

Caracterização da Unidade Curricular / Characterization of the Curricular Unit

Designação da Unidade Curricular (UC) / Title of Curricular Unit (CU): Linguagens e Teoria da Computação / Languages and Computation Theory

Área científica da UC / CU Scientific Area: Informática / Computer Science

Semestre / Semester: 3º

Número de créditos ECTS / Number of ECTS credits: 6

Carga horária por tipologia de horas / Workload by type of hours: TP: 22,5; PL: 45; OT: 9; O: 13,5

Carga letiva semanal / Weekly letive charge: 4,5h

Docente responsável / Responsible professor: Jorge Manuel de Azevedo Pereira Simões, Doutor

Objetivos de aprendizagem (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

Nesta unidade curricular os alunos aprofundam os seus conhecimentos de algoritmia e deverão ficar aptos a desenvolver programas em linguagem Java e Python. No final da disciplina os alunos deverão ainda conhecer e saber aplicar os principais algoritmos de pesquisa e ordenação de elementos de um conjunto, sabendo avaliar a complexidade de cada um deles. Pretende-se ainda que os alunos saibam como avaliar a complexidade de algoritmos simples e também como selecionar o algoritmo mais adequado a cada problema particular. Os alunos deverão ficar aptos a desenvolver programas que envolvam estruturas de dados complexas, como listas e árvores, e saber executar operações sobre essas estruturas de dados.

Intended learning outcomes (knowledge, skills and competences to be developed by the students):

In this course, students deepen their knowledge of algorithms and should be able to develop programs in Java and Python language. At the end of the course, students should also know how to apply the main search algorithms and ordering elements of a set, knowing how to assess the complexity of each one of them. It is also intended that students know how to assess the complexity of simple algorithms, and also how to select the most suitable algorithm for each particular problem. Students should be able to develop programs that involve complex data structures, such as lists and trees, and know how to perform operations on these data structures.

Conteúdos programáticos:

1. Revisões da linguagem Java

2. Análise da complexidade de algoritmos

- 2.1. Notação O
- 2.2. Algoritmos polinomiais e algoritmos exponenciais
- 2.3. Referência a problemas NP-completos

3. Métodos de pesquisa e ordenação

- 3.1. Revisão de algoritmos de pesquisa
- 3.2. Revisão de algoritmos de ordenação
- 3.3. Comparação do desempenho de algoritmos de pesquisa e ordenação

4. Estruturas de dados lineares

- 4.1. Vectores, registos e pilhas
- 4.2. Listas ligadas

5. Tabelas de dispersão

- 5.1. Implementação de tabelas de dispersão
- 5.2. Funções de dispersão
- 5.3. Métodos de resolução de colisões

6. Estruturas de dados não lineares

- 6.1. Definição de árvore
- 6.2. Árvores binárias
- 6.3. Tipos de árvores binárias
- 6.4. Heaps. Algoritmo HeapSort

Syllabus:

1. Java language revisited

2. Analysis of the complexity of algorithms

- 2.1. Big Oh notation
- 2.2. Polynomial Algorithms and Exponential Algorithms
- 2.3. Reference to NP-complete problems

3. Search and sorting methods

- 3.1. Review of search algorithms
- 3.2. Review of sorting algorithms
- 3.3. Comparing the performance of search and sorting algorithms

4. Linear data structures

- 4.1. Vectors, Records and Stacks
- 4.2. linked lists

Sem Validade
Administrativa

5. Hash tables

- 5.1. Implementation of hash tables
- 5.2. Hash functions
- 5.3. Collision Resolution Methods

6. Nonlinear data structures

- 6.1. Tree definition
 - 6.2. Binary trees
 - 6.3. Types of binary trees
 - 6.4. Heaps. HeapSort Algorithm
-

