

### Informação Geral:

<b>Unidade Orgânica</b>	Escola Superior de Tecnologia e Gestão	<b>Ano Letivo</b>	2017/2018
<b>Curso</b>	Licenciatura em Engenharia Informática (D) [9119]	<b>Grau</b>	Licenciatura
<b>Ano Curricular</b>	1	<b>Período</b>	S2
<b>UC/Módulo</b>	Programação II	<b>ECTS</b>	7
<b>Área Científica</b>	Ciências de Engenharia	<b>Carácter</b>	Obrigatório
<b>Horas Totais</b>	189	<b>T</b> 0	<b>TP</b> 30
		<b>PL</b> 45	<b>TC</b> 0
		<b>S</b> 0	<b>E</b> 0
		<b>OT</b> 0	

*T - Ensino Teórico; TP - Teórico Prático; PL - Prático e Laboratorial; TC - Trabalho de Campo; S - Seminário; E - Estágio; OT - Orientação Tutorial*

### Docente responsável:

José Manuel Magno Lopes

### Docentes que lecionam a unidade curricular:

António Carlos Alves Urbano ( 75,00 horas semanais de contacto: TP: 30,00; PL: 45,00; )

José Manuel Magno Lopes ( 75,00 horas semanais de contacto: TP: 30,00; PL: 45,00; )

Catarina Isabel Ferreira Viveiros Tavares Reis ( 120,00 horas semanais de contacto: TP: 30,00; PL: 45,00; PL: 45,00; )

Sónia Maria Almeida da Luz ( 135,00 horas semanais de contacto: PL: 45,00; PL: 45,00; PL: 45,00; )

Marco Paulo Monteiro Ferreira ( 45,00 horas semanais de contacto: PL: 45,00; )

João Pedro Ferreira Ramos ( 90,00 horas semanais de contacto: PL: 45,00; PL: 45,00; )

### Pré-requisitos:

Não tem.

### Idioma:

Português e Inglês

### Enquadramento:

Esta UC permite ao Estudante aprofundar os conhecimentos de Programação, nomeadamente adotando o paradigma da orientação aos objetos para a resolução de problemas concretos. O Estudante adquire, igualmente, competências na utilização das principais ferramentas e técnicas para realizar, sob uma perspetiva prática, a implementação/codificação de programas em projetos de software. É uma UC que promove a maturação de pequenos grupos (equipas) de trabalho e pretende introduzir as questões subjacentes ao processo de desenvolvimento de software com vista à comercialização de um produto atendendo às necessidades dos clientes deste tipo de produtos.

### Objetivos de aprendizagem:

C1 - Capacidade para modelar problemas de acordo com o paradigma orientado aos objetos

C2 - Capacidade para implementar um projeto de acordo com uma modelação

C3 - Reconhecer a necessidade de reutilização de código

## Programa:

### Conteúdos Programáticos:

- Conceitos Básicos da P.O.O.
- Encapsulamento
- Herança
- Polimorfismo
- Interfaces
- Identificação de comportamento semelhante
- Definição de classes puramente abstratas (interfaces) que funcionam como marcador
- Utilização genérica dos objetos "marcados" com a mesma Interface
- Tipos Abstratos de Dados
- Exceções e Serialização
- Conceção de uma estrutura de tratamento de erros para um contexto específico
- Persistência recorrendo a Streams
- modo texto
- modo binário
- Recorrer à serialização como abordagem para persistir a informação em modo binário

### Fundamentação da coerência dos conteúdos programáticos com os objetivos/competências da unidade curricular:

- Conceitos Básicos da P.O.O. (C1; C2 e C3)
- Encapsulamento (C1; C2 e C3)
- Herança (C1; C2 e C3)
- Polimorfismo (C1; C2 e C3)
- Interfaces (C1; C2 e C3)
- Tipos Abstratos de Dados (C1; C2 e C3)
- Exceções e Serialização (C1; C2 e C3)

## Metodologia de Ensino / Aprendizagem:

### Presencial:

Ensino teórico-prático (TP): apresentação e discussão dos assuntos referidos no conteúdo programático. Elaboração de exercícios

Ensino prático-laboratorial (PL): Estudo acompanhado e desenvolvimento de um projeto

Atendimento presencial aos estudantes: Sessões de orientação pessoal para conduzir o processo de aprendizagem e esclarecerem-se dúvidas

### Autónoma:

1. Estudo
  - 1.1 Leitura de excertos de bibliografia recomendada pela unidade curricular
  - 1.2 Resolução dos exercícios recomendados pela unidade curricular
2. E-aprendizagem
  - 2.1 Consulta de material de apoio relativo à unidade curricular

### Recursos Específicos:

1. Ensino teórico prático - sala de aula normal
2. Ensino prático laboratorial - laboratório de informática
3. Atendimento presencial aos estudantes - gabinete ou sala de aula normal

## Avaliação:

### Descrição:

Avaliação Contínua/Periódica:

50% - Componente Teórica

20% - Prova Escrita 1 - individual (mínimo 8)

30% - Prova Escrita 2 - individual (mínimo 8)

50% - Componente Prática

20% - Trabalho Laboratorial - 2 estudantes (mínimo 8)

30% - Teste Prático - individual (mínimo 8)

Avaliação Final:

50% - Componente Teórica

20% - Prova Escrita 1 - individual (mínimo 8)

30% - Prova Escrita 2 - individual (mínimo 8)

50% - Componente Prática

50% - Teste Prático - individual (mínimo 8)

A nota de cada componente (teórica e/ou prática) poderá ser salvaguardada, caso o estudante assim o entenda.

Melhoria:

Integral - deverá realizar o exame final completo (ambas as componentes teórica e prática).

**Número de elementos de avaliação final:** 3

**Número de elementos de avaliação contínua/periódica:** 4

## Demonstração da coerência das metodologias de ensino com os objetivos de aprendizagem da unidade curricular:

Ensino teórico-prático (TP): apresentação e discussão dos assuntos referidos no conteúdo programático. Elaboração de exercícios (C1 e C3)

Ensino prático-laboratorial (PL): Estudo acompanhado e desenvolvimento de um projeto (C1; C2 e C3)

Atendimento presencial aos estudantes: Sessões de orientação pessoal para conduzir o processo de aprendizagem e esclarecerem-se dúvidas (C1; C2 e C3)

## Bibliografia:

### Recomendada:

Apontamentos fornecidos nas aulas.

### Complementar:

Bruce Heckel, Thinking in Java, 4th edition, 2006, Prentice-Hall  
F. Mário Martins, JAVA 8 - POO + Construções Funcionais, 2017, FCA  
The Java Tutorial, Sharon Biocca Zakhour, Sowmya Kannan, Raymond Gallardo, 5th edition, 2013, Java Series