

---

**Ano Letivo** 2021-22

---

**Unidade Curricular** ARQUITETURA DE COMPUTADORES

---

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

---

**Unidade Orgânica** Faculdade de Ciências e Tecnologia

---

**Código da Unidade Curricular** 14781042

---

**Área Científica** CIÊNCIA DE COMPUTADORES

---

**Sigla**

---

**Código CNAEF (3 dígitos)** 4.8.1

---

**Contributo para os Objetivos de Desenvolvimento Sustentável - ODS (Indicar até 3 objetivos)** 4.8.10

---

**Línguas de Aprendizagem** Português

**Modalidade de ensino**

Presencial

**Docente Responsável**

Helder Aniceto Amadeu de Sousa Daniel

DOCENTE	TIPO DE AULA	TURMAS	TOTAL HORAS DE CONTACTO (*)
Helder Aniceto Amadeu de Sousa Daniel	T; TP	T1; TP1	28T; 28TP
João Miguel Gago Pontes de Brito Lima	TP	TP2; TP3	56TP

\* 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	28T; 28TP	156	6

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

**Precedências**

Sem precedências

**Conhecimentos Prévios recomendados**

Sistemas Digitais

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

Compreender a estrutura e operação dos processadores atuais, bem como de um computador:

*CPU, multicore, SMP, GPU, TPU.*

Compreender como as construções das linguagens de alto nível são implementadas em linguagem *assembly* e código de máquina.

Adquirir conceitos que os permitam programar em linguagem *assembly*.

### **Conteúdos programáticos**

Introdução à arquitetura dos processadores: CISC vs RISK and SISD vs SIMD vs MIMD

*Single cycle processors, multi-cycle processors, pipelining, data path*

Arquiteturas Von Neumann e Harvard

Paralelismo nas instruções:

Pipeline

Super escalar

*Hyper threading*

Thread parallelism:

Symmetric multiprocessing

Multicore

Processamento vetorial e extensões SIMD

GPUs

Unidades tensoriais (TPUs)

---

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

Todos os conceitos básicos são introduzidos nas aulas teóricas, sendo aplicados nas aulas laboratoriais.

*A avaliação, em época normal e recurso é obtida pela média ponderada entre o exame e os trabalhos práticos:*

*classificação final = 70% Exame + 30 % Trabalho prático*

*Em qualquer outra época a avaliação é apenas constituída por exame.*

Obtém-se aprovação se classificação final for maior ou igual a 9,5 valores.

Os exames consistem de prova escrita, que inclui questões teóricas e questões práticas semelhantes aos exercícios das aulas teórico-práticas e práticas.

Todos os alunos regularmente inscritos são admitidos ao exame de época normal e recurso.

#### **Bibliografia principal**

J.L. Hennessy and D.A. Patterson (2019). "Computer architecture, a quantitative approach, 6th ed.", Morgan Kaufmann publishers, Elsevier Inc.

D.A. Patterson and J.L. Hennessy (2018) "Computer Organization and Design, the hardware/software interface, RISK-V ed.", Morgan Kaufmann publishers, Elsevier Inc.

---

**Academic Year** 2021-22

---

**Course unit** COMPUTER ARCHITECTURE

---

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

---

**Faculty / School** FACULTY OF SCIENCES AND TECHNOLOGY

---

**Main Scientific Area**

---

**Acronym**

---

**CNAEF code (3 digits)** 4.8.1

---

**Contribution to Sustainable Development Goals - SGD (Designate up to 3 objectives)** 4, 8, 10

---

**Language of instruction** Portuguese

---

**Teaching/Learning modality** Face to face learning

**Coordinating teacher** Helder Aniceto Amadeu de Sousa Daniel

Teaching staff	Type	Classes	Hours (*)
Helder Aniceto Amadeu de Sousa Daniel	T; TP	T1; TP1	28T; 28TP
João Miguel Gago Pontes de Brito Lima	TP	TP2; TP3	56TP

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

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

Digital Systems

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

*Understand the structure and operation of today's processors, as well as computers:*

*CPU, multicore, SMP, GPU, TPU.*

Understand how high level language constructs are translated into assembly and machine code.

*Acquire fundamental concepts to program in assembly language.*

## Syllabus

*Intro to processor architecture: CISC vs RISC and SISD vs SIMD vs MIMD*

*Single cycle processors, multi-cycle processors, pipelining, data path*

*Von Neumann Architecture and Harvard Architecture*

*Instruction parallelism:*

*Pipeline*

*Super scalar*

*Hyper threading*

*Thread parallelism:*

*Symmetric multiprocessing*

*Multi core*

*Vectorial processing and SIMD stream processing*

*GPUs*

*Tensor units ( TPUs )*

---

## Teaching methodologies (including evaluation)

*All required knowledge is presented in the lectures, being applied, and developed in lab classes .*

*Assessment, in normal and recourse seasons have 2 components:*

*Final grade = 70 % written exam + 30 % practical assignments*

*In any other season, assessment is just the grade of a written exam.*

*Approval if final grade  $\geq 9.5$  (rounded to 10)*

*The written exams include theoretical questions and programming exercises like the ones addressed in lab classes.*

### **Main Bibliography**

J.L. Hennessy and D.A. Patterson (2019). "Computer architecture, a quantitative approach, 6th ed.", Morgan Kaufmann publishers, Elsevier Inc.

D.A. Patterson and J.L. Hennessy (2018) "Computer Organization and Design, the hardware/software interface, RISK-V ed.", Morgan Kaufmann publishers, Elsevier Inc.