

Informática e Tecnologias Multimédia

Licenciatura, 1º Ciclo

Plano: Despacho n.º 12419/2016 - 14/10/2016

Ficha da Unidade Curricular: Programação Orientada por Objetos

ECTS: 6; Horas - Totais: 162.0, Contacto e Tipologia, TP:30.0; PL:45.0; OT:5.0;

Ano|Semestre: 2|S1; Ramo: Tronco comum;

Tipo: Obrigatória; Interação: Presencial; Código: 814312

Área Científica: Informática

Docente Responsável

Sandra Maria Gonçalves Vilas Boas Jardim

Docente e horas de contacto

Fernando Sérgio Hortas Rodrigues - Equiparado Assistente 1º Triénio: TP: 30 + PL: 45

Sandra Maria Gonçalves Vilas Boas Jardim - Professora Adjunta: OT: 5

Objetivos de Aprendizagem

Proporcionar uma introdução à Análise e Desenho Orientados ao Objeto e metodologias iterativas e evolutivas, através do Unified Process e UML e ainda introduzir a utilização de padrões de desenho. A linguagem Java é utilizada nos tópicos referidos quando necessário. Visão geral das metodologias XP e SCRUM.

Conteúdos Programáticos

Metodologias Ágeis XP and SCRUM; Unified Modeling Language; Análise e desenho orientados por objetos; Unified Process; Padrões de Desenvolvimento.

Conteúdos Programáticos (detalhado)

1 - Categorias de Metodologias de Desenvolvimento de Software

- Categorias e características das metodologias de desenvolvimento de software
- Visão geral de metodologias de desenvolvimento de software populares: Waterfall, Scrum, Extreme Programming, Unified Process

2 – Desenho e Análise orientados por objetos (OO)

- O que é a Análise e o Desenho
- Análise e Desenho orientados por objetos
- A Modelação visual - UML

3 – A UML - Unified Modeling Language

- Princípios da Modelação
- Perspetivas de Arquiteturas orientadas a objetos
- Formas de utilização da UML
- Modelo conceptual da UML
- Os vários tipos de Entidades, Relações e Mecanismos Comuns
- Os diagramas da UML

- 4 – Processos Iterativos, Evolutivos e Ágeis
 - Características e vantagens do desenvolvimento Iterativo e Evolutivo
 - O Processo Unificado (UP)
 - Fases, Iterações e Milestones
 - As disciplinas

- 5 – Fase de Conceção (Inception)
 - Características
 - Duração
 - Artefactos que se iniciam nesta fase

- 6 – Requisitos Evolutivos
 - O que são Requisitos
 - Categorias de Requisitos
 - Requisitos Evolutivos vs. Requisitos do tipo Waterfall
 - Como encontrar requisitos

- 7 – Casos de Uso (UCs)
 - Atores, Cenários e Casos de Uso
 - Para que servem os Use Cases
 - Formatos de UCs (Resumido, Casual, Detalhado)
 - Evolução dos UCs nas várias iterações

- 8 – Fase de Elaboração
 - As várias iterações da fase de Elaboração
 - Os UCs durante as várias iterações
 - Artefactos iniciados na fase de elaboração

- 9 – Modelos de Domínio
 - O que são e para que servem
 - Classes conceptuais
 - Métodos para encontrar classes conceptuais
 - Atributos
 - Atributos vs Classes
 - Modelar com Classes de Descrição
 - Associações
 - Métodos para encontrar associações

- 10 – Diagramas de Sequência do Sistema (DSSs)
 - O que são e para que servem
 - Relação entre DSSs e Use Cases
 - Nomeação de Eventos de Sistema e Operações
 - Contractos de Operação e suas características
 - Como criar e escrever Contratos de Operação
 - Os contratos de operação no contexto do UP

- 11 – Requisitos para o Desenho
 - Motivação para a passagem para as atividades de desenho
- 12 – Diagramas de Interação

- Notação dos Diagramas de Sequência e Comunicação
- Notação dos Diagramas de Interação

13 – Diagramas de Classe

- O que são e para que servem
- Notação dos diagramas de classe
- Diagramas de Classes de Desenho
 - Classificadores
 - Como mostrar atributos no diagrama de classes
 - Anotações
 - Operações e Métodos
 - Estereótipos, *Profiles* e *Tags*
 - Propriedades,
 - Generalização, classes e métodos abstratos
 - Dependências
 - Interfaces
 - Composição e Agregação
 - *Constraints*
 - Associação qualificada
 - Classe de associação
 - Classe *Singleton*
 - Compartimentos definidos pelo utilizador
 - Relação entre diagramas de interação e digramas de classes

14 – GRASP – General *Responsibility Assignment Software Patterns*

- Desenho de objetos com responsabilidades
- UML vs. Princípios de desenho
- Desenho de objetos: Entradas, Atividades e Saídas
- Responsabilidades e Orientação por Responsabilidade no desenho
- Metodologia GRASP para desenho OO
- Relações entre GRASP, Responsabilidades e UML
- Padrões
 - O que são e para que servem
 - Aplicação do GRASP ao desenho de objetos
 - *Creator*
 - *Expert*
 - Low Coupling
 - Controller
 - High Cohesion

Metodologias de avaliação

Época de Frequência:

Trabalhos Práticos (20%)

Projeto Prático (80%)

Os alunos poderão ser convocados para discussão do Projeto Prático. A falta à discussão implica a atribuição da nota de zero (0) valores e conseqüente reprovação na época correspondente.

Épocas de Exame, Recurso, Especial e Trabalhador-Estudante:

Trabalhos Práticos (20%)

Projeto Prático (80%)

O Projeto tem discussão obrigatória. A falta à discussão implica a atribuição da nota de zero (0) valores e consequente reprovação na época correspondente.

Software utilizado em aula

Visual Paradigm

Bibliografia recomendada

- Larman, C. (2004). *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development*. -: Prentice Hall
- Booch, G. e Rumbaugh, J. e Jacobson, I. (2005). *The Unified Modeling Language User Guide*. -: Addison Wesley
- Nunes, M. e O'Neill, H. e Ramos, P. (2010). *Exercícios de UML*. Lisboa: Lidel

Coerência dos conteúdos programáticos com os objetivos

No Cap. 1 são tipificadas as várias MDS e são descritas genericamente as metodologias XP e SCRUM.

No Cap. 2 são apresentados os conceitos gerais de Análise e Desenho OO.

No Cap. 3 É apresentada a UML, em detalhe.

Do Cap. 4 ao Cap. 13 é apresentada a metodologia UP acompanhada de exemplos concretos.

No Cap. 14 é apresentada uma introdução aos Padrões de Software.

Metodologias de ensino

Aulas teórico-práticas expositivas onde se descrevem os conceitos fundamentais. Aulas práticas de resolução de casos práticos e aplicação dos conceitos a cenários de utilização real.

Coerência das metodologias de ensino com os objetivos

Aulas teórico-práticas expositivas onde se descrevem os conceitos fundamentais. Aulas práticas de resolução de casos práticos e aplicação dos conceitos a cenários de utilização real.

Língua de ensino

Português

Pré-requisitos

Conhecimento de uma linguagem de programação orientada a objetos.

Docente Responsável



Diretor de Curso, Comissão de Curso



Conselho Técnico-Científico

