instituto superior de engenharia de lisboa

Ficha de Unidade Curricular A3ES
Programação
Licenciatura em Engenharia Informática e de Computadores

2025-26

1. Designação da unidade curricular

[3222] Programação / Programming

2. Sigla da área científica em que se insere	IC
3. Duração	Unidade Curricular Semestral
4. Horas de trabalho	162h 00m
5. Horas de contacto	Total: 67h 30m das quais T: 48h 00m, h 00m | TP: 67h 30m, 7h 30m | P: 12h 00m, h 00m
6. % Horas de contacto a distância	Sem horas de contacto à distância
7. ECTS	6
8. Docente responsável e respetiva carga letiva na Unidade Curricular	[889] Pedro Alexandre de Seia e Cunha Ribeiro Pereira | Horas Previstas: 67.5 horas
9. Outros docentes e respetivas cargas letivas na unidade curricular	[1245] José Luís Falcão Cascalheira | Horas Previstas: 67.5 horas [1550] Nuno Miguel da Costa de Sousa Leite | Horas Previstas: 135 horas [1603] Ana Sofia Querido Rito Rebelo | Horas Previstas: 135 horas [2207] Rodolfo Miguel Carvalho Morgado | Horas Previstas: 67.5 horas

10. Objetivos de aprendizagem e a sua compatibilidade com o método de ensino (conhecimentos, aptidões e competências a desenvolver pelos estudantes).

Os estudantes que terminam com sucesso esta unidade curricular serão capazes de:
1. Demonstrar o conhecimento sobre as construções elementares de uma linguagem de programação multi-paradigma e com tipificação estática
2. Construir pequenos programas que resolvam problemas simples descritos em linguagem natural.
3. Escrever, testar, depurar e corrigir pequenos programas.
4. Produzir documentação técnica onde se justificam as decisões tomadas nos programas construídos.
5. Utilizar eficazmente ferramentas para desenvolver programas e para elaborar relatórios.

10. Intended Learning objectives and their compatibility with the teaching method (knowledge, skills and competences by the students).

Students who successfully complete this course unit be able to:
1. Demonstrate know-how on the basic constructs of a statically typed and multi-paradigm programming language.
2. Build small programs aimed at solving simple problems described in natural language.
3. Write, test, debug and fix small programs.
4. Write reports justifying decisions made while building computer programs.
5. Make effective use of the tools to produce small computer programs and to write technical reports.

11. Conteúdos programáticos

I. Valores, tipos e variáveis; Mutabilidade e imutabilidade; Expressões; Entrada/Saída de dados; Decisão binária e decisão múltipla; Funções; Ativação de funções e passagem de parâmetros; Funções como parâmetros; Expressões lambda.
II. Construções de controlo de execução no estilo imperativo e no estilo funcional; Recursão e co-recursão.
III. Coleções de dados; Operações sobre coleções: filter, map e fold; Algoritmos elementares sobre coleções.
IV. Tipos primitivos e definição de tipos; Composição de dados homogénea (coleções); Composição de dados heterogênea (objetos).

11. Syllabus

I. Values, Types and variables; Mutability and immutability; Expressions, Input/Output; Binary and multiple decision instructions. Functions; Function activation and parameters passing; Functions as parameters; Lambda expressions.
II. Execution control constructs: in both imperative and functional programming styles; Recursion and corecursion.
III. Data collections; Basic operations on collections: filter, map, fold; Elementary algorithms on data collections.
IV. Primitive types and type definition; Homogeneous data composition (collections); heterogeneous data composition (objects).

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

Esta unidade curricular representa para a maioria dos alunos um primeiro contacto com a programação, que se pretende motivador sem descurar o formalismo e o rigor, sendo a base da formação de software do curso. São introduzidos conceitos e vocabulário fundamental da programação imperativa e funcional.
Os objetivos 1, 2 e 3 são alcançados através dos itens I até IV do conteúdo programático.
O desenvolvimento de programas durante as sessões práticas permite atingir os objetivos 4 e 5.

12. Evidence of the syllabus coherence with the curricular unit´s intended learning outcomes

This curricular unit represents for most students a first contact with programming, which is intended to be motivating without neglecting formalism and rigor, being the basis of the course's software training.
Fundamental concepts and vocabulary of imperative and functional programming are introduced.
Objectives 1, 2 and 3 are achieved through items I to IV of the syllabus.
The development of programs during practical sessions allows achieving objectives 4 and 5.

13. Metodologias de ensino e de aprendizagem específicas da unidade curricular articuladas com o modelo pedagógico

Previstas 30 aulas durante o semestre (15 de 3 horas e 15 de 1,5 horas). Quatro das aulas de 3 horas são de prática laboratorial e as restantes teóricas e teórico-práticas. As aulas destinam-se à apresentação dos temas e de exemplos práticos de aplicação. Os tópicos principais são ainda explorados com a realização de trabalhos práticos para desenvolver pequenos programas.

13. Teaching and learning methodologies specific to the curricular unit articulated with the pedagogical model

30 lectures are planned during the semester (15 of 3 hours and 15 of 1.5 hours). Four of the 3-hour lectures are laboratory practice and the rest are theoretical and theoretical-practical. The classes are intended to present the themes and practical examples of application. The main topics are further explored by carrying out practical work to develop small programs.

14. Avaliação

Os resultados da aprendizagem (1) e (2) são avaliados individualmente através do exame final escrito e das fichas realizadas durante o semestre. Com os trabalhos práticos realizados em grupo são avaliados os resultados da aprendizagem (3) e (5). Os resultados da aprendizagem (4) e (5) são avaliados na discussão final dos trabalhos de grupo.
A nota final corresponde a 40% do exame final, 40% dos trabalhos (nota da discussão final) e 20% das fichas realizadas durante o período letivo (média das 3 melhores das 4 realizadas), sendo as fichas e os trabalhos pedagogicamente fundamentais.

14. Assessment

Learning outcomes (1) and (2) are assessed individually through the final written exam and short written tests carried out during the semester. With practical work carried out in groups, learning outcomes (3) and (5) are assessed. Learning outcomes (4) and (5) are evaluated in the final discussion of group work.
The final grade corresponds to 40% of the final exam, 40% of practical works (final discussion grade) and 20% of the evaluation sheets maded during the academic period (average of the 3 best of the 4 completed), with the evaluation sheets and practical works are pedagogically fundamental.

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

Nas sessões teóricas e teórico-práticas são apresentados os mecanismos básicos da linguagem de programação e são construídos, testados e corrigidos pequenos programas com a participação dos alunos. Para realizar os diversos trabalhos práticos são usadas ferramentas de complexidade gradualmente crescente (linha de comando, editor simples, ambiente integrado, depurador de erros) e são escritos relatórios que fundamentam as decisões tomadas.

15. Evidence of the teaching methodologies coherence with the curricular unit´s intended learning outcomes

In theoretical and theoretical-practical sessions, the basic mechanisms of the programming language are presented and small programs are built, tested and corrected with the participation of students. To carry out the various practical tasks, tools of increasing complexity are used (command line, simple editor, integrated environment, debugger) and reports are created to support the decisions made.

16. Bibliografia de consulta/existência obrigatória

I. Galata; J. Howard; D. Lucas and E Shapiro, Kotlin Apprentice: Beginning Programming with Kotlin, thirth edition, Razeware LLC, 2021. ISBN-13: 9781950325375; ISBN-10: 1950325377

B. Eckel; S. Isakova, Atomic Kotlin, Mindview LLC, 2021. ISBN-13: 978-0981872551; ISBN-10: 0981872557

P. Pereira, Conceitos de Programação em Kotlin, Texto de apoio produzido especialmente para esta unidade curricular. (https://2324moodle.isel.pt/mod/resource/view.php?id=139893)

17. Observações

Unidade Curricular Obrigatória
Unidade Curricular comum ao(s) curso(s) de LMATE

Data de aprovação em CTC: 2025-07-30 Data de aprovação em CTC: Data de aprovação em CTC: 2024-07-17

Data de aprovação em CP: 2025-07-29 Data de aprovação em CP: Data de aprovação em CP: 2024-06-26