Ficha de Unidade Curricular A3ES
Computação Distribuída
Mestrado em Engenharia Informática e de Computadores
2024-25

---

1. Caracterização da Unidade Curricular

1.1 Designação
[4117] Computação Distribuída / Distributed Computing

1.2 Sigla da área científica em que se insere
IC

1.3 Duração
Unidade Curricular Semestral

1.4 Horas de trabalho
162h 00m

1.5 Horas de contacto
Total: 67h 30m das quais T: 22h 30m | TP: 20h 00m | P: 25h 00m

1.6 ECTS
6

1.7 Observações
Unidade Curricular Obrigatória
Unidade Curricular comum ao(s) curso(s) de MEIM

---

2. Docente responsável [710] Luís Manuel da Costa Assunção

---

3. Docentes e respetivas cargas letivas na unidade curricular
[710] Luís Manuel da Costa Assunção | Horas Previstas: 67.5 horas
[1551] José Manuel de Campos Lages Garcia Simão | Horas Previstas: 67.5 horas

---

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

1. Descrever e discutir as vantagens, os problemas e os desafios que se colocam no desenvolvimento de aplicações usando o paradigma da computação distribuída;
2. Conhecer os padrões de arquitetura, de interação e comunicação entre as partes das aplicações distribuídas;
3. Conhecer e aplicar algoritmos relacionados com Tempo (relógios lógicos), ordenação e coordenação de eventos e consensos em computação distribuída;
4. Desenvolver aplicações distribuídas, com recurso a middleware e API para acesso a componentes e serviços remotos, usando modelos de chamadas remotas;
5. Saber utilizar o alojamento de serviços em infraestruturas distribuídas, locais e em Clouds públicas com recurso a Máquinas Virtuais e Contentores.

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

1. Know how to describe and discuss the advantages, problems and challenges that arise in the development of applications using the distributed computing paradigm.
2. Know the patterns of architecture, interaction, and communication between the parts of distributed applications.
3. Know and apply algorithms related to Time (logic clocks), ordering and coordination of events and consensus in distributed computing.
4. Develop distributed applications using middleware and API to access to remote components and services, doing remote invocations.
5. Know how to deploy services in distributed local infrastructures and in public Clouds using Virtual Machines and Containers.

5. Conteúdos programáticos

1. Caracterização, Potencialidades e Desafios: Comunicação; Concorrência; Resiliência; Escalabilidade; Elasticidade; Replicação; Teorema CAP; 8 Falácias da Computação Distribuída;
2. Execução de componentes distribuídas: Máquinas físicas e virtuais; Containers; infraestruturas Cloud.
3. Middleware de comunicação e interação: Sockets; Objetos distribuídos (Java RMI); Remote Procedure Call (gRPC); Serviços REST e Microservices; Callbacks, Comunicação assíncrona em streaming; Serviços
Stateless/Stateful; Message Queuing; Publish/Subscribe e event-driven (RabbitMQ);
4. Tempo, ordenação de eventos, coordenação e consensos. Relógios lógicos e vetoriais na ordenação causal de mensagens. Exclusão mútua e consensos baseados em eleições distribuídas. Comunicação por grupos e multicast com eventos de membership (Spread Toolkit);
5. Concretização de aplicações em infraestruturas locais e infraestruturas Cloud. Experimentação com máquinas virtuais e Containers (Google Cloud Platform e Docker)

5. Syllabus

1. Characterization, Potential and Challenges: Communication; Concurrency; Resilience; Scalability; Elasticity; Replication; CAP Theorem; 8 Fallacies of Distributed Computing;
2. Execution of the distributed components: Physical and Virtual machines; Containers; Cloud Infrastructures;
3. Middleware for communication and interaction: Sockets; Distributed Objects (Java RMI); Remote Procedure Call (gRPC), REST and Microservices; Callbacks, Asynchronous communication and Streaming; Stateless/Stateful Components; Message Queuing; Publish/Subscribe and event-driven (RabbitMQ);
4. Time, event ordering, coordination, and consensus. Logical and vector clocks for causal ordering of messages. Mutual exclusion and consensus based on distributed elections. Group communication and multicast with membership events (Spread Toolkit);
5. Implementation of applications on local and Cloud infrastructures. Practical laboratory with Virtual Machines and Containers (Google Cloud Platform and Docker)

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

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), (3) e (4) contribuem para os objetivos (1), (2), (3) e (4), 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) e (5) contribuem os conteúdos programáticos (2) e (5), onde o aluno tem de demonstrar a funcionalidade das soluções desenvolvidas nos trabalhos práticos.

6. Evidence of the syllabus coherence with the curricular unit´s intended learning outcomes

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), (3) and (4) contribute to objectives (1), (2), (3) and (4), 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) and (5), contribute the programmatic contents (2) and (5), where the student is challenged to evaluate and to demonstrate the functionality of the solutions developed in practical works.

7. Metodologias de ensino (avaliação incluída)

Metodologia de ensino teórico-prática, consistindo na apresentação e discussão dos temas e a concretização de exemplos que demonstrem os conceitos envolvidos. Por cada tema são propostos desafios a realizar em aulas práticas para os alunos encontrarem soluções, consolidando a aprendizagem.
A avaliação é distribuída com exame final. Os resultados de aprendizagem (1), (2) e (3) são avaliados através de um exame de avaliação individual. Os resultados de aprendizagem (4) e (5) são avaliados com dois trabalhos de projeto com relatório de concretização e a demonstração/discussão em sala de aula.
A avaliação é composta por:
1) Nota de Exame final (NE)>=9,5), pedagogicamente fundamental, com duração de 2 horas;
2) Trabalho de projeto 1 (NP1): Desenvolvimento de um sistema distribuído cliente/servidor
3) Trabalho de projeto 2 (NP2>=9,5), pedagogicamente fundamental, com o desenvolvimento de um sistema distribuído envolvendo os middleware estudados.

Nota final (NF)=0,5xNE+0,2xNP1+0,3xNP2.

7. Teaching methodologies (including assessment)

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 to students so that they find solutions, thus consolidating learning. Assessment is distributed with a final exam.
The learning outcomes (1), (2) and (3) are assessed through a individual assessment final examination.
The learning outcomes (4) and (5) are evaluated in laboratory work and two work projects including a completion report and a classroom demonstration and discussion of its functionality.
The evaluation is composed by:
1) Individual final Exam Grade (EG)>=9.5), pedagogically fundamental, lasting 2 hours;
2) Work Project 1 Grade (P1G): Development of a client/server distributed system
3) Work Project 2 Grade (P2G>=9.5), pedagogically fundamental, consisting of the development of a distributed system involving all studied middlewares

Final Grade (FG)=0.5xEG+0.2xP1G+0.3xP2G.

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

Usando uma metodologia de ensino teórico-prática, são apresentados conceitos, modelos, padrões e arquiteturas das aplicações baseadas no paradigma da computação distribuída. A aposta sistemática de demonstrar e concretizar exemplos de aplicação com plataformas tecnológicas existentes, contribui coerentemente para os objetivos de aprendizagem consolidado pela avaliação de trabalhos práticos laboratoriais.

8. Evidence of the teaching methodologies coherence with the curricular unit´s intended learning outcomes

Using a theoretical and practical teaching methodology, concepts, models, patterns and architectures of distributed computing applications are presented. The systematic approach to demonstrate concrete application examples with existing technological platforms, contributes coherently to learning objectives consolidated by practical laboratories and work assessment.

9. Bibliografia de consulta/existência obrigatória

George Coulouris, Jean Dollimore, Tim Kindberg and Gordon Blair, Distributed Systems: Concepts and Design, Fifth Edition, published by Addison Wesley, May 2011
Letha Hughes Etzkorn, Introduction to Middleware: Web Services, Object Components, and Cloud Computing, CRC Press, 2017
Martin Kleppmann, Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems, 1st Edition, 2017

Slides de suporte às aulas e artigos selecionados sobre cada tópico

10. Data de aprovação em CTC 2024-07-17 2024-07-17

11. Data de aprovação em CP 2024-06-26 2024-06-26