

Licenciatura: Engenharia Informática	Área Científica: 481 – Ciências Informáticas
Unidade curricular / Curricular Unit: Sistemas Multimédia I / Multimedia Systems I	ECTS: 6
Duração: Semestral	Horas de Contacto (Teórico Práticas): 60 (48 TP + 12 OT)
Professor(a) responsável / Responsible academic staff	Doutor Helder Rodrigo Pinto
e-mail institucional	helder.pinto@my.istec.pt
Outros professores / Other academic staff	Dr. Sérgio Nogueira
Objetivos de aprendizagem / Learning outcomes of the curricular unit (conhecimentos, aptidões e competências a desenvolver pelos estudantes)	
<p>No final os estudantes que a completarem deverão ser capazes de:</p> <ol style="list-style-type: none"> 1. Distinguir Metodologias de Desenvolvimento de Software Waterfall e Agile. 2. Identificar os intervenientes e celebrações do SCRUM. 3. Identificar requisitos funcionais e não funcionais para soluções e plataformas digitais usando o modelo FURPS+. 4. Desenhar Diagramas de Casos de Uso, identificando os atores, funcionalidades e associações. 5. Gerir versões de código, num contexto de desenvolvimento de software colaborativo, recorrendo a ferramentas como Git. 6. Aplicar procedimentos de planeamento e gestão de projetos de software, recorrendo à ferramenta WBS (Work Breakdown Structure). 7. Representar agregados, entidades, valores e relações de um sistema informático, pela modelação de domínio de um sistema, recorrendo a Domain-Driven Design (DDD). 8. Trabalhar em equipa no desenvolvimento de um projeto de Engenharia Informática. 9. Resolver problemas de forma criativa, resiliente e autónoma. 10. Desenvolver a comunicação assertiva e o pensamento crítico. <p>(English)</p> <p>Upon completion, students should be able to:</p> <ol style="list-style-type: none"> 1. Distinguish between Waterfall and Agile software development methodologies. 2. Identify SCRUM stakeholders and celebrations. 3. Identify functional and non-functional requirements for digital solutions and platforms using the FURPS+ model. 4. Design Use Case Diagrams, identifying actors, functionalities, and associations. 5. Manage code versions in a collaborative software development context using tools such as Git. 6. Apply software project planning and management procedures using the WBS (Work Breakdown Structure) tool. 7. Represent aggregates, entities, values, and relationships of a computer system by modelling a system domain using Domain-Driven Design (DDD). 8. Work as part of a team to develop a Computer Engineering project. 9. Solve problems creatively, resiliently and autonomously. 10. Develop assertive communication and critical thinking skills. 	
Conteúdos programáticos / Syllabus	
<ol style="list-style-type: none"> 1. Metodologias de Desenvolvimento de Software <ol style="list-style-type: none"> 1.1. Gestão de Projetos 1.2. WBS 1.3. Waterfall 1.4. Agile 1.5. SCRUM 	

2. Engenharia de Requisitos
 - 2.1. Requisitos funcionais
 - 2.2. Requisitos não funcionais
 - 2.3. Modelo FURPS+
 - 2.4. Diagrama de Casos de Uso
3. Gestão de Versões de Código
 - 3.1. Git
 - 3.2. Github
4. Modelo de Domínio
 - 4.1. Agregados
 - 4.2. Entidades
 - 4.3. Valores
 - 4.4. Relações
5. Projeto de Engenharia Informática
 - 5.1. Planeamento
 - 5.2. Implementação Colaborativa
 - 5.3. Documentação

(English)

1. Software Development Methodologies
 - 1.1. Project Management
 - 1.2. WBS
 - 1.3. Waterfall
 - 1.4. Agile
 - 1.5. SCRUM
2. Requirements Engineering
 - 2.1. Functional Requirements
 - 2.2. Non-Functional Requirements
 - 2.3. FURPS+ Model
 - 2.4. Use Case Diagram
3. Code Version Management
 - 3.1. Git
 - 3.2. Github
4. Domain Model
 - 4.1. Aggregates
 - 4.2. Entities
 - 4.3. Values
 - 4.4. Relationships
5. Computer Engineering Project
 - 5.1. Planning
 - 5.2. Collaborative Implementation
 - 5.3. Documentation

Demonstração da coerência dos conteúdos programáticos com os objetivos da unidade curricular / Demonstration of the syllabus coherence with the curricular unit's objectives

A Unidade Curricular (UC) de Sistemas Multimédia I (SMUL1) adota uma abordagem centrada num projeto interdisciplinar que integra conhecimentos de Tecnologias de Internet II (TINT2), Programação III (PROG3), Bases de Dados (BDDAD) e Teoria das Probabilidades e Modelos de Simulação (TEPMS). Esta metodologia, baseada em casos práticos e projetos, é perfeitamente coerente com os objetivos, potenciando a aplicação e consolidação das competências.

Os objetivos de reconhecer e identificar metodologias de desenvolvimento de *software* (Objetivo 1, Competências 1, 2 e 7), nomeadamente Agile e SCRUM, são intrínsecos à estrutura da UC. O desenvolvimento do projeto em Sprints com *User Stories* e a avaliação ao longo dos Sprints espelham a prática destas metodologias.

A necessidade de levantar e identificar requisitos funcionais e não funcionais (Objetivo 2, Competência 3) para o sistema é diretamente satisfeita pela análise das *User Stories* fornecidas em cada Sprint. A capacidade de desenhar Diagramas de Casos de Uso (Objetivo 3, Competência 4) é crucial para formalizar e modelar o sistema.

A utilização de ferramentas de gestão de versões como Git (Objetivo 4, Competência 5) é uma exigência natural do trabalho colaborativo em equipa, fomentado pelo projeto. O planeamento e gestão de projetos (Objetivo 5, Competência 6), incluindo o uso de WBS, é praticado nas fases de *Sprint Planning* e na distribuição de tarefas.

A representação de um sistema informático através de Modelo de Domínio (DDD) (Objetivo 6, Competência 7) é aplicada na conceptualização e implementação do projeto.

Finalmente, o trabalho em equipa (Competência 8), a resolução criativa e autónoma de problemas (Competência 9) e o desenvolvimento da comunicação e pensamento crítico (Competência 10) são continuamente exercitados na dinâmica dos Sprints, nas *Sprint Reviews* e na defesa final do projeto, culminando numa avaliação que reflete a aquisição integral destas competências.

(English)

The Multimedia Systems I (SMUL1) course adopts an interdisciplinary project-based approach that integrates knowledge from Internet Technologies II (TINT2), Programming III (PROG3), Databases (BDDAD) and Probability Theory and Simulation Models (TEPMS). This methodology, based on practical cases and projects, is perfectly consistent with the objectives, enhancing the application and consolidation of skills.

The objectives of recognising and identifying software development methodologies (Objective 1, Skills 1, 2 and 7), namely Agile and SCRUM, are intrinsic to the structure of the CU. The development of the project in Sprints with User Stories and the evaluation throughout the Sprints reflect the practice of these methodologies.

The need to raise and identify functional and non-functional requirements (Objective 2, Competence 3) for the system is directly met by the analysis of the User Stories provided in each Sprint. The ability to design Use Case Diagrams (Objective 3, Competence 4) is crucial for formalising and modelling the system.

The use of version management tools such as Git (Objective 4, Competence 5) is a natural requirement of collaborative teamwork, fostered by the project. Project planning and management (Objective 5, Competence 6), including the use of WBS, is practised in the Sprint Planning phases and in the distribution of tasks.

The representation of a computer system using Domain-Driven Design (DDD) (Objective 6, Competence 7) is applied in the conceptualisation and implementation of the project.

Finally, teamwork (Competence 8), creative and autonomous problem solving (Competence 9) and the development of communication and critical thinking (Competence 10) are continuously exercised in the dynamics of Sprints, Sprint Reviews and the final defence of the project, culminating in an assessment that reflects the full acquisition of these competences.

Metodologias de ensino e avaliação / Teaching methodologies including evaluation

As aulas desta UC assumem um carácter teórico-prático e são lecionadas num contexto baseado em casos práticos e projetos, complementadas com demonstração da aplicação prática com recurso a exercícios, bem como orientação e mentoria em trabalhos. O Projeto Interdisciplinar é desenvolvido segundo metodologia Agile, com 3 Sprints e User Stories bem definidas, que os estudantes devem analisar, planear nos Sprint Planning; implementar e testar ao longo do Sprint; e defender nas Sprint Review.

(English)

The classes in this CU are theoretical and practical in nature and are taught in a context based on practical cases and projects, complemented by demonstrations of practical application using exercises, as well as guidance and mentoring in assignments.

The Interdisciplinary Project is developed according to Agile methodology, with 3 Sprints and well-defined User Stories, which students must analyse, plan in Sprint Planning; implement and test throughout the Sprint; and defend in the Sprint Review.

Demonstração da coerência das metodologias de ensino com os objetivos da unidade curricular / Demonstration of the coherence between the teaching methodologies and the learning outcomes

As metodologias de ensino da UC SMUL1, profundamente teórico-práticas e centradas em casos práticos e num projeto interdisciplinar Agile, estabelecem uma coerência robusta com os objetivos de aprendizagem.

O desenvolvimento do projeto em Sprints com *User Stories* (Metodologia Agile) capacita diretamente os estudantes a reconhecer e implementar metodologias de desenvolvimento de *software* ágeis (Objetivos 1 e 7, Competências 1 e 2). A análise das *User*

Stories em cada *Sprint Planning* promove o levantamento e representação de requisitos funcionais e não funcionais, e o desenho de Diagramas de Casos de Uso (Objetivos 2 e 3, Competências 3 e 4).

A natureza colaborativa dos Sprints, aliada à orientação e mentoria e à necessidade de gerir versões de código com Git (Objetivo 4, Competência 5), reforça o trabalho em equipa (Competência 8). O planeamento inerente a cada Sprint alinha-se com o objetivo de planear e gerir projetos de *software* (Objetivo 5, Competência 6).

A aplicação prática contínua, com demonstrações e exercícios, fomenta a resolução de problemas de forma criativa e autónoma (Competência 9) e desenvolve a comunicação assertiva e o pensamento crítico (Competência 10), competências cruciais para a defesa dos projetos e interação em equipa. Esta abordagem garante que os conhecimentos são aplicados e consolidados num ambiente que simula cenários reais da engenharia de software.

(English)

The teaching methodologies of UC SMUL1, which are deeply theoretical and practical and focused on practical cases and an interdisciplinary Agile project, establish a robust consistency with the learning objectives.

The development of the project in Sprints with User Stories (Agile Methodology) directly enables students to recognise and implement agile software development methodologies (Objectives 1 and 7, Competencies 1 and 2). The analysis of User Stories in each Sprint Planning promotes the survey and representation of functional and non-functional requirements, and the design of Use Case Diagrams (Objectives 2 and 3, Competencies 3 and 4).

The collaborative nature of Sprints, combined with guidance and mentoring and the need to manage code versions with Git (Objective 4, Competency 5), reinforces teamwork (Competency 8). The planning inherent in each Sprint aligns with the objective of planning and managing software projects (Objective 5, Competency 6).

Continuous practical application, with demonstrations and exercises, encourages creative and autonomous problem solving (Competency 9) and develops assertive communication and critical thinking (Competency 10), crucial skills for defending projects and team interaction. This approach ensures that knowledge is applied and consolidated in an environment that simulates real-world software engineering scenarios.

Bibliografia / Bibliography

Fundamental

Documentação da UC

Nunes, M., & O'Neill, H. (2004). Fundamentos de UML. FCA

Miguel, A. (2018). Gestão de Projetos de Software. FCA

Vaughn, V. (2013). Implementing Domain Driven Design. Addison Wesley

Complementar / Complementary

Evans, E. (2003). Domain-Driven Design: Tackling Complexity in the Heart of Software. Addison Wesley