

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

1. Caracterização da Unidade Curricular.

- 1.1. **Designação da unidade curricular** (1.000 carateres).
Computação Distribuída/Distributed Computing
- 1.2. **Sigla da área científica em que se insere** (100 carateres).
INF
- 1.3. **Duração**¹ (100 carateres).
Semestral
- 1.4. **Horas de trabalho**² (100 carateres).
162h
- 1.5. **Horas de contacto**³ (100 carateres).
67,5h: T(22,5h); TP(20h); PL(25h)
- 1.6. **ECTS** (100 carateres).
6
- 1.7. **Observações**⁴ (1.000 carateres).
Obrigatória
- 1.7. **Remarks** (1.000 carateres).
Mandatory

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

LUIS MANUEL DA COSTA ASSUNÇÃO, 67.5horas

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

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

(1.000 carateres).

Pretende-se que os alunos que terminem com sucesso a unidade curricular sejam capazes de:

1. Descrever e discutir as vantagens, os problemas e desafios que se colocam no desenvolvimento de aplicações usando o paradigma da computação distribuída;
2. Conhecer os padrões de arquitetura e modelos de interação e comunicação entre as partes das aplicações distribuídas, incluindo a composição de serviços por orquestração e coreografia;
3. Desenvolver aplicações usando o paradigma da computação distribuída no acesso a componentes e serviços alojados em infraestruturas distribuídas, sejam elas em infraestruturas locais ou nas infraestruturas de *Cloud* públicas mais conhecidas;
4. Analisar novos paradigmas emergentes na área da Computação Distribuída.

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

Students who successfully complete the course unit are able to:

1. Describe and discuss the advantages, problems and challenges that arise in the development of applications using the distributed computing paradigm;
2. Know the architecture patterns and interaction and communication models between the parts of the distributed applications, including the composition of services by orchestration and choreography;
3. Develop applications using the distributed computing paradigm in accessing components and services hosted in distributed infrastructures, whether they be in local infrastructures or in the most well-known public cloud infrastructures;
4. Analyze emerging new paradigms in the Distributed Computing area.

5. Conteúdos programáticos (1.000 carateres).

1. Características dos sistemas distribuídos, nomeadamente a escalabilidade/elasticidade, concorrência e tolerância a falhas. Falácias da computação distribuída; Teorema CAP. Tempo e coordenação em sistemas distribuídos. Relógios lógicos e ordenação de mensagens. Algoritmos de exclusão mútua e de consenso baseados em eleições. Comunicação por grupos;
2. Padrões, arquiteturas e modelos de interação na computação distribuída: Cliente/Servidor; Arquiteturas *n-tier*; filas de mensagens; *Publishing/Subscribe*; *Peer-to-Peer*; Arquiteturas HPC e *Job Scheduling*;
3. Granularidade, comunicação e interação das componentes distribuídas: Sockets; Objetos distribuídos; Serviços web (REST/SOAP); Comunicação assíncrona e *Callbacks*; Componentes *Stateless/Statefull*;
4. Conceito de orquestração e coreografia. Padrões de *workflow*;
5. Concretização prática sobre plataformas, tanto em infraestruturas locais como nas infraestruturas de *Cloud* públicas mais conhecidas.

5. Syllabus (1.000 characters).

1. Characteristics of distributed systems, namely scalability/elasticity, concurrency and fault tolerance. Distributed computing fallacies; Theorem CAP. Time and coordination in distributed systems. Logical clocks and message ordering. Algorithms of mutual exclusion and consensus based on elections. Group communication;
2. Patterns, architectures and interaction models in distributed computing: Client / Server; N-tier architectures; message queues; Publishing / Subscribe; Peer-to-Peer; HPC and Job Scheduling architectures;
3. Granularity, communication and interaction of the distributed components: Sockets; Distributed objects; Web services (REST/SOAP); Asynchronous communication and callbacks; Components Stateless/Statefull;
4. Concept of orchestration and choreography. Workflow patterns;
5. Practical implementation on platforms, both in local infrastructures as well as in the most popular public cloud infrastructures.

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

Tendo como grande objetivo alicerçar unidades curriculares posteriores que necessitam dos fundamentos essenciais na área da computação distribuída, os objetivos enunciados podem ser sumarizados na aquisição de competências fundamentais, nomeadamente as características, as diferenças e os desafios da computação distribuída face à computação centralizada. Assim, os conteúdos programáticos (1), (2) e (3) contribuem para os objetivos (1), (2) e (3), pois ao serem apresentados exemplos concretos usando tecnologias que implementam os diversos modelos de interação entre as partes das aplicações distribuídas, o aluno adquire competências que lhe permitem desenvolver soluções concretas avaliadas através da realização de trabalhos práticos. Para o objetivo (4) contribuem os conteúdos programáticos (4) e (5), onde o aluno é desafiado a analisar e discutir tecnologias e soluções emergentes.

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

Having as a great objective to create a base used in later curricular units that need the essential foundations in the area of distributed computing, the stated objectives can be summarized in the acquisition of fundamental competences, namely the characteristics, the differences and the challenges of the distributed computing face to the centralized computing. Thus, the programmatic contents (1), (2) and (3) contribute to objectives (1), (2) and (3), since concrete examples are presented using technologies that implement the various models of interaction between the parts of the distributed application, where students acquire skills that allow them to develop concrete solutions evaluated through the accomplishment of practical works. For the objective (4), the programmatic contents (4) and (5) contribute, where the student is challenged to analyze and discuss emerging technologies and solutions.

7. Metodologias de ensino (avaliação incluída) (1.000 carateres).

Aulas teórico-práticas (15 aulas de 3 horas e 15 de 1,5 horas) com 67,5 horas de contacto.

A metodologia consiste na apresentação e discussão dos temas e a concretização de exemplos que demonstrem os conceitos envolvidos. Por cada tema são propostos desafios para que os alunos encontrem soluções, consolidando assim a aprendizagem.

Os resultados de aprendizagem (1), (2) e (3) são avaliados através de um exame escrito de avaliação

individual. O resultado de aprendizagem (3) é avaliado em trabalhos laboratoriais e uma discussão final desses trabalhos. O resultado de aprendizagem (4) é avaliado através de um relatório e apresentação na aula sobre um tópico emergente na área da computação distribuída.

A avaliação é composta por:

1) Exame final individual com duração de 2 horas;

2) Aulas práticas com resolução de exercícios e trabalhos práticos, apresentados e discutidos nas últimas aulas do semestre.

Nota Final: 50% Exame com nota ≥ 10 ; 50% Prática (trabalhos de projeto obrigatórios com nota ≥ 10)

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

Theoretical-practical classes (15 lessons of 3 hours and 15 of 1.5 hours) with 67.5 hours of contact.

The methodology consists in the presentation and discussion of the topics and the concretization of examples that demonstrate the concepts involved. For each topic, challenges are proposed so that students find solutions, thus consolidating learning.

The learning outcomes (1), (2) and (3) are assessed through a written individual assessment examination. The learning outcome (3) is evaluated in laboratory work and a final discussion of these works. The learning outcome (4) is evaluated through a report and presentation in class on an emerging topic in the distributed computing area.

The evaluation is composed by:

1) Individual final exam with 2 hours;

2) Practical classes with resolution of exercises and practical work, presented and discussed in the last classes of the semester.

Final Grade: 50% Exam with grade ≥ 10 ; 50% Practice (mandatory practical work with note ≥ 10)

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

Usando uma metodologia de ensino teórico-prática, são apresentados conceitos, modelos e arquiteturas das aplicações baseadas no paradigma da computação distribuída. Os objetivos (1), (2) e (3), avaliados através de um exame escrito individual, são coerentemente abordados durante a apresentação e demonstração das características dos sistemas distribuídos face a sistemas centralizados. A aposta sistemática de demonstrar e concretizar exemplos de aplicação com as tecnologias existentes, contribui coerentemente para o objetivo (2) e (3) consolidado pela avaliação de trabalhos práticos laboratoriais. A metodologia de sistematicamente indicar fontes de informação e o desafio de os alunos realizarem um pequeno trabalho de pesquisa e síntese sobre tópicos emergentes na área da computação distribuída conduz coerentemente ao objetivo (4).

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

Using a theoretical and practical teaching methodology, concepts, models, and architectures of distributed computing applications are presented. The objectives (1), (2) and (3), assessed through an individual written exam, are coherently addressed during the presentation and demonstration of the characteristics of distributed systems faced to centralized systems. The systematic approach to demonstrate concrete application examples with existing technologies, contributes to the objective (2) consolidated by practical laboratory work assessment. The methodology of indicating sources of information (references) and the students challenge to research and write a synthesis on an emerging topic in distributed computing leads to the objective (4).

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

- George Coulouris, Jean Dollimore, Tim Kindberg, Gordon Blair Distributed Systems, Concepts and Design, 5th Edition, ISBN 978-0-273-76059-7, Pearson, 2012.
- Introduction to Middleware: Web Services, Object Components, and Cloud Computing, Letha Hughes Etzkorn, CRC Press, 2017
- *Cloud Computing: Theory and Practice (2nd Edition)*, Dan C. Marinescu, Ed. Morgan Kaufmann/Elsevier, 2018

¹ Anual, semestral, trimestral, ...

² Número total de horas de trabalho.

³ 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).

⁴ Assinalar sempre que a unidade curricular seja optativa.