

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

1. Caracterização da Unidade Curricular.

- 1.1. **Designação da unidade curricular** (1.000 carateres).
Infraestruturas de Sistemas Distribuídos / Distributed Systems Infrastructures
- 1.2. **Sigla da área científica em que se insere** (100 carateres).
IC
- 1.3. **Duração¹** (100 carateres).
Semestral
- 1.4. **Horas de trabalho²** (100 carateres).
162
- 1.5. **Horas de contacto³** (100 carateres).
67.5H (T:22,5H, TP:45H)
- 1.6. **ECTS** (100 carateres).
6
- 1.7. **Observações⁴** (1.000 carateres).
- 1.7. **Remarks** (1.000 carateres).

2. Docente responsável e respetiva carga letiva na Unidade Curricular (preencher o nome completo) (1.000 carateres). António Luís Freixo Guedes Osório, 67,5H

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).

São objetivo de aprendizagem (1) promover competências de avaliação de requisitos e decisão arquitetural no desenvolvimento de sistemas informáticos distribuídos para domínios de aplicação críticos, considerando a heterogeneidade, adaptabilidade à evolução tecnológica, dependências tecnológicas, reutilização e aspetos de normalização (interoperabilidade); Exploração de abordagens orientadas a serviços (SOA) ou micros serviços num quadro aberto de componentes (sistemas computacionais autónomos e heterogéneos) e diversidade de paradigmas computacionais e de comunicação (RPC, MOM, Tuple-spaces) (2) e (3). Competências no desenvolvimento de sistemas complexos tolerantes a falhas, a coordenação com base nos modelos X/Open ou consensos (Paxos) (4). Experimentação sobre tecnologias/especificações OSGi, Java/JINI, JEE, Cloud entre outras, no desenvolvimento de demonstradores que permitem confrontar o aluno com a complexidade do desenvolvimento de sistemas informáticos distribuídos (4).

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

The learning outcomes include (1) promote competencies to evaluate requirements and architectural decisions in the development of distributed informatics (computing) systems for critical applications, considering heterogeneity, adaptability to technological evolution, technological dependencies, reuse and standardization aspects (interoperability); Exploration of Service Oriented Approaches (SOA) or microservices in an open framework of components (autonomous and heterogeneous computational systems) and diversity of computational and communication paradigms (RPC, MOM, Tuple-spaces) (2) and (3). Competencies in the development of complex fault-tolerant systems, coordination based on X/Open or consensus (Paxos) models are key aspects (4). Experimentation considering technologies/specifications OSGi, Java/Jini, JEE, Cloud technologies, among others, in the development of demonstrators that confronts the students with the

complexity of developing distributed informatics (computing) systems (4).

5. Conteúdos programáticos (1.000 carateres).

1. Caracterização de requisitos, arquiteturas e estratégias tecnológicas no desenvolvimento de infraestruturas de sistemas distribuídos, considerando sistemas críticos, normas e especificações abertas; 2. Estudo de Infraestruturas de suporte a arquiteturas orientadas aos serviços (SOA) ou microserviços como estratégia de modularidade e reutilização; 3. Estudo de Infraestruturas de sistemas baseados em mensagens (MOM) e partilha de informação, nomeadamente usando mecanismo *tuple-space* entre serviços; 4. Estudo de Infraestruturas de suporte ao processamento transacional e modelo X/Open, coordenação baseada em consensus (Paxos) na coordenação de sistemas distribuídos; 5. Infraestruturas de suporte a sistemas informáticos confiáveis (*reliability*) e escaláveis; estudo de casos sobre tecnologias e infraestruturas de referência (e.g., OSGi/Web Services, Java/JINI, Java/JEE, Zookeeper/kafka, Cloud).

5. Syllabus (1.000 characters).

1. Characterization of requirements, architectures and technological strategies in the development of distributed systems infrastructures, considering critical systems, standards, and open specifications; 2. Study of Infrastructures supporting service-oriented architectures (SOA) as a strategy of modularity and reuse; 3. Study of Messaging-based Systems Infrastructures (MOM) and information sharing, namely using tuple-space mechanism between services; 4. Study of infrastructures supporting transactional processing, model X/Open, and coordination based on consensus (Paxos) in the coordination of distributed systems; 5. Infrastructures supporting reliable and scalable computer systems; case studies on reference technologies and infrastructures (e.g., OSGi/ Web Services, Java/JINI, Java/JEE, Zookeeper/Kafka, Cloud).

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

A complexidade do desenvolvimento de sistemas informáticos num crescendo de requisitos de integração de elementos heterogéneos, requer abordagens fundamentadas sobre opções arquiteturais. Aspetos de qualidade requerem mecanismos que garantam tolerância a falhas (*reliability*) e resposta eficiente a alterações de escala (*scalability*). Assente em conceitos de sistemas distribuídos, revisitados e discutidos no contexto de interações entre elementos distribuídos, a abordagem dota o aluno das ferramentas para o desenvolvimento fundamentado de sistemas informáticos. A coordenação aparece como elemento central no garante da confiabilidade da rede complexa de interações e gestão complexa de eventos entre elementos (computacionais) distribuídos.

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

The complexity of the development of computer systems in a growing integration requirement of heterogeneous elements requires approaches based on architectural options. Quality aspects require mechanisms that guarantee fault tolerance (*reliability*) and efficient response to scalability. Based on the concepts of distributed systems, revisited and discussed in the context of interactions among distributed elements, it provides the student with tools for the development of computer systems. Coordination appears as a central element in ensuring the reliability of the complex network of interactions and complex management of events among distributed (computational) elements.

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

O aluno desenvolve exercícios simples onde é requerida a modelação e implementação de componentes distribuídas envolvendo tecnologias heterogéneas, na consolidação dos conceitos (1) e (2). Com uma ênfase para tecnologias em código aberto (Zookeeper/Kafka, Java/JEE, JINI), pretende-se complementar as competências do aluno obtidas em unidade curricular precedente de sistemas distribuídos onde o aluno adquiriu conceitos elementares (3). O modelo transacional X/Open, coordenação distribuída e respetivas interfaces são validadas num quadro de aplicação envolvendo elementos de um sistema informático distribuído (4). Um trabalho em grupo integrando diferentes paradigmas permite aos alunos a concretização selecionando os quadros tecnológicos de forma fundamentada (5). Um teste permite consolidar a medida do desempenho na componente individual.

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

The student develops simple exercises where modeling and implementation of distributed components involving heterogeneous technologies to the consolidation of concepts (1) and (2). With an emphasis on open

source implementations (Zookeeper/Kafka, Java/JEE, JINI), it is intended to complement the student's competencies obtained in a previous curricular unit of distributed systems where the student acquired elementary concepts (3). Validation of X/Open transactional model, distributed coordination and respective interfaces in an application framework involving elements of a distributed informatics (computing) system (4). A group work integrating different paradigms allows the students to materialize acquired competencies by selecting the proper technological frameworks (5). A test allows consolidating a measure for the individual performance.

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

Um conjunto de exercícios de complexidade crescente permite aos alunos a familiarização com o quadro de desenvolvimento adotado (Java/OSGi, Maven, git). Com base em exemplos de referência o aluno é desafiado no desenho da arquitetura na resposta um problema de desenvolvimento de um sistema informático com elementos distribuídos. O desenvolvimento ou utilização de mecanismos de coordenação da concorrência (e.g., um *two-phase locking manager*) em trabalhos práticos consolidam competências no desenvolvimento de sistemas informáticos (com elementos distribuídos).

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

A group of exercises of increasing complexity allows students to familiarize themselves with the adopted development framework (Java / OSGi, Maven, git). Based on reference examples the student is challenged in the design of the architecture in response to a problem of development of informatics (computer) systems with distributed elements. The development or use of concurrency coordination mechanisms (e.g., a two-phase locking manager) in practical work consolidates skills in the development of computer systems (with distributed elements).

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

Distributed Systems: Principles and Paradigm, Autores: Andrew S Tanenbaum, Maarten Van Steen
Editora: Pearson; 2nd edition (26 July 2013), ISBN-10: 1292025522.

Reliable Distributed Systems Technologies, Web Services, and Applications, Autor: Kenneth Birman
Editora: Springer (2005), ISBN: ISBN-10: 0387215093.

Principles of Transaction Processing, Autor: Philip A. Bernstein, Editora: Morgan Kaufmann, (2009)
ISBN-10: 1558606238.

ZooKeeper: Distributed Process Coordination, Autores: Flavio Junqueira, Benjamin Reed
Editora: O'Reilly Media; (December 2013), ISBN-10: 1449361307.

Kafka - The Definitive Guide, Autores: Neha Narkhede, Gwen Shapira, O'Reilly (October 2017)
ISBN-10: 1491936169.

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.