
Ano Letivo 2021-22

Unidade Curricular COMPILADORES

Cursos ENGENHARIA INFORMÁTICA (1.º ciclo)

Unidade Orgânica Faculdade de Ciências e Tecnologia

Código da Unidade Curricular 14781061

Área Científica CIÊNCIA DE COMPUTADORES

Sigla

Código CNAEF (3 dígitos) 481

Contributo para os Objetivos de Desenvolvimento Sustentável - ODS (Indicar até 3 objetivos) 04;08;10

Línguas de Aprendizagem Português-PT

Modalidade de ensino

Preferencialmente presencial.

Em caso de confinamento devido a COVID-19, as aulas serão leccionadas remotamente usando plataformas como Zoom e Discord.

Docente Responsável

Fernando Miguel Pais da Graça Lobo

DOCENTE	TIPO DE AULA	TURMAS	TOTAL HORAS DE CONTACTO (*)
Fernando Miguel Pais da Graça Lobo	PL; T	T1; PL1; PL2	28T; 56PL

* Para turmas leccionadas conjuntamente, apenas é contabilizada a carga horária de uma delas.

ANO	PERÍODO DE FUNCIONAMENTO*	HORAS DE CONTACTO	HORAS TOTAIS DE TRABALHO	ECTS
3º	S2	28T; 28PL	156	6

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

Precedências

Sem precedências

Conhecimentos Prévios recomendados

Programação, Programação Orientada a Objectos, Algoritmos e Estruturas de Dados, Matemática Discreta, Arquitectura de Computadores.

Objetivos de aprendizagem (conhecimentos, aptidões e competências)

- compreender os objetivos, arquitetura e abrangência de um compilador
- perceber a integração das diversas fases de compilação de linguagens imperativas
- ser capazes de especificar a sintaxe e a semântica de uma linguagem de programação
- saber utilizar os principais algoritmos e estruturas de dados usados na implementação de compiladores
- desenvolver componentes de um compilador para uma linguagem imperativa simples

Conteúdos programáticos

1. Conceitos introdutórios, panorâmica geral de um compilador
 2. Linguagens formais e autómatos finitos
 3. Análise lexical
 4. Análise sintática (parsing)
 5. Análise semântica
 6. Ambientes de execução
 7. Geração de código
-

Metodologias de ensino (avaliação incluída)

Nas aulas teóricas o professor faz a exposição da matéria usando o quadro e o computador para apresentar os conteúdos. As aulas práticas servem para consolidar os conceitos apresentados nas aulas teóricas, através da resolução de problemas e discussão de questões relacionadas com os trabalhos práticos.

A avaliação é feita por frequência conforme estipulado pelo Artº 9, ponto 1-b) do Regulamento de Avaliação da UAAlg, e consite em:

- trabalhos práticos (30%)
 - exame final (70%)
 - É necessário uma nota não inferior a 7,0 valores no trabalho prático para serem admitidos a exame final.
 - Os trabalhos práticos são feitos individualmente ou em grupo de 2 pessoas. O trabalho é composto por várias partes, cada qual com o seu prazo de entrega. A nota prática do grupo é convertida numa nota prática individual no momento da discussão dos trabalhos (que ocorrerá na última semana de aulas). A nota prática é, portanto, individual, e depende do desempenho de cada elemento do grupo na discussão.
-

Bibliografia principal

Livro de texto principal

- Compiler Principles, Techniques, and Tools, 2nd Edition, Addison Wesley, 2007.
Alfred V. Aho, Monica Lam, Ravi Sethi, Jeffery D. Ullman.

Referência complementar

Programming Language Pragmatics, 4th Edition, 2015.
- Michael L. Scott.

Referência complementar em língua portuguesa

Compiladores - da teoria à prática, Editora FCA, 2014
- Pedro Reis Santos, Thibault Langlois.

Academic Year 2021-22

Course unit COMPILERS

Courses INFORMATICS (COMPUTER SCIENCE) (1st Cycle)

Faculty / School FACULTY OF SCIENCES AND TECHNOLOGY

Main Scientific Area

Acronym

CNAEF code (3 digits) 481

Contribution to Sustainable Development Goals - SGD (Designate up to 3 objectives) 04;08;10

Language of instruction Portuguese-PT

Teaching/Learning modality

Preferably face-to-face.

In case of confinement due to COVID-19, lectures will take place remotely via platforms such as Zoom and Discord.

Coordinating teacher

Fernando Miguel Pais da Graça Lobo

Teaching staff	Type	Classes	Hours (*)
Fernando Miguel Pais da Graça Lobo	PL; T	T1; PL1; PL2	28T; 56PL

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

Contact hours

T	TP	PL	TC	S	E	OT	O	Total
28	0	28	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

Computer programming, Object Oriented Programming, Algorithms and Data Structures, Discrete Mathematics, Computer Architecture.

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

- understand the goals and architecture of a compiler
- understand the various phases of compiler construction
- be capable of specifying the syntax and semantics of a programming language
- know the major algorithms and data structures used in the implementation of a compiler
- develop components of a compiler for a simple imperative language

Syllabus

1. Introductory concepts, overview of a compiler
2. Formal languages and finite automata
3. Lexical analysis
4. Syntax analysis (parsing)
5. Semantic analysis
6. Run-Time Environments
7. Code Generation

Teaching methodologies (including evaluation)

In the main T lectures, the course materials are explained to students along with illustrative examples.

In the P lectures students have hands-on experience on the course materials doing exercises and working on their projects.

The grading follows "avaliação por frequência" as stated in Art. 9, 1-b) of the 'Regulamento de Avaliação' of UAAlg, and consists of:

- programming project (30%)
- final exam (70%)
- Students need a grade greater or equal to 7,0 in the project to be admitted to the final exam.
- The programming project has to be done individually or in a group of a maximum of 2 people. The project is composed of several parts, each with its own deadline. Each part corresponds to a task required in the development of a compiler. The project (practical) grade of the group will be converted into a student grade upon the mandatory individual project discussion that will take place in the last week of lectures of the semester.

Main Bibliography

Main textbook

- Compiler Principles, Techniques, and Tools, 2nd Edition, Addison Wesley, 2007.
Alfred V. Aho, Monica Lam, Ravi Sethi, Jeffery D. Ullman.

Additional reference

Programming Language Pragmatics, 4th Edition, 2015.
- Michael L. Scott.

Book in portuguese for those not so familiar with english:

Compiladores - da teoria à prática, Editora FCA, 2014
- Pedro Reis Santos, Thibault Langlois.