

Ficha de Unidade Curricular (FUC)

1. Unidade curricular

Programação / Computer Programming

2. Docente responsável e respetiva carga letiva na unidade curricular (preencher nome completo)

Manuel Matos	
--------------	--

3. Outros docentes e respetivas cargas letivas na unidade curricular

--	--

4. Objetivos da aprendizagem (conhecimentos, aptidões e competências a desenvolver pelos estudantes)

É objectivo aprenderem a utilizar programas científicos, diversificados no uso, tratamento e representação de dados.

Terem a capacidade de resolver ligações de interfaces entre programa e equipamento. Saberem identificar ficheiros com formato normalizado de importação e exportação de dados.

Terem a habilidade de elaborar pequenos programas que vão ao encontro das suas necessidades enquanto profissionais na área da biologia.

Após a aprovação na unidade curricular, o aluno deverá possuir a capacidade de:

1. Fazer e compreender programas descritos em Matlab
2. Fazer e analisar programas em Python
3. Ler e escrever ficheiros (csv e xml) em Python e importar e exportar os mesmos em Matlab
4. Compreender a representação Unicode UTF-8
5. Estabelecer ligações a equipamentos GPIB e ler dados utilizando o Python
6. Ser capaz de utilizar o R para ler dados csv e representá-los graficamente.

(nc=883<1000)

To learn the use of diverse scientific programs that process and represent data.

Have the capacity of solving connections that interface equipment and programs.

Know the format of standard file representation of data used for import and export.

Gain the skill of building small programs to solve particular problems in the biological area.

After approval in this syllabus the student should be able to:

1. Develop and understand programs written in MatLab.
2. Develop and analyse programs in Python.
3. Read and write files (csv and xml) and import and export them into Matlab.
4. Understand the UTF-8 character code representation.
5. Establish connections through the GPIB bus using Python
6. Use R to read csv data files and make graphical plots.

(nc=739<1000)

5. Conteúdos programáticos

I - Introdução ao MatLab, variáveis, vector, matriz, controlo de fluxo, gráficos, e ficheiros
 II - Programação orientada por objectos utilizando o Python, classes, objectos, instância, métodos, herança, excepções, iteradores, e ficheiros. Algoritmia e estrutura de dados.
 III - Mapeamento de caracteres UTF-8
 IV - Barramento de instrumentação – GPIB (IEEE-488)
 V - Representação gráfica de dados biológicos utilizando o R, vector, matriz, ficheiros, e script
 VI - O editor de texto
 VII - Estrutura de ficheiros csv, xml

(nc=512<1000)

I - Introduction to MatLab, variables, vector, matrix, flux control, graphics and files
 II - Object based programming with Python, class, object, instance, methods, inheritance, exceptions, iterators, and files. Data structures and algorithms.
 III - UTF-8 character encoding
 IV - GPIB (IEEE-488) instrumentation bus
 V - Graphical representation of biological data using R, vector, matrix, files, and script
 VI - A text editor
 VII - File formats of type csv and xml

(nc=458<1000)

6. Demonstração da coerência dos conteúdos programáticos com os objetivos da unidade curricular

O Matlab é abordado no ponto (I) do programa na perspectiva de familiarizar o aluno com uma ferramenta a usar em outras unidades curriculares.
 O Python, escolhido pelo uso crescente entre profissionais que não são de informática, abordado no ponto (II) é a linguagem de programação base para o estudo de algoritmos, estruturas de dados, leitura e escrita de ficheiros.
 O R, abordado no ponto (V), amplamente divulgado na comunidade científica por ser gratuito, rápido, leve e ter muitos módulos de instalação simples, é usado para tratar dados, e fazer gráficos.
 Em outras unidades curriculares, a ligação de equipamento com a aplicação dedicada é feita através de ficheiros e ou barramento de instrumentação, daí a ser necessário a compreensão de ficheiros tipo csv e xml, do mapeamento de caracteres UTF-8, e dos barramentos USB e GPIB, abordados nos pontos (III, IV, VI e VII) do programa.

(nc=889<1000)

Matlab is presented in topic (I) as an introductory tool that students will use on other curricular units.
 Python, chosen due to its' increased use among professionals that are not from a computer science area, presented in topic (II), is the basic programming language used for the study of algorithms and data structures, reading and writing files.
 R, presented in (V), popular within the scientific community due to its freeware, lightweight, and simple installation of new packages, is used mainly for data manipulation and graphics.
 Other curricular units use equipment with dedicated applications that interface through an instrumentation bus and data files, thus it is compulsory to understand the csv and xml file formats, UTF-8 character encoding and the USB and GPIB bus. This is presented in topics (III, IV, VI e VII).

(nc=828<1000)

7. Metodologias de ensino (avaliação incluída)

As aulas são teórico práticas. Durante a semana haverá duas aulas, uma mais teórica com exercícios para consolidar os conhecimentos, e outra mais prática em laboratório, com um guião escrito. O Python é a linguagem de programação base sobre a qual incidirão os restantes pontos.

Cada semana terá um trabalho prático realizado no laboratório e que será avaliado com 0 (não entregou ou não fez o suficiente), 0.5 (aceitável) e 1 (bom). Estes trabalhos (TP) irão valer 60% da nota prática. É necessário realizar pelo menos 80% desses trabalhos práticos para ter aproveitamento.

Haverá um trabalho prático (TF) obrigatório a realizar fora do âmbito das aulas e que corresponderá a 40% da avaliação prática. Incluído na avaliação deste trabalho está uma discussão do mesmo.

No fim das aulas um teste (T) avaliará a componente teórica da disciplina.

A nota final da disciplina será de 40% teórica e 60% prática.

Classificação final $CF = 0.4T + 0.6 (0.6 TP + 0.4 TF)$; ($CF \geq 9.5$, $T \geq 9.5$, $TP \geq 8$)

(nc=991<1000)

Lessons are theoretical-practical. Each week has two lessons: one of them, more theoretical with exercises to provide examples of applications, and the other more practical with a written guide to help through the lab session. Python is the programming language used as the basis that supports the other themes of the syllabus.

Each week, a practical exercise, done in the lab, will have the following score: 0 (not delivered or not done), 0.5 (acceptable) and 1 (correct). These exercises (TP) will weight 60% of the practical mark. Necessary to deliver at least 80% of the exercises to be endorsed for approval.

An extra practical compulsory exercise (TF) done out of class weighted with 40% of the practical mark. Included in this mark is a discussion of the work.

At the end of the syllabus there will be a test (T) to evaluate the theoretical contents.

The final mark will be 40% theory and 60% practice.

Final Classification $CF = 0.4T + 0.6 (0.6 TP + 0.4 TF)$; ($CF \geq 9.5$, $T \geq 9.5$, $TP \geq 8$)

(nc=996<1000)

8. Demonstração da coerência das metodologias de ensino com os objetivos de aprendizagem da unidade curricular

Através das aulas teórica-práticas é exposta a matéria que permite aos alunos ter um fundamento base dos temas. Os exercícios práticos propostos, resolvidos no laboratório, irão incidir sobre os diversos pontos do programa, com combinações diversas dos temas de modo a atingir os objectivos de aprendizagem.

Os trabalhos realizados durante o semestre incentivam o aluno a realizar um estudo contínuo e permitem consciencializar o aluno do progresso na aprendizagem.

O teste final permite avaliar o aluno perante os fundamentos teóricos da unidade curricular.

O trabalho prático final, proposto a meio do semestre, e com uma discussão após a entrega do mesmo no fim do semestre, permite avaliar a compreensão dos temas na sua globalidade.

(nc=735<3000)

The theoretical-practical classes expose the basic knowledge that will give the students the fundamental basis of the syllabus. The proposed lab sessions with the practical exercises, will cover the whole topics of the syllabus, with several combinations of the topics so that the objectives can be fulfilled.

The practical exercises solved throughout the semester will motivate the student to a continuous study and to be aware of the learning progress.

The final test allows evaluation of the theoretical basis of the curricular unit.

The final practical exercise is proposed in the middle and submitted at the end of the semester. After submission, the student is subject to a discussion which allows the evaluation of the themes as a whole.

(nc=741<3000)

9. Bibliografia principal

M. Lutz, Learning Python - Powerful Object-Oriented Programming, O'Reilly Media, 2009
J. V. Guttag, Introduction to Computation and Programming Using Python, MIT Press, 2013
J.M. Hughes, Real World Instrumentation with Python - Automated Data Acquisition and Control Systems, O'Reilly Media, 2010
W. F. Punch, R. Enbody, The Practice of Computing Using Python, Addison-Wesley, 2013
T. Siau, A. Bayen, An Introduction to MATLAB Programming and Numerical Methods for Engineers, Academic Press, 2015

(nc=493<3000)