

INSTITUTO POLITÉCNICO DE BEJA
ESCOLA SUPERIOR DE TECNOLOGIAS E GESTÃO
CURRICULUM DO CURSO DE ENGENHARIA INFORMÁTICA
CURRICULUM DO CURSO DE INFORMÁTICA DE GESTÃO
CURRICULUM DO CURSO DE ENGENHARIA DE REABILITAÇÃO E ACESSIBILIDADE

DESCRITOR DE UNIDADE CURRICULAR

DESIGNAÇÃO: Programação II

ANO: 1

SEMESTRE: 2

ÁREA CNAEF: 481

CRÉDITOS: 7,0

TEMPO DE TRABALHO DO ESTUDANTE EM HORAS:

Total:	Contacto: 60							
	Teóricas	Teórico Práticas	Práticas e Laboratório	Trabalho de Campo	Seminário	Estágio	Orientação Tutória	Outras*
189		30	30					

DESCRIÇÃO RESUMIDA DA UNIDADE CURRICULAR:

Esta unidade é a continuação de Programação I introduzindo o processamento de ficheiros, as interfaces gráficas e a programação por eventos, bem como, alguns conceitos avançados da programação orientada pelos objectos. Em particular a herança, polimorfismo e a utilização e definição de *patterns* elementares.

TEMAS PROGRAMÁTICOS:

- Herança.
- Polimorfismo.
- Sobrecarga e redefinição de operações.
- Relações *is-a* versus *has-a*.
- Hierarquias de classes.
- Classes contentor.
- *Input-Output*.
- Interfaces textuais com o utilizador.
- Interfaces gráficas.
- Programação por eventos.
- Tratamento de erros (excepções).
- Design de Sistemas.
- *Patterns* elementares.
- Diagramas de comportamento.

BIBLIOGRAFIA DE BASE:

- David J. Barnes & Michael Kölling, "Objects First with Java - A Practical Introduction using BlueJ", 2nd Edition, Prentice Hall / Pearson Education, 2004.
- Cay S. Horstmann, "Java Concepts", 4th edition, John Wiley & Sons, 2005.
- The Java Tutorial, <http://java.sun.com/docs/books/tutorial/>.
- Joshua Bloch, "Effective Java Programming Language Guide", Addison-Wesley Professional, 2001.
- Kathy Sierra, "Heads First Java", 2nd edition, O'Reilly Media, 2005.

OBJECTIVOS EDUCACIONAIS:

- Sabe discutir e identificar os conceitos de encapsulamento, abstracção, herança e polimorfismo.
- Saber optar e justificar a opção por relações de *is-a* ou *has-a*.
- Desenhar, codificar e testar pequenos programas com interface gráfica que respondem aos eventos do utilizador e que utilizam tratamento de excepções, aplicando também as competências adquiridas em Programação 1.
- Partindo de um enunciado em linguagem natural, identificar e aplicar os conceitos e técnicas estudadas nesta disciplina e em Programação 1.
- Conhecer e aplicar *patterns* elementares.

ESTRATÉGIAS DE ENSINO / APRENDIZAGEM:

- Nas aulas teórico-práticas expõem-se os conteúdos e propõe-se a realização de trabalhos de modelação e programação. Estes trabalhos são realizados em grupos de dois, três ou quatro alunos.
- Nas aulas práticas os alunos resolvem problemas de modelação e programação, em grupos de dois, utilizando *pair programming* e desenvolvimento conduzido por testes, e sob supervisão do docente. Este discute possíveis resoluções com os alunos e esclarece dúvidas apresentadas. Os alunos continuam a resolução dos problemas propostos nas suas horas de trabalho autónomo.
- Ao longo de todo o semestre é utilizada uma plataforma de ensino à distância que permite várias formas de interacção entre alunos e entre estes e os docentes.

AVALIAÇÃO:

- Realização de programas em horas de contacto, continuados em horas autónomas.
- Realização de programas em horas autónomas.
- Realização de testes escritos.
- Apresentação de trabalhos.
- Avaliação de um ou mais relatórios escritos.