Ano Lectivo	2016/17
Curso	Engenharia Informática
Unidade Curricular	Algorítmos e Estruturas de Dados
Língua de ensino	Português
ECTS/tempo de trabalho (horas)	

ECTS	Total	Horas de contacto semestral						
		T	TP	PL	S	TC	O	OT
7.5	205		60	30				

T - Teóricas; TP - Teórico-práticas; PL - Prática-laboratorial; S - Seminário; OT - Orientação tutorial; TC - Trabalho de campo; E - Estágio; O* - Outras horas caraterizadas como Ensino Clínico ao abrigo da Diretiva nº 77/453/CEE de 27 Junho adaptada pela Diretiva 2005/36/CE;

Docente Responsável/Carga letiva [nome completo e e-mail]	Valentim Alberto Correia Realinho / vrealinho@estgp.pt
Outros Docentes e respetivas cargas letivas [nome completo e e-mail]	Sérgio Duarte Correia / scorreia@estgp.pt
	Valentim Alberto Correia Realinho / vrealinho@estgp.pt
Pré-requisitos [competências à entrada; pré-requisitos; precedências]	Não
Objetivos da aprendizagem (conhecimentos, aptidões e competências) a desenvolver pelos estudantes, operacionalização dos objetivos e medição do seu grau de cumprimento)	Estudo das várias estruturas de dados, algoritmos de pesquisa, de ordenação, e de grafos. Capacidade de conceção e análise de algoritmos. Criar competências para a utilização de Tipos Abstratos de Informação no desenvolvimento de programas. Conhecer e saber construir desde as estruturas de dados elementares até às avançadas. Conhecer e saber construir os vários algoritmos de ordenação e pesquisa e as várias otimizações. Criar competências para saber escolher o melhor algoritmo para um determinado problema. Conhecer as várias propriedades dos grafos. Conhecer os principais algoritmos dos grafos.
Conteúdos Programáticos [estrutura de conteúdos a desenvolver para o total de horas previsto]	Tipos Abstratos de Informação (TAI). Fundamentos da Análise de Algoritmos (Crescimento de Funções, Notação Big-Oh, complexidade computacional). Estruturas de dados elementares, árvores e recursão, tipos abstratos de informação elementares (Pilhas, Filas, Grafos, Strings, Conjuntos). Ordenação: Métodos elementares; Quicksort; Mergesort; Priority Queues e Heapsort. Pesquisa: Symbol Tables e Binary Search Trees; Balanced Trees; Hashing. Grafos: Propriedades e tipos de grafos; Pesquisa em grafos; Digraphs (grafos direccionais) e DAG's (grafos direccionais acíclicos); Minimum Spanning Trees (Árvores de custo mínimo); Menor caminho.
Demonstração da coerência dos conteúdos programáticos com os objectivos da unidade curricular	O conteúdo relacionado com os Tipos Abstratos de Informação (TAI) vai permitir ao aluno perceber o que é um TAI, como o utilizar corretamente e as vantagens de o utilizar. Isto permite cumprir o objetivo relacionado com criar competências para a utilização do TAI. O conteúdo sobre os fundamentos da análise de algoritmos são ferramentas que permitem analisar os algoritmos estudados. Isto irá permitir ao aluno desenvolver capacidades de conceção a análise de algoritmos e também criar competências para saber escolher o melhor algoritmo para cada problema concreto. Os restantes objetivos resultam diretamente do estudo das estruturas de dados, da ordenação, da pesquisa e dos grafos.
Metodologias de ensino (avaliação incluída) [indicar os produtos, critérios e pesos de avaliação] (máx1000 carateres)	**1 - Metodologias de ensino** Aulas teórico-práticas com exposição da matéria e apresentação de exemplos/casos de estudo. Aulas práticas com elaboração de exercícios sobre a matéria dada e implementação de exercícios de programação.

	2 - Avaliação por frequência Um Teste (40%) com toda a matéria. O teste tem nota mínima de 8. Dois trabalhos práticos (20% + 40%). Os trabalhos têm nota mínima de 10. Os trabalhos são feitos em grupos de 2 a 3 alunos. O primeiro trabalho prático é para o aluno adquirir prática de programação com ponteiros, estruturas de dados dinâmicas e compilação separada. Segundo trabalho prático de programação com Tipos Abstratos de Informação, estruturas de dados avançadas e algoritmos (ordenação, pesquisa e grafos). Os trabalhadores estudantes são dispensados das aulas práticas, mas ficam obrigados a realizar os trabalhos práticos. **3 - Avaliação por Exame** Igual à avaliação de frequência. O aluno pode aproveitar os trabalhos práticos das avaliações anteriores, desde que atinjam a nota mínima.
Demonstração da coerência das metodologias de ensino com os objectivos da aprendizagem da unidade curricular	As aulas teórico-práticas permitem apresentar os conceitos teóricos relacionados com a análise de algoritmos. Mas também apresentar e estudar as estruturas de dados, os algoritmos de ordenação, os algoritmos de pesquisa, e os algoritmos de grafos. Isto permite cumprir os objetivos identificados como "conhecer" e "criar competências". As aulas práticas permitem implementar e/ou usar estruturas de dados e algoritmos sempre no contexto dos tipos abstratos de informação (TAI). E os trabalhos práticos permitem cimentar aqueles conceitos. Estas componentes práticas permitem atingir os objetivos relacionados com o "saber construir" e o "criar competências para utilizar os TAI.
Bibliografia Principal	Robert Sedgewick, "Algorithms in C, Parts 1-4: Fundamental, Data Structures, Sorting, Searching", 3ª Edição, Addison Wesley Professional, 1998. 004.4 SDGI Robert Sedgewick, "Algorithms in C, Part 5: Graph Algorithms", 3ª Edição, Addison Wesley Professional, 2002. 004.4 SDG II Thomas Cormen, Charles Leiserson, Ronald Rivest, Clifford Stein, "Introduction to Algorithms", 2ª Edição, MIT Press, 2001. 004 IA Jon Kleinberg and Éva Tardos, "Algorithm Design", Addison Wesley, 2005. 004.4 KLN Rocha, A., Estruturas de Dados e Algoritmos em C, FCA – Editora de Informática, 20082. Baptista, L., "Programação Estruturada Linguagem ANSI-C", ESTGP, 1996. Gonçalves, J., "Programação com Linguagem C", Edições Sílabo, 1993. 004.4 GNC Kernighan, B. e Ritchie, D., "The C Programming Language", Second Edition Prentice Hall, 1988.
Bibliografia Complementar	
Situações especiais [estudantes com estatuto especial]	**1 - Avaliação por frequência** **2 - Avaliação por Exame**