

### Informação Geral:

<b>Unidade Orgânica</b>	Escola Superior de Tecnologia e Gestão	<b>Ano Letivo</b>	2015/2016
<b>Curso</b>	Licenciatura em Engenharia Informática	<b>Grau</b>	Licenciatura
<b>UC/Módulo</b>	Tópicos Avançados de Engenharia de Software	<b>ECTS</b>	6
<b>Área Científica</b>	Sistemas de Informação	<b>Carácter</b>	Obrigatório
<b>Horas Totais</b>	162	<b>T</b> 30	<b>TP</b> 0
		<b>PL</b> 45	<b>TC</b> 0
		<b>S</b> 0	<b>E</b> 0
		<b>OT</b> 0	

*T - Ensino Teórico; TP - Teórico Prático; PL - Prático e Laboratorial; TC - Trabalho de Campo; S - Seminário; E - Estágio; OT - Orientação Tutorial*

### Docente responsável:

Catarina Isabel Ferreira Viveiros Tavares Reis

### Docentes que lecionam a unidade curricular:

Catarina Isabel Ferreira Viveiros Tavares Reis ( 75,00 horas semanais de contacto: T: 30,00; PL: 9,00; PL: 36,00; )

### Pré-requisitos:

Não tem.

### Idioma:

Português e Inglês

### Enquadramento:

Esta UC permite ao Estudante aprofundar os conhecimentos de Engenharia de Software, nomeadamente acerca das metodologias ágeis aplicadas ao desenvolvimento de software tais como o XP, Scrum e Kanban (Lean). Permite ainda ao Estudante aferir dos principais factores que influenciam a adopção de uma metodologia ágil versus uma metodologia plan-driven.

O Estudante adquire igualmente competências na utilização das principais ferramentas e técnicas para realizar, sob uma perspectiva ágil, a engenharia de requisitos, o desenho de software, a implementação/codificação de programas, os testes de software e a gestão de riscos em projetos de software.

É uma UC que favorece igualmente a maturação de equipas de software e os vários papéis que dela fazem parte, e ainda a interação com clientes deste tipo de produtos.

### Objetivos de aprendizagem:

C1 - Conhecimento e compreensão da perspectiva ágil no desenvolvimento de software;

C2 - Capacidade de relacionar os conceitos das metodologias ágeis de desenvolvimento de software com os conceitos da gestão de projetos tradicional;

C3 - Capacidade de identificar as principais ferramentas e técnicas utilizadas pelas metodologias ágeis de desenvolvimento de software;

C4 - Capacidade de aplicar os conceitos, ferramentas e técnicas num ambiente real de desenvolvimento de software segundo uma metodologia ágil;

C5 - Capacidade de gestão de equipas de desenvolvimento software segundo uma perspectiva ágil;

C6 - Capacidade de identificar, avaliar e resolver os riscos de um projecto de software segundo uma perspectiva ágil.

## Programa:

### Conteúdos Programáticos:

1. Introdução às metodologias ágeis de desenvolvimento de software
2. A metodologia eXtreme Programming
3. A metodologia Scrum
4. A metodologia Kanban
5. Gestão Ágil de projectos de software
6. Arquiteturas de software
7. Desenho de software - padrões de desenho e refactoring
8. Testes de software segundo uma perspectiva ágil

### Fundamentação da coerência dos conteúdos programáticos com os objetivos/competências da unidade curricular:

Os conteúdos programáticos lecionados contribuem para as competências gerais estabelecidas para a UC da seguinte forma:

1. Introdução às metodologias ágeis de desenvolvimento de software (C1, C2, C3, C4)
2. A metodologia eXtreme Programming (C1, C2, C3, C4, C5)
3. A metodologia Scrum (C1, C2, C3, C4, C5)
4. A metodologia Kanban (C1, C2, C3, C4, C5)
5. Gestão Ágil de projectos de software (C1, C2, C3, C4, C5, C6)
6. Arquiteturas de software (C1, C2, C3)
7. Desenho de software - padrões de desenho e refactoring (C1, C2, C3)
8. Testes de software segundo uma perspectiva ágil (C1, C2, C3)

## Metodologia de Ensino / Aprendizagem:

### Presencial:

1. Ensino teórico
  - 1.1 Apresentação dos conceitos e princípios do programa da UC
  - 1.2 Exemplificação com problemas de desenvolvimento de software reais
  - 1.3 Aplicação das técnicas e ferramentas das metodologias ágeis a problemas de desenvolvimento de software
  - 1.4 Análise crítica dos resultados obtidos
2. Ensino prático e laboratorial
  - 2.1 Realização de um projeto de desenvolvimento de software adoptando uma metodologia ágil;
  - 2.2 Integração de entrevistas e de interações com o cliente real do produto de software;
  - 2.3 Apresentação e discussão pública do produto de software desenvolvido.
3. Orientação tutorial
  - 3.1 Sessões de orientação pessoal, em pequenos grupos para conduzir o processo de aprendizagem e esclarecerem-se dúvidas

### Autónoma:

1. Estudo
  - 1.1 Leitura de excertos de bibliografia recomendada pela unidade curricular
  - 1.2 Resolução dos exercícios recomendados pela unidade curricular
  - 1.3 Desenvolvimento do projeto prático
2. E-aprendizagem
  - 2.1 Consulta de material relativo à unidade curricular

### Recursos Específicos:

1. Ensino teórico - sala de aula normal

- 2. Ensino teórico-prático - sala de aula normal
- 3. Ensino prático e laboratorial – laboratório de engenharia de software
- 4. Orientação tutorial – gabinete ou sala de aula normal

### Avaliação:

#### Descrição:

Avaliação periódica:

- Componente teórica: 30% (mínimos de 10):
  - 2 provas escritas teóricas (15% + 15%)
- Componente prática: 70%:
  - 2 projetos práticos com defesa pública (prova oral) obrigatória (cada um) (35%+35%)

Avaliação final:

- Componente teórica: 30%
  - prova escrita teórica (mínimos de 10);
- Componente prática: 70%
  - prova prática no computador;

Os estudantes que assim pretendam, estão dispensados da realização dos elementos de avaliação incluídos no método de avaliação da unidade curricular, desde que estes tenham sido realizados no próprio ano letivo ou no ano letivo imediatamente anterior, e que os estudantes tenham obtido os mínimos.

As melhorias de classificação devem ser realizadas integralmente, não sendo salvaguardadas notas parcelares para este efeito.

**Número de elementos de avaliação final:** 2

**Número de elementos de avaliação contínua/periódica:** 4

### Demonstração da coerência das metodologias de ensino com os objetivos de aprendizagem da unidade curricular:

1. Ensino teórico

- 1.1 Apresentação dos conceitos e princípios das metodologias ágeis (C1, C2)
- 1.2 Exemplificação com problemas de desenvolvimento de software reais (C1, C2)
- 1.3 Aplicação das técnicas e ferramentas das metodologias ágeis a problemas de desenvolvimento de software (C3, C4, C5, C6)
- 1.4 Análise crítica dos resultados obtidos (C3, C4, C5, C6)

2. Ensino prático e laboratorial

- 2.1 Realização de um projeto de desenvolvimento de software adoptando uma metodologia ágil; (C4, C5, C6)
- 2.2 Integração de entrevistas e de interações com o cliente real do produto de software; (C4, C5, C6)
- 2.3 Apresentação e discussão pública do produto de software desenvolvido. (C4, C5)

3. Orientação tutorial

- 3.1 Sessões de orientação pessoal, em pequenos grupos para conduzir o processo de aprendizagem e esclarecerem-se dúvidas (C1, C2, C3, C6)

### Bibliografia:

#### Recomendada:

Apresentações fornecidas nas aulas teóricas e práticas.

I. Sommerville, Software Engineering, 10th edition, Pearson Education Limited, 2015.

K. Beck e C. Andres, Extreme Programming Explained: Embrace Change, Second Edition, Addison-Wesley Professional, 2004

E. Gamma, R. Helm, R. Johnson e J. Vlissides, Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley Professional, 1994.

#### Complementar:

IEEE Computer Society, SWEBOK - Software Engineering Body of Knowledge, 2004.

R. Craig e S. Jaskiel, Systematic Software Testing, Artech House Publishers, 2002.

T. Pigoski, Practical Software Maintenance: Best Practices for Managing Your Software Investment, Wiley, 1996. Leon, Software Configuration Management Handbook, Second Edition, Artech House Publishers, 2004.

PMI, Project Management Body of Knowledge (PMBok), Fifth Edition, 2013