

Caraterização da Unidade Curricular / Characterisation of the Curricular Unit

Designação da Unidade Curricular / Curricular Unit:	[31821488] Engenharia de Software II [31821488] Software Engineering II		
Plano / Plan:	Licenciatura em Engenharia Informática V2		
Curso / Course:	Licenciatura em Engenharia Informática Computer Sciences Engineering		
Grau / Diploma:	Licenciado		
Departamento / Department:	Departamento de Informática		
Unidade Orgânica / Organic Unit:	Escola Superior de Tecnologia e Gestão de Viseu		
Área Científica / Scientific Area:	Ciências Informáticas		
Ano Curricular / Curricular Year:	3		
Período / Term:	S2		
ECTS:	6		
Horas de Trabalho / Work Hours:	0159:00		
Horas de Contacto/Contact Hours:			
(T) Teóricas/Theoretical:	0019:30	(TC) Trabalho de Campo/Fieldwork:	0000:00
(TP) Teórico-Práticas/Theoretical-Practical:	0000:00	(OT) Orientação Tutorial/Tutorial Orientation:	0000:00
(P) Práticas/Practical:	0052:00	(E) Estágio/Internship:	0000:00
(PL) Práticas Laboratoriais/Practical Labs:	0000:00	(O) Outras/Others:	0000:00
(S) Seminário/Seminar:	0000:00		

Docente Responsável / Responsible Teaching

[3047] Carlos Augusto Da Silva Cunha

Docentes que lecionam / Teaching staff

[3047] CARLOS AUGUSTO DA SILVA CUNHA

[3326] LUIS CARLOS LOPES SOARES

[3337] PEDRO MIGUEL DE OLIVEIRA MARTINS

Objetivos de Aprendizagem

O1. Desenvolver software baseado em padrões.

O2. Desenhar e implementar testes de software.

Learning Outcomes of the Curricular Unit

O1. Pattern-Oriented software architecture.

O2. Test-driven design and development.

Conteudos Programáticos (Lim:1000)

- 1 Padrões de Desenho
 - 1.1 Padrões de Criação
 - 1.2 Padrões Comportamentais
 - 1.3 Padrões Estruturais
- 2 Fundamentos de Testes de Software
 - 2.1 Testes White-box
 - 2.2 Testes Black-box
 - 2.3 Test-driven Development
- 3 Testes de Unidade
 - 3.1 Testes de Interface
 - 3.2 Testes de Estruturas de Dados
 - 3.3 Testes de Condições Limite
 - 3.4 Testes de Caminhos de Execução
 - 3.5 Testes de Caminhos de Tratamento de Erros
- 4 Testes de Integração
 - 4.1 Testes *Bottom-Up*
 - 4.2 Testes *Top-Down*
 - 4.3 *Stubs* , *Mocks* e *Drivers*
- 5 Testes de Sistema
 - 5.1 Testes de Performance: Robustez, Disponibilidade e Capacidade
 - 5.2 Testes de Aceitação
 - 5.3 Testes de Instalação

Syllabus (Lim:1000)

- 1 Design Patterns
 - 1.1 Creational Patterns
 - 1.2 Behavioural Patterns
 - 1.3 Structural Patterns
- 2 Software Testing Fundamentals
 - 2.1 White-box Testing
 - 2.2 Black-box Testing
 - 2.3 Test-driven Development
- 3 Unit Testing
 - 3.1 Interface Testing
 - 3.2 Data Structure Testing Limit Conditions Testing
 - 3.3 Limit Conditions Testing
 - 3.4 Execution Paths Testing
 - 3.5 Error Handling Paths Testing
- 4 Integration Testing
 - 4.1 Bottom-up testing
 - 4.2 Top-down testing
 - 4.3 Stubs, Mocks and Drivers
- 5 System-level Testing
 - 5.1 Performance Testing: Robustness, Availability and Capacity
 - 5.2 Acceptance Tests
 - 5.3 Deployment Tests

Metodologias de Ensino (Avaliação incluída; Lim:1000)

Aulas de caráter essencialmente prático, com apelo constante à participação, desenvolvimento do espírito crítico e de iniciativa, e procura da excelência nos trabalhos realizados.

Ponderação, em todas as épocas de avaliação:

- **trabalhos práticos** (35%, mínimos: 10/20);
- **prova escrita** (50%, mínimos: 10/20);
- **tarefas resolvidas presencialmente em aula** (15%, mínimos para a Época Normal: 12/20).

Parte da classificação dos trabalhos práticos resulta da sua avaliação contínua (apenas em Época Normal). Todos os trabalhos são passíveis de defesa individual.

Teaching Methodologies (Including evaluation; Lim:1000)

The theoretical classes are expositive. The materials cover all learned content and present illustrative examples.

Students apply the theoretical concepts learned in theoretical classes to practical scenarios in the practical classes. In these classes, we use a set of worksheets, which describe the tasks that students must perform and contain a description of the theoretical concepts needed to solve the presented situations.

Evaluation is based on written tests and practical works. The final mark is calculated as follows:

- Written Test: 50% (minimum of 8 points)
- Practical Works: 35% (minimum of 10 points)
- Assessment worksheets: 15% (Normal Epoch: minimum of 12 points in the assessment worksheets)

Bibliografia de Consulta (Lim:1000)

- Roger S. Pressman, (2005) Software engineering : A practitioner's approach, Boston, McGraw-Hill, (004.41 PRE)
- Gerreiro, S. (2015) Introdução à Engenharia de Software, FCA Editora de Informática, ISBN: 978-972-722-795-2
- Mauro Nunes & Henrique O Neill, (2001) Fundamental de UML, FCA Editora de Informática, (004.41 NUN)
- Spillner, A. et al. (2014). Software Testing Foundations: A Study Guide for the Certified Tester Exam (Rocky Nook Computing), Rocky Nook, ISBN: 978-1937538422
- Liu, H. (2009). Software Performance and Scalability, John Wiley & Sons, ISBN: 978-0470462539
- Molyneaux, I. (2014). The Art of Application Performance Testing: From Strategy to Tools, O'Reilly, ISBN: 978-1491900543
- Stuttard, D. (2011). The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws, John Wiley & Sons, ISBN: 978-1118026472
- Wysopal, C. et al. (2006). The Art of Software Security Testing: Identifying Software Security Flaws, Addison Wesley, ISBN: 978-0321304865

Bibliography (Lim:1000)

- Roger S. Pressman, (2005) Software engineering : A practitioner's approach, Boston, McGraw-Hill, (004.41 PRE)
- Gerreiro, S. (2015) Introdução à Engenharia de Software, FCA Editora de Informática, ISBN: 978-972-722-795-2
- Mauro Nunes & Henrique O Neill, (2001) Fundamental de UML, FCA Editora de Informática, (004.41 NUN)
- Spillner, A. et al. (2014). Software Testing Foundations: A Study Guide for the Certified Tester Exam (Rocky Nook Computing), Rocky Nook, ISBN: 978-1937538422
- Liu, H. (2009). Software Performance and Scalability, John Wiley & Sons, ISBN: 978-0470462539
- Molyneaux, I. (2014). The Art of Application Performance Testing: From Strategy to Tools, O'Reilly, ISBN: 978-1491900543
- Stuttard, D. (2011). The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws, John Wiley & Sons, ISBN: 978-1118026472
- Wysopal, C. et al. (2006). The Art of Software Security Testing: Identifying Software Security Flaws, Addison Wesley, ISBN: 978-0321304865

Observações

«Observações»

Observations

«Observations»

Observações complementares