

| | |
|---|---|
| LICENCIATURA: Engenharia Informática | ÁREA CIENTÍFICA: Engenharia Informática |
| UNIDADE CURRICULAR/CURRICULAR UNIT: Programação II / Programming II | ECTS: 6 |
| DURAÇÃO: Semestral | HORAS DE CONTACTO TEÓRICO PRÁTICAS: 60 (48 TP+12 OT) |

OBJETIVOS DE APRENDIZAGEM / LEARNING OUTCOMES OF THE CURRICULAR UNIT

Para concluir com sucesso esta unidade curricular, os estudantes deverão demonstrar possuir os seguintes conhecimentos e capacidades:

1. Estabelecer um paralelismo entre uma aplicação tradicional vs. aplicações implementadas numa abordagem orientada a objetos.
2. Enumerar as vantagens da utilização do paradigma P.O.O.
3. Enumerar as características de uma linguagem O.O.
4. Formular e Modelar um problema numa abordagem O.O., incluindo o número, cardinalidade e tipo de relações entre as entidades (classes) envolvidas.
5. Definir e distinguir corretamente os mecanismos de Encapsulamento, Herança, Polimorfismo, Overriding e Overloading
6. Utilizar corretamente os mecanismos de Encapsulamento, Herança, Polimorfismo (incluindo Overriding) e Overloading ao desenhar e criar classes.
7. Interpretar um modelo em UML e construir diagramas básicos “use case” e “classes”
8. Utilizar corretamente o editor/compilador/depurador do C#.
9. Utilizar todos elementos do C# (dados, operadores, estruturas de decisão e repetição, funções, etc.)
10. Utilizar características importantes do C#, tais como Captura e Gestão de Exceções, Ficheiros e Streams
11. Criar aplicações em C# para Consola e WPF.
12. Entender e utilizar o conceito de RIA (Rich Internet Application)
13. Utilizar uma base de dados simples em SQL (SQL Server ou Access)

(English)

To successfully complete this curricular unit, students must demonstrate the following knowledge and skills:

1. Establish a parallelism between a traditional application vs. applications implemented in an object-oriented approach.
2. Enumerate the advantages of using the P.O.O. paradigm.
3. Enumerate the characteristics of a P.O.O. language.

4. Formulate and Model a problem in a P.O.O. approach, including the number, cardinality and type of relations between the entities (classes) involved.
5. Define and distinguish correctly the mechanisms of Encapsulation, Inheritance, Polymorphism, Overriding and Overloading
6. Use correctly the mechanisms of Encapsulation, Inheritance, Polymorphism (including Overriding) and Overloading when designing and creating classes.
7. Interpret a UML model and build basic "use case" and "classes" diagrams
8. Use correctly the C# editor/compiler/debugger.
9. Use all C# elements (data, operators, decision and repetition structures, functions, etc.)
10. Use important C# features such as Exception Capture and Management, Files and Streams
11. Create C# applications for Console and WPF.
12. Understand and use the concept of RIA (Rich Internet Application)
13. Use a simple SQL database (SQL Server or Access)

CONTEÚDOS PROGRAMÁTICOS / SYLLABUS

1. A linguagem C#
2. O ambiente de desenvolvimento Framework.NET
3. Conceitos de programação orientada a objetos – Classes, Herança, Polimorfismo
4. Modelação simples de aplicações orientadas a objetos
5. Detecção e correção de Exceções
6. Desenho e Implementação de Interfaces Gráficas
7. A tecnologia WPF – Windows Presentation Foundation – em aplicações multimédia
8. A linguagem XAML – Extensible Application Markup Language
9. Manipulação de dados, objetos e contentores WPF
10. Aplicações RIA em WPF
11. Ligação a Bases de Dados SQL

(English)

1. The C# language
2. The Framework.NET development environment
3. Concepts of object oriented programming - Classes, Inheritance, Polymorphism
4. Simple modeling of object-oriented applications
5. Exception detection and correction
6. Design and Implementation of Graphical Interfaces
7. The WPF - Windows Presentation Foundation - technology in multimedia applications

8. The XAML - Extensible Application Markup Language
9. Manipulation of WPF data, objects and containers
10. RIA applications in WPF
11. Linking to SQL Databases

**DEMONSTRAÇÃO DA COERÊNCIA DOS CONTEÚDOS PROGRAMÁTICOS COM OS OBJETIVOS DA UNIDADE CURRICULAR/
DEMONSTRATION OF THE SYLLABUS COHERENCE WITH THE CURRICULAR UNIT'S OBJECTIVES**

Nesta unidade curricular serão ministrados conteúdos que permitam ao estudante utilizar de forma eficiente a linguagem de programação C#, transformar problemas propostos de forma tradicional em abordagens orientadas a objetos e desenvolver competências no domínio da modelação e implementação de programas Orientados a Objetos, incluindo representação de um sistema simples em UML. Para o efeito, serão abordados os conceitos de programação orientada a objetos, nomeadamente o encapsulamento, a herança e o polimorfismo (pontos 1 a 3), serão introduzidos os conceitos de modelação de aplicações orientadas a objeto e respetiva representação na forma de modelos UML (ponto 4), bem como a utilização de interfaces gráficas simples, baseadas em XAML/WPF (pontos 6 a 8). A unidade será concluída com uma breve revisão de SQL e ilustrar-se-á a ligação entre a aplicação e a base de dados (pontos 9 a 11). Para consolidação teórica e prática, será desenvolvida na aula, uma aplicação em arquitetura 3-Tier.

(English)

In this course unit the students will be taught contents that allow them to use efficiently the C# programming language, transform problems proposed in a traditional way in object-oriented approaches and develop skills in modeling and implementing object-oriented programs, including representation of a simple system in UML. To this end, the concepts of object-oriented programming will be covered, namely encapsulation, inheritance and polymorphism (sections 1 to 3), the concepts of modelling object-oriented applications and their representation in the form of UML models will be introduced (section 4), as well as the use of simple graphical interfaces, based on XAML/WPF (sections 6 to 8). The unit will be concluded with a brief review of SQL and will illustrate the connection between the application and the database (sections 9 to 11). For theoretical and practical consolidation, an application in 3-Tier architecture will be developed in class.

METODOLOGIAS DE ENSINO E AVALIAÇÃO / TEACHING METHODOLOGIES INCLUDING EVALUATION

As aulas assumirão um carácter teórico-prático e decorrerão numa sala preparada com software e equipamento informático. Serão lecionadas num contexto baseado em projetos práticos, com demonstrações constantes.

De acordo com o Regulamento de Funcionamento do ISTECS Porto a avaliação é efetuada através de um exame final obrigatório. Na classificação final, poderão ser considerados elementos de avaliação contínua, tais como testes, trabalhos individuais ou em grupo, assim como a participação nas aulas presenciais e com recursos de aprendizagem proporcionados por sistemas de e-learning.

O estudante que realize os trabalhos práticos propostos nas aulas e nas condições aprovadas, poderá prescindir da realização da Prova prática final.

(English)

The classes will assume a theoretical-practical character, and will take place in a room prepared with software and computer equipment. They will be taught in a context based on practical projects, with constant demonstrations.

According to the ISTEC Porto Operating Regulations, the evaluation is carried out through a mandatory final examination. In the final classification, elements of continuous evaluation may be considered, such as tests, individual or group work, as well as participation in face-to-face classes and learning resources provided by e-learning systems.

The student who performs the practical work proposed in the classes and in the approved conditions, may dispense with the completion of the final practical test.

**DEMONSTRAÇÃO DA COERÊNCIA DAS METODOLOGIAS DE ENSINO COM OS OBJETIVOS DA UNIDADE CURRICULAR /
DEMONSTRATION OF THE COHERENCE BETWEEN THE TEACHING METHODOLOGIES AND THE LEARNING OUTCOMES**

A metodologia usada na unidade curricular visa contribuir para que o estudante aprofunde os seus conhecimentos e capacidades no domínio do desenvolvimento de aplicações orientadas a objeto em arquiteturas 3-Tier, com utilização de base de dados SQL, culminando com o desenvolvimento de trabalhos de grupo e individuais.

Dado ser um paradigma de programação prevalente com muita relevância no mercado de trabalho, e pelo facto de introduzir conceitos avançados de desenvolvimento de aplicações desktop, é crítico dotar os estudantes da licenciatura de engenharia multimédia, do tipo de conhecimentos e competências necessário nestas tecnologias . Importa, assim, que as aulas tenham um carácter teórico-prático que garanta o conhecimento sustentado e a sua aplicação concreta em contextos práticos.

(English)

The methodology used in the curricular unit aims to contribute to the student's deepening their knowledge and skills in the development of object-oriented applications in 3-Tier architectures, using SQL database, culminating with the development of group and individual work.

Given that it is a prevalent programming paradigm with much relevance in the labour market, and due to the fact that it introduces advanced concepts of desktop application development, it is critical to endow students of the multimedia engineering degree with the type of knowledge and skills required in these technologies . It is therefore important that the classes have a theoretical and practical character that ensures sustained knowledge and its concrete application in practical contexts.

BIBLIOGRAFIA / BIBLIOGRAPHY

FUNDAMENTAL / ESSENTIAL:

Chowdhury, K. (2018). *Windows Presentation Foundation Development Cookbook*. Packt

Hamilton, K., & Miles, R. (2006). *Learning UML 2.0*. O'Reilly Media.

COMPLEMENTAR/ COMPLEMENTARY:

Wikibooks. (2023). *C#*. http://en.wikibooks.org/wiki/C_Sharp_Programming

Wikibooks. (2023). *P.O.O.* http://en.wikibooks.org/wiki/Object_oriented_programming

INTERNET:

Acesso a publicações da especialidade, gratuitamente, através da rede SPRINGER:

<https://link.springer.com/>