

---

**Ano Letivo** 2018-19

---

**Unidade Curricular** LÓGICA E COMPUTAÇÃO

---

**Cursos** ENGENHARIA INFORMÁTICA (1.º ciclo)

---

**Unidade Orgânica** Faculdade de Ciências e Tecnologia

---

**Código da Unidade Curricular** 14781053

---

**Área Científica** CIÊNCIA DE COMPUTADORES

---

**Sigla**

---

**Línguas de Aprendizagem** Português-PT

---

**Modalidade de ensino** Presencial

---

**Docente Responsável** Daniel da Silva Graça

DOCENTE	TIPO DE AULA	TURMAS	TOTAL HORAS DE CONTACTO (*)
Daniel da Silva Graça	T; TP	T1; TP1; TP2	30T; 60TP

\* 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	S2	30T; 30TP	168	6

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

### Precedências

Sem precedências

### Conhecimentos Prévios recomendados

Recomenda-se que o aluno tenha conhecimentos básicos de Matemática (incluindo noções elementares de teoria de conjuntos, de grafos, e de funções), assim como conhecimentos básicos de programação.

### Objetivos de aprendizagem (conhecimentos, aptidões e competências)

Com esta disciplina pretende-se dar aos alunos fundamentos sólidos de Lógica e de Teoria da Computação. Em particular pretende-se desenvolver o raciocínio lógico e a capacidade de abstração dos alunos, para que mais facilmente possam compreender conceitos avançados utilizados em programação/informática, e para que possam desenvolver estratégias e metodologias que lhes permitam tornarem-se mais eficazes na resolução de problemas de índole lógica ou algorítmica. Pretende-se ainda que os alunos tenham noções sobre os limites teóricos da computação, e que compreendam que existem problemas que não podem ser resolvidos através de computadores, quer se coloque limites nos recursos a utilizar (tempo e/ou memória) ou não.

### Conteúdos programáticos

- 1) Lógica Proposicional: sintaxe, semântica, sistemas dedutivos (dedução natural), resultados de correção e completude.
- 2) Lógicas de primeira ordem (cálculo de predicados): sintaxe, semântica, sistemas dedutivos (dedução natural), resultados de correção e completude.
- 3) Introdução à Teoria da Computabilidade: máquinas de Turing, tese de Church-Turing, problemas não-computáveis (problema da paragem).
- 4) Complexidade Computacional: definições básicas, as classes P e NP, o problema "P = NP?", problemas NP-completos.

### **Metodologias de ensino (avaliação incluída)**

A unidade curricular será ministrada com recurso a aulas presenciais. Haverá aulas teóricas para apresentar os conceitos teóricos, e aulas teórico-práticas onde será exemplificado a aplicação dessa teoria, através de exemplos e resolução de problemas. Será facultada uma sebenta de apoio especialmente concebida para a disciplina, contendo toda a teoria relevante, bem como exemplos resolvidos. Também será fornecida uma lista de problemas, alguns dos quais serão resolvidos nas aulas, e outros serão deixados para que os alunos os resolvam em casa, de acordo com o previsto na distribuição das horas de trabalho independentes da unidade curricular.

A avaliação será efetuada por meio de exame final, sendo realizados dois testes escritos durante o semestre. Se a média das notas dos testes for igual ou superior a 9,5 valores, o aluno será dispensado de exame final. Poderá ser exigida uma prova complementar aos alunos que obtenham uma nota igual ou superior a 16 valores nos testes/exame.

---

### **Bibliografia principal**

[1] Apontamentos fornecidos pelo docente.

[2] Michael Huth and Mark Ryan, Logic in Computer Science: Modelling and Reasoning About Systems, 2nd edition, Cambridge University Press, 2004.

[3] A. G. Hamilton, Logic for Mathematicians, 2nd edition, Cambridge University Press, 1988.

[4] M. Sipser. Introduction to the Theory of Computation, 3rd edition, Cengage Learning, 2012.

[5] M. R. Garey and D. S. Johnson. Computers and Intractability: a Guide to the Theory of NP-Completeness. W. H. Freeman & Co., 1979.

[6] J. E. Hopcroft, R. Motwani, and J. D. Ullman. Introduction to Automata Theory, Languages, and Computation, 3rd edition, Addison-Wesley, 2006.

---

**Academic Year** 2018-19

---

**Course unit** LOGIC AND COMPUTATION

---

**Courses** INFORMATICS (COMPUTER SCIENCE) (1st Cycle)

---

**Faculty / School** Faculdade de Ciências e Tecnologia

---

**Main Scientific Area** CIÊNCIA DE COMPUTADORES

---

**Acronym**

---

**Language of instruction** Portuguese-PT

---

**Teaching/Learning modality** Presential

---

**Coordinating teacher** Daniel da Silva Graça

---

Teaching staff	Type	Classes	Hours (*)
Daniel da Silva Graça	T; TP	T1; TP1; TP2	30T; 60TP

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

### Contact hours

T	TP	PL	TC	S	E	OT	O	Total
30	30	0	0	0	0	0	0	168

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

We recommend that students have a basic knowledge of mathematical concepts such as set theory, graphs, or functions, and an elementary knowledge of how to program.

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

This course aims to give students a solid foundation in Logic and in the Theory of Computation. In particular, it is intended to develop the students' logical reasoning and abstraction capabilities so that they can more easily grasp advanced notions used in programming/computer science, and it is also intended to develop strategies and methodologies which the students can use in order to increase their effectiveness when solving logical or algorithmic problems. Another objective is that the students understand the theoretical limits of computation and that they become aware that there are problems which cannot be solved with the use of computers either if limits are put on the resources used (time, memory) or not.

### Syllabus

- 1) Propositional Logic: syntax, semantics, deductive systems (natural deduction), soundness and completeness results.
- 2) First-order logics (predicate calculus): syntax, semantics, deductive systems (natural deduction), soundness and completeness results.
- 3) Introduction to the Theory of Computation: Turing machines, Church-Turing thesis, non-computable problems (the Halting Problem).
- 4) Computational Complexity: basic definitions, classes P and NP, the problem "P = NP?", NP-complete problems.

### Teaching methodologies (including evaluation)

The course will be taught using lectures. There will be (theoretical) lectures to present the theoretical concepts and practical classes where the theory will be exemplified through examples and problem solving. Lecture notes (in Portuguese) specially designed to support the course and containing all the relevant theory and worked examples will be provided to the students. A list of problems will also be provided. Some of the problems will be solved in the classroom, and others will be left for the students to solve out of classroom. The evaluation is done by a final exam. During the semester there will be two tests. If the average grade of the two tests is equal or superior to 9,5 values, the student will be dispensed of the final exam. A complementary test may be performed for students having a grade equal or superior to 16 values in the tests or exam.

### Main Bibliography

- [1] Lecture notes supplied by the teacher (in Portuguese).
- [2] Michael Huth and Mark Ryan, Logic in Computer Science: Modelling and Reasoning About Systems, 2nd edition, Cambridge University Press, 2004.
- [3] A. G. Hamilton, Logic for Mathematicians, 2nd edition, Cambridge University Press, 1988.
- [4] M. Sipser. Introduction to the Theory of Computation, 3rd edition, Cengage Learning, 2012.
- [5] M. R. Garey and D. S. Johnson. Computers and Intractability: a Guide to the Theory of NP-Completeness. W. H. Freeman & Co., 1979.
- [6] J. E. Hopcroft, R. Motwani, and J. D. Ullman. Introduction to Automata Theory, Languages, and Computation, 3rd edition, Addison-Wesley, 2006.