

## Anexo II – Modelo de Ficha de Unidade Curricular

### Ficha de Unidade Curricular – (Versão A3ES 2018-2023)

#### 1. Caracterização da Unidade Curricular.

- 1.1. **Designação da unidade curricular** (1.000 carateres).  
Arquitetura de Computadores / Computer Architecture
- 1.2. **Sigla da área científica em que se insere** (100 carateres).  
IC
- 1.3. **Duração<sup>1</sup>** (100 carateres).  
Semestral
- 1.4. **Horas de trabalho<sup>2</sup>** (100 carateres).  
162 h
- 1.5. **Horas de contacto<sup>3</sup>** (100 carateres).  
Total - 67,5 h  
TP - 46,5 h  
PL - 21,0 h
- 1.6. **ECTS** (100 carateres).  
6
- 1.7. **Observações<sup>4</sup>** (1.000 carateres).  
Comum aos cursos de LEIC, LEETC e LEIRT
- 1.7. **Remarks** (1.000 carateres).  
Common to LEIC, LEETC, and LEIRT courses

#### 2. Docente responsável e respetiva carga letiva na Unidade Curricular (preencher o nome completo) (1.000 carateres).

Tiago Miguel Braga da Silva Dias

#### 3. Outros docentes e respetivas cargas letivas na unidade curricular (1.000 carateres).

#### 4. Objetivos de aprendizagem (conhecimentos, aptidões e competências a desenvolver pelos estudantes).

(1.000 carateres).

Os estudantes que terminarem com sucesso esta unidade curricular serão capazes de:

1. Demonstrar conhecimento em arquiteturas de computadores;
2. Demonstrar conhecimento na estrutura interna de um processador;
3. Escrever programas de pequena complexidade usando a linguagem *assembly*;
4. Utilizar ferramentas para produção de programas escritos em linguagem *assembly* e para a depuração dos seus erros;
5. Demonstrar conhecimento no mapeamento de memória física no espaço de endereçamento do processador;

6. Demonstrar conhecimento no mapeamento de dispositivos periféricos no espaço de endereçamento do processador e na sua utilização, tais como portos paralelos de entrada/saída ou temporizadores/contadores;

7. Demonstrar conhecimento na utilização do mecanismo de interrupção dos processadores.

**4. Intended learning outcomes (knowledge, skills and competences to be developed by the students). (1.000 characters).**

Students who successfully complete this course will be able to:

1. Demonstrate knowledge of computer architectures;
2. Demonstrate knowledge of the internal structure of a processor;
3. Write low complexity programs using the assembly language;
4. Use software development tools for production and debugging of assembly programs;
5. Demonstrate knowledge of physical memory mapping onto the address space of a processor;
6. Demonstrate knowledge about peripheral usage and its mapping onto the address space of a processor, namely input / output parallel ports and timers / counters;
7. Demonstrate knowledge of processor interrupt mechanisms and of its usage.

**5. Conteúdos programáticos (1.000 characters).**

I. Introdução: áreas de aplicação de computadores; conceito de programa e modelo de execução; componentes de um computador; evolução tecnológica.

II. Arquitetura de um processador: unidade de processamento de dados, unidade lógico-aritmética e tipos de registos; unidade de controlo; codificação de instruções; modelos Harvard e Von-Neumann.

III. Programação *assembly*: arquitetura do conjunto de instruções, tipos de instruções e operandos, modos de endereçamento; rotinas e convenções de utilização; pilha e sua utilização; produção e teste de programas.

IV. Subsistema de memória: tecnologias de memórias; espaço de endereçamento e barramentos de dados, endereço e controlo; organização de módulos de memória; hierarquia de memória.

V. Subsistema de entrada/saída: tipos de dispositivos e suas características; portos paralelos; temporizadores/contadores.

VI. Mecanismo de interrupção: conceitos de exceção e interrupção; vetorização e níveis de prioridade; processamento de pedidos de interrupção.

**5. Syllabus (1.000 characters).**

I. Introduction: application areas of computers; concept of computer program and its execution model; components of a computer; technological evolution of processors and memories.

II. Processor architecture: datapath, arithmetic and logic unit and types of registers; control unit; instruction coding; the Harvard and Von Neumann models.

III. Assembly programming: instruction set architecture, types of instructions, operands and addressing modes; concept of routine and calling conventions; concept of stack and its implementation; tools for the development and test of programs.

IV. Memory subsystem: memory technologies, processor address space; data, address and control buses; organization of memory modules; memory hierarchy.

V. Input/output subsystem: types and characteristics of input/output devices; parallel ports; counters/timers;

VI. Interrupt mechanism: concepts of exception and interrupt; interrupt vectorization and priority levels; processing of interrupt requests.

**6. Demonstração da coerência dos conteúdos programáticos com os objetivos de aprendizagem da unidade curricular (1.000 caracteres).**

Esta UC tem como objetivos principais habilitar os alunos para compreenderem e utilizarem sistemas computacionais baseados em microprocessadores.

Começa-se por apresentar o conceito de sistema computacional baseado em microprocessador e os seus principais componentes (I). Depois, introduz-se um processador didático de 16 bits, que suporta o estudo dos restantes tópicos. São analisados os principais módulos funcionais do processador e identificadas as preocupações no seu desenho (II). Tendo em conta as características deste processador, abordam-se os tópicos da programação em *assembly* e os processos de geração, carregamento em memória e execução de programas (III). No âmbito dos outros subsistemas, é abordada a interface e a decodificação de memória física mapeada no espaço de endereçamento do processador (IV) e são introduzidos os periféricos fundamentais, i.e. portas paralelas e temporizadores/contadores (V). Por último, é estudado o mecanismo de interrupção deste processador (VI).

**6. Evidence of the syllabus coherence with the curricular unit's intended learning outcomes (1.000 characters).**

The main goal of this CU is to enable students to understand and use microprocessor-based computer systems.

First, the concept of a computer system based on a microprocessor and the main components of such systems are presented (I). Then, a 16-bit didactic processor is introduced. This processor supports the study of the remaining course topics. Such study starts with the analysis of the main functional modules of the processor (II). Then, assembly programming is addressed. This includes also the program building process, as well as the loading of the programs into memory and its execution and debugging processes (III). The memory and I/O subsystems are studied afterwards. Such study covers not only the mapping of the physical memory and peripherals onto the processor address space but also the corresponding interfaces (IV). Moreover, two fundamental peripherals are also studied: parallel I/O ports and timers/counters (V). Finally, the processor interruption mechanism is analyzed (VI).

**7. Metodologias de ensino (avaliação incluída) (1.000 caracteres).**

Ensino teórico-prático (TP) com recurso à prática laboratorial (PL) para demonstração dos principais conceitos. Estão previstas 30 aulas durante o semestre: 15 de 3h e 15 de 1,5h.

As aulas TP visam a apresentação e explicação dos temas, sendo os alunos levados a laboratório após a exposição de alguns tópicos para experimentarem as valências adquiridas. Ademais, durante o semestre são realizados trabalhos práticos para consolidação da aprendizagem. Esses trabalhos são realizados em grupo.

Os resultados da aprendizagem (2) a (7) são avaliados através dos trabalhos práticos e dos relatórios realizados no semestre (P). Os resultados da aprendizagem (2), (3), (5), (6) e (7) são também avaliados através de teste escrito individual (T), realizado no final do semestre. Todos os resultados da aprendizagem são ainda avaliados numa discussão final individual (D), a realizar no final do semestre.

A nota final é definida por  $0,4*P+0,4*T+0,2*D$ . A classificação mínima exigida em P, T e D é 10 valores.

**7. Teaching methodologies (including assessment) (1.000 characters).**

Theoretical and practical (TP) teaching supported by laboratory classes (PL) for the demonstration of the main syllabus topics. During the semester there will be 30 classes: 15 classes of 3h and 15 of 1.5h.

The TP classes are devoted to the presentation and the explanation of the syllabus contents. The PL classes are used to conduct practical activities demonstrating the application of some syllabus topics, as well as to provide support for the course's assignments. The students perform these activities in groups.

The learning outcomes (2) to (7) are assessed through the course's assignments and the corresponding reports (P). The learning outcomes (2), (3), (5), (6) and (7) are assessed in an individual written test (T), taken at the end of the semester. All learning outcomes are also evaluated in a final viva voce examination (D).

The final grade is calculated according to the following formula:  $0,4*P+0,4*T+0,2*D$ . A minimum classification of 10 values is required in P, T and D.

**8. Demonstração da coerência das metodologias de ensino com os objetivos de aprendizagem da unidade curricular (3.000 caracteres).**

Nas aulas teórico-práticas são apresentados e discutidos os conceitos teóricos de forma interativa. Com o propósito de facilitar a compreensão dos conteúdos programáticos, também são analisados casos de estudo relevantes e resolvidos exercícios sobre problemas concretos. Os alunos são incentivados a participar, quer respondendo a questões colocadas pelo docente, quer colocando questões ou dando contributos para a resolução dos problemas propostos. Esta abordagem, com curtos períodos de exposição da teoria e acompanhada da resolução de exercícios, conduz a que os alunos assimilem os principais conceitos da UC. O teste escrito individual permite avaliar a aprendizagem destes conceitos.

As atividades práticas realizadas em laboratório visam ajudar os alunos a consolidar os conceitos teóricos apresentados nas aulas teórico-práticas e a apoiar a realização dos trabalhos práticos da UC. Os relatórios destes trabalhos, em particular do último trabalho, em que os alunos são confrontados com a especificação de um sistema para desenhar e implementar uma solução, permitem aferir as boas práticas no desenvolvimento de programas em linguagem *assembly* e avaliar a capacidade dos alunos para utilizarem microprocessadores no desenvolvimento de sistemas computacionais de baixa complexidade.

A realização da discussão oral final individual permite visa avaliar, com maior rigor, a qualidade dos resultados globais da aprendizagem de cada aluno.

**8. Evidence of the teaching methodologies coherence with the curricular unit's intended learning outcomes (3.000 characters).**

The theoretical concepts of the syllabus contents are presented in the theoretical-practical lectures by using an interactive methodology. Such methodology comprehends short periods of theory exposition, the discussion of relevant case studies and practical problem-solving activities. Altogether this approach allows the students to acquire the main concepts of the course. Such learning outcomes are assessed in an individual written examination.

The practical activities carried out in the laboratory help the students consolidating the theoretical concepts and contribute to the successful accomplishment of the course's assignments. The written reports of such assignments allow evaluating the student's skills to write good computer programs using the assembly language and their ability to use microprocessors in the development of low complexity computational systems.

The final individual viva voce examination allows the instructor to evaluate each student, in detail, regarding all the learning outcomes.

**9. Bibliografia de consulta/existência obrigatória (1.000 caracteres).**

Harris, S. L. & Harris, D. M. (2015). Digital Design and Computer Architecture: ARM Edition, Waltham, MA: Morgan Kaufmann Publishers, ISBN: 978-0128000564.

Patterson, D. A. & Hennessy, J. L. (2016). Computer Organization and Design: The Hardware/Software Interface - ARM Edition, Cambridge, MA: Morgan Kaufmann Publishers, ISBN-13: 978-0128017333.

Mano, M. & Kime, C. (2015). Logic and Computer Design Fundamentals, 5ª Edição, Upper Saddle River, N.J.: Pearson Education, ISBN-13: 978-0133760637.

---

<sup>1</sup> Anual, semestral, trimestral, ...

<sup>2</sup> Número total de horas de trabalho.

<sup>3</sup> Discriminadas por tipo de metodologia adotado (T - Ensino teórico; TP - Ensino teórico-prático; PL - Ensino prático e laboratorial; TC - Trabalho de campo; S - Seminário; E - Estágio; OT - Orientação tutorial; O - Outro).

<sup>4</sup> Assinalar sempre que a unidade curricular seja optativa.