

<b>LICENCIATURA:</b> Engenharia Multimédia	<b>ÁREA CIENTÍFICA:</b> Engenharia Informática
<b>UNIDADE CURRICULAR/CURRICULAR UNIT:</b> Programação de Computadores III / Computer Programming III	<b>ECTS:</b> 6
<b>DURAÇÃO:</b> Semestral	<b>HORAS DE CONTACTO TEÓRICO PRÁTICAS:</b> 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. Conhecer e saber instalar o ambiente de desenvolvimento XCode;
2. Saber usar a interface playground para desenvolver algoritmos;
3. Conhecer os vários tipos de dados da linguagem Swift e saber criar e usar variáveis e constantes;
4. Conhecer os operadores da linguagem e saber usar expressões;
5. Saber usar arrays unidimensionais e multidimensionais;
6. Compreender o conceito de set e saber executar operações com sets;
7. Saber declarar e aceder a dictionaries;
8. Compreender a estrutura e saber utilizar os statements de decisão e de repetição;
9. Saber utilizar funções na linguagem Swift;
10. Compreender o significado e saber utilizar enumerations;
11. Compreender os conceitos associados e saber utilizar classes e objetos;
12. Saber desenvolver aplicações (“apps”) de complexidade moderada e avançada no sistema operativo iOS, usando a linguagem Swift.
13. Desenvolva interfaces utilizando Storyboards.
14. Desenvolva interfaces utilizando SwiftUI.
15. Navegação entre ecrã.
16. Aceder a APIs utilizando a linguagem Swift.
17. Realizar projetos de trabalho em grupo que permitam desenvolver competências e atitudes de responsabilização, de solidariedade e de cooperação;
18. Utilizar, de forma criativa e crítica, conhecimentos, capacidades e atitudes na resolução de problemas concretos no âmbito dos conteúdos programáticos.

(English)

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

1. Know and know how to install the XCode development environment;
2. Know how to use the playground interface to develop algorithms;
3. Know the various types of data in Swift language and know how to create and use variables and constants;
4. Know the language operators and know how to use expressions;
5. Know how to use one-dimensional and multidimensional arrays;
6. Understand the concept of set and know how to perform operations with sets;
7. Know how to declare and access dictionaries;
8. Understand the structure and know how to use decision and repetition statements;
9. Know how to use functions in Swift language;
10. Understand the meaning and know how to use enumerations;
11. Understand the associated concepts and know how to use classes and objects;
12. Know how to develop applications (“apps”) of moderate and advanced complexity on the iOS operating system, using the Swift language;
13. Develop interfaces using Storyboards;
14. Develop interfaces using SwiftUI;
15. Navigation between screens;
16. Access APIs using the Swift language;
17. Carry out group work projects that enable the development of skills and attitudes of responsibility, solidarity and cooperation;
18. Use, creatively and critically, knowledge, skills and attitudes in solving concrete problems within the scope of the syllabus.

#### CONTEÚDOS PROGRAMÁTICOS / SYLLABUS

1. Desenvolvimento de aplicações para iOS – Uma visão geral
  - 1.1. Tecnologias para desenvolvimento de aplicações iOS
  - 1.2. Obter e instalar o Xcode e o iOS SDK
  - 1.3. O Apple Developer Program
  - 1.4. O Xcode
2. Playground user interface
  - 2.1. Criar um primeiro programa (com playground)
  - 2.2. Introduzir código na área de edição
  - 2.3. As áreas do playground IDE
3. Variáveis e constantes
  - 3.1. Declarar variáveis e constantes

- 3.2. Type annotations
- 3.3. Atribuição de nomes às variáveis
- 3.4. Imprimir strings e variáveis
- 3.5. Optionals
- 3.6. Exemplo – Declaração e utilização de variáveis (playground)
- 4. Tipos de dados
  - 4.1. Integers
  - 4.2. Floating-point
  - 4.3. Conversões entre Integers e Floating-points
  - 4.4. Booleans
  - 4.5. Tuples
- 5. Operadores
  - 5.1. Operador de atribuição
  - 5.2. Operadores aritméticos
  - 5.3. Operador Remainder
  - 5.4. Operações com diferentes tipos de dados
  - 5.5. Operadores de incremento e decremento
  - 5.6. Os operadores unários + e –
  - 5.7. Operadores aritméticos de atribuição
  - 5.8. Precedência e associatividade de operadores
  - 5.9. Operadores de comparação
  - 5.10. Operadores lógicos
  - 5.11. Operador condicional
  - 5.12. Operador Nil Coalescing
  - 5.13. Operadores Range
- 6. Strings e Characters
  - 6.1. Declarar e inicializar strings “vazias”
  - 6.2. Concatenação de strings
  - 6.3. Interpolation
  - 6.4. Representação interna das strings
  - 6.5. Caracteres especiais em literais String
  - 6.6. Extended Grapheme Clusters
  - 6.7. Comparação de strings
  - 6.8. Determinar o comprimento das strings
- 7. Arrays
  - 7.1. Declarar arrays
  - 7.2. Declarar arrays com um valor pré-definido

- 7.3. Criar arrays por adição de dois arrays
- 7.4. Criar um array com um Array Literal
- 7.5. Aceder a arrays e alterar o conteúdo
- 7.6. Arrays multidimensionais
- 8. Sets
  - 8.1. Declarar um set
  - 8.2. Criar um set com um array literal
  - 8.3. Criar um set com o range operator
  - 8.4. Aceder e alterar um set
  - 8.5. Operações fundamentais com Sets
  - 8.6. Determinar a associação e a igualdade entre sets
  - 8.7. Equality
  - 8.8. Subsets e Strict Subsets
  - 8.9. Supersets e Strict Supersets
  - 8.10. Disjoint
- 9. Dictionaries
  - 9.1. Declarar um dictionary
  - 9.2. Declarar um empty dictionary
  - 9.3. Criar um Dictionary
  - 9.4. Determinar se um Dictionary tem ou não elementos
  - 9.5. Determinar o número de elementos num Dictionary
  - 9.6. Aceder aos elementos de um Dictionary
  - 9.7. Alterar elementos de um Dictionary
  - 9.8. Adicionar elementos a um Dictionary
  - 9.9. Remover elementos de um Dictionary
  - 9.10. Nested dictionaries
- 10. Estruturas de decisão
  - 10.1. O statement if
  - 10.2. Utilizar os operadores lógicos
  - 10.3. Comparar e utilizar optionals
  - 10.4. A cláusula else
  - 10.5. O statement if...else if
  - 10.6. O statement switch
  - 10.7. Interval Matching case
  - 10.8. Tuples Matching case
  - 10.9. Switch...case – utilização de variáveis/constantes temporárias
- 11. Switch...case – utilização da cláusula where

- 11.1. Estruturas de repetição
- 11.2. while e repeat...while
- 11.3. For...in loops 11.3. for loops
- 11.4. Utilizar for...in loops em vez de for loops
- 11.5. Incrementar e apresentar o index com for...in loop
- 11.6. Decrementar e apresentar o index com for...in loop
- 11.7. Utilizar a função enumerate()
- 11.8. Iteração não-sequencial – cláusula where
- 11.9. Iteração não-sequencial – função stride()
- 11.10. Percorrer arrays multidimensionais
- 11.11. Control Transfer Statements

## 12. Funções

- 12.1. Definir e chamar funções
- 12.2. Parâmetros da função (Function parameters)
- 12.3. Valores de parâmetros pré-definidos
- 12.4. Valor de retorno da função (Function return value)
- 12.5. Funções com múltiplos valores de retorno
- 12.6. Argument Labels e Parameter Names
- 12.7. Variadic Parameters 12.8. In-Out Parameters
- 12.8. Function Types
- 12.9. Function Types como Parameters Types
- 12.10. Function Types como valor de retorno
- 12.11. Nested Functions

## 13. Enumerations

- 13.1. A sintaxe das Enumerations
- 13.2. Testar valore de uma Enumeration com um Switch
- 13.3. Associated Values
- 13.4. Raw Values
- 13.5. Enumerations recursivas

## 14. Classes e Structures

- 14.1. Comparação entre Classes e Structures
- 14.2. Structures e Enumerations são “Value Types”;
- 14.3. Classes são “Reference Types”;
- 14.4. Escolher entre Classes e Structures
- 14.5. Comportamento na atribuição e cópia de Strings, Arrays e Dictionaries
- 14.6. Propriedades
- 14.7. Métodos

14.8. Herança

14.9. Inicialização e De inicialização

14.10. Exemplos de utilização de classes e objetos em Swift

15. Navegação

15.1. Desenvolvimento de interfaces utilizando SwiftUI.

15.2. Navegação entre ecrãs.

16. APIs

16.1. Aceder a APIs utilizando a linguagem Swift.

17. Desenvolver aplicações para IOS.

*(English)*

1. iOS Application Development - An Overview

1.1. Technologies for iOS application development

1.2. Get and install Xcode and iOS SDK

1.3. The Apple Developer Program

1.4. The Xcode

2. Playground user interface

2.1. Create a first program (with playground)

2.2. Enter code in the editing area

2.3. The IDE playground areas

3. Variables and constants

3.1. Declare variables and constants

3.2. Type annotations

3.3. Assigning names to variables

3.4. Print strings and variables

3.5. Optionals

3.6. Example - Declaration and use of variables (playground)

4. Types of data

4.1. Integers

4.2. Floating point

4.3. Conversions between Integers and Floating-points

4.4. Booleans

4.5. Tuples

5. Operators

5.1. Assignment operator

5.2. Arithmetic operators

- 5.3. Remainder Operator
- 5.4. Operations with different types of data
- 5.5. Increment and Decrement Operators
- 5.6. The unary operators + and –
- 5.7. Arithmetic Assignment Operators
- 5.8. Operator precedence and associativity
- 5.9. Comparison Operators
- 5.10. Logical Operators
- 5.11. Conditional operator
- 5.12. Nil Coalescing Operator
- 5.13. Range Operators
6. Strings and Characters
  - 6.1. Declare and initialize “empty”; strings
  - 6.2. String concatenation
  - 6.3. Interpolation
  - 6.4. Internal representation of strings
  - 6.5. Special Characters in String Literals
  - 6.6. Extended Grapheme Clusters
  - 6.7. String Comparison
  - 6.8. Determine the length of strings
7. Arrays
  - 7.1. Declare arrays
  - 7.2. Declare arrays with a predefined value
  - 7.3. Create arrays by adding two arrays
  - 7.4. Create an Array with an Array Literal
  - 7.5. Accessing arrays and changing content
  - 7.6. Multidimensional arrays
8. Sets
  - 8.1. Declare a set
  - 8.2. Create a set with an array literal
  - 8.3. Create a set with the range operator
  - 8.4. Access and change a set
  - 8.5. Fundamental Set Operations
  - 8.6. Determine association and equality between sets
  - 8.7. Equality
  - 8.8. Subsets and Strict Subsets
  - 8.9. Supersets and Strict Supersets

## 8.10. Disjoint

## 9. Dictionaries

- 9.1. Declare a dictionary
- 9.2. Declare an empty dictionary
- 9.3. Create a Dictionary
- 9.4. Determining whether or not a Dictionary has elements
- 9.5. Determining the Number of Elements in a Dictionary
- 9.6. Accessing Dictionary Elements
- 9.7. Changing Elements of a Dictionary
- 9.8. Adding Elements to a Dictionary
- 9.9. Remove Elements from a Dictionary
- 9.10. Nested dictionaries

## 10. Decision structures

- 10.1. The if statement
- 10.2. Use logical operators
- 10.3. Compare and use optionals
- 10.4. The else clause
- 10.5. The if...else if statement
- 10.6. The switch statement
- 10.7. Interval Matching case
- 10.8. Tuples Matching Case
- 10.9. Switch...case - use of temporary variables/constants
- 10.10. Switch...case – using the where clause

## 11. Repeating structures

- 11.1. While and repeat...while
- 11.2. For...in loops 11.3. for loops
- 11.3. Use for...in loops instead of for loops
- 11.4. Increment and present the index with for...in loop
- 11.5. Decrement and display the index with for...in loop
- 11.6. Using the enumerate() function
- 11.7. Non-sequential iteration - where clause
- 11.8. Non-sequential iteration - stride() function
- 11.9. Scroll through multidimensional arrays
- 11.10. Control Transfer Statements

## 12. Functions

- 12.1. Define and call functions
- 12.2. Function parameters

- 12.3. Predefined parameter values
- 12.4. Function return value
- 12.5. Functions with multiple return values
- 12.6. Argument Labels and Parameter Names
- 12.7. Variadic Parameters 12.8. In-Out Parameters
- 12.8. Function Types
- 12.9. Function Types as Parameters Types
- 12.10. Function Types as return value
- 12.11. Nested Functions
- 13. Enumerations
  - 13.1. The Syntax of Enumerations
  - 13.2. Testing the Value of an Enumeration with a Switch
  - 13.3. Associated Values
  - 13.4. Raw Values
  - 13.5. recursive enumerations
- 14. Classes and Structures
  - 14.1. Comparison between Classes and Structures
  - 14.2. Structures and Enumerations are “Value Types”;
  - 14.3. Classes are “Reference Types”;
  - 14.4. Choosing between Classes and Structures
  - 14.5. Behavior in assigning and copying Strings, Arrays and Dictionaries
  - 14.6. Properties
  - 14.7. Methods
  - 14.8. Heritage
  - 14.9. Initialization and Deinitialization
  - 14.10. Examples of using classes and objects in Swift
- 15. Navigation
  - 15.1. Interface development using SwiftUI.
  - 15.2. Navigation between screens.
- 16. APIs
  - 16.1. Access APIs using the Swift language.
- 17. Develop applications for IOS.

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

A computação móvel tem tido um enorme incremento nos últimos anos com a explosão da utilização de dispositivos móveis de elevadas capacidades de computação, comunicação, armazenamento e interface. As aplicações são cada vez maiores indo desde o entretenimento e redes sociais até ao trabalho empresarial.

Os objetivos desta unidade curricular visam a aquisição de conhecimento e prática dos conceitos essenciais do projeto, arquitetura e implementação de aplicações, principalmente empresariais, em dispositivos móveis, utilizando APIs de grande divulgação corrente, pelo que os conteúdos mencionados são coerentes com os objetivos de 1 a 18, para responder aos problemas e desafios da grande diversidade de capacidades e formas de interação dos dispositivos móveis atuais.

*(English)*

Mobile computing has had a huge increase in recent years with the explosion in the use of mobile devices with high computing, communication, storage and interface capabilities. Applications span more and more fields from entertainment and social media to business work.

The main objectives of this course aim to acquire knowledge and practice of the essential concepts of design, architecture and implementation of applications, mainly business, in mobile devices, using APIs of great current dissemination, therefore, the contents mentioned are consistent with objectives 1 to 18, to respond to the problems and challenges of the great diversity of capacities and forms of interaction of current mobile devices.

**METODOLOGIAS DE ENSINO E AVALIAÇÃO / TEACHING METHODOLOGIES INCLUDING EVALUATION**

A metodologia de ensino baseia-se em exposições teóricas para apresentação dos conceitos científicos, seguidas de aplicações práticas de utilização e desenvolvimento de algoritmos e aplicações mobile. A apresentação dos elementos da linguagem Swift é concretizada através da resolução de exemplos apresentados e resolvidos pelo professor, seguidos de pequenos exercícios e projetos desenvolvidos pelos estudantes. inicialmente através da utilização do *playground* do ambiente de desenvolvimento. Esta metodologia, baseada numa evolução gradual, permitirá que os estudantes possam acompanhar o desenvolvimento dos conteúdos e criar aplicações de complexidade crescente.

De acordo com o Regulamento de Funcionamento do ISTEC-Porto a avaliação é efetuada através de um exame escrito individual e 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.

(English)

The methodology is based on theoretical presentations for the presentation of scientific concepts, followed by practical applications of use and development of algorithms and mobile applications. The presentation of the elements of the Swift language is achieved through the resolution of examples presented and solved by the teacher, followed by small exercises and projects developed by students. initially through the use of the development environment playground. This methodology, based on a gradual evolution, will allow students to follow the development of the subject and create applications of increasing complexity. According to ISTEC-Porto's internal regulations, assessment is made through an individual and compulsory written examination. In the final classification, elements of continuous assessment may be considered, such as tests, individual or group work, as well as the participation in the presential classes and in learning resources provided by e-learning systems.

**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 com componente prática é adequada à natureza dos objetivos definidos para a unidade curricular. A participação dos estudantes na resolução de problemas de programação e de aplicações de complexidade crescente, permite a aquisição dos conhecimentos e das capacidades de forma sustentada através do desenvolvimento de "apps" para os vários dispositivos do sistema operativo iOS.

(English)

The methodology with a practical component is appropriate to the nature of the objectives set for the course unit. The participation of students in solving programming problems and applications of increasing complexity, should allow the acquisition of knowledge and skills that enable the development of "apps" for the various devices of the iOS operating system.

**BIBLIOGRAFIA / BIBLIOGRAPHY**

FUNDAMENTAL / ESSENTIAL:

Apple Inc. (2020). The Swift Programming Language (Swift 5.3). <https://books.apple.com/gb/book/the-swift-programming-language-swift-5-3/id881256329>

Hoffman, J. (2015). Mastering Swift 4. Packt Publishing

Marcelino, L., & Silva, C. (2017). Desenvolvimento em Swift para iOS. FCA

COMPLEMENTAR/ COMPLEMENTARY:

Gray, A. (2015). Swift Pocket Reference Programming for iOS and OS X. O'Reilly Media, Inc.

Manning, J., Buttfield-Addison, P., Nugent, T., & Wooldridge, J. (2014). Swift Development with Cocoa: Developing for the Mac and iOS App Stores. O'Reilly Media, Inc.

Neuberg, M. (2017). iOS 11 Programming Fundamentals with Swift. O'Reilly Media, Inc.

Knott, M. (2014). Beginning Xcode. APRESS

Pierce, T., & Wooldridge, D. (2014). The Business of iOS App Development. APRESS.

Rogers, T. M. (2015). Swift Recipes - A Problem-Solution Approach. APRESS.

INTERNET:

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

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