

CURSO	Licenciatura em Informática		
UNIDADE CURRICULAR	Programação III	Obrigatória	X
		Opcional	
ÁREA CIENTÍFICA	Engenharia Informática		

Ano: 1º	Semestre: 1º	ECTS: 6	Horas de Contacto teórico práticas: 60
---------	--------------	---------	--

### OBJETIVOS DA APRENDIZAGEM

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

1. Conhecer e saber instalar o ambiente de desenvolvimento XCode;
2. Saber usar interface playground para desenvolver programas;
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. Compreender a estrutura das strings e saber usar funções de strings;
6. Saber usar arrays unidimensionais e multidimensionais;
7. Compreender o conceito de set saber executar operações com sets;
8. Saber declarar e aceder a dictionaries;
9. Compreender a estrutura e saber utilizar os statements de decisão e de repetição;
10. Saber utilizar funções na linguagem Swift;
11. Compreender o significado e saber utilizar enumerations;
12. Compreender os conceitos associados e saber utilizar classes e objetos;
13. Saber desenvolver aplicações ("apps") de complexidade moderada no sistema operativo iOS, usando a linguagem Swift.

## PROGRAMA

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
  - 10.10. Switch...case – utilização da cláusula where
11. Estruturas de repetição
  - 11.1. while e repeat...while
  - 11.2. 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
  - 12.9. Function Types
  - 12.10. Function Types como Parameters Types
  - 12.11. Function Types como valor de retorno
  - 12.12. Nested Functions
- 13. Enumerations
  - 13.1. A sintaxe das Enumerations
  - 13.2. Testar valor 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 Deinitialização
  - 14.10. Exemplos de utilização de classes e objetos em Swift
- 15. Desenvolver aplicações
  - 15.1. Aplicação 1 - Calculadora
  - 15.2. Aplicação 2 – Jogo Bull's Eye
  - 15.3. Aplicação 3 – Checklists

## **DEMONSTRAÇÃO DE COERÊNCIA ENTRE CONTEÚDOS PROGRAMÁTICOS E RESULTADOS DA APRENDIZAGEM**

Os pontos 1.1 a 1.4 permitem concretizar o objetivos 1. O objetivo 2 é alcançável através do conteúdos com o números 2.1 a 2.3. Os pontos do programa de 3.1 a 3.6 e de 4.1 a 4.5 permitem alcançar o objetivo 3. O objetivo 4 é atingido através dos pontos 5.1 a 5.11. Os pontos do conteúdo programático de 6.1 a 6.8 permitem atingir o objetivo 5. O objetivo 6 é alcançável através dos conteúdos programáticos 7.1 a 7.6. Os pontos do programa de 8.1 a 8.10 permitem concretizar o objetivo 7. O objetivo 8 é atingido com os pontos 9.1 a 9.10. Os conteúdos programáticos com os números 10.1 a 10.10 e 11.1 a 11.11 permitem atingir o objetivo permitem atingir o objetivo 9. O objetivo 10 é alcançável com o conteúdos 12.1 a 12.3. Os pontos 13.1 a 13.5 permitem atingir o objetivo 11. O objetivo 12 é alcançável através dos pontos 14.1 a 14.10. Finalmente, o objetivo 13 é atingido com os pontos 15.1 a 15.3.

## **METODOLOGIA DE ENSINO E AVALIAÇÃO**

Todas as aulas são de natureza teórico-prática. Estão previstas 60 horas de contato. O tempo total de trabalho do aluno corresponde a 162 horas. 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 alunos. inicialmente através da utilização do playground do ambiente de desenvolvimento. Esta metodologia, baseada numa evolução gradual, permitirá que os alunos possam acompanhar o desenvolvimento da matéria e criar aplicações de complexidade crescente.

De acordo com o Regulamento de Funcionamento do ISTECS 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 em recursos de aprendizagem proporcionados por sistemas de e-learning.

## **DEMONSTRAÇÃO DE COERÊNCIA ENTRE METODOLOGIAS DE ENSINO E RESULTADOS DE APRENDIZAGEM**

A participação dos alunos na resolução de problemas de programação e de aplicações de complexidade crescente, deverá permitir a aquisição dos conhecimentos e das capacidades que permitam o desenvolvimento de “apps” para os vários dispositivos do sistema operativo iOS.

## **BIBLIOGRAFIA**

### **Fundamental:**

MARCELINO, Luís SILVA, Catarina; Desenvolvimento em Swift para iOS; FCA  
Hoffman, Jon., Mastering Swift 4;Packt

Apple Inc (2017). The Swift Programming Language (Swift 4). Retirado de [https://developer.apple.com/library/content/documentation/Swift/Conceptual/Swift\\_Programming\\_Language/](https://developer.apple.com/library/content/documentation/Swift/Conceptual/Swift_Programming_Language/)

**Complementar:**

KNOTT, Matthew; Beginning Xcode; Apress

MANNING, Jonathon et al; Swift Development with Cocoa; O'Reilly Media

ROGERS, T. Michael; Swift Recipes - A Problem-Solution Approach; Apress

GRAY, Anthony; Swift Pocket Reference Programming for iOS and OS X; O'Reilly Media

PIERCE, Taylor et al; The Business of iOS App Development; Apress

NEUBERG, Matt; iOS 11 Programming Fundamentals with Swift; O'Reilly Media

**INTERNET:**

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

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