

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

Ano: 2º	Semestre: 2º	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 Python;
2. Compreender a natureza e a estrutura da linguagem Python bem como a organização e modo de execução do código;
3. Compreender e saber utilizar os vários tipos de dados da linguagem;
4. Compreender e saber utilizar as estruturas de decisão e de repetição;
5. Compreender e saber utilizar funções;
6. Compreender e saber utilizar a programação por objetos em Python;
7. Compreender e saber utilizar a linguagem Python para desenvolver aplicações na plataforma Raspberry Pi.

PROGRAMA

1. Introdução à linguagem Python
 - 1.1. Origens e evolução da linguagem
 - 1.2. Características da linguagem
 - 1.3. Modos de execução de código
2. Plataforma de desenvolvimento
 - 2.1. IDEs para Python
 - 2.2. Instalação de plataformas de desenvolvimento
 - 2.3. Execução de código
 - 2.4. Exemplos
3. Organização de execução de código Python
 - 3.1. Módulos e packages
 - 3.2. Names e namespaces
 - 3.3. Scopes
 - 3.4. Objetos e classes
4. Tipos de dados intrínsecos
 - 4.1. Tipos mutáveis e imutáveis
 - 4.2. Números
 - 4.2.1. Inteiros
 - 4.2.2. Booleans
 - 4.2.3. Floating-point
 - 4.2.4. Complexos

- 4.2.5. Frações
- 4.3. Strings e bytes
 - 4.3.1. Conceito
 - 4.3.2. Operações básicas
 - 4.3.3. Exemplos
- 4.4. Tuples
- 4.5. Listas
- 4.6. Byte arrays
- 4.7. Set types
- 4.8. Dicionários
- 4.9. O módulo Collections
- 5. Estruturas de decisão
 - 5.1. If
 - 5.2. If... else
 - 5.3. Ifl...elif
 - 5.4. Operador ternário
- 6. Estruturas de repetição
 - 6.1. Loop for
 - 6.1.1. Iterações em ranges
 - 6.1.2. Iterações em sequências
 - 6.2. Loop while
 - 6.3. Statements Break e Continue
 - 6.4. Utilização da clausula else
- 7. Exemplos de utilização conjunta de statement de decisão e repetição
- 8. Funções
 - 8.1. Conceito de função
 - 8.2. Utilidade das funções
 - 8.3. Criação de funções em Python
 - 8.4. Scopes
 - 8.5. Passagem de argumentos
 - 8.6. Valores retornados
 - 8.7. Funções recursivas
 - 8.8. Funções anónimas
 - 8.9. Funções built-in
- 9. Programação Orientada a Objetos
 - 9.1. Conceitos introdutórios
 - 9.2. Classes em Python
 - 9.3. Inicializar instâncias
 - 9.4. Herança e composição
 - 9.5. Acesso a classe base
 - 9.6. Herança múltipla
 - 9.7. Classes e métodos static
 - 9.8. Private methods

- 9.9. Operator overloading
- 9.10. Polimorfismo
- 10. Projeto de programação em Python para Smart Homes com RaspberryPi.

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. Numa primeira fase, a apresentação dos elementos da linguagem é concretizada através da resolução de exemplos apresentados e resolvidos pelo professor, seguidos de pequenos exercícios e projetos desenvolvidos pelos alunos. Esta metodologia, baseada numa evolução gradual, permitirá que os alunos possam acompanhar o desenvolvimento da matéria e criar aplicações de complexidade crescente. Numa segunda fase, de características fundamentalmente aplicacionais, os alunos deverão adquirir familiaridade com um dispositivo RaspberryPi e implementar aplicações usando a linguagem Python.

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 aprendizagem da linguagem através de pequenos projetos práticos e a prática laboratorial usando dispositivos RaspberryPi são de molde a permitir que os objetivos da unidade curricular sejam atingidos.

BIBLIOGRAFIA

Fundamental:

COSTA, Ernesto. (2015). Programação em Python. Lisboa: FCA

Complementar:

BORGES, Luiz (2015); PYTHON para desenvolvedores, NOVATEC, S. Paulo.

BHASKAR, Chaudhary (2015); Tkinter GUI Application Development Blueprints, PACKT, Birmingham.
MARTINS, João. (2015). PYTHON – Introdução à programação utilizando múltiplos paradigmas. Lisboa: IST Press.
PERKINS, Jacob (2014); PYTHON 3 Text Processing with NLTK 3 Cookbook, PACKT, Birmingham.
VASCONCELOS, José (2015); PYTHON – Algoritmia e Programação Web, FCA, Lisboa.

INTERNET:

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

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