
1. Caracterização da Unidade Curricular

1.1 Designação

[1629] Projeto / Project

1.2 Sigla da área científica em que se insere

INF

1.3 Duração

Unidade Curricular Semestral

1.4 Horas de trabalho

324h 00m

1.5 Horas de contacto

Total: 90h 00m das quais TP: 13h 30m | P: 45h 00m | S: 9h 00m | O: 22h 30m

1.6 ECTS

12

1.7 Observações

Unidade Curricular Obrigatória

2. Docente responsável

[1457] Paulo Manuel Trigo Cândido da Silva

3. Docentes e respetivas cargas letivas na unidade curricular

Não existem docentes definidos para esta unidade curricular

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

1. Avaliar e escolher propostas (alternativas) de projetos de aplicação e integração de conhecimento adquirido no curso.
2. Formular, autonomamente, objectivos e seu (re)planeamento e monitorização visando concretizar o projeto.
3. Explorar propostas (inovadoras) em temas do projeto, apreciar originalidade das técnicas e testá-las para fundar escolhas; implementar (ou integrar) técnicas para alcançar os objectivos.
4. Adoptar e adaptar (ao projeto) metodologias de desenvolvimento (de software), desenho baseado em padrões e técnicas para realização de testes; implementar soluções que incorporem e evidenciem estes aspectos.
5. Expor e defender publicamente o seu trabalho e arguir trabalho exposto por outros.
6. Escrever relatório expondo de modo organizado, claro e sintético, os objectivos do projeto, o enquadramento com trabalhos relacionados, as técnicas implementadas (ou integradas), a eventual originalidade dessas técnicas (no confronto com alternativas) e apreciação de resultados.



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

1. Evaluate and choose (alternative) project proposals that aim to apply and integrate the knowledge acquired during the course.
2. Formulate, autonomously, goals and its (re)planning and monitoring to accomplish the project.
3. Explore (innovative) proposals, in the project's subjects, assess the originality of the techniques and test them to ground the choices; implement (or integrate) techniques to achieve the goals.
4. Adopt and adapt (to the project) development methodologies (of software), pattern-based design and testing techniques; implement solutions that incorporate and make evidence of those aspects.
5. Expose and defend publicly his work and argue the work exposed by others.
6. Write a project report that exposes in an organized, clear and synthetic way the project objectives, the relation with related work, the techniques implemented (or integrated), the possible originality of those techniques (in comparison with alternatives), and the assessment of results.

5. Conteúdos programáticos

- I. Fundamentos dos processos de desenvolvimento incrementais, especificação de requisitos, caracterização do risco, escalonamento de prioridades; definição de objectivos e tarefas, gestão do tempo considerando a urgência e importância das tarefas, construção de planos de curto e médio prazo.
- II. Aplicação de padrões para especificação do modelo de análise a partir dos requisitos e tradução explícita desses modelos na implementação (código); desenho de arquitetura candidata e sua validação.
- III. Especificação de testes e sua implementação e integração com a validação de requisitos; controlo e gestão de versões.
- IV. Noção de relatório de projeto (e de dissertação científica) quanto à estrutura de capítulos (mais usuais) e conteúdo (esperado) em cada capítulo (nessa estrutura); notações para identificação de grafismos, formulações, e referências bibliográficas (e.g., via LaTeX); diferenças entre o relatório de projeto e um manual de utilizador ou a descrição de uma interface aplicacional.



5. Syllabus

- I. Fundamentals of incremental development processes, specification of requirements, risk characterization, priority scheduling, definition of goals and tasks, time management considering the urgency and importance of tasks, construction of short and medium term plans.
- II. Application of patterns to specify the analysis model from the requirements and explicit translation, of those models, into the implementation (code); design of the candidate architecture and its validation.
- III. Specification of tests and its implementation and integration with the requirements? validation; version control and management.
- IV. Concept of project report (and of scientific dissertation) in what concerns the (most usual) structure of chapters and he (expected) content of each chapter (within that structure); notations to identify graphics, formulas, and bibliographic references (e.g., via LaTeX); differences between project report and user manual or the description of an application programming interface.

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

Esta UC proporciona experiência de aplicação e integração dos conhecimentos do curso. Isto concretiza-se de dois modos: a) na abordagem às atividades de concepção, desenho, implementação e operação (CDIO) de um projeto (cf., itens I, II, III), e b) fomentando o aprofundar de temas curriculares e estimulando a capacidade de enquadrar e confrontar soluções atuais com ideias próprias (cf., itens II e IV).

A motivação para desenvolver o projeto de modo disciplinado e sistemático foca, para além dos aspectos tecnológicos, a gestão de tempo e risco, interação em equipa, teste, validação e manutenção de sistemas (itens, I, II, III).

A perspectiva de cada projeto, como um contributo que incrementa, de modo coerente, experiência e conhecimento a trabalho (projetos) anterior, é mantida pelo(s) docente(s) orientador(es) de cada projeto e suportada pela qualidade dos relatórios produzidos (cf., item IV) e pela capacidade para reutilizar o código desenvolvido (cf., todos itens, ênfase para o II).



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

This UC provides experience of application and integration of the knowledge acquired in the course. This is achieved in two ways: a) in the execution of the tasks to conceive, design, implement and operate (CDIO) a project (cf., items I, II, III), and b) fostering the deepening of curricula subjects and stimulating the ability to frame and confront current solutions with own ideas (cf., items II, IV).

The motivation to develop the project in a disciplined and systematic way focuses, apart from technological aspects, the management of time and risk, the team interaction, the testing, validation and maintenance of the system (items I, II, II).

The perspective of each project, as a contribution that increases, consistently, experience and knowledge to previous work (projects), is maintained by the teacher(s) that supervise each project and it is supported by the quality of the produced reports (cf., item IV) and the capability to reuse the implemented code (cf., all items, emphasis to II)

7. Metodologias de ensino (avaliação incluída)

Avaliação distribuída com exame final, incluindo 2 componentes:

- Participação-FEIM (Fórum em Engenharia Informática e Multimédia)
- Prova-Final com Júri (exame final)

Trabalhos em grupo (preferência 2 alunos).

Participação-FEIM: avaliação de valor binário (em {1, 0}), a q_{FEIM} , indicando presença, ou ausência, de qualidade; a atribuição é feita:

- após apresentação
- por docente da UC e pelo(s) orientador(es) do trabalho
- "peso" docente UC igual ao de todos orientadores
- ausência de consenso implica posição do coordenador da UC ou LEIM
- $q_{FEIM}=0$, para alunos que não realizem esta componente

Prova-Final pedagogicamente fundamental: Júri com 1 arguente proposto pelo(s) orientador(es), até 2 orientadores, responsável da UC (ou em quem ele delegue)

Duração ≤ 1 h: apresentação ≤ 15 min, arguição ≤ 45 min

Critérios: relatório (R); produto (P); apresentação (A); discussão (D)

Nota Prova-Final (NP): $R*30\% + P*40\% + A*15\% + D*15\%$ ($NP \geq 9,50$)

Nota Final (NF): $NP * (90\% + q_{FEIM}*10\%)$

Aprovação: $NF \geq 9,50$



**7. Teaching methodologies
(including assessment)**

Assessment distributed with final exam, including 2 components:

- Participation-FEIM (Forum on Eng. of Inf. and Mult.)
- Final-Exam with Panel

Group work (likely 2 students).

Participation-FEIM: assessment assigns binary value (in {1, 0}), to q_{FEIM} , indicating presence, or absence, of quality; the assignment is made:

- after presentation
- by UC teacher and the work supervisor(s)
- UC teacher "weight" equals to that of all supervisors
- lack of consensus implies the position of UC or LEIM coordinator
- $q_{FEIM} = 0$, for students that do not perform this component

Final-Exam pedagogically fundamental: Panel with 1 examiner proposed by supervisor(s), up to 2 supervisors, UC coordinator (or whom he delegates)

Duration $\leq 1h$: presentation $\leq 15min$, oral-exam $\leq 45min$

Criteria: report (R); product (P); presentation (A); discussion (D)

Exam-Score (NP): $R*30\% + P*40\% + A*15\% + D*15\%$

Final-Score (NF): $NP * (90\% + q_{FEIM}*10\%)$

Approval: $NF \geq 9,50$



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

Nas aulas TP iniciais são expostas as propostas de projeto; exposição feita pelos docentes proponentes e os alunos têm cerca de 2 semanas para escolher (e propor); cf., objetivo de aprendizagem (OA) 1. Restantes TP apresentam os conteúdos programáticos (ênfase OA 2, 4) com problemas ilustrativos a resolver pelos alunos, na sala, antes de apresentada a solução. O problema da escrita do relatório (cf., OA 6) é também abordado em TP.

Nas aulas PL desenvolve-se o trabalho de projeto; a competência para percorrer as várias fases do projeto vai sendo incrementalmente amadurecida em contexto de experimentação e visando alcançar os objetivos definidos por cada grupo de trabalho (cf., OA 2, 3, 4).

Nas aulas S cada grupo apresenta o seu trabalho (cf., OA 3, 5) aos docentes da UC e ao(s) orientador(es). Na sequência os docentes da UC emitem apreciação qualitativa do estado do trabalho e comentários de síntese.

As aulas S servem de suporte à preparação da apresentação no FEIM que por sua vez representa um ponto de situação e um momento de preparação para o exame final. Uma adequada preparação e motivação para o FEIM contribui também para a projeção de cada trabalho no espaço das empresas que também expõem os seus objetivos no decurso do FEIM.

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

The initial TP classes present the project proposals; presentations are made by the proponent teachers and students have about 2 weeks to choose (and propose), cf., learning outcomes (LO) 1. The remaining TP present the syllabus (emphasis LO 2, 4) with illustrative problems to be solved by students, in classroom, before the solution presentation. The problem of writing the report (cf., LO 6) is also addressed in TP.

In classes PL the project work is developed; the competence to move through the various project activities is incrementally matured in an experimental context and aimed at achieving the goals defined by each working group (cf., LO 2, 3, 4).

In classes S each group presents his work (cf., LO 3, 5) to the UC teaches and to the supervisor(s). Thereafter the UC teachers transmit a qualitative appreciation on the state of the work and a feedback synthesis.

The S classes serve as a preparation for the presentation in FEIM, which in turn represents a work-status? point and a moment of preparation for the final exam. Adequate preparation and motivation for the FEIM also contributes to propagate each work in the space of the companies that also expose their objectives during the FEIM.

9. Bibliografia de

consulta/existência obrigatória

1. Kruchten, P. (2003). The Rational Unified Process: An Introduction (3rd ed.). Addison-Wesley.
2. Larman, C. (2004). Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development (3rd ed.). Prentice Hall. available-online
3. Eckel., B. (2017). Python 3 Patterns, Recipes and Idioms; r1.0. available-online
4. Osmani, A. (2017). Learning JavaScript Design Patterns; vol 1.7.0. O'Reilly. available-online
5. Scott, B. and Neil, T. (2009). Designing Web Interfaces: Principles and Patterns for Rich Interactions; O'Reilly.
6. Grand, M. (2003). Patterns in Java: A Catalog of Reusable Design Patterns Illustrated with UML (2nd ed.). Wiley Publishing.
7. Gamma, E., Helm, R., Johnson R., and Vlissides, J. M. (1994). Design Patterns: Elements of Reusable Object-Oriented Software; Addison-Wesley.
8. Lampion, L. (1994). LaTeX: A Document Preparation System (2nd ed.). Addison-Wesley.

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

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