

TeSP - Informática

Técnico Superior Profissional

Plano: Despacho n.º 8838/2020 de 14-09-2020 + Despacho n.º 3463/2023 de 16/03/2023

Ficha da Unidade Curricular: Programação Orientada por Objetos

ECTS: 6; Horas - Totais: 162.0, Contacto e Tipologia, TP:70.0;

Ano | Semestre: 2 | S1

Tipo: Obrigatória; Interação: Presencial; Código: 614221

Área de educação e formação: Ciências informáticas

Docente Responsável

Fernando Sérgio Hortas Rodrigues

Professor Adjunto

Docente(s)

Fernando Sérgio Hortas Rodrigues

Professor Adjunto

Mário Rui Betes Rodrigues Lopes

Assistente Convidado

Objetivos de Aprendizagem

Permitir a aprendizagem do paradigma de orientação a objetos através da linguagem de programação C#.

Objetivos de Aprendizagem (detalhado)

1. Adquirir conhecimentos sobre o paradigma da orientação por objetos, tais como abstração, encapsulamento, hereditariedade e polimorfismo.
2. Adquirir conhecimentos sobre a linguagem de programação C# e da .Net
3. Adquirir conhecimentos sobre métodos de persistência de dados, através da utilização de frameworks para acesso a bases de dados.

Conteúdos Programáticos

C#: .NET; Tipos; Expressões; Classes; Métodos; Construtores; Hereditariedade; Polimorfismo; Coleções; Interfaces; Exceções; Eventos; Delegações; Sobreposição de Métodos; Streams de ficheiros; ADO.NET; EF; GUI.

Conteúdos Programáticos (detalhado)

1. Conceitos gerais

- 1.1. Introdução à .NET
- 1.2. Exploração do ambiente de trabalho Visual Studio .NET
- 1.3. Criação de um Projeto de uma Aplicação Windows

2. Fundamentos da linguagem C#

- 2.1. Perceber os fundamentos da linguagem C#
- 2.2. Os vários tipos predefinidos do C#
- 2.3. Variáveis, Constantes e Enumerados
- 2.4. Arrays unidimensionais, multidimensionais e Jagged Arrays
- 2.5. Conversões implícitas e explícitas
- 2.6. Expressões e Operadores
- 2.7. Instruções Condicionais
- 2.8. Instruções Iterativas

3. Criar Objetos em C#

- 3.1. Value Types e Reference Types
- 3.2. Boxing e Unboxing
- 3.3. Classes e Objetos
 - 3.3.1. Sintaxe e Semântica
 - 3.3.2. Definição de Classes
 - 3.3.3. Criação e Inicialização de Objetos
- 3.4. Namespaces
- 3.5. Access Modifiers (Scope)
- 3.6. Métodos
 - 3.6.1. Sintaxe e Semântica
 - 3.6.2. Parâmetros, variáveis locais e scope
 - 3.6.3. Passagem de parâmetros por valor e por referência
 - 3.6.4. Overloading
- 3.7. Construtores
 - 3.7.1. Sintaxe e Semântica
 - 3.7.2. Overload de Construtores
 - 3.7.3. Inicialização de Construtores
- 3.8. Inicialização de Classes
- 3.9. Variáveis readonly
- 3.10. Membros de Classe Estáticos

4. Técnicas de Programação Orientadas a Objetos em C#

- 4.1. Programação Estruturada vs. Orientada por Objetos
- 4.2. Encapsulamento
- 4.3. Propriedades e Propriedades Automáticas

4.4. Hereditariedade

4.4.1. Classes Derivadas

4.4.2. Acesso a Construtores da Classe Base

4.4.3. Classes Seladas

4.4.4. Métodos Virtuais

4.4.5. Classes Abstratas

4.4.6. Propriedades Abstratas

4.5. Polimorfismo

5. Programar em C#

5.1. Instrução foreach

5.2. Indexação de Objetos (Indexers/Propriedades Indexadas)

5.3. Interfaces

5.4. Operadores is e as

5.5. Coleções

5.5.1. Implementação de Tipos Enumeráveis

5.5.1.1. A interface IEnumerable

5.5.1.2. Implementação explícita de IEnumerator

5.5.1.3. Iteradores Yield

5.5.1.4. Iteradores Nomeados (Named Iterators)

5.5.2. ArrayList

5.5.3. Queue

5.5.4. Stack

5.5.5. Hash Table

5.6. Exceções (Exception Handling)

5.7. Eventos e Delegações (Delegates and Events)

6. Implementação de Aplicações Baseadas em .NET, em C#

6.1. Sobreposição de Métodos (Overriding Methods) na biblioteca System.Object

6.2. Formatação de Cadeias de Caracteres e Números

6.3. Streams e ficheiros

7. Frameworks para Mapeamento Objeto-Relacional

7.1 Entity Framework (EF)

7.1.1 Introduction to EF

7.1.2 Comerçar pela Base de Dados

7.1.3 Comerçar pelo Código

7.1.4 Constraints, Chaves e Associações

7.1.5 Acesso a dados e operações CRUD

8. C# Avançado

8.1. Genéricos

8.2. Tipos anónimos

8.3. Métodos Anónimos

8.4. Lambdas

Metodologias de avaliação

Época de Frequência:

Testes (T1 e T2), com peso 20% cada

Trabalho Prático (TP), com peso 60%

$$\text{NotaFinal} = (T1*0,2 + T2*0,2 + TP*0,6)$$

Outras épocas:

Exame (E), com peso 40%

Trabalho Prático (TP), com peso 60% e apresentação-defesa obrigatória.

$$\text{NotaFinal} = (E*0,4 + nTP*0,6)$$

Obs:

- É condição necessária, mas não suficiente, a obtenção da nota mínima de 7 valores na componente de Testes/Exame e obtenção da nota mínima de 10 valores na componente de Trabalho Prático, caso contrário o aluno reprova nessa época de avaliação.

- O não cumprimento da condição necessária anterior, mesmo que a média final obtida seja maior ou igual a 9,5 valores implica a reprovação à UC nessa época de avaliação, sendo que neste caso a nota final que constará na pauta será de 9 valores.

- A obtenção de nota mínima na componente Teste/Exame em época de Frequência, em caso de reprovação nessa época, pode ser aproveitada para época Normal de Exame.

- A obtenção de nota mínima na componente Trabalho Prático em qualquer época, em caso de reprovação nessa época, pode ser aproveitada para a época de exame seguinte.

-A falta do aluno à apresentação-defesa implica a atribuição de zero (0) valores na componente Trabalho Prático.

- O aluno obtém aprovação à UC, de acordo com o disposto nos Pontos 11 e 12 do Artigo 11º do Regulamento Académico do IPT

Software utilizado em aula

Visual Studio

MS Sql Server

C# Jupyter Notebooks

Estágio

Não aplicável.

Bibliografia recomendada

- Japikse, P. e Troelsen, A. (2022). *Pro C# 10 with .NET 6: Foundational Principles and Practices*

in Programming.. Apress. 11th Ed., ISBN: 9781484278680
- Loureiro, H. (2017). *C# 7.0 com Visual Studio..* FCA - Editora de Informática. ISBN: 9789727228683
- Smith, J. (2021). *Entity Framework Core in Action..* Manning Publications Co.. 2nd Ed., ISBN: 9781617298363

Coerência dos conteúdos programáticos com os objetivos

Objetivo 1: Capítulos 3 a 6 e capítulo 8;

Objetivo 2: Capítulo 1 a 6 e capítulo 8;

Objetivo 3: Capítulo 7.

Metodologias de ensino

Aulas teóricas expositivo-participativas onde se descrevem e discutem com os alunos os conceitos fundamentais. Aulas práticas de resolução de: casos práticos; exercícios; aplicação dos conceitos apreendidos a cenários de utilização real.

Coerência das metodologias de ensino com os objetivos

Aulas teórico-práticas expositivas e participativas onde se descrevem e discutem com os alunos, os conceitos fundamentais. Aulas práticas de resolução de casos práticos e/ou exercícios com exemplos mais complexos e completos, para uma consolidação profunda dos conceitos transmitidos e ainda a aplicação dos conceitos apreendidos a cenários de utilização real em ambiente simulado.

Língua de ensino

Português

Pré-requisitos

Considera-se importante que os alunos tenham conhecimentos de programação estruturada e algoritmia básica.

Programas Opcionais recomendados

Não aplicável.

Observações

Objetivos de Desenvolvimento Sustentável:

- 4 - Garantir o acesso à educação inclusiva, de qualidade e equitativa, e promover oportunidades de aprendizagem ao longo da vida para todos;
 - 9 - Construir infraestruturas resilientes, promover a industrialização inclusiva e sustentável e fomentar a inovação;
 - 10 - Reduzir as desigualdades no interior dos países e entre países;
-

Docente responsável
