

### Informação Geral:

<b>Unidade Orgânica</b>	Escola Superior de Tecnologia e Gestão	<b>Ano Letivo</b>	2015/2016
<b>Curso</b>	Licenciatura em Engenharia Informática	<b>Grau</b>	Licenciatura
<b>UC/Módulo</b>	Programação Avançada	<b>ECTS</b>	7
<b>Área Científica</b>	Engenharia Informática - Sistemas de Informação	<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:

Patrício Rodrigues Domingues

### Docentes que lecionam a unidade curricular:

Patrício Rodrigues Domingues ( 78,00 horas semanais de contacto: TP: 30,00; PL: 45,00; PL: 3,00; )

Vitor Manuel de Jesus Carreira ( 78,00 horas semanais de contacto: PL: 39,00; PL: 39,00; )

Carlos Fernando de Almeida Grilo ( 81,00 horas semanais de contacto: PL: 6,00; PL: 33,00; PL: 42,00; )

### Pré-requisitos:

Nenhum.

### Idioma:

Português

### Enquadramento:

Esta UC proporciona ao estudante a aquisição de competências gerais e avançadas na área da programação.

### Objetivos de aprendizagem:

C1 - Conhecimentos avançados sobre programação  
 C2 - Reforçar a capacidade de análise e resolução de problemas específicos da programação  
 Específicas  
 C3 - Reforçar competências sobre programação em ambiente Linux  
 C4 - Obter conhecimentos sólidos sobre os fundamentos dos sistemas concorrentes  
 C5 - Obter conhecimentos básicos sobre os fundamentos dos sistemas distribuídos  
 C6 - Adquirir capacidade para o desenvolvimento de aplicações que envolvam o paradigma concorrente  
 C7 - Aplicação da aprendizagem em novas situações e contexto  
 C8 - Capacidade em estudar autonomamente e em manter-se atualizado  
 C9 - Capacidade de trabalhar em equipa  
 C10 - Capacidade para realizar projetos

## Programa:

### Conteúdos Programáticos:

1. Programação concorrente
  - 1.1 Introdução à programação concorrente
  - 1.2 Processos e threads
  - 1.3 Mecanismos de comunicação entre threads
  - 1.4 Concorrência, exclusão mútua e sincronização
2. Programação distribuída
  - 2.1 Introdução aos sistemas distribuídos
  - 2.2 Representação da informação
  - 2.3 Programação por sockets
  - 2.4 Estudo de protocolos aplicativos

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

1. Programação concorrente
  - 1.1 Introdução à programação concorrente (C1, C3, C4)
  - 1.2 Processos e threads (C1, C3, C4, C6)
  - 1.3 Mecanismos de comunicação entre threads (C1, C3, C4, C6)
  - 1.4 Concorrência, exclusão mútua e sincronização (C1, C2, C3, C4, C6)
2. Programação distribuída
  - 2.1 Introdução aos sistemas distribuídos (C1, C3, C5)
  - 2.2 Representação da informação (C1, C5)
  - 2.3 Programação por sockets (C1, C5, C6, C7)
  - 2.4 Estudo de protocolos aplicativos (C1, C2, C5, C6, C7)

## Metodologia de Ensino / Aprendizagem:

### Presencial:

- EP1. Ensino teórico: Apresentação dos conceitos e princípios respeitantes à programação  
EP2. Ensino prático: Exemplificação e aplicação a sistemas reais  
EP3. Ensino laboratorial: Resolução de exercícios

### Autónoma:

- AA.1. Consulta de material relativo à unidade curricular  
AA.2. Resolução dos exercícios recomendados para a unidade curricular  
AA.3. Apoio ao projeto prático

### Recursos Específicos:

Laboratório de informática.

## Avaliação:

### Descrição:

AP=Avaliação periódica  
AP.1. Dois testes escritos teóricos T (mín. 8/20 val.)  
AP.2. Um projeto prático com duas etapas P1, P2 (mín. 9/20 em 0,4xP1+0,6xP2)  
AP.3. Classificação final:  $CF = 0,50xT + 0,2xP1 + 0,3xP2$   
AE=Avaliação por exame  
AE.1. Duas partes: escrita (PE) e prova prática ao computador (PC)  
AE.2. Classificação final:  $CF = 0,50xPE + 0,50xPC$  (mín: PE=8,0, PC=9,0)  
Transitam para a avaliação final, e para cada componente, a classificação da componente se essa for maior ou igual ao respetivo mínimo.  
Melhoria. Obtida a aprovação à UC em época anterior, a tentativa de melhoria da nota final pode ser feita por i) melhoria à componente teórica ou ii) melhoria à componente prática ou iii) melhoria a ambas as componentes. Para efeitos de cálculo da nota final da tentativa de melhoria é considerada, para cada componente, a nota obtida na tentativa mais recente.

Número de elementos de avaliação final: 2

Número de elementos de avaliação contínua/periódica: 4

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

- EP=Ensino Presencial  
EP1. Ensino teórico: Apresentação dos conceitos e princípios respeitantes à programação (C1, C2, C3)  
EP2. Ensino prático: Exemplificação e aplicação a sistemas reais (C2, C7, C10)  
EP3. Ensino laboratorial: Resolução de exercícios (C2, C3, C6, C10)

AA=Aprendizagem Autônoma  
AA.1.Consulta de material relativo à unidade curricular (C1, C7, C8)  
AA.2.Resolução dos exercícios recomendados para a unidade curricular (C2, C3, C5, C6, C7, C8)  
AA.3.Apoio ao projeto prático (C2, C3, C5, C6, C7, C9, C10)

## **Bibliografia:**

### **Recomendada:**

Programação Avançada - Apontamentos das aulas teóricas", Patrício Domingues, ESTG/IPLeiria.  
"Programação Avançada - Fichas das aulas práticas", docentes das aulas práticas, ESTG/IPLeiria.  
"Operating Systems Concepts", A. Silberschatz, 8th edition, 2011 (ISBN:1118112733)  
"UNIX Network Programming, Volume 2: Interprocess Communications", R. Stevens, Prentice Hall, 2nd edition, 1999, ISBN 0 13 081081-9  
"UNIX Network Programming, Volume 1: Networking APIs: Sockets and XTI", R. Stevens, Prentice Hall, 2nd edition, 1998, ISBN 0-13-490012-X

### **Complementar:**

"The C Programming Language", Kernighan & Ritchie, Prentice-Hall, 2ª edição, 1998, ISBN: 978-013-110-362-7  
"Operating Systems - Internals and Design Principles", William Stallings, 7ª edição, 2011, ISBN : 013230998X