
1. Designação da unidade curricular

[2798] Infraestruturas de Sistemas Distribuídos / Distributed Systems Infra-structures

2. Sigla da área científica em que se insere

IC

3. Duração

Unidade Curricular Semestral

4. Horas de trabalho

162h 00m

5. Horas de contacto

Total: 60h 00m das quais T: 20h 00m | TP: 20h 00m | P: 20h 00m

6. % Horas de contacto a distância

Sem horas de contacto à distância

7. ECTS

6

8. Docente responsável e respetiva carga letiva na Unidade Curricular

[440] António Luís Freixo Guedes Osório | Horas Previstas: N/D

9. Outros docentes e respetivas cargas letivas na unidade curricular

Não existem docentes definidos para esta unidade curricular

10. Objetivos de aprendizagem e a sua compatibilidade com o método de ensino (conhecimentos, aptidões e competências a desenvolver pelos estudantes).

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 microserviços 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).

10. Intended Learning objectives and their compatibility with the teaching method (knowledge, skills and competences by the students).

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 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).

11. Conteúdos programáticos

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).

11. Syllabus

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).

12. Demonstração da coerência dos conteúdos programáticos com os objetivos de aprendizagem da unidade curricular

A complexidade do desenvolvimento de sistemas informáticos num crescendo de requisitos de integração de elementos heterogêneos, requer abordagens fundamentadas sobre opções arquiteturais. Aspectos 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.

12. Evidence of the syllabus coherence with the curricular unit's intended learning outcomes

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.

13. Metodologias de ensino e de aprendizagem específicas da unidade curricular articuladas com o modelo pedagógico

O aluno desenvolve exercícios simples onde é requerida a modelação e implementação de componentes distribuídas envolvendo tecnologias heterogêneas, 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).

13. Teaching and learning methodologies specific to the curricular unit articulated with the pedagogical model

The student develops simple exercises where modeling and implementing distributed components involving heterogeneous technologies to consolidate 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).

14. Avaliação

Dois trabalhos em grupo ($NP \geq 9,5$, média aritmética) integrando diferentes paradigmas que permitem aos alunos a concretização selecionando os quadros tecnológicos de forma fundamentada (5). Um exame final ($NE \geq 9,5$) permite consolidar a medida do desempenho na componente individual. A avaliação é distribuída com exame final. Todas as componentes são pedagogicamente fundamentais. A nota final é $NF = 0,5 \times NP + 0,5 \times NE$.

14. Assessment

Two group works ($NP \geq 9,5$) integrating different paradigms allow the students to materialize acquired competencies by selecting the proper technological frameworks (5). A final exam ($NE \geq 9,5$) allows for consolidating a measure of individual performance. Assessment is distributed with a final exam. All components are pedagogically fundamental. The final grade is $NF = 0,5 \times NP + 0,5 \times NE$.

15. Demonstração da coerência das metodologias de ensino com os objetivos de aprendizagem da unidade curricular

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).

15. Evidence of the teaching methodologies coherence with the curricular unit's intended learning outcomes

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).

**16. Bibliografia de
consulta/existência obrigatória**

1. Andrew S Tanenbaum, Maarten Van Steen (2013), Distributed Systems: Principles and Paradigm, Pearson, 2nd edition, ISBN-10: 1292025522.
2. Kenneth Birman (2005), Reliable Distributed Systems Technologies, Web Services, and Applications, Springer, ISBN: ISBN-10: 0387215093.
3. Philip A. Bernstein (2009), Principles of Transaction Processing, Morgan Kaufmann, ISBN-10: 1558606238.
4. Flavio Junqueira, Benjamin Reed (2013), ZooKeeper: Distributed Process Coordination, O'Reilly Media, ISBN-10: 1449361307.
5. Neha Narkhede, Gwen Shapira (2017), Kafka - The Definitive Guide, O'Reilly, ISBN-10: 1491936169.

17. Observações

Unidade Curricular Obrigatória

Data de aprovação em CTC:

Data de aprovação em CP: