

	GUIA DE FUNCIONAMENTO DA UNIDADE CURRICULAR (GFUC)	MODELO PED.008.03
---	---	-----------------------------

Curso	Engenharia Informática						
Unidade curricular (UC)	Engenharia de Software II						
Ano letivo	2023-2024	<i>Ano</i>	3.º	<i>Período</i>	1.º semestre	<i>ECTS</i>	6
Regime	Obrigatório	<i>Tempo de trabalho (horas)</i>			Total: 168	Contacto: 75	
Docente(s)	Doutora Maria Clara dos Santos Pinto Silveira						
<input type="checkbox"/> <i>Responsável da UC ou</i> <input checked="" type="checkbox"/> <i>Coordenador(a) Área/Grupo Disciplinar</i> <input type="checkbox"/> <i>Regente (cf. situação de cada Escola)</i>	Doutora Maria Clara dos Santos Pinto Silveira						

GFUC PREVISTO

1. OBJETIVOS DE APRENDIZAGEM

Após a conclusão da UC, os alunos deverão ser capazes de:

1. *Projetar, executar e gerir o ciclo de vida do desenvolvimento de um sistema de software recorrendo a diferentes abordagens;*
2. *Elaborar a documentação técnica de um projeto usando a linguagem UML (Unified Modeling Language) e Essence - SEMAT;*
3. *Aplicar padrões de casos de uso no levantamento e análise de requisitos;*
4. *Validar as funcionalidades do sistema face aos requisitos especificados.*

2. CONTEÚDOS PROGRAMÁTICOS

1. *Engenharia de Software e a problemática do desenvolvimento de software. Boas práticas no desenvolvimento de software.*
2. *O processo de desenvolvimento de software: ciclo de vida de software na perspetiva orientada para objetos: Rational Unified Process; SCRUM; DevOps; SecDevOps; MDA -Model-Driven Architecture: definição, abordagem, ferramentas, casos de sucesso; Melhoria do processo: CMMI - Capability Maturity Model Integration.*
3. *Especificação de requisitos de software. Gestão de requisitos. Modelo em espiral para a engenharia de requisitos.*
4. *Padrões de Software: introdução ao conceito de padrão; estudo e aplicação de padrões para casos de uso.*
5. *UML - Unified Modeling Language: casos de uso, modelação da estrutura, do comportamento e da arquitetura. Essence – SEMAT: arquitetura, dimensões, elementos-chave (Alfa, produtos tangíveis, atividades, competências)*

<p>POLI ESCOLA SUPERIOR TECNOLOGIA GESTÃO TÉCNICO GUARDA</p>	<p>GUIA DE FUNCIONAMENTO DA UNIDADE CURRICULAR (GFUC)</p>	<p>MODELO PED.008.03</p>
--	--	-------------------------------------

6. *Ciclo de vida de utilização de Ferramentas CASE. Integração de ferramentas.*
7. *Métricas no processo de desenvolvimento de Software.*
8. *Verificação e validação do software.*

3. DEMONSTRAÇÃO DA COERÊNCIA DOS CONTEÚDOS PROGRAMÁTICOS COM OS OBJETIVOS DA UC

1. *Os Conteúdos 1, 2, 3, 4, 5, 6, 7 e 8 estão coerentes com o Objetivo 1, pois focam as diferentes abordagens.*
2. *Os Conteúdos 3, 4, 5 e 6 estão coerentes com o Objetivo 2, pois são apresentadas as diversas formas de documentar os requisitos, lecionada a linguagem UML, bem como as ferramentas necessárias.*
3. *Os Conteúdos 3 e 4 estão coerentes com o Objetivo 3, dado que se ministram os padrões para especificação de requisitos na abordagem “Patterns for Effective Use Cases”.*
4. *O Conteúdo 8 está coerente com o Objetivo 4, dado que é usado o Modelo em V que faz o paralelismo entre as entregas do processo de desenvolvimento de software e as entregas do processo de testes.*

4. BIBLIOGRAFIA PRINCIPAL

Obrigatória:

1. *Textos de apoio e diapositivos das aulas fornecidos pelo docente e disponibilizados na plataforma de e-learning.*
2. *Ivar Jacobson; Harold "Bud" Lawson; Pan-Wei Ng; Paul E. McMahon; Michael Goedicke. The Essentials of Modern Software Engineering: Free the Practices from the Method Prisons; ACM Books, 2019.*
3. *Nunes, Mauro; O'Neill, Henrique. Fundamental de UML, 5ª Ed., FCA Editora, 2007.*
4. *Sommerville, I. (2020). Engineering software products. London, UK: Pearson.*
5. *Adolph, Steve; Bramble, Paul. Patterns for Effective Use Cases, Addison-Wesley Pearson Education, 2003.*
6. *Unified Modeling Language™, Resource Page, www.uml.org.*

Recomendada:

7. *Jacobson, Ivar; Booch, Grady; Rumbaugh, James; The Unified Software Development Process; Addison –Wesly; 1999.*
8. *Rubin, Kenneth S. Essential Scrum: A Practical Guide to the Most Popular Agile Process, 2012, Addison-Wesley Professional.*

<p>POLI ESCOLA SUPERIOR TECNOLOGIA GESTÃO TÉCNICO GUARDA</p>	<p>GUIA DE FUNCIONAMENTO DA UNIDADE CURRICULAR (GFUC)</p>	<p>MODELO PED.008.03</p>
--	--	-------------------------------------

9. O'Neil, H., Nunes, M., Ramos, P. *Exercícios de UML*, FCA, 2010.

10. Penzenstadler, B., Betz, S., Venters, C.C., Chitchyan, R., Porras, J., Seyff, N., Duboc, L., & Becker, C. (2018). *Everything is INTERRELATED: Teaching Software Engineering for Sustainability*. *IEEE/ACM 40th International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET)*, 153-162.

11. Reis, L. Cagica, L., Silveira, C., Russo, N., & Marques A. (2021). *Inovação e Sustentabilidade em Tecnologias de Informação e Comunicação*. Lisboa: Silabo, ISBN: 978-989-561-146-1.

12. Santos, Vitor (2018). *Criatividade em Sistemas de Informação*. Lisboa, Editora FCA. ISBN: 978-972-722-891.

13. Sommerville, Ian. *Software Engineering (10th edition)*. Slides disponíveis em: <https://software-engineering-book.com/slides/>.

14. Torres, J., Julio, D., Silveira, C., & Reis, L. (2022). *Diabetes Tracker and Volunteer+ Software Engineering for Sustainability*. In L. Reis, L. Carvalho, C. Silveira, & D. Brasil (Ed.), *Digitalization as a Driver for Smart Economy in the Post-COVID-19 Era* (pp. 198-227). IGI Global.

5. METODOLOGIAS DE ENSINO (REGRAS DE AVALIAÇÃO)

Metodologias de ensino:

1. *Aprendizagem ativa*
2. *Lição expositiva*
3. *Trabalho de projeto*
4. *Estudo de casos*
5. *Resolução de problemas.*

Regras de avaliação:

Avaliação contínua (Frequência): o estudante está obrigado à presença em pelo menos 1/2 das aulas para poder ser avaliado na época de avaliação contínua. Os estudantes com o estatuto trabalhador-estudante não têm presenças obrigatórias.

1. 80% (16 valores) - *Atividades realizadas, preferencialmente, durante as aulas, envolvendo trabalhos individuais/em grupo/desenvolvimento de um projeto em conjunto com a Unidade Curricular (UC) Programação para a Internet. Os estudantes com o estatuto trabalhador-estudante terão de realizar essas atividades, mesmo que fora das aulas, para poderem ser avaliados nesta componente.*

<p>POLI ESCOLA SUPERIOR TECNOLOGIA GESTÃO TÉCNICO GUARDA</p>	<p>GUIA DE FUNCIONAMENTO DA UNIDADE CURRICULAR (GFUC)</p>	<p>MODELO PED.008.03</p>
--	--	-------------------------------------

2. 20% (4 valores) - Entrega do relatório (grupo) em pdf e protótipo relativo ao Projeto em conjunto com a UC Programação para a Internet. O protótipo tem um peso de 2 valores. Apresentação e defesa **individual** (obrigatória).

Data de entrega do relatório do projeto: 6 de janeiro de 2024.

Data da apresentação e defesa (obrigatória): 9 ou 12 de janeiro 2024.

A avaliação em cada uma das Unidades Curriculares é feita de forma independente, com base nos elementos previamente definidos e na informação obtida nas aulas, onde é feita a orientação, o controlo, supervisão e a avaliação do projeto desenvolvido e do seu progresso. Os alunos que já aprovaram/não estão a frequentar a UC Programação para a Internet, terão de desenvolver e entregar um protótipo da aplicação (plataforma de desenvolvimento à escolha do aluno e aprovada pelo docente).

Exame época normal /Exame época recurso/Exame época especial: para o estudante que não tenha obtido aproveitamento na avaliação contínua ou não a tenha realizado.

100% (20 valores) - Teste escrito.

6. DEMONSTRAÇÃO DA COERÊNCIA DAS METODOLOGIAS DE ENSINO COM OS OBJETIVOS DA UC

1. A aprendizagem ativa está coerente com os objetivos pois permite aos estudantes desenvolverem competências colaborativas, aumentarem o interesse pelos conteúdos programáticos, para além da introdução da criatividade e do pensamento crítico que pode ser aplicado tanto na fase de especificação de requisitos como na de implementação de projetos de software.
2. Lição expositiva está coerente com os objetivos devido à necessidade de apresentar os conteúdos teóricos aos estudantes, nomeadamente os vários modelos de processo, os padrões, as métricas e os níveis de maturidade.
3. Trabalho de projeto está coerente com os objetivos visto que o trabalho final abrange todas as etapas do processo de desenvolvimento de software, pelo que obriga à aplicação de todos os conceitos abordados ao longo do semestre a um caso prático.
4. Estudo de casos permite analisar documentação de outros projetos, analisar casos de sucesso de empresas que usam determinadas abordagens (por exemplo MDA), conhecer empresas certificadas CMMI, entre outros.

<p>POLI ESCOLA SUPERIOR TECNOLOGIA GESTÃO</p> <p>TÉCNICO GUARDA</p>	<p>GUIA DE FUNCIONAMENTO DA UNIDADE CURRICULAR (GFUC)</p>	<p>MODELO PED.008.03</p>
--	--	-------------------------------------

5. *Resolução de problemas está coerente com os objetivos pois a aplicação de conteúdos teóricos a exercícios práticos de inspiração realista, relacionados com os conteúdos (modelar sistemas, aplicar padrões, desenhar testes) ajuda a consolidar a matéria, realçando o saber fazer.*

7. REGIME DE ASSIDUIDADE

O estudante está obrigado à presença em pelo menos 1/2 das aulas para poder ser avaliado na época de avaliação contínua e no exame da época normal. Os estudantes com o estatuto trabalhador-estudante não têm presenças obrigatórias.

8. CONTACTOS E HORÁRIO DE ATENDIMENTO

Contactos: Maria Clara Silveira, Gab.21 | e-mail: mclara@ipq.pt

Atendimento: 3ª feira das 9h -11h e 14h30 às 17h ou combinar por e-mail.

DATA

19 de setembro de 2023