

Ficha de Unidade Curricular

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores
Laboratório de Programação
Ocorrência de: 2012/2013

[Página oficial](#)

Informação Geral

Unidade curricular: Laboratório de Programação
Código: EEC0030
Curso: MIEEC, 4º, 49 estudantes (em 04.10.2012)
Ano letivo: 2012/2013
Semestre: 1S
Créditos: 6 ECTS
Horas/Semanas: 1x2T, 3x2P
Professores: [João Correia Lopes](#)

Língua de Ensino

Suitable for English-speaking students

Objetivos, Competências e Resultados de Aprendizagem

1. ENQUADRAMENTO

Especificação, desenvolvimento e manutenção de aplicações de software com interface gráfica em arquiteturas cliente/servidor.

2. OBJETIVOS ESPECÍFICOS

Esta unidade curricular tem por objetivos:

- Promover a aquisição de conceitos, métodos e técnicas de Engenharia de Software e dotar os estudantes da capacidade de os aplicar na conceção e desenvolvimento de sistemas de software.
- Dotar os estudantes de conhecimentos práticos na utilização de ferramentas de desenvolvimento de software adequadas à metodologia a usar e que permitam o acompanhamento do desenvolvimento do produto durante todo o seu ciclo de vida, incluindo a depuração, teste e documentação de código na linguagem de programação Java.

3. PRÉ-REQUISITOS

Conhecimentos em Programação Orientada por Objetos.

4. DISTRIBUIÇÃO

Componente Científica: 40%
Componente Tecnológica 60%

5. RESULTADOS DA APRENDIZAGEM

Ao completar esta unidade curricular o estudante deve ser capaz de:

- Identificar e documentar os requisitos de um Sistema de Software utilizando "histórias de utilizador"
- Descrever os casos de utilização utilizando UML
- Realizar um protótipo de interface com o utilizador
- Identificar e documentar os requisitos suplementares
- Obter o modelo conceptual do domínio utilizando UML
- Obter modelos de processos de negócio utilizando UML
- Descrever a arquitetura utilizando UML
- Validar a arquitetura através de um protótipo vertical
- Modelar o comportamento de objetos utilizando UML
- Modelar a estrutura de classes utilizando UML
- Elaborar a Ajuda ao Utilizador
- Elaborar o Manual de Instalação
- Codificar classes em Java usando as API standard
- Realizar versões de componentes de software
- Documentar o código Java utilizando Javadoc
- Testar o código utilizando Junit
- Utilizar uma ferramenta de elaboração colaborativa de documentação
- Utilizar um IDE na manutenção do software
- Utilizar um sistema de controlo de versões

Programa

- Introdução à Engenharia de Software e à modelação ágil.
- Introdução à Engenharia de requisitos. Documentação de requisitos.
- Linguagem de modelação UML.
- Projeto de software orientado por objetos. Modelação de arquitetura, estrutura de classes e comportamento.
- Projeto de interfaces.
- Codificação em Java.
- Verificação, validação e teste de software.
- Manutenção de software. Gestão de configurações. Controlo de versões.
- Gestão e planeamento de projetos.

Bibliografia Principal

- Scott Ambler, *The Object Primer*, Cambridge University Press, 3rd Edition, 2004, ISBN: 978-0-521-54018-6 [Biblioteca](#)
- Bruce Eckel, *Thinking in Java*, Prentice Hall, 4th Edition, 2006, ISBN: 0131-87248-6 [Biblioteca](#)

Bibliografia Complementar

- Alberto Manuel Rodrigues da Silva e Carlos Alberto Escaleira Videira, *UML, metodologias e ferramentas CASE*, 2ª Edição, Volume 1, Maio 2005, Centro Atlântico Editora, ISBN: 989-615-009-5 [Biblioteca](#)
- Russ Miles e Kim Hamilton, *Learning UML 2.0*, O'Reilly, 2006, ISBN: 978-0-596-00982-3 [Biblioteca](#)
- Ian Sommerville, *Software engineering*, Addison-Wesley, 9th Edition, 2010, ISBN: 0-321-31379-8 [Biblioteca](#)
- Mauro Nunes e Henrique O'Neill, *Fundamental de UML*, 3ª edição, FCA - Editora Informática, 2004, ISBN: 978-972-722-481-4 [Biblioteca](#)

Métodos de Ensino

A unidade curricular tem aulas teóricas (2 horas por semana) e aulas laboratoriais (2 horas por semana) em sala de computadores.

Os tempos letivos das aulas teóricas são usados para descrever brevemente as várias fases de desenvolvimento de software e os artefactos associados e para apresentar e exemplificar a utilização das metodologias e ferramentas a usar no trabalho prático. São ainda indicadas referências de leitura.

Nas aulas laboratoriais os estudantes trabalham, em grupos de 5, num projeto de software.

Software

- [Enterprise Architect](#) (Windows)
- [Umbrello](#) (Linux)
- [IDE NetBeans](#) (Linux e Windows)
- [Dokuwiki](#)
- [ArgoUML](#)
- [SVN](#)

Palavras-chave

Ciências Físicas > Ciência de computadores > Programação

Ciências Físicas > Ciência de computadores > Programação > Engenharia de software

Modo de Avaliação

Avaliação distribuída sem exame final

Componentes de Avaliação e Ocupação registadas

Descrição	Tipo	Tempo (horas)	Data de Conclusão
Participação presencial (estimativa)	Aulas	56	
TP1: Protótipo de Interfaces com o Utilizador	Trabalho laboratorial ou de campo	10	2012-10-02
TP2: Especificação de Requisitos	Trabalho laboratorial ou de campo	7	2012-10-16
TP3: Projeto de Alto Nível	Trabalho laboratorial ou de campo	14	2012-11-06
TP4: Protótipo	Trabalho laboratorial ou de campo	18	2012-11-13
TP5: Projeto Detalhado	Trabalho laboratorial ou de campo	19	2012-12-04
TP6: Produto	Trabalho laboratorial ou de campo	30	2012-12-11
TP7: Apresentação do Produto	Participação Presencial	2	2012-12-14
TP8: Desempenho Individual	Trabalho		
FT1: Modelo de Casos de Utilização	Exercício	1,5	2012-10-08
FT2: Modelo Conceptual do Domínio	Exercício	1,5	2012-10-29
FT3: Modelo de Arquitetura	Exercício	1,5	2012-11-19
FT4: Java	Exercício	1,5	2012-11-26

Descrição	Tipo	Tempo (horas)	Data de Conclusão
		Total: 162	

Obtenção de Frequência

O trabalho prático (TP) será avaliado através da documentação apresentada, da aplicação desenvolvida e do desempenho individual nas aulas (TP1 a TP8).

Os conceitos teóricos são avaliados através da resposta individual a folhas de perguntas (FT1 a FT4), a realizar em algumas aulas laboratoriais.

A aprovação na unidade curricular está condicionada à obtenção de 50% em cada um dos trabalhos de avaliação prática (TP1 a TP8) e de 40% na média das folhas de perguntas relacionadas com a matéria teórica (FT1 a FT4).

Esta unidade curricular, dada a sua natureza laboratorial, não é passível de avaliação em momento único, pelo que não poderá ser substituída por exame de recurso.

Cálculo da Classificação Final

A nota final será calculada usando a fórmula: $NOTA = 80\% TP + 20\% FT$

sendo:

$$TP = (TP1 + 3* TP2 + 2* TP3 + 2* TP4 + 2* TP5 + 8* TP6 + TP7 + TP8) / 20$$

$$FT = (FT1 + FT2 + FT3 + FT4) / 4$$

A classificação da componente prática (TP) pode variar de elemento para elemento do mesmo grupo em mais ou menos 2 valores, com base na opinião dos docentes e na autoavaliação a realizar internamente em cada grupo.

Provas e Trabalhos Especiais

Após a demonstração do trabalho prático, pode ter lugar uma prova oral para estudantes em que seja difícil avaliar a participação no trabalho prático em grupo.

Avaliação Especial (TE, DA, ...)

A avaliação prática e teórica, realizada nas aulas laboratoriais durante o semestre de funcionamento da unidade curricular, é exigida a todos os estudantes, independentemente do regime de inscrição e da necessidade de obter avaliação de frequência.

Os trabalhadores estudantes e equivalentes dispensados das aulas devem, com periodicidade a combinar com os docentes, apresentar a evolução dos seus trabalhos, assim como devem fazer a apresentação destes, simultaneamente com os estudantes ordinários, e realizar as provas teóricas e práticas de avaliação individual previstas.

Melhoria de Classificação Final/Distribuída

A nota final pode ser melhorada apenas através da frequência de numa nova edição da unidade curricular.

Observações

— JCL

From:

<https://paginas.fe.up.pt/~jlopes/> - JCL

Permanent link:

<https://paginas.fe.up.pt/~jlopes/doku.php/teach/lpro/201213/ficha>

Last update: **02/09/2013 18:25**

