ISEL
ADEETC
Área Departamental de Engenharia Electrónica e Telecomunicações e de Computadores

Ficha de Unidade Curricular

Curso	Licenciatura em Engenharia Informática e de Computadores							
Designação UC	Modelação e Padrões de Desenho							
Área Científica	IC			**Observações**	Optativa. Partilhada com outros cursos.			
Ano	2	**Semestre**	4	**Duração**[1]	Semestral			
ECTS	6	**Horas de trabalho**[2]	162	**Horas de contacto**[3]	TP	T	P	PL
					67,5			

Docente Responsável	Fernando Miguel Gamboa de Carvalho

Objetivos de aprendizagem (conhecimentos, aptidões e competências a desenvolver pelos estudantes).
(1.000 carateres).

Os estudantes que terminam com sucesso esta unidade curricular serão capazes de:
(1) Descrever os principais padrões de desenho orientados por objetos
(2) Compreender o modelo de programação assíncrono e usar APIs assíncronas
(3) Usar abstrações de ordem superior que representam os efeitos ou, o estado de controlo de um programa, tais como: *Streams*, Futuros, Promessas e Continuações.
(4) Desenvolver componentes de software assentes em abordagens assíncronas não-bloqueantes e que disponibilizam APIs assíncronas.

Conteúdos programáticos ***(1.000 carateres).***

1. Introdução aos padrões de desenho orientados por objetos (OO).
2. Parametrização de comportamento e Funções de 1ª classe.
3. Funções de ordem superior (e.g. *andThen*, *memoize*, etc) e composição de funções.
4. Simplificação de padrões de desenho OO.
5. APIs fluentes (e.g. comparing, reversed, *thenComparing*, etc).
6. Métodos *default*. Aplicação em padrões de desenho e composição de interfaces.
7. Introdução aos *Streams* como sequências de elementos com processamento *on-demand*.
8. Encadeamento, operações de redução imutáveis e mutáveis.
9. Geração de *Streams*.
10. Modelo de programação assíncrono.
11. Efeitos essenciais na programação e sua relação entre o modelo síncrono e assíncrono.
12. Desenho de APIs assíncronas. Futuros, Promessas e Continuações.

Demonstração da coerência dos conteúdos programáticos com os objetivos de aprendizagem da unidade curricular ***(1.000 carateres).***

Esta unidade curricular tem o objetivo de introduzir técnicas avançadas de programação que combinam o paradigma de programação por objetos com abordagens usadas no paradigma de programação funcional reativa (https://en.wikipedia.org/wiki/Functional_reactive_programming).
A programação funcional reativa é um paradigma de programação orientado à propagação de alterações (fluxos de dados assíncronos) e que usa construções da programação funcional.
Estas abordagens oferecem um idioma de desenvolvimento de sistemas escaláveis, fáceis de manter e de evoluir.

[1] Anual, semestral, trimestral, …

[2] Número total de horas de trabalho

[3] Discriminadas por tipo de metodologia adotado (T - Ensino teórico; TP - Ensino teórico-prático; PL - Ensino prático e laboratorial; TC - Trabalho de campo; S - Seminário; E - Estágio; OT - Orientação tutorial; O - Outro)

Metodologias de ensino (avaliação incluída) ***(1.000 carateres).***

Ensino teórico-prático, estando previstas 30 aulas durante o semestre a que correspondem 67,5 horas de contacto (15 aulas de 3 horas e 15 de 1,5 horas). O tempo total de trabalho do estudante é de 160 horas. As aulas destinam-se à apresentação dos temas e de exemplos práticos de aplicação.
Por cada tema tópico são apresentados nas aulas problemas e discutidas diferentes soluções com os alunos através da sua modelação. A solução final é implementada na aula.
Os tópicos principais são ainda explorados através da realização de fichas de exercícios e de trabalhos. Os resultados da aprendizagem são avaliados através: (1) do teste escrito (T), (2) das fichas de exercícios (F) e (3) da discussão dos trabalhos (P). A nota final será a melhor classificação obtida através da seguinte fórmula: 40% [T] + 20%[F] + 40% [P].

Demonstração da coerência das metodologias de ensino com os objetivos de aprendizagem da unidade curricular ***(3.000 carateres).***

Os temas são introduzidos através da apresentação de problemas que são debatidos com os alunos, sob orientação do docente e solucionados através do desenho de modelos no quadro, que traduzem as sugestões da audiência. Por fim é implementada a solução que melhor responde aos requisitos de facilidade de manutenção, escalabilidade e evolução.
As fichas de exercícios pretendem replicar o mesmo tipo de problemas apresentados nas aulas, mas aplicados num novo contexto, devendo ser resolvidos autonomamente pelo aluno e com o apoio do docente no esclarecimento de dúvidas.
Os trabalhos reúnem um conjunto de desafios que deverão ser solucionados através da aplicação dos conhecimentos adquiridos. A avaliação deste projeto é baseada numa discussão onde são discutidas as soluções desenvolvidas pelo aluno.

Bibliografia de consulta/existência obrigatória ***(1.000 carateres).***

R. Urma, M. Fusco, A. Mycroft, *Modern Java in Action - Lambdas, streams, functional and reactive programming*, Manning Publications, 2018, ISBN 9781617293566