Eduardo Borges (2014). Python para Desenvolvedores. Novatec
- Goldwasser, M. H., & Letscher, D. (2008). *Object-oriented programming in Python*. Pearson Prentice Hall.
- Lukaszewski A. (2010). MySQL for Python. Packt Publishing.
- Lutz, M. (2013). *Learning python: Powerful object-oriented programming*. "O'Reilly Media, Inc."
- Reges, S., Stepp, M., & Obourn, A. (2018). *Building Python Programs*. Pearson.
- Romano, F. (2015). Learning Python. Packt Publishing.
- Summerfield, M. (2008), Programming in Python 3: A Complete Introduction to the Python Language. Addison-Wesley Professional.