Mapa de Formação de 2º Ciclo em Informática

Documento Enquadrador da Oferta Formativa de 2º Ciclo em Informática
Maio 2007

Escola de Engenharia
Universidade do Minho

Enquadramento

Comissão Científica do Departamento de Informática

Assinala-se em 2006 o 30º aniversário do lançamento do primeiro projecto pedagógico da UM na área da Informática — a Licenciatura em *Engenharia de Produção - ramo Sistemas*. Tratou-se de um projecto de caracter pioneiro, cujos frutos viriam a revelar-se de enorme impacto, não apenas no tecido socio-económico regional, como na dinamização de toda uma área de investigação, desenvolvimento e negócio no contexto português. De então para cá, a oferta pela UM de formação superior nesta área expandiu-se e diversificou-se em projectos consistentes e continuados, escudados em investigação científica própria, sólida, com reconhecimento internacional, e revelando, por entre as vicissitudes de todos os caminhos percorridos, uma inegável coerência estratégica. As actuais licenciaturas em Engenharia de Sistemas e Informática, Informática de Gestão, Matemática e Ciências de Computação, Engenharia Electrónica Industrial e Computadores e Engenharia de Comunicações, assim como os inúmeros cursos de Mestrado existentes, são excelentes testemunhos dessa diversidade e dessa coerência.

No interior da Escola de Engenharia esta área abrange hoje 2 departamentos, 4 cursos de licenciatura que, no seu todo, excedem os 2000 alunos inscritos, 5 mestrados e correspondentes cursos de especialização, cerca de uma centena de alunos de doutoramento. Em vários desses projectos pedagógicos esta área cruza-se ainda com as áreas afins da Electrónica e dos Sistemas nas quais a Universidade atinge similares níveis de inovação e excelência.

É neste contexto de grande vitalidade e atractividade externa que foi lançado, há cerca de dois anos, um esforço no sentido de repensar a globalidade da oferta formativa em Informática e Sistemas de Informação na Universidade, procurando, em particular, construir uma proposta integrada de formação ao nível do 2º Ciclo *articulada*, *diversificada* e *flexível*.

Articulada para suscitar sinergias e, dessa forma, potenciar a experiência vasta e consolidada que nesta área a Universidade dispõe, de forma a cada vez melhor, competir adequadamente com instituições congéneres, atraindo, em diferentes faixas etárias, os estudantes mais qualificados no país e fora dele.

Diversificada para poder atingir, de foma coerente e útil, diferentes tipos de públicos e necessidades do tecido humano e social, adequando-se aos requisitos de uma sociedade em constante mutação e crescente integração global.

Flexível para ser capaz de se adequar à crescente heterogeneidade dos estudantes do Ensino Universitário no que concerne à motivação, proveniência, idade, formação anterior, disponibilidade e expectativas.

Este esforço foi participado pelos Departamentos de Informática (DI), Sistemas de Informação (DSI) e Produção e Sistemas (DPS). Contou, igualmente, com a colaboração do Departamento de Matemática da Escola de Ciências e com o Instituto de Educação e Psicologia.

Dele nasceu o desenho de 3 Mapas de Formação que concretizam a oferta formativa nas 3 áreas de intervenção principal destes Departamentos (a Informática, os Sistemas de Informação e a Engenharia de Sistemas), concebidos segundo um conjunto de princípios e estrutura similares, cada um deles aberto à cooperação inter-departamental. O presente documento articula os resultados desse trabalho para a área da *Informática* numa visão integradora que, recolhendo o espírito da Declaração de Bolonha e suas implicações a longo termo, tem na excelência dos profissionais que forma o único critério de aferição do seu sucesso.

Princípios

A Formação de 2º Ciclo em Informática na Universidade do Minho rege-se pelos seguintes princípios:

1. Define-se como um programa de ensino de excelência, confluência de sinergias múltiplas entre equipas docentes qualificadas, centros de investigação reconhecidos internacionalmente e parcerias fortes com o tecido económico-social.

2. Integra-se num Mapa Pedagógico global que articula a totalidade da oferta formativa em Informática ao nível dos três ciclos de formação superior, no qual se definem os diversos percursos conducentes aos diferentes graus e diplomas previstos na legislação. Este mapa oferece a cada estudante de 2º Ciclo a possibilidade de desenhar o seu próprio percurso formativo no seio de uma oferta ampla e diversificada, proporcionando-lhe igualmente uma imagem clara das suas balizas e dos processos de certificação associados.

3. Em particular, o subconjunto da formação de 2º Ciclo em Informática certificado com o grau de Mestre organiza-se em dois tipos cursos de Mestrado[1] que combinam diferentes *perfis de especialidade* com uma tipologia de percursos formativos distintos.

   Como se explica em detalhe na secção 4 deste documento, a arquitectura do MEI e do MI será concebida a partir de um conjunto de *Unidades Curriculares de Especialidade*, dotadas de 30 ECTS, cada uma das quais configura uma formação coerente numa área aplicacional específica, enunciada na triplíce vertente dos *fundamentos*, das *tecnologias* e das *aplicações* e integrando uma componente horizontal de projecto. Estas Unidades Curriculares constituem a principal inovação pedagógica no desenho do mapa de oferta formativa aqui apresentado.

   Apesar dessa matriz comum, os dois cursos distinguem-se pelas seguintes características:

[1] que, numa primeira fase, corresponderão a dois cursos distintos: *Mestrado de Informática* (MI) e o *Mestrado de Engenharia Informática* (MEI).

- O MEI, sendo um Mestrado em Engenharia que se propõe completar a formação de 1º Ciclo de modo conducente à certificação profissional em Engenharia, apresenta um conjunto fixo de *perfis de especialidade* que integram contéudos e competências necessárias para tal certificação. Tais perfis, que serão fixados em diálogo com a Ordem dos Engenheiros, incluem formação específica orientada ao desenvolvimento de competências para o planeamento, realização e avaliação de projectos de investigação e desenvolvimento em Engenharia. Por fim, a gênese da dissertação de Mestrado terá como base um *Projecto de Investigação e Desenvolvimento em Engenharia* com envolvência efectiva de uma parceria industrial.
- O MI, por outro lado, não define perfis formativos específicos, permitindo que cada aluno combine de forma livre duas das Unidades Curriculares de Especialização, acima referidas, à sua escolha. A virtualidade deste curso reside na flexibilidade curricular que proporciona. Tal flexibilidade torna possível o desenho de currículos orientados ao aluno com relevância para um público alvo que busque re-conversão ou especialização em domínios específicos correspondentes a re-orientações de interesses profissionais ou de áreas de negócio do tecido empresarial.

4. Toma consciência tanto da heterogeneidade das motivações e expectativas de quantos buscam o Ensino Superior, como da diversidade de desafios que uma sociedade complexa coloca hoje à missão da Universidade. Nesse sentido alia, na definição de todos os seus percursos formativos, *exigência*, na coerência científico-pedagógica que os orienta, e *flexibilidade*, nas estruturas e modos de organização que os configuram.

5. Reconhece, em particular, a importância estratégica de tomar seriamente a Universidade como *um lugar onde se volta*, procurando responder em termos de formação contínua a novos públicos e novos desafios.

6. Procura potenciar, em termos organizativos, o modelo matricial da Universidade para reunir competências, optimizar recursos e suscitar sinergias, num projecto integrador que lhe permitirá manter e aprofundar, nos anos que se vão seguir, a capacidade de liderança e inovação que lhe são hoje reconhecidos.

Arquitectura

A oferta formativa em Informática ao nível do 2º Ciclo é organizada a partir de um conjunto de Unidades Curriculares de acordo com seguinte classificação:

Unidade Curricular de Especialidade (30 ECTS) Configura uma proposta de formação coerente numa área aplicacional específica, enunciada na triplíce vertente dos *fundamentos*, das *tecnologias* e das *aplicações* e integrando uma componente horizontal de projecto. Estas Unidades Curriculares, designadas por UCE30, podem ter dois tipos de incidência:

- *Tecnológica:* formação numa área tecnológica ou científica específica da Informática

- *Profissional:* formação para uma profissão ou forma específica de exercício profissional em Informática.

Cada UCE30 é avaliada por um exame único e pelo desempenho na componente de projecto integrado. Cada UCE30 tem alocada uma equipa docente responsável pela planificação e gestão científica-pedagógica da mesma. Essa equipa é constituida por um mínimo de 4 doutores que a *patrocinam* do ponto de vista científico-pedagógico, sendo pelo menos 3 docentes de Departamentos de Ciência e Tecnologia da Universidade do Minho. Todas as responsabilidades lectivas associadas à UCE30 são assumidas por esta equipa, sem prejuízo de, eventualmente, outros docentes virem a colaborar na lecionação de módulos específicos. Cada doutor não poderá patrocinar mais do que uma UCE30. As UCE30 serão, regra geral, oferecidas ao longo de dois semestres lectivos.

Projecto de Dissertação Unidade curricular de projecto, orientada ao desenvolvimento de competências profissionalizantes específicas e integrando a realização de um projecto individual conducente à elaboração de uma dissertação em um dos dois formatos seguintes:

Projecto de Investigação : dissertação de cariz científico-tecnológico, envolvendo a capacidade de aplicação original de métodos de pesquisa a problemas e/ou aplicações da Informática.

Projecto de Investigação e Desenvolvimento em Engenharia : dissertação de cariz científico-tecnológico associada a um projecto de engenharia, envolvendo, preferencialmente, uma componente de exercício probatório da profissão de engenheiro.

Unidade Curricular de Reconversão (15 ou 30 ECTS) Dotada de uma estrutura similar à prevista para as Unidades Curriculares de Especialidade, restringe o seu âmbito a áreas de formação básica em Informática e destina-se a estudantes que

- provenientes de uma formação de 1º Ciclo numa área restrita de Informática, pretendam adquirir competências equivalentes em outra área.
- provenientes de outras formações científicas ou tecnológicas, que não em Informática, pretendam adquirir competências equivalentes a uma formação de 1º Ciclo em Informática.

Cursos de Mestrado

O subconjunto da formação de 2º Ciclo em Informática certificado com o grau de Mestre organiza-se em tipos de cursos de Mestrado que combinam diferentes *perfis de especialidade* com uma tipologia de percursos formativos distintos. Estes *perfis* correspondem a 60 ECTS e são obtidos pela realização de duas UCE30. Propoem-se, assim, os cursos seguintes:

- *Mestrado de Informática* (MI)
- *Mestrado de Engenharia Informática* (MEI)

a que são associados *perfis de especialidade* determinados

- por um *perfil de formação especializada em engenharia*, no caso do MEI;
- pelo par de UCE30, realizados pelo aluno na parte curricular do curso, no caso do MI.

As designações escolhidas[2] permitem oferecer cursos como

- MEI em *P*, onde *P* designa um perfil de especialidade vertical, por exemplo MEI em *Engenharia de Software* ou MEI em *Redes e Serviços de Comunicações*.
- MI em *E* e *F*, onde *E* e *F* designam UCE30; por exemplo, MI *em Computação Gráfica e Sistemas Inteligentes* ou MI *em Bioinformática e Métodos Formais*.

1. Mestrado de Engenharia Informática (MEI). Este curso representa o subconjunto da Formação de 2º Ciclo em Informática acreditado pela Ordem dos Engenheiro. Assim, supõe

- a certificação prévia num curso de 1º Ciclo em Engenharia Informática, ou equivalente;
- a realização de uma *Dissertação* com base num *Projecto de Investigação e Desenvolvimento em Engenharia*;
- a certificação da parte escolar do curso a um dos perfis cuja acreditação pela Ordem dos Engenheiros será garantida.

É a seguinte a estrutura curricular do MEI:

ECTCS	Mestrado de Engenharia Informática (MEI)
30	UC Especialidade (a)
30	UC Especialidade (a)
15	Seminário (b)
45	Dissertação (com base num *Projecto de Investigação e Desenvolvimento em Engenharia*)

(a) A escolher da bolsa comum de pares de UCE30 correspondentes a perfis de especialização propostos no Mapa de Formação de 2º Ciclo em Informática da Universidade do Minho e acreditados pela Ordem dos Engenheiros.
(b) UC que inclui a frequência em regime de seminário de módulos oferecidos na bolsa de UCE em Informática complementada com formação específica, dita *de projecto*, orientada ao desenvolvimento de competências para o planeamento, realização e avaliação de projectos de investigação e desenvolvimento em Engenharia. Esta unidade inclui módulos dedicados à gestão e planificação de projectos, metodologias de investigação, gestão de equipas, ética e deontologia profissional, cidadania e ambiente, expressão oral e escrita, entre outros.

[2] Atente-se na semelhança entre o papel a desempenhar pela designação MI ou MEI e o vulgarmente atribuído às designações MBA ou MSC.

2. Mestrado de Informática (MI). Mestrado de cariz científico-tecnológico que acomoda uma grande diversidade de percursos formativos em Informática e cujo principal objectivo é certificar percursos especificamente desenhados por cada aluno de acordo com as suas necessidade e/ou expectativas profissionais. Assim, neste curso são relaxados os requisitos relativos à composição da componente curricular e ao tipo de projecto subjacente à Dissertação. A realização do Mestrado de Informática supõe, pois,

- a certificação prévia num curso de 1º Ciclo em Informática, ou equivalente;
- a realização de uma *Dissertação* com base num *Projecto de Investigação* ou num *Projecto de Investigação e Desenvolvimento em Engenharia*;
- a realização de duas UCE30 livremente escolhidas a partir da oferta disponível no Mapa de Formação em Informática da Universidade do Minho.

É a seguinte a estrutura curricular do MI:

ECTCS	Mestrado de Informática (MI)
30	UC Especialidade (a)
30	UC Especialidade (a)
15	Seminário (b)
45	Dissertação (com base num *Projecto de Investigação ou Investigação e Desenvolvimento em Engenharia*)

(a) A escolher da bolsa comum de UCE30 propostas no Mapa de Formação de 2º Ciclo em Informática da Universidade do Minho.
(b) UC que inclui a frequência em regime de seminário de módulos oferecidos na bolsa de UCE em Informática complementada com formação específica, dita *de projecto*, orientada ao desenvolvimento de competências para o planeamento, realização e avaliação de projectos de investigação. Esta unidade inclui módulos dedicados a métodos de investigação, ética e deontologia profissional, cidadania e ambiente, expressão oral e escrita, entre outros.

O *rationale* para a introdução de um curso com as características do MI no Mapa Pedagógico de formação de 2º Ciclo em Informática, é o reconhecimento de que a complexidade e constante mutação do tecido socio-económico exige percursos formativos que, simultaneamente,

- seja possível certificar em termos de um conjunto preciso de *competências* profissionais em Informática,
- contemplem uma ampla diversificação da oferta disponível em termos dos *conteúdos* específicos a partir dos quais essas competências se desenvolvem.

A acentuação que, no espírito da Declaração de Bolonha, é colocada nas *competências*, torna possivel, a partir de uma oferta ampla mas coerente de UCE30, o desenho de currículos orientados ao aluno. Fá-lo, precisamente, na medida em que *competências* similares podem ser obtidas

através de formações que apresentem alguma diversidade em termos de *conteúdos*. Esta possibilidade parece ser essencial quando se pretende atingir um público alvo que busque re-conversão ou especialização em domínios muito específicos correspondentes a re-orientações de interesses profissionais ou de áreas de negócio do tecido empresarial. Note-se, de resto, que o curso que esta proposta visa adequar era já ele essencialmente orientado à reconversão de profissionais e que, em várias ocasiões, os alunos sugeriram, como principal alteração que valeria a pena introduzir no curso, a oferta de percursos alternativos para resposta a necessidades específicas quer dos próprios quer dos seus empregadores.

O aluno pode, assim, especializar-se, através de uma formação vertical intensiva, que inclui obrigatoriamente uma componente de projecto, em duas áreas que tenham directamente a ver com os seus interesses pessoais ou profissionais. Permite ainda, que com um mesmo curso, se responda a exigências de formação distintas que a envolvente socio-económica constantemente coloca à Universidade.

Poderá questionar-se se este tipo de formação é internacionalmente identificável ou mesmo se não conduz a alguma diluição do perfil de profissional resultante. Relativamente à primeira objecção pode-se aduzir o seguinte:

- No domínio da formação em Informática, onde nos últimos anos, na Europa como nos EUA, se tem assistido a um declínio relativo da procura, não existem ainda padrões de organização curricular de referência, coexistindo antes uma pluralidade de propostas que na sua maioria estão ainda em fase experimental. Um marco importante neste processo foi o *European Computer Sceince Summit*, que em Outubro de 2005, em Zurique, reuniu pela primeira vez mais de uma centena de Directores de Departamento de Informática de toda a Europa. Entre as grandes linhas propostas nessa cimeira, sublinha-se a dissociação dos processos formativos das respectivas certificações, o entendimento integrado da noção de especialização (abrangendo conteúdos, métodos e aplicações) e a necessidade de flexibilização de percursos e diversificação de públicos-alvo.

- Uma análise curricular de cursos similares em Universidades europeias de referência, torna manifesto que, ao contrário do que acontece no caso do 1º Ciclo, não existem curricula com pretensão de "universalidade". Tal explica-se pela enorme diversidade de áreas de especialização relevantes em Informática e pela vontade explícita de adequar os cursos a diferentes exigências de mercado.

- Uma organização curricular de *percurso livre*, similar à que se pretende para o MI, surge como uma possibilidade sempre considerada na oferta formativa em Informática nas escolas de referência na Europa. Em vários casos estudados [3] reconhece-se que o objectivo do 2º Ciclo em Informática é, essencialmente, permitir uma especialização sólida num domínio da área, assumindo uma formação de base ministrada no 1º Ciclo e enfatizando uma preparação *flexível* para um mercado de trabalho em permanente mutação.

[3] nemadamente na *Ecole Polytéchnique Fédérale de Lausanne*, Suiça, na *Technishen Universitat Muenchen*, Alemanha, e no *Imperial College of Science and Technology*, Reino Unido.

- Em particular, uma organização curricular com flexibilidade análoga à proposta para o MI, surge no modelo proposto pela Universidade de Oxford para a formação pós-graduada em Engenharia de Software (disponível em `http://www.softeng.ox.ac.uk/courses/subjects.html`): existem disponíveis 5 temas (com uma organização semelhante à que aqui se propõe para as UCE30).

Relativamente à segunda objecção, sublinha-se que

- O MI, ao estruturar-se através de duas UCE com o carácter acima referido, visa precisamente levar a especialização muito a sério, assegurando uma formação que é ao mesmo tempo vertical nos conteúdos e muito ampla ao nível das aplicações, metodologias e projecto. Aquilo que se pretende atingir está, efectivamente, nos antípodas de uma formação generalista em que formalmente se cobrisse uma vasta gama de tópicos, mas com um grau de profundidade e maturação reduzido e essencialmente livresco.

- Por outro lado, as UCE com base nas quais cada aluno organiza o perfil formativo que mais se lhe adeque, não se definem ao sabor de modas tecnológicas ou conjunturais, mas antes correspondem a áreas base, claramente identificadas e reconhecidas na comunidade académica internacional e no mercado, estruturantes deste domínio científico-tecnológico. A relevância de cada uma delas, e consequentemente, dos perfis profissionais que a sua composição pode originar, é pois indiscutível, assim como a sua marca é claramente identificável tanto em termos académicos como industriais.

Candidaturas e Contactos

Podem-se candidatar os detentores de um curso de 1º ciclo em Informática ou equivalente.

No caso dos cursos das universidades portuguesas, esses cursos correspondem quer aos cursos de Licenciatura antes e depois dos recentes processos de adequação ao modelo de Bolonha, quer à realização de 180 créditos ECTS de um curso de mestrado integrado nesta área.

Mais informações (incluindo documentação necessária às candidaturas) estão disponíveis em `http://msc.di.uminho.pt` ou por e-mail para `msc-info@di.uminho.pt`.

Conteúdo

1

ERS
Engenharia de Redes e Serviços

Sumário

O presente documento descreve uma UCE30 (="Unidade Curricular de Especialidade"), creditada em 30 ECTS, oferecida pelo Grupo de Comunicações por Computador do Departamento de Informática em colaboração com o Departamento de Sistemas de Informação, no 2º ciclo da re-estruturação curricular da oferta formativa em Tecnologias de Informação e Comunicação da Universidade do Minho, no espírito de Bolonha.

A UCE30 ERS, "Engenharia de Redes e Serviços", visa fornecer formação fundamental e especializada sobre um leque seleccionado de tópicos acerca do desenvolvimento e fornecimento de aplicações e serviços de rede. É constituída por cinco módulos de formação relacionados entre si, quatro usados para enfatizar as aplicações e serviços convencionais, sendo o quinto, no formato de seminários, dedicado a novos serviços e aplicações.

Este proposta complementa uma outra UCE30, "Tecnologias e Protocolos de Infra-Estrutura", oferecida pelo mesmo Grupo. Embora seja desejável frequentar a TPI antes da ERS, os pré-requisitos desta última são apenas fundamentos de comunicações e redes de computadores, usualmente adquiridos na formação de 1º ciclo.

Coordenação: *Joaquim Macedo, Bruno Dias, António Costa, Maria João Nicolau.*

1.1 Objectivos

A UCE30 ERS visa fornecer conhecimento especializado em Engenharia de Redes e Serviços, promovendo a formação e aquisição de capacidades e competências para a definição, desenvolvimento e gestão de aplicações e serviços de rede, quer sobre plataformas de comunicação fixas quer móveis. É dada atenção especial à formação detalhada em àreas científicas como a segurança e a gestão de redes, e o desenvolvimento, produção e suporte de aplicações e serviços em redes de comunicações envolvendo conteúdos multimédia. O conteúdo temático desta

unidade curricular, que deve conduzir a uma aprendizagem avançada e coerente nessas matérias, distribui-se por cinco elementos de aprendizagem (6 ECTS cada) relacionados entre si, definidos e assegurados, nas suas componentes científica e pedagógica, pelo grupo de Comunicações por Computador do Departamento de Informática, com colaboração do Departamento de Sistemas de Informação.

1.2 Caracterização Global

1.2.1 Regime e Escolaridade

A UCE apresentada neste documento é oferecida em regime semestral, que sendo mais intensivo e concentrado no tempo, é adequado para uma oferta em formação contínua, permitindo, além disso, a compatibilização com a estrutura de um Mestrado Europeu ERASMUS-MUNDUS, onde esta UCE também é proposta. A sua articulação é realizada em 5 módulos, um dos quais assumindo o formato de seminários CSN1, em que a colaboração com entidades externas ao Departamento de Informática e ao Departamento de Sistemas de Informação assume relevância intencional. Esta estrutura modular, que se detalha na secção 3 deste documento, tem a seguinte escolaridade: 8T + 8TP para os 4 módulos nucleares e 4T + 4TP para a componente dedicada a um tópico avançado e integrador, em formato de seminários.

1.2.2 Áreas Científicas

Áreas Científicas: Informática (30 ECTS)

Áreas Científicas (classificação ACM): As áreas científicas classificadas no *ACM Computing Curricula (1998)* são as seguintes:

C. Computer Systems Organization

- C.2 COMPUTER-COMMUNICATION NETWORKS
  - C.2.2 Network Protocols
  - C.2.3 Network Operations
  - C.2.4 Distributed Systems

E. Data

- E.3 DATA ENCRYPTION
- E.4 CODING AND INFORMATION THEORY

I. Computing Methodologies

- I.4 IMAGE PROCESSING AND COMPUTER VISION

1.2.3 Requisitos

1.2.4 Resultados de Aprendizagem

- Identificar, compreender e discutir os conceitos subjacentes à segurança de redes e às arquitecturas mais utilizadas, induzindo capacidades relevantes para a definição de soluções para protecção das redes e dos serviços de comunicações.

- Reconhecer e compreender as actividades inerentes à gestão de redes, saber escolher e aplicar tecnologias de gestão disponíveis, atendendo às especificidades de cada cenário de implementação.

- Efectuar a análise, concepção e desenvolvimento de aplicações em rede em contextos de tempo real, sendo capaz de seleccionar e usar plataformas avançadas de programação distribuída.

- Compreender os príncipios, técnicas e algoritmos usados para representação, compressão e processamento de conteúdos multimédia.

- Tendo como base a formação num tópico avançado e integrador (que no contexto da UCE30 ERS se realiza no formato de seminários), desenvolver capacidades adicionais, nomeadamente identificar problemáticas emergentes das tecnologias abordadas nos elementos nucleares de aprendizagem e descrever as estratégias e mecanismos actuais que, mais relevantemente, contribuam para a sua resolução.

1.3 Estrutura Curricular

1.3.1 Estrutura Base

ERS - Engenharia de Redes e Serviços - (30 ECTS):

- Segurança em Redes (6 ECTS),

- Gestão de Redes e Serviços (6 ECTS),

- Programação Distribuída e de Tempo-Real (6 ECTS),

- Serviços e Sistemas Multimédia (6 ECTS),

- Seminários ERS(6 ECTS).

1.3.2 Métodos de Ensino

- Exposição de conceitos e análise de casos de estudo.
- Trabalho de grupo em pequenos casos de estudo, com recurso, sempre que tal se mostrar pertinente, a ferramentas informáticas específicas, nomeadamente de simulação.
- Trabalho de grupo em implementações de casos, com recurso a equipamentos de comunicação, com orientação directa da equipa docente afecta à UCE.

1.3.3 Avaliação

O processo de avaliação visa aferir não apenas os conhecimentos e competências adquiridas, mas também a capacidade revelada na sua integração e aplicação nos casos de estudo e seminários. Assim, existirá um exame final que abordará matérias dos diferentes módulos, em média pesada com a avaliação contínua e com o resultado do trabalho dos seminários (ferramentas, relatório, apresentação oral, etc), que funcionará como componente integradora sobre um tópico avançado.

1.3.4 Contéudos Programáticos

Os conteúdos programáticos são agrupados em 4 módulos temáticos, nucleares e complementados com a formação num tema avançado e integrador, em formato de seminários. Nesta fase, os conteúdos programáticos são apresentados apenas em língua inglesa (tal como constam da proposta Erasmus Mundus) e conforme os detalhes apresentados em seguida.

ERS - NETWORKS AND SERVICES ENGINEERING (30 ECTS)

Objectives:

The Core Unit ERS aims at providing core background on Computer Networks, promoting knowledge and skills in defining, deploying and managing network services, both in fixed and mobile network environments. Special attention is given to hot topics in today's networks such as security and management issues, distributed multimedia applications and systems. In this way, the corresponding syllabus is envisioned to offer advanced learning outcomes in these subjects.

1. **Network Security**

   2nd Semester, 1st Year, 6 ECTS Contents:

   (i) Security and criptography principles: authentication, key management techniques and access control. Security at network level. Security of applications and services. Information systems security. Security planning.

   (ii) Multimedia security: conditional access systems, digital radio, DVD copyright, watermarking (architectures, legal sigues, copyright protection, information theory and game theory for watermarking, robust hashing and applications).

   Learning outcomes:

(i) to identify and discuss the concepts of network security and the major standardized architectures envisioned to protect networks and services;

(ii) to understand the principles and functionality of cryptography, authentication and management techniques for access control;

(iii) at the end of the course, the students should be able to define and deploy secure network infrastructures and services, adapted to the investment and required security level.

Bibliography:

* S. Malik, "Network Security Principles and Practices". Cisco Press. 2002.

* M. Rhodes-Ousley, et al. "Network Security: The Complete Reference", McGraw-Hill, 2003.

* C. Kaufman et al., "Network Security: Private Communication in a Public World". 2ed., Prentice Hall, 2002.

* G. DeLaet, G. Schauwers, "Network Security Fundamentals", Cisco Press Fundamentals Series, 2004.

2. Management of Networks and Services

2nd Semester, 1st Year, 6 ECTS

Contents:

(i) Network and capacity planning. Network management principles and architectures.

(ii) OSI and INMF/SNMP management frameworks: data model, architecture and management entities, communication paradigms, management objects and related standards.

(iii) Services management. Quality of service as a recent and one of the most relevant issues in network management.

(iv) Additional technologies: web services, mobile management agents, management by delegation, management policies. Current best practices.

Learning outcomes:

(i) to plan network design and resources provisioning according to specific requirements and constrains;

(ii) to demonstrate basic knowledge on important network management standards and technologies;

(iii) to identify the main network management tasks, including the use of public domain and proprietary management tools;

(iv) to select and apply appropriate technologies for accomplishing network management tasks, identifying costs and limitations in different implementation scenarios;

(v) to analyse and discuss the most relevant approaches inherent to management architectures and mechanisms, resulting from on-going research.

Bibliography:

* B. Dias, "Network Services Magement Framework", PhD Thesis, Universidade do Minho, 2005.

* D. Mauro, K. Schmidt, "Essential SNMP, 2nd Ed.", O'Reilly, 2003.

* J. Kretchmar, "Open Source Network Administration", Prentice Hall, 2003.

* J. Vasseur, M. Pickavet, P. Demeester, "Network Recovery", Morgan Kaufmann, 2004.

* J. Strassner, "Policy-Based Network Management", Morgan Kaufmann, 2003.

* M. Subramanian, "Network Management An introduction to principles and practice", Addison Wesley, 1999.

* S. Morris, "Network Management, MIBs and MPLS: Principles, Design and Implementation", Addison Wesley, 2003.

* W. Stallings, "SNMP, SNMPv2, SNMPv3 and RMON 1 and 2", Addison-Wesley, 1998.

3. Real-time and Distributed Programming

2nd Semester, 1st Year, 6 ECTS

Contents:

(i) Network programming: packet capturing and transmission (raw sockets, libpcap, libnet). Programming BSD sockets. Concurrent programming and client-server model. Remote procedure call (RPC/XDR, CORBA, JavaRMI). Programming Web applications and Web Services (SOAP, JSP, Servlets).

(ii) Real Time Java, ORBs, and DDS specification. Applications: Telecomander, real time user location in wireless networks, real time switching and planning.

Learning outcomes:

(i) to identify application scenarios, analyse requirements, design and be able to develop programs for distributed programming in real-time contexts using the most modern techniques and environments such as CORBA, Web Services and Real Time Java;

(ii) to specify and implement software components for network devices, applying theoretical concepts in networking software development;

(iii) to differentiate abstraction levels in network applications programming;

(iv) at the end of the course, the student should be able to select the most effective environment and programming language, in the context of the requirements of the real-time distributed application to be developed.

Bibliography:

* P. Dibble, "Real-Time Java Platform Programming", Sun Microsystems Press, 2002.

* M. Donahoo et al., "TCP/IP Sockets in C: Practical Guide for Programmers", Practical Guides Series, 2000.

* D. Comer et al., "Internetworking with TCP/IP, Vol. III: Client-Server Programming and Applications - Linux/Posix SocketsVersion", Prentice-Hall, 2000.

* J. Snell et al., "Programming Web Services with SOAP", O'Reilly Media, 2001.

* M. Aleksy, "Implementing Distributed Systems with Java and CORBA", Springer, 2005.

4. **Multimedia Systems and Applications**

2nd Semester, 1st Year, 6 ECTS

Contents:

(i) Development and Properties of Multimedia Systems. Classification of Media. Multimedia Computing.

(ii) Text Compression and Processing. Digital Audio Compression and Processing. Digital Image and Video Compression and Processing.

(iii) Models and algorithms for content-based networking.

Learning outcomes:

(i) to identify and compare different media;

(ii) to recognize the requirements and select the appropriate tools for multimedia data acquisition and storage;

(iii) to define, discuss and compare several loss or lossless media compression methods, algorithms and standards;

(iv) to apply suitable methods and tools for media compression;

(v) to define, discuss, compare and apply basic temporal signal, audio, image and video processing tools;

(vi) to select and compare suitable tools and devices for multimedia content creation, management, storage and visualization.

Bibliography:

* M. Mandal, "Multimedia Signals and Systems", Kluwer, 2001.

* F. Halsall, "Multimedia Communications: Applications, Networks, Protocols and Standards", Addison-Wesley, 2000.

5. **Seminars ERS**

2nd Semester, 1st Year, 6 ECTS

Contents:

These seminars will address emerging topics related to ERS core unit. They may run under an intensive or progressive (along the semester) regime.

Learning outcomes: it will depend on the nature and contents of the lectured seminars.

2

TPI
Tecnologias e Protocolos de Infra-Estrutura

Sumário

O presente documento descreve uma UCE30 (="Unidade Curricular de Especialidade"), creditada em 30 ECTS, oferecida pelo Grupo de Comunicações por Computador do Departamento de Informática em colaboração com o Departamento de Sistemas de Informação, no 2^o ciclo da re-estruturação curricular da oferta formativa em em Tecnologias de Informação e Comunicação da Universidade do Minho, no espírito de Bolonha.

A UCE30 TPI, "Tecnologias e Protocolos de Infra-Estrutura", visa fornecer formação fundamental e especializada em Redes de Computadores. Promove o conhecimento e as aptidões em tecnologias de rede actuais e emergentes, redes de acesso ou de core, fixas ou móveis com ou sem fios. Consiste em cinco módulos de formação relacionados entre si, sendo um deles no formato de seminários dedicados a tecnologias, protocolos ou teorias emergentes.

Esta proposta é complementada por uma outra UCE30, Engenharia de Redes e Serviços*, oferecida pelo mesmo Grupo.*

Coordenação: *Alexandre Santos, Paulo Carvalho, Pedro Sousa, M. Solange Lima, Vasco Freitas.*

2.1 Objectivos

A UCE30 TPI visa fornecer formação fundamental e especializada em Redes de Computadores, promovendo o conhecimento e as aptidões em tecnologias actuais e emergentes em ambientes de interligação em rede, tanto em contextos móveis como em contextos de ligação fixa. O conteúdo temático desta unidade curricular, que deve conduzir a uma aprendizagem avançada nessas matérias, estrutura-se em cinco elementos de aprendizagem (6 ECTS cada) relacionados entre si,

definidos e assegurados, nas suas componentes científica e pedagógica, pelo grupo de Comunicações por Computador do Departamento de Informática, com colaboração do Departamento de Sistemas de Informação.

Esta Unidade TPI foca, principalmente, tecnologias e protocolos correspondentes aos primeiros níveis protocolares (até à camada de rede) das pilhas OSI e TCP/IP.

2.2 Caracterização Global

2.2.1 Regime e Escolaridade

A UCE apresentada neste documento é oferecida em regime semestral, que sendo mais intensivo e concentrado no tempo, é adequado para uma oferta em formação contínua, permitindo, além disso, a compatibilização com a estrutura de um Mestrado Europeu ERASMUS-MUNDUS, onde esta UCE também é proposta. A sua articulação é realizada em 5 módulos, um dos quais assumindo o formato de seminários CSN1, em que a colaboração com entidades externas ao Departamento de Informática e ao Departamento de Sistemas de Informação assume relevância intencional. Esta estrutura modular, que se detalha na secção 3 deste documento, tem a seguinte escolaridade: 8T + 8TP para os 4 módulos nucleares e 4T + 4TP para a componente dedicada a um tópico avançado e integrador, em formato de seminários.

2.2.2 Áreas Científicas

Áreas Científicas: Informática (30 ECTS)

Áreas Científicas (classificação ACM): As áreas científicas classificadas segundo o *ACM Computing Curricula (1998)* são as seguintes:

C. Computer Systems Organization

C.2 COMPUTER-COMMUNICATION NETWORKS

C.2.1 Network Architecture and Design

C.2.2 Network Protocols

C.2.3 Network Operations

C.2.5 Local and Wide-Area Networks

C.2.6 Internetworking

C.4 PERFORMANCE OF SYSTEMS

2.2.3 Requisitos

2.2.4 Resultados de Aprendizagem

- Explicar os principais conceitos teóricos subjacentes às principais tecnologias de acesso e core, sabendo exemplificar e discutir o uso dessas tecnologias em cenários de redes distintos;

- Identificar e descrever os diversos tipos de redes móveis existentes, atendendo à sua arquitectura, estrutura protocolar, função e aplicabilidade.

- Seleccionar os serviços de rede mais apropriados para suportar uma dada aplicação em ambiente móvel.

- Identificar e discutir as diferenças e conceitos subjacentes aos protocolos IPv4 e IPv6, e compreender os princípios e funcionalidades inerentes à mobilidade, concretizando para IPv4 e IPv6.

- Reconhecer a motivação para a integração de serviços com requisitos de qualidade variados, e planear essa integração com base em mecanismos e protocolos específicos.

- Tendo como base a formação num tópico avançado e integrador (que no contexto da UCE30 TPI se realiza no formato de seminários), desenvolver capacidades adicionais, nomeadamente identificar problemáticas emergentes das tecnologias abordadas nos elementos nucleares de aprendizagem e descrever as estratégias e mecanismos actuais que, mais relevantemente, contribuam para a sua resolução.

2.3 Estrutura Curricular

2.3.1 Estrutura Base

TPI - Tecnologias e protocolos de Infra-Estrutura - (30 ECTS):

- Tecnologias de Rede de Acesso e Core (6 ECTS),

- Redes de Comunicações sem Fios e Móveis (6 ECTS),

- Redes IP Avançadas (6 ECTS),

- Redes Multi-serviço (6 ECTS),

- Seminários TPI (6 ECTS).

2.3.2 Métodos de Ensino

- Exposição de conceitos e análise de casos de estudo.
- Trabalho de grupo em pequenos casos de estudo, com recurso, sempre que tal se mostrar pertinente, a ferramentas informáticas específicas, nomeadamente de simulação.
- Trabalho de grupo em implementações de casos, com recurso a equipamentos de comunicação, com orientação directa da equipa docente afecta à UCE.

2.3.3 Avaliação

O processo de avaliação visa aferir não apenas os conhecimentos e competências adquiridas, mas também a capacidade revelada na sua integração e aplicação nos casos de estudo e seminários. Assim, existirá um exame final que abordará matérias dos diferentes módulos, em média pesada com a avaliação contínua e com o resultado do trabalho dos seminários (ferramentas, relatório, apresentação oral, etc), que funcionará como componente integradora sobre um tópico avançado.

2.3.4 Contéudos Programáticos

Os conteúdos programáticos são agrupados em 4 módulos temáticos, nucleares e complementados com a formação num tema avançado e integrador, em formato de seminários. Nesta fase, os conteúdos programáticos são apresentados apenas em língua inglesa (tal como constam da proposta Erasmus Mundus) e conforme os detalhes apresentados em seguida.

TPI - INFRASTRUCTURE TECHNOLOGIES AND PROTOCOLS (30 ECTS)

Objectives:

The Core Unit TPI aims at providing core background on Computer Networks, promoting knowledge and skills in current and emerging technologies and protocols, both in fixed and mobile network environments. In this way, the corresponding syllabus is envisioned to offer advanced learning outcomes in these subjects.

1. **Access and Core Network Technologies**

   1st Semester, 1st Year, 6 ECTS

   Contents:

   (i) Protocol and access technologies: xDSL, PPP, Cable Modems and DOCSIS, Powerline, Ethernet (IEEE 802.3), Fast Ethernet (IEEE 802.3u), Giga and 10Giga Ethernet, Wireless Lan (802.11x) (overview), Token Ring (IEEE 802.5) and Token Bus (IEEE 802.4).

   (ii) Metropolitan and Wide Area Network (WAN) technologies: FDDI (ANSI X3T9.5), DQDB (IEEE 802.6), Frame Relay, Assynchrounous Transfer Mode (ATM), Switched Multimegabit Data Service (SMDS), Multiprotocol Label Switching (MPLS) and Generalised MPLS.

(iii) Physical framing: Digital Hierarchies, SONET, OC-x and PONs.

Learning outcomes:

(i) to explain the main theorectical concepts underpinning the major access and core network technologies;

(ii) to exemplify and discuss the use of such technologies over distinct networking scenarios;

(iii) to understand the principles of physical framing.

Bibliography:

* A. Gumaste, T. Antony, "First Mile Access Networks and Enabling Technologies", Cisco Press, 2004.

* G. Kramer, "Ethernet Passive Optical Networks", McGraw-Hill, Communications Engineering, 2005.

* A. Gillespie, "Access Networks: Technology and V5 Interfacing", Artech House Publishers, 1999.

* S. Ovadia, "Broadband Cable TV Access Networks: From Technologies to Applications", Prentice Hall, 2001.

* S. Shepard, "Metro Area Networking", McGraw-Hill, Networking Professional, 2003.

* D. Minoli et al., "Ethernet-Based Metro Area Networks", McGraw-Hill, 2002.

2. Wireless and Mobile Communication Networks

1st Semester, 1st Year, 6 ECTS

Contents:

(i) Wide area mobile networks: introduction to cellular networks (GSM, UMTS, 4G), architectures and evolution; main services (voice, SMS, CSD, HSCSD, GPRS, EDGE). Simulation and evaluation of CDMA-UMTS. Description and evaluation of MIMO architectures in cellular networks.

(ii) Satellite mobile networks. "Trunking"(Tetra) networks.

(iii) Wireless Local Area Networks (WLANs) reference models (IEEE802.11 and HyperLAN): overall architecture, radio interfaces, propagation and protocol stacks.

(iv) Wireless Personnel Area Networks (WPANs) reference models (Bluetooth and IEEE802.15): overall architecture, main concepts and examples and protocol stacks.

(v) Short-range wireless transmission systems: optical systems (IrDA) and radio-based systems (Bluetooth).

(vi) Integration and interoperability of mobile networks: association and handover, global authentication, security issues.

(vii) Intelligent network platforms. Access to network services (OSA/Parlay, Mexe, GMLC). Technologies and services to support applications (SMS, WAP, MMS, J2ME). Positioning and Geo-location in mobile networks.

Learning outcomes:

(i) to describe the several types of mobile networks, explaining their evolution and discussing their funcionality, complementarity and use;

(ii) to describe the architecture of cellular networks, interrelating the main components of the architecture and understanding their role;

(iii) to characterize the underlying protocol stacks of WLANs and WPANs; to select the most convenient network services to support applications in mobile environments.

Bibliography:

* H. Holma and A. Toskala editors, "WCDMA for UMTS - Rádio Access For Third Generation Mobile Communications", John Wiley & Sons, 2000.

* J. Tisal, "GSM cellular radio telephony", John Wiley & Sons, 1997.

* G. Held, "Data over wireless networks Bluetooth, WAP, and wireless LANs", McGraw-Hill, 2001.

* P. Nicopolitidis et al., "Wireless Networks", John Wiley & Sons, 2003.

* Y. Lin, I. Chlamtac, "Wireless and mobile network architectures", John Wiley & Sons, 2001.

3. Advanced IP Networks

1st Semester, 1st Year, 6 ECTS

Contents:

(i) Internetworking principles. IPv4 and IPv6: data units, addressing, unicast and multicast routing algorithms and protocols; IPv6 extensions and functionalities.

(ii) Introduction to mobility in IP networks: impact of mobility in the protocol stack. Concepts of macro e micromobility. Mobile IPv4 e IPv6: addressing, routing, roaming, fast handover and recovery. Quality of service and security issues in Mobile IP Networks.

Learning outcomes:

(i) to identify and discuss the concepts underlying IPv4 and IPv6 protocols, and their main characteristics and functionality;

(ii) to understand the principles and funcionality of Mobile IP, explaining its concretization in IPv4 and IPv6.

Bibliography:

* W. Stallings, "Data and Computer Communications, 7th Ed.", Prentice Hall, 2003.

* S. Hagen, "IPv6 Essentials", OReilly, 2002.

* C. Perkins, "Mobile IP - Design Principles ans Practices", Addison-Wesley, 1998.

* D. Wisely, P. Eardley, L. Burness, "IP for 3G - Networking Technologies for Mobile Communications", Wiley, 2002.

* J. Chen, T. Zhang, "IP-based Next-Generation Wireless Network", Wiley, 2004.

* H. Soliman, "Mobile IPv6 - Mobility in a wireless Internet", Addison-Wesley, 2004.

4. **Multiservice Networking**

1st Semester, 1st Year, 6 ECTS

Contents:

(i) Service integration: motivation and principles.

(ii) Quality of Service (QoS): principles, parameters, architectures (e.g. Intserv, Diffserv). Traffic control and congestion avoidance mechanisms. QoS routing. Services and services contracts specification, configuration and management.

(iii) Reliable and unreliable transport services: unicast and multicast cases. Protocols oriented to QoS and real-time support (e.g. RTP/RTCP/RTSP). Resource reservation and signaling protocols (e.g. RSVP, H.323, SIP). QoS mapping. Voice over IP and Video/TV over IP.

Learning outcomes:

(i) to recognize the need for service integration and discuss the problematic of measuring/evaluating and guaranteeing QoS in the Internet;

(ii) to explain and differentiate current QoS architectures, technologies and mechanisms, and exemplify its effective deployment;

(iii) to understand the problem of network congestion and to identify the mechanisms and their best practices in order to avoid it;

(iv) to select adequate transport services based on the characteristics of applications and user services;

(v) to identify, define and relate the main concepts and algorithms for supporting real-time applications with group communication and QoS requirements;

(vi) to identify, understand and choose the main signaling and resource reservation solutions in the Internet;

(vii) to understand theoretical and practical concepts behind services such as VoIP and Video/TV over IP.

Bibliography:

* H. Chao, X. Guo, "Quality of Service Control in High-Speed Networks", Wiley, 2001. * S. Jha, M. Hassan, "Engineering Internet QoS", Artech House Inc., 2002. * Z. Wang, "Internet QoS: Architectures and Mechanisms for Quality of Service", The Morgan Kaufmann Series in Networking, 2001.

5. Seminars TPI

1st Semester, 1st Year, 6 ECTS

Contents:

The seminars will address emerging topics related to the TPI core unit. They may run under an intensive or progressive (along the semester) regime.

Learning outcomes: it will depend on the nature and contents of the lectured seminars.

3

CMU
Computação Móvel e Ubíqua

Sumário

A Unidade Curricular de Especialidade em Computação Móvel e Ubíqua tem como objectivo formar profissionais com conhecimentos sólidos e competências avançadas na concepção, realização e avaliação de soluções informáticas para sistemas caracterizados pela forte mobilidade dos seus utilizadores e pela ubiquidade dos serviços de computação e comunicações. A Computação Móvel e Ubíqua representa uma mudança de paradigma, de uma era marcada pelos sistemas informáticos centrados no computador de secretária , para uma era marcada pela ubiquidade das comunicações e dos dispositivos computacionais, que assim passam a ser parte integrante do espaço físico em que vivemos e das nossas mais variadas tarefas do dia-a-dia. Esta mudança de paradigma tem naturalmente um impacto muito forte na concepção e concretização de sistemas informáticos, originando novos desafios em áreas tão diversas como a Engenharia de Software, a Interacção Humano-Computador ou os Sistemas Distribuídos. Tendo em conta que os desafios associados à computação móvel e ubíqua se caracterizam por uma forte multi-disciplinaridade, esta UCE30, ainda que centrada cientificamente nos sistemas distribuídos e de comunicações, pode ser do potencial interesse de alunos com perfis muito diversos, que poderão encontrar aqui um complemento adequado a sua formação numa área mais específica. A UCE30 em Computação Móvel e Ubíqua resulta de uma adaptação de algumas disciplinas do Mestrado em Sistemas Móveis no sentido de constituir uma oferta de formação coerente e devidamente integrada no 2º ciclo da re-estruturação curricular da oferta formativa em Informática da Universidade do Minho, no espírito do Processo de Bolonha.

Coordenação: *Rui José, Adriano Moreira, Helena Rodrigues, Pedro Branco.*

3.1 Contexto e Objectivos

A Computação Móvel e Ubíqua representa um novo passo na evolução da Informática, uma mudança de uma era marcada pelo computador de secretária e pelo correspondente paradigma de

computação *desktop*, para uma era marcada pela ubiquidade das comunicações e pela relação de um para muitos entre as pessoas e os dispositivos computacionais, que assim passam a ser parte integrante do nosso espaço físico e uma presença constante nas mais variadas tarefas do nosso dia-a-dia. O resultado é uma clara transição de paradigma com um forte impacto na concepção e concretização de sistemas informáticos, originando novos desafios em áreas tão diversas como, por exemplo, as Comunicações, a Engenharia de Software, a Interacção Humano-Computador ou os Sistemas Distribuídos.

A UCE30 em Computação Móvel e Ubíqua visa desenvolver competências avançadas na concepção, desenvolvimento e avaliação de sistemas informáticos baseados nos paradigmas da computação móvel e ubíqua. A formação proporcionada é assumidamente multi-disciplinar, pretendendo-se com isso alcançar um tratamento integrado das questões dos sistemas móveis e ubíquos e explorar o estabelecimento de pontes entre as áreas-fronteira das várias disciplinas de referência. A estrutura desta UCE30 inclui 5 módulos de aprendizagem, sendo que quatro desses módulos são temáticos, com 5 ECTS cada, e o outro é um módulo de projecto integrador com 10 ECTS. Nos módulos temáticos serão desenvolvidos os principais conceitos teóricos subjacentes a esta UCE30, sendo esse processo sempre interligado com o projecto integrador.

Esta UCE30 resulta da evolução e integração de algumas disciplinas do Mestrado em Sistemas Móveis, beneficiando por isso da experiência científica e pedagógica desenvolvida durante as várias edições desse mestrado. Além disso, e seguindo as recomendações mais recentes no que se refere à formação de 2^{o}ciclo, esta UCE30 é também fortemente apoiada em actividade de investigação já consolidadas através de vários anos de projectos com financiamento nacional e europeu nas áreas dos sistemas móveis e ubíquos. Essas actividades são um elemento importante não só por contribuírem para a formação científica dos docentes mas também pelo enquadramento e recursos que podem proporcionar a esta UCE30. Por outro lado, também se espera que esta UCE30 seja ao mesmo tempo uma forma de promover a capacidade de I&D na área dos Sistemas Móveis e a capacidade de concepção de serviços e aplicações inovadoras. Embora a indústria de software e serviços seja o mercado-alvo mais natural, espera-se que as competências proporcionadas por esta UCE30 possam também ser do interesse de diversos outros sectores, como por exemplo as empresas operadoras de serviços móveis ou os grandes clientes de TICs, com especial incidência nos sectores da distribuição, comércio e transportes.

A natureza multi-disciplinar da Computação Móvel e Ubíqua faz com que esta UCE30 possa ser do interesse de alunos com perfis bastante variados, sendo essa potencial diversidade encarada como uma mais-valia importante no seu próprio funcionamento. UCEs na área das comunicações podem complementar esta formação no sentido de uma maior orientação para os serviços móveis e para o mercado das operadoras móveis. As UCEs em Computação Gráfica ou em Sistemas Inteligentes podem ser especialmente adequadas numa perspectiva de desenvolvimento de competências na criação de ambientes inteligentes e interactivos. As UCEs em Engenharia de Aplicações e em Sistemas Distribuídos podem complementar uma formação mais vocacionada para a concepção de serviços e aplicações distribuídas.

3.2 Caracterização Global

3.2.1 Regime e Escolaridade

Esta UCE30 poderá funcionar em regime semestral ou anual de acordo com o enquadramento que venha a ser definido no âmbito dos cursos de pós-graduação que venha a integrar. No que se refere à escolaridade, e tendo como base a existência de 4 módulos e um projecto integrador, a escolaridade semanal, calculado com base num regime de funcionamento semestral, é a seguinte: 8T + 4TP + 4PL +4OT (para acompanhamento presencial do desenvolvimento do projecto integrador).

3.2.2 Áreas Científicas

Áreas Científicas: Informática (30 ECTS)

Áreas Científicas (classificação ACM):

- C. Computer Systems Organization/ C.2 COMPUTER-COMMUNICATION NETWORKS/ C.2.1 Network Architecture and Design
- C. Computer Systems Organization/C.2 COMPUTER-COMMUNICATION NETWORKS/C.2.4 Distributed Systems
- D. Software/D.1 PROGRAMMING TECHNIQUES/D.1.3 Concurrent Programming
- D. Software/D.2 SOFTWARE ENGINEERING/ D.2.11 Software Architectures
- D. Software/ D.2 SOFTWARE ENGINEERING / D.2.12 Interoperability
- H. Information Systems/ H.5 INFORMATION INTERFACES AND PRESENTATION/ H.5.2 User Interfaces

3.2.3 Requisitos

Embora esta não seja uma UCE30 vocacionada para o ensino da programação, a sua forte componente prática e de projecto implica alguma experiência de programação distribuída, particularmente em Java e C, sendo o nível de competências aquele que normalmente se associa a uma formação de primeiro ciclo na área das TICs.

3.2.4 Resultados de Aprendizagem

- Explicar os fundamentos gerais da computação ubíqua e aplicá-los na concepção, desenvolvimento e operação de sistemas informáticos.

- Explicar o funcionamento dos principais serviços de comunicações para utilizadores móveis e avaliar a sua utilização em ambientes de mobilidade.
- Explicar os principais conceitos envolvidos na concepção de interfaces humano-computador e a sua importância na usabilidade do sistema.
- Identificar os diferentes tipos de modelos para arquitecturas de software e seleccionar o suporte de sistema mais adequado para o desenvolvimento de uma solução móvel ou ubíqua específica
- Planear e executar em equipa sistemas móveis ou ubíquos de complexidade média, tendo em conta os vários aspectos multidisciplinares envolvidos e avaliando um conjunto vasto de abordagens e tecnologias alternativas do ponto de vista da sua aplicabilidade num cenário específico.
- Conceber e aplicar uma estratégia de avaliação de usabilidade no âmbito de um processo de desenvolvimento de um sistema de computação móvel ou ubíqua

3.3 Estrutura Curricular

3.3.1 Estrutura Base

A UCE30 em Computação móvel e ubíqua encontra-se organizada de acordo com os seguintes módulos:

- Sistemas de computação ubíqua (5 ECTS),
- Redes e serviços de comunicações móveis (5 ECTS),
- Interacção e usabilidade (5 ECTS),
- Suporte de sistema para ambientes móveis e ubíquos (5 ECTS),
- Projecto (10 ECTS).

3.3.2 Métodos de Ensino

O plano de formação desta UCE30 integra um conjunto variado de metodologias de ensino com as quais se pretende proporcionar uma experiência de aprendizagem mais rica, capaz de apelar à diversidade de estilos de aprendizagem dos alunos, e maximizar a forte interdependência entre o teórico e o prático que caracteriza a computação móvel e ubíqua. Nesse sentido, e para além do clássico método expositivo que será utilizado na apresentação geral de conceitos, merecem um forte destaque os métodos de aprendizagem activa, com os quais se pretende alcançar um maior envolvimento dos alunos no processo de aprendizagem. Esse envolvimento é particularmente relevante dado o conjunto muito significativo de objectivos de alto nível incluídos nos objectivos

de aprendizagem desta UCE, o que necessariamente implica também a prática durante as aulas de actividades como a análise ou a avaliação. As sessões presenciais irão portanto incluir diversas actividades de trabalho como questões abertas, análise de casos específicos, análise de vídeos, apresentações sobre artigos ou temas do programa. Adicionalmente, a aprendizagem far-se-á também com base na experiência de aprendizagem proporcionada pelo projecto integrador. Mais do que servir para aplicação prática de conceitos já abordados nas teóricas, o projecto deverá ser sobretudo um referencial para toda a UCE30 permitindo também que a experiência resultante da sua realização proporcione um contexto de aprendizagem motivador para a abordagem dos fundamentos teóricos.

3.3.3 Avaliação

A avaliação desta UCE30 é uma média pesada entre um exame final, que visa sobretudo aferir o conhecimento adquirido nos vários módulos, uma avaliação contínua calculada com base no trabalho desenvolvido ao longo do período lectivo (e.g. relatórios) e a avaliação do projecto integrador. A diversidade de métodos de avaliação permite aferir não só os conhecimentos adquiridos mas também a capacidade de os relacionar e aplicar na concretização de sistemas informáticos. Pretende-se também ao incluir uma avaliação contínua, incluir uma componente de avaliação formativa que permita aos alunos obter ainda antes do final do período lectivo algum feedback sobre o seu progresso e desempenho.

3.3.4 Conteúdos Programáticos

SISTEMAS DE COMPUTAÇÃO UBÍQUA

Descrição.
Este módulo aborda os fundamentos da computação ubíqua, não apenas na perspectiva de explicar os principais conceitos associados a esse paradigma computacional, mas sobretudo numa perspectiva de desenvolvimento das competências necessárias para concretizar em sistemas reais os conceitos e abordagens apresentados. Assim, serão analisados os principais desafios na realização desses sistemas e analisadas diversas abordagens alternativas que permitem concretizar a visão da computação como tecnologia ubíqua.

Programa resumido.

- Fundamentos de Computação Ubíqua
- Fusão entre ambientes físicos e virtuais
- Técnicas, tecnologias e sistemas de localização
- Modelos de interacção

- Sistemas de Computação Ubíqua
- Desafios ao Desenvolvimento em ambientes reais
- Implicações de privacidade e outras questões sociais

Resultados de Apreendizagem Específicos:

- Explicar os princípios gerais da Computação Ubíqua e analisar sistemas informáticos do ponto de vista da sua aplicação desses princípios
- Discutir a um nível profissional os principais factores tecnológicos e sociais que enquadram a transição de um paradigma desktop para a Computação Ubíqua.
- Explicar a contribuição das áreas tecnológicos mais relevantes para a concretização da visão da Computação Ubíqua.
- Aplicar abordagens de referência na criação de sistemas informáticos de complexidade média segundo os princípios do paradigma de Computação Ubíqua
- Analisar o impacto da experiência de utilização e das normas sociais na criação de aplicações de Computação Ubíqua.

REDES E SERVIÇOS DE COMUNICAÇÕES MÓVEIS

Descrição.
As redes móveis e os serviços que suportam são um elemento de base em sistemas de computação móvel e ubíqua, sendo portanto necessário um conhecimento adequado das características dos vários tipos de rede e do tipo de funcionalidade que proporcionam. Este módulo aborda um conjunto variado de tecnologias de redes móveis e abrange questões que vão desde os meios de transmissão utilizados e as suas implicações até aos serviços de valor acrescentado suportados pelos operadores de redes móveis celulares. A formação proporcionada deverá permitir o desenvolvimento de serviços avançados assentes em tecnologias de comunicações móveis.

Programa resumido.

- Redes móveis de área alargada: Redes móveis celulares (GSM, UMTS, redes de 4ª geração). Conceito de rede celular e as várias gerações de redes móveis celulares. Arquitectura das redes móveis celulares. Principais serviços suportados (voz, SMS, CSD, HSCSD, GPRS, EDGE). Redes móveis por satélite. Redes de "trunking"(Tetra).
- Redes de área local sem fios e redes de área pessoal: Modelos de referência. Redes de área local sem fios (IEEE802.11 e HyperLAN). Arquitectura, interfaces de rádio, características de propagação e pilhas protocolares. Redes de área pessoal (Bluetooth e IEEE802.15). Conceito e exemplos, arquitectura e pilhas protocolares das redes de área pessoal.

- Sistemas de transmissão sem fios de curto alcance: Sistemas ópticos (IrDA). Sistemas por rádio (Bluetooth).

- Integração e interoperabilidade de redes móveis: Associação e handover. Autenticação global. Aspectos de segurança.

- Serviços das plataformas de rede: Plataformas de rede inteligente. Acesso a serviços de rede (OSA/Parlay, Mexe, GMLC). Tecnologias e serviços de suporte a aplicações (SMS, WAP, MMS, J2ME). Posicionamento e localização geográfica em redes móveis.

Resultados de Apreendizagem Específicos:

- Descrever na generalidade os diversos tipos de redes móveis existentes do ponto de vista da sua função, discutir a sua complementaridade, e explicar a forma como evoluíram ao longo do tempo;

- Descrever a arquitectura das redes móveis celulares e relacionar os seus componentes com a função que cada um desempenha no funcionamento global do sistema;

- Descrever a arquitectura e as pilhas protocolares das redes de área local e pessoal sem fios, bem como dos sistemas de transmissão sem fios de curto alcance;

- Seleccionar os serviços de rede mais apropriados para suportar uma dada aplicação em ambiente móvel.

INTERACÇÃO E USABILIDADE

Descrição.
Uma consequência fundamental da utilização de dispositivos móveis é a quebra de muitos dos pressupostos que estão na base do paradigma desktop. O modelo de interacção é um aspecto em que isso de faz notar com particular incidência, obrigando por isso a abordagens radicalmente diferentes na concepção de interfaces que tenham em conta a variedade de modalidades de interacção disponíveis, a diversidade de contextos em que a interacção pode ocorrer e a possibilidade de colaboração entre vários utilizadores. Adicionalmente, o facto de se pretender desenvolver tecnologia para integrar o dia-a-dia das pessoas obriga a abordagens de concepção que integrem desde início a avaliação da usabilidade e a aceitação social dessas tecnologias. Este módulo proporciona a formação de competências base para o desenvolvimento de mecanismos de interacção adequadas às características dos novos paradigmas computacionais e para a sua avaliação no contexto da experiência de utilização que proporcionam.

Programa resumido.

- Fundamentos de Interacção Humano-Computador

- Interacção Humano-Computador em Ambientes Ubíquos
- Usabilidade no processo de concepção de software
- Modelação do utilizador
- Análise de tarefas
- Técnicas e metodologias de avaliação

Resultados de Apreendizagem Específicos:

- Explicar os principais conceitos envolvidos na concepção de um interface humano-computador.
- Descrever as principais estratégias utilizadas na concepção de interfaces humano-computador em ambientes de computação móvel e ubíqua
- Explicar a noção de usabilidade e os desafios que levanta ao desenvolvimento de software
- Classificar os diferentes métodos de avaliação em IHC no contexto dos sistemas ubíquos e conceber uma estratégia de avaliação da interacção para um sistema ubíquo.

SUPORTE DE SISTEMA PARA AMBIENTES MÓVEIS E UBÍQUOS

Descrição.
Este módulo terá um foco particular nos requisitos e sistemas de middleware para sistemas móveis e ubíquos. Pretende-se apresentar as principais abordagens na concepção de sistemas de suporte para este tipo de ambientes, bem como um conjunto de plataformas de middleware de referência, presentes no mercado ou em investigação, cujas características são especialmente relevantes para este tipo de cenários. Serão também abordados novos sistemas de middleware que pretendem suportar aplicações cuja ambiente de execução é assumidamente um espaço físico e os serviços a ele associados.

Programa resumido.

- Fundamentos de Middleware
- Tecnologias de middleware
- Desafios para a Computação Móvel e Ubiqua
- Requisitos para sistemas distribuídos móveis
- Limitações do middleware para sistemas distribuídos tradicionais
- Plataformas e toolkits para computação ubíqua

- Middleware sensível ao contexto
- Selecção de serviços dependentes do contexto

Resultados de Apreendizagem Específicos:

- Explicar os requisitos específicos impostos pela necessidade de suporte à mobilidade nas tecnologias de middleware.
- Avaliar a aplicabilidade de middleware em cenários específicos de construção de sistemas móveis ou ubíquos.
- Explicar as funcionalidades mais comuns em sistemas de middleware cujo ambiente de programação esteja associado a espaços físicos.

PROJECTO

Descrição.
O projecto assume-se como um componente central desta UCE, já que desempenha um papel integrador entre as várias áreas de conhecimento. Esse papel central assenta na convicção de que na área dos sistemas móveis e computação ubíqua a experimentação prática é fundamental para uma aprendizagem efectiva e para o desenvolvimento de uma visão crítica e abrangente das reais possibilidades e implicações das muitas tecnologias e abordagens envolvidas. O módulo de projecto irá funcionar sempre em coordenação muito próxima com os restantes módulos, pretendo-se que desta interligação resultem forte sinergias. Por um lado o projecto será, como é natural, um local privilegiado para a aplicação prática de conceitos teóricos. Mas ao mesmo tempo também se pretende que o projecto seja um contributo importante para as actividades das aulas teóricas ao permitir que os temas e experiências concretos resultantes da sua execução possam ser usados como exemplo e referência na aprendizagem dos conceitos teóricos ou como ilustração de casos concretos de conceitos abstractos. É também expectável que as necessidades e dificuldades sentidas no projecto sejam usadas como motivação para a introdução de novos temas, ou seja os alunos já se depararam com um determinado problema no projecto, e mais tarde vão perceber durante os módulos teóricos que esse problema é comum a outros sistemas e que existem abordagens já estabelecidas para lidar com eles.

4

ACS
Análise e Concepção de Software

Sumário

A análise e a concepção são actividades incluídas no processo de desenvolvimento de software. O desenvolvimento refere-se às fases do ciclo de vida responsáveis pelo projecto de construção de sistemas, incluindo ainda a implementação (também designada de construção). Excluem-se, por exemplo, os estudos de viabilidade económica, as tarefas de manutenção e a utilização efectiva do sistema.

O módulo de **Análise e Concepção de Software** *(ACS) tem como principal objectivo dotar os estudantes de competências metodológicas no contexto do desenvolvimento (e mais concretamente nas tarefas de análise e concepção) de sistemas de software de elevada complexidade, para obter soluções correctas e fiáveis recorrendo aos princípios básicos da engenharia.*

Este módulo organiza-se em torno de 5 áreas de conhecimento (Knowledge Areas - KAs), definidas em [SWEBOK 2005], cujas unidades curriculares que as consubstanciam correspondem a 30 ECTS. A avaliação é feita por exame global único e aborda os assuntos tratados em todas as 5 unidades curriculares.

Coordenação: *.João M. Fernandes, Fernando M. Martins, Olga M. Pacheco, António J. Esteves, Ricardo J. Machado (DSI).*

4.1 Objectivos

A disciplina da engenharia de software cobre não só os aspectos técnicos da construção dos sistemas de software, mas também aspectos de gestão, tais como a liderança de equipas de programação, o planeamento das actividades e a análise económica dos projectos de desenvolvimento tecnológico.

A engenharia de software pode ser definida de várias formas. Das várias sugestões já feitas, salientam-se as seguintes:

- "The establishment and use of sound engineering principles (methods) in order to obtain economically software that is reliable and works on real machines" [Bauer 1972].

- "Software engineering is that form of engineering that applies the principles of computer science and mathematics to achieving cost-effective solutions to software problems" [CMU/SEI-90-TR-003].

- "The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software" [IEEE 1990].

Todas estas definições incluem uma ideia central que implica que a engenharia de software é muito mais que codificar programas numa dada linguagem e inclui questões de qualidade, económicas, de gestão de recursos humanos, bem como conhecimentos e aplicação de princípios metodológicos. A engenharia de software baseia-se em princípios da ciência da computação e da matemática, mas, como ramo da engenharia, ultrapassa largamente estes princípios e recorre igualmente a um conjunto alargado de outras disciplinas (engenharia de computadores, gestão, gestão de projectos, gestão da qualidade, ergonomia em software e engenharia de sistemas). A engenharia de software preocupa-se com a criação de software de qualidade, segundo uma abordagem sistemática, controlada e eficiente. Assim, há necessariamente uma ênfase na análise, especificação, concepção e manutenção dos sistemas de software. Adicionalmente, devem abordar-se questões relacionadas com a gestão, a qualidade, o uso de normas e a gestão das capacidades dos projectistas.

Esta UCE cobre a componente mais ligada ao desenvolvimento técnico de aplicações de software, deixando, para uma 2ª UCE a definir, as matérias mais directamente relacionadas com o processo de software, a gestão do processo e dos projectos, e a definição de índices de qualidade. Essas 2 UCEs formariam, de forma coerente e totalmente integrada, um currículo de 2º ciclo em ENGENHARIA DE SOFTWARE, segundo os padrões actualmente aceites pela comunidade desta área científica.

4.2 Caracterização Global

4.2.1 Regime e Escolaridade

A UCE apresentada neste documento poderá ser oferecida em regime semestral ou anual de acordo com o público a que se destine. O regime anual será preferencialmente adoptado para a formação inicial, durante a qual num mesmo ano um aluno poderá realizar duas UCE, enquanto o regime semestral, mais intensivo mas concentrado no tempo, parece ser mais adequado para oferta em formação contínua ou quando exigido pela estrutura de Mestrados Erasmus-Mundus que venha a integrar. Atendendo à sua articulação em 4 módulos e um projecto integrado, conforme se detalha a seguir, a escolaridade proposta é a seguinte: 8T + 8TP + 8P (sendo as 8P correspondentes à componente de projecto integrado).

4.2.2 Áreas Científicas

Áreas Científicas: Engenharia de Software (30 ECTS)

Áreas Científicas (classificação SWEBOK2005 & SE2004) :

SWEBOK 2005:
KA #1 “Software Requirements”
KA #2 “Software Design”
KA #3 “Software Construction”
KA #9 “Software Engineering Tools and Methods”

SE 2004:
MAA — Software Modeling & Analysis
DES — Software Design
VAV — Software V&V
PRF — Professional Practice

4.2.3 Requisitos

Para frequentar este módulo, os alunos devem possuir as competências associadas às duas seguintes SEEK (Software Engineering Educational Knowledge) Areas [SE 2004]:

- CMP — Computing Essentials (Computer Science foundations, Construction technologies, Construction tools, Formal construction methods)
- FND — Mathematical & Engineering Fundamentals (Mathematical foundations, Engineering foundations for software, Engineering economics for software)

Estas competências são, grosso modo, obtidas pelos alunos que frequentarem as licenciaturas de 1º ciclo em TICE da U.Minho. Em concreto, espera-se que os alunos sejam capazes de construir programas usando métodos rigorosos de especificação e linguagens de programação e de pôr a executar os programas escritos, usando as ferramentas adequadas (editores, compiladores, interpretadores, depuradores, IDEs).

O aluno interessado por este módulo será aquele que, após formação na área da programação de computadores, pretende complementar essas suas competências tecnológicas, com uma vertente mais vocacionada para a execução de actividades de desenvolvimento de software, segundo uma abordagem de engenharia (engenheiro de software, engenheiro de requisitos, analista de sistemas, arquitecto de software).

4.2.4 Resultados de Aprendizagem

- Captar os requisitos dum sistema junto dos *stakeholders*, documentá-los de forma a garantir que eles descrevem correctamente o sistema pretendido, e validá-los à custa, por exemplo, da execução de inspecções ou de revisões formais.

- Modelar formalmente os requisitos de um sistema de software e raciocinar dentro dos modelos produzidos, utilizando ambientes de prototipagem de especificações formais e tendo uma percepção clara do lugar destes métodos no desenvolvimento de software.
- Construir e avaliar protótipos de sistemas interactivos em função de requisitos e objectivos de usabilidade definidos.
- Construir uma arquitectura de componentes de software que permitam responder de forma eficaz à necessária concretização dos requisitos elencados, bem como possibilitem uma manutenção e evolução controlada.
- Construir, em equipa, sistemas de software complexos, de acordo com o pretendido e a funcionar correctamente, através da combinação de actividades de análise, concepção, codificação, validação e teste.

4.3 Estrutura Curricular

4.3.1 Estrutura Base

A presente UCE organiza-se em torno de 4 módulos temáticos, correspondendo cada um a 5 ECTS, articulados entre si por um projecto integrado, que garante a experimentação e aplicação prática dos resultados da aprendizagem. Esta estrutura é apresentada na Fig. 1. Os módulos referidos são:

Unidade curricular	ECTS
Análise e Modelação de Requisitos (AMR)	5
Arquitecturas de Software (AS)	5
Métodos Formais no Projecto de Software (MFPS)	5
Usabilidade e Interacção (UI)	5
Projecto Integrado (PI)	10

4.3.2 Métodos de Ensino

- Exposição de conceitos e análise de casos de estudo.
- Trabalho de grupo em exercícios e pequenos estudos de caso, em certos casos com recurso a ferramentas informáticas específicas.
- Trabalho de projecto em grupo com orientação directa da equipa docente afecta à UCE.

4.3.3 Avaliação

O processo de avaliação visa aferir não apenas os conhecimentos e competências mínimas adquiridas, mas também a capacidade revelada na sua integração e aplicação no projecto de engenharia

de software. Assim, é constituída por um exame final único em média pesada com a avaliação contínua em prática laboratorial e o resultado (artefactos de software, ferramentas, relatório, apresentação) do projecto integrado.

4.3.4 Contéudos Programáticos

ANÁLISE E MODELAÇÃO DE REQUISITOS — AMR

Descrição:

[SWEBOK 2005] KA #1 "Software Requirements"

[SE 2004] MAA — Software Modeling & Analysis

A área de requisitos de software trata a aquisição, análise, especificação, validação e manutenção de requisitos do software. Entende-se por requisitos as propriedades que os sistemas (ainda em projecto) devem vir a manifestar aquando da sua realização. Os requisitos do software exprimem as necessidades e as constrições que são colocadas a um sistema de software e que devem ser tidas em conta aquando do seu desenvolvimento. Esta área é já reconhecida como de primordial importância para a indústria de software, dado o impacto que as suas actividades geram na estabilização e gestão de todo o processo de desenvolvimento de software.

Programa resumido:

- Processo de engenharia de requisitos;
- Levantamento de requisitos;
- Análise de requisitos;
- Especificação e modelação de requisitos;
- Validação de requisitos;
- Gestão de requisitos.

Resultados de Apreendizagem Específicos:

- definir qual a intervenção que a equipa de engenharia de requisitos deve executar ao nível de todo os processo de engenharia de software, explicitando o envolvimento formal dos *stakeholders*, ao longo de todo o processo de engenharia de requisitos.
- tratar a forma como devem ser capturados os requisitos, bem como as técnicas que devem ser utilizadas para ajudar as diversas fontes de requisitos (humanos e não humanos) a "libertá-los" correctamente.

- detectar e resolver conflitos entre os requisitos capturados, definir a fronteira do sistema em projecto e como ele interactua com o seu meio ambiente, bem como transformar requisitos do sistema em requisitos de software.
- tratar o documento de requisitos do ponto de vista da sua estrutura, qualidade e verificabilidade, normalmente organizado em duas partes distintas: (1) o documento de definição de requisitos que descreve os requisitos do utilizador; (2) a especificação dos requisitos de software que estabelece o acordo entre os clientes e os fornecedores do sistema de software.
- examinar o documento de requisitos a fim de garantir que ele descreve o sistema pretendido, à custa, por exemplo, da execução de inspecções (ou revisões formais) do documento, ou da prototipagem rápida das suas interfaces.
- gerir a alteração dos requisitos, garantir uma semântica bem definida para os mesmos, assim como rastreá-los ao longo de todo o processo de desenvolvimento.

ARQUITECTURAS DE SOFTWARE — AS

Descrição:

[SWEBOK 2005] KA #2 "Software Design"

[SE 2004] DES — Software Design

A concepção de software é simultaneamente o processo de definição da arquitectura, dos componentes, das interfaces e de outras características do sistema (ou componente), e o resultado desse processo. Na óptica do processo e no âmbito do ciclo de vida de desenvolvimento de software, a concepção de software consiste na actividade em que os requisitos de software são analisados com o intuito de produzir uma descrição da estrutura interna e da organização do sistema. Na óptica do produto, a concepção de software (o resultado final do processo) deve descrever a arquitectura do sistema (i.e., como o sistema pode ser decomposto e organizado em componentes), as interfaces entre os componentes, bem como os próprios componentes com um nível de pormenor que permita a construção destes.

Programa resumido:

- Definição das componentes de uma arquitectura software;
- Decomposição modular. Estruturação do um sistema software complexo;
- Modelos de controlo e fluxo de informação;
- Padrões estruturais e de comportamento;
- Frameworks orientadas ao domínio e de componentes reutilizáveis;

- Desenvolvimento orientado à reutilização
- Aplicações multi-camada (especifidades e padrões necessários);
- Estratégias e mecanismos de integração;

Resultados de Apreendizagem Específicos:

- avaliar a importância de uma arquitectura de software no contexto geral de disponibilização de uma solução.
- definir os diferentes tipos de modelos arquitecturais a nível de estrutura do sistema, decomposição por camadas e por elementos de controlo.
- analisar soluções existentes e o seu grau de flexibilidade e reutilização.
- analisar e avaliar as qualidades intrínsecas à arquitectura (integridade conceptual, correcção e completude, e construibilidade), os atributos que se revelam em tempo de concepção (capacidade de modificação, portabilidade, reusabilidade, integrabilidade e capacidade de teste) e os atributos que só revelam em tempo de execução (desempenho, segurança, disponibilidade, usabilidade e funcionalidade).
- utilizar frameworks específicas e mecanismos de especialização
- definir aplicações com indepêndencia de camadas e modos de a assegurar.
- analisar as condicionantes dos modelos de programação e escolher as notações e linguagens mais adequadas para representar artefactos de concepção de software, nomeadamente a organização estrutural da concepção e o comportamento do software.
- medir os impactos dos mecanismos de integração na definição de uma arquitectura.
- reutilizar soluções pré-definidas e garantidamente testadas e eficazes no âmbito do domínio do problema recorrendo, por exemplo, à utilização de estruturas arquitecturais e perspectivas, estilos arquitecturais, padrões de concepção, famílias de programas e frameworks.

MÉTODOS FORMAIS NO PROJECTO DE SOFTWARE — MFPS

Descrição:

[SWEBOK 2005] KA #9 "Software Engineering Tools and Methods" (sub-área software engineering methods)

A área dos métodos formais inclui todas as aplicações da matemática discreta a problemas de engenharia de software. Os métodos formais, aplicados ao desenvolvimento de software, recorrem a linguagens formais para descrever os artefactos de software (especificações, arquitecturas e código), permitindo a prova formal de propriedades, bem como a reificação dos mesmos para suporte à implementação.

Esta unidade curricular, irá focar-se no estudo do desenvolvimento de sistemas de software baseado em Métodos Formais, com ênfase para as técnicas de especificação baseadas em modelos da matemática discreta, que conduzem ao cálculo de sistemas de informação com garantia de correcção e à certificação de software por construção correcta e/ou verificação.

Programa resumido:

- Introdução ao problema do controlo de qualidade em 'software'. Especificação formal — porquê e para quê? Introdução ao binómio especificação *vs* implementação.
- Introdução Métodos Formais: taxonomia, ciclo de vida, certificação. Discussão de áreas típicas de aplicação: sistemas críticos e confiáveis; áreas da saúde e da segurança; qualidade de serviços; sistemas de informação avançados.
- Linguagens e métodos para especificação formal. Antecendentes históricos. Do método de Viena (VDM) ao 'standard' ISO/IEC 13817-1 (VDM-SL).
- Estudo da notação VDM-SL. Modelação por conjuntos, sequências e funções.
- Propriedades invariantes. Obrigações de prova.
- Prototipagem e animação. Ambientes de desenvolvimento formal. Experiência com a utilização do ambiente VDMTOOLS.
- Especificação formal por objectos (VDM++). Integração com UML.

Resultados de Apreendizagem Específicos:

- Modelar os requisitos de um sistema de software numa linguagem de especificação formal construtiva.
- Raciocionar dentro dos modelos produzidos de forma rigorosa, identificando propriedades e obrigações de prova.
- Utilizar um ambiente de prototipagem de especificações formais.
- Ter uma percepção clara do lugar destes métodos no projecto de sistemas informáticos complexos: áreas de aplicação, formas de introdução, relação com os níveis de modelação semi-formais (e.g., UML) e com o desenvolvimento de código.

USABILIDADE E INTERACÇÃO — UI

Descrição:

[SWEBOK 2005] KA #2 "Software Design"(sub-área Key Issues in Software Design)

[SE 2004] DES — Software Design / VAV — Software V&V

A Usabilidade é uma propriedade da interacção entre um dado sistema e os seus utilizadores definida pela norma ISO 9241-11 como "a eficácia, eficiência e satisfação com que determinados utilizadores, atingem determinados objectivos em ambientes específicos". Assegurar a usabilidade de um sistema software corresponde, assim, a desenvolvê-lo por forma a que os utilizadores possam atingir os seus objectivos da forma o mais eficiente e satisfatória possível. Questões como a definição e avaliação de objectivos de usabilidade, análise de tarefas, ou definição das metáforas e paradigmas de interacção mais apropriados a um dado contexto, têm impacto em todo o processo de desenvolvimento, desde a análise de requisitos até à implementação, passando pela concepção e avaliação dos sistemas. A Engenharia da Usabilidade agrega um conjunto de métodos, técnicas e ferramentas, derivadas da área de Interacção Humano-Computador, que permitem tornar a usabilidade um factor central no desenvolvimento de software.

Programa resumido:

- Fundamentos de IHC – Modelos e Teorias (modelos cognitivos, o computador, modelos de interacção).
- Engenharia da usabilidade – definição do problema (estudo dos utilizadores, análise de tarefas); definição de requisitos e objectivos de usabilidade; concepção do sistema (estilos de interacção, desenho conceptual, guidelines); avaliação de usabilidade (avaliação empírica, avaliação analítica, o *Usability Maturity Model* – ISO 18529).
- Prototipagem rápida de interfaces – diferentes tipos de protótipos; *Toolkits*; ambientes de desenvolvimento; prototipagem e avaliação.
- Arquitecturas software para sistemas interactivos – identificação de padrões, arquitecturas para aplicações interactivas móveis.

Resultados de Apreendizagem Específicos:

- identificar os diferentes tipos de modelos relevantes no processo de engenharia de usabilidade;
- aplicar técnicas de análise de usabilidade;
- desenvolver uma interface com utilização de toolkits e componentes existentes, criando para tal uma arquitectura software de suporte ao sistema interactivo.

PROJECTO INTEGRADO — PI

Descrição:

[SWEBOK 2005] KA #3 "Software Construction" e KA #9 "Software Engineering Tools and Methods" (sub-área software engineering tools)

[SE 2004] PRF — Professional Practice

Com esta unidade curricular pretende-se integrar, num único projecto de cariz implementacional (relativo à execução das actividades de desenvolvimento de software), as competências desenvolvidas nas restantes unidades curriculares deste módulo. A fase de implementação é um acto fundamental da engenharia de software, ou seja, consiste na construção de software de acordo com o pretendido e a funcionar correctamente, através da combinação de actividades de codificação, validação e teste. A implementação de software está intimamente ligada à concepção, visto que a primeira deve transformar em suporte tecnológico (código) as arquitecturas concebidas e descritas pela segunda. Esta transformação tende a ser cada vez mais automática, pois existem sub-tarefas perfeitamente repetitivas e mecanicistas. Assim, é na implementação de software que a utilização de ferramentas se mostra mais crítica.

Considera-se desejável que o projecto integrado funcione segundo a abordagem PLEE (Project-Led Engineering Education). Assim, as temáticas a ministrar nas restantes unidades curriculares devem contribuir directamente para a execução do projecto, o que obriga a uma coordenação entre todos os docentes envolvidos, de forma a que esta unidade curricular seja, de facto, um projecto de desenvolvimento de software que integra um conjunto muito alargado das valências previstas. Em especial, espera-se que o enfoque das restantes disciplinas seja fortemente condicionado pelas temáticas incluídas no projecto.

Programa resumido:

- Desenvolvimento de um projecto integrador com a inclusão de todos os conhecimentos focados nas restantes unidades curriculares
- Tratamento, numa óptica de projecto, da componente de testes do sistema software, abrangendo os testes unitários de integração e carga.

Resultados de Apreendizagem Específicos:

- construir, em equipa, sistemas de software complexos, de acordo com o pretendido e a funcionar correctamente, através da combinação de actividades de análise, concepção, codificação, validação e teste.
- aplicar, em contextos práticos, os conhecimentos adquiridos no módulo, em projecto de exigência similar ao que os encontrarão na indústria, quando profissionais.
- utilizar ferramentas (meta-CASE, frameworks IDEs), nomeadamente no contexto das actividades menos criativas e nas quais o ser humano tem mais tendência para cometer erros.

- usar normas, para garantir a uniformidade processual e notacional e para permitir a interoperabilidade e portabilidade das soluções.

5

MFES
Métodos Formais em Engenharia de Software

Sumário

O presente documento descreve uma UCE30 (="Unidade curricular de Especialidade de 30 ETCS") que o grupo de Teoria e Métodos Formais da Universidade do Minho tenciona oferecer a nível de 2^o ciclo da re-estruturação curricular da oferta formativa do Departamento de Informática segundo o Processo de Bolonha.

Esta UCE30 resulta de mais de 20 anos de experiência no ensino, investigação a aplicação de métodos rigorosos na construção de software. Pretende responder aos desafios que nos dias de hoje decorrem na só da ubiquidade do suporte informático na sociedade mas também da globalização da prestação de serviços em informática. O recurso aos fundamentos matemáticos da modelação, raciocínio e teste de soluções dão a esta oferta formativa um marca especial, colocando-a desde já no patamar da exigência que o profissional terá de satisfazer no futuro da sociedade da informação.

Seguindo o padrão definido dentro do Departamento de Informática, a UCE consta de 4 módulos de 5 ECTS cada um, articulados entre si por um "bus" laboratorial (de 10 ECTS) que garante a experimentação e aplicação prática dos resultados da aprendizagem. A integração de conhecimentos reflectir-se-á também na avaliação, que constará de um único exame final (para toda a matéria teórica da UCE) em média pesada com a avaliação em prática laboratorial e projectos concretos.

A instância deste esquema para a presente UCE é a que consta da Fig. 5.1. O sucesso deste modelo de formação depende antes mais da receptividade por parte do seu principal mercado alvo — a indústria de software. Espera-se que esta indústria, em particular a de âmbito local e regional, se associe à academia e assim se feche o ciclo entre produção e absorção de recursos humanos de qualidade, promovendo o Minho como região de conhecimento e exigência.

Coordenação: *Luís S. Barbosa, José B. Barros, Manuel Alcino Cunha, José N. Oliveira, Joost Visser.*

5.1 Contexto e Objectivos

A UCE que aqui se propõe pretende consolidar uma experiência de vinte anos no ensino de métodos formais para o desenvolvimento de aplicações de software. Os cinco módulos que o compoem — quatro temáticos e um projecto integrador de índole exclusivamente laboratorial — estão intimamente ligadas e exprimem os vários vectores de que depende o projecto fiável de aplicações à escala industrial segundo a visão (*mission statement*) da UCE.

No seu conjunto, os módulos que fazem parte desta UCE pretendem realizar o desígnio de que é possível afixar o carimbo

aos artefactos de *software* desenvolvidos segundo os seus princípios.

Na sua componente prática, a UCE preocupa-se com a prototipagem das ideias e seus modelos, testando-os atempada e exaustivamente antes de se proceder à fase de cálculo e implementação, aumentando a interacção com o cliente e poupando erros de perspectiva ou infantilidades de concepção. Em suma: *saber modelar e calcular — sim, mas também saber testar*.

Naturalmente, um tal programa de trabalho só é viável se alicerçado em três premissas metodológicas: o recurso ao binómio especificação-implementação (de outra forma, o teste resvalaria para o estafado "debug" de última hora), o recurso a linguagens formais (de outra forma o cálculo seria vacuoso ou impossível) e o recurso a ambientes sofisticados de computação simbólica. Este último ingrediente do método proposto tem bastante tradição local e é de prever a reanimação de algumas ideias antigas, agora viabilizadas por conhecimentos acrescidos e novos recursos humanos.

Tal como noutras disciplinas de engenharia e à imagem do que aconteceu já na tecnologia do "hardware", a *arquitectura de software* deverá ser o destino último de todos os esforços científicos e pedagógicos a desenvolver, na certeza de que só a esse nível se pode *pensar em grande* e, verdadeiramente, fazer engenharia com reutilização, elegância e eficácia.

Uma última palavra para as virtualidades de se associar a ciência da análise e teste de programas aos métodos formais de programação: numa altura em que tudo está já feito, ainda que mal-feito muitas vezes, é de esperar a evolução das tecnologias da informação para uma fase com contornos "geriáticos": o que se fez mal-feito acabará mais tarde ou mais cedo por adoecer e precisar de reparo, restauro ou de cuidados intensivos, até; o que de bem-feito se fez pode vir a precisar de ser explicado a quem dele beneficia, quando se perder a memória de quem o fez. Nesta perspectiva de evolução, a componente de engenharia reversa assumirá mais importância

que a directa, e é de esperar que sejam uma vez mais os métodos formais a garantir fiabilidade naquilo que se vier a fazer, desde que alicerçados na capacidade de inspecção de código legado. O módulo que se propõe nesse segmento consolida trabalhos de investigação recente na área e prolonga-se a aspectos tão importantes do projecto de *software* como o da cobertura de baterias de teste, a análise de risco, a inferência de métricas, etc.

5.2 Caracterização Global

5.2.1 Regime e Escolaridade

A UCE apresentada neste documento poderá ser oferecida em regime semestral ou anual de acordo com o público a que se destine. O regime anual será preferencialmente adoptado para a formação inicial, durante a qual num mesmo ano um aluno poderá realizar duas UCE, enquanto o regime semestral, mais intensivo mas concentrado no tempo, parece ser mais adequado para oferta em formação contínua ou quando exigido pela estrutura de Mestrados *Erasmus-Mundus* que venha a integrar.

Atendendo à sua articulação em 4 módulos e um projecto integrador, conforme se detalha a seguir, a escolaridade proposta é a seguinte: 8T + 8TP + 8P (sendo as 8P correspondentes à componente de projecto integrador).

5.2.2 Áreas Científicas

Áreas Científicas: Informática (30 ECTS)

Áreas Científicas (classificação ACM): É a seguinte a distribuição dos 30 ECTS da UCE pelas áreas científicas classificadas no ACM *Computing Curricula* (1998), apresentadas por ordem lexicográfica:

- Software/SOFTWARE ENGINEERING/Metrics — 5
- Software/SOFTWARE ENGINEERING/Requirements/Specifications — 5
- Software/SOFTWARE ENGINEERING/Software Architectures — 5
- Software/SOFTWARE ENGINEERING/Software/Program Verification — 5
- Software/SOFTWARE ENGINEERING/Testing and Debugging — 5
- Theory of Computation/LOGICS AND MEANINGS OF PROGRAMS/Specifying and Verifying and Reasoning about Programs — 5

5.2.3 Requisitos

- Experiência em programação, incluindo, preferencialmente, alguma familiaridade com a programação declarativa (funcional ou lógica).
- Conhecimentos de lógica e matemática discreta a nível de primeiro ciclo.

5.2.4 Resultados de Aprendizagem

- Criar, rever, analisar, classificar, animar, testar e transformar *modelos* abstractos de problemas e sistemas em Engenharia de Software.
- Transformar especificações de sistemas de informação complexos em implementações sobre diversos tipos de tecnologias.
- Modelar, analisar, classificar e transformar diferentes padrões de interacção, estratégias de modularização (componentes, objectos, serviços) e esquemas de organização arquitectural do software.
- Seleccionar técnicas de análise, planear e executar projectos de teste de software.
- Planear, executar e avaliar a qualidade de projectos de modelação e desenvolvimento de software, recorrendo a métodos rigorosos e diferentes tecnologias de aplicação.
- Desenvolver de forma integrada a função de concepção e projecto em Engenharia.

5.2.5 Métodos de Ensino

- Exposição de conceitos e análise de casos de estudo.
- Trabalho de grupo em exercícios e pequenos estudos de caso, em certos casos com recurso a ferramentas informáticas específicas.
- Trabalho de projecto em grupo com orientação directa da equipa docente afecta à UCE.

5.2.6 Avaliação

O processo de avaliação visa aferir não apenas os conhecimentos e competências mínimas adquiridas, mas também a capacidade revelada na sua integração e aplicação no projecto de engenharia de software. Assim, é constituída por um exame final único em média pesada com a avaliação contínua em prática laboratorial e o resultado (ferramentas, relatório, apresentação) do projecto integrador.

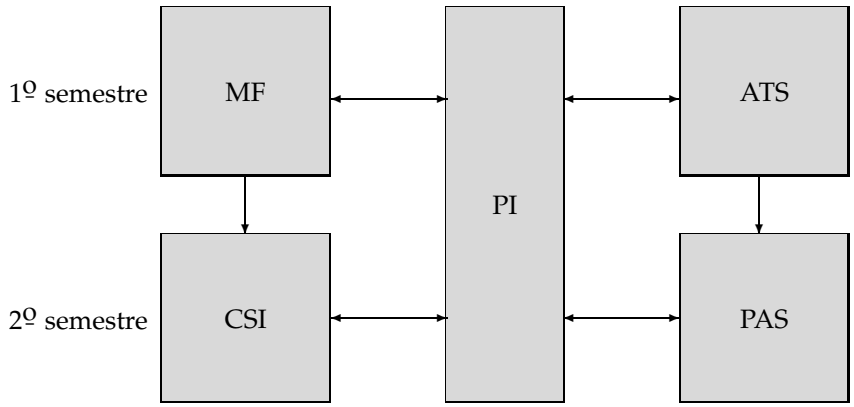

Figura 5.1: Estrutura curricular da UCE30 *Métodos Formais em Engenharia de Software*. Legenda: *MF* - Métodos Formais; *CSI* - Cálculo de Sistemas de Informação; *PAS* - Processos e Arquitectura de Software; *ATS* - Análise e Teste de Software

5.3 Conteúdos Programáticos

Os contéudos programáticos desta UCE são agrupados em 4 módulos temáticos e um de projecto integrador, conforme a seguir se detalha.

5.3.1 Métodos Formais (MF)

Descrição. A ciência tem a ver com o perceber-se *como o mundo e as suas coisas funcionam* e a tecnologia com o garantir que *certas coisas que se desejam* acontecem de forma fiável e repetível. A necessidade de basear a segunda em resultados da primeira é uma constante da evolução das disciplinas de engenharia, a única de facto capaz de ultrapassar as limitações do puro experimentalismo e do amadorismo, e de promover a transmissão segura do conhecimento.

Ora o resultado científico recorre, por natureza, à fórmula matemática para poder ser expresso e, assim, ser susceptível de raciocínio. É assim que as engenharias "clássicas", como a civil, a mecânica, a electrotécnica, a electrónica, etc, recorrem sistematicamente a disciplinas da matemática (algumas já centenárias) para a arrumação dos seus corpos de conhecimento. São exemplos conhecidos de todos nós o cálculo diferencial/integral, a análise numérica, a geometria analítica, etc, etc.

Com o advento das chamadas *Tecnologias da Informação* esta situação alterou-se. Na verdade, essas tecnologias desde cedo mostraram alguma dificuldade em sobreviver sobre tal herança científica, pelo facto de esta estar demasiado comprometida com a modelização de entidades da vida real que, como o *espaço* e o *tempo*, são por natureza *contínuas* e *infinitas*. Ora o suporte computacional capaz de mecanizar esses modelos é, por limitação física, *discreto* e *finito*. O que faz com que sejam a *matemática discreta*, a *lógica formal*, a *álgebra universal*, etc. as disciplinas em que o projecto científico do 'software' se alicerça.

Contudo, essas disciplinas encerram uma dificuldade — a de, por serem demasiado *descritivas*, os raciocínios dificilmente escalarem a problemas de *tamanho real*. Existem duas alternativas para sair dessa dificuldade: apostar em ferramentas de verificação (demonstradores de teoremas, etc) ou apostar em notações mais económicas e susceptíveis ao raciocínio. Este módulo segue a segunda destas vias, sendo a sua base o ensino do cálculo de relações binárias dito 'pointfree' (sem variáveis) e a sua essência a chamada *transformada 'pointfree'* que converte expressões da lógica e matemática discreta como forma de agilizar o raciocínios necessários.

Este módulo estrutura-se, pois, a dois níveis — descrição e cálculo. No primeiro, ensina-se a especificar 'software' através da criação de modelos abstractos sobre os quais se pode raciocinar e garantir propriedades desejáveis ('model-oriented specification'). Recorre-se à linguagem de especificação VDM-SL (ISO/IEC standard 13817-1) para esse efeito e a ferramentas como as VDMTools (IFAD/CSK) para animar e testar esses modelos abstractos. No segundo, ensina-se a converter esses modelos para a notação relacional e a raciocinar sobre eles. O exercício estende-se a outras áreas da modelação como, por exemplo, a formulação de uma semântica para diagramas ER (entidade-relação), etc.

Pela sua natureza propedêutica, o módulo é central às que a acompanham neste módulo. A sua articulação com o Projecto Integrado (PI) é feita ao nível da prototipagem e animação de modelos VDM-SL no ambiente VDMTOOLS® - *for Quality Software on Schedule* da IFAD/CSK.

Programa resumido

- Introdução à especificação formal como método de *controlo de qualidade* em 'software'. Binómio *especificação /implementação*. O "standard" ISO/IEC 13817-1 (VDM-SL).

- Ciclo de vida do desenvolvimento formal de 'software'. Especificação formal construtiva. Modelação de um problema. Prototipagem e animação. Validação por teste. Importância da verificação formal das propriedades de um modelo. Obrigações e métodos de prova. Sub-especificação e não-determinismo. Relação (especificação) versus função (implementação).

- Introdução ao cálculo de relações. Inclusão, composição e intersecção de relações. Conversa de uma relação. Formulação de propriedades em notação "pointfree".

- Taxonomia de relações binárias. As funções vistas como casos particulares de relações. Representação de predicados lógicos por relações binárias. Ordens. Estruturação do cálculo relacional com base em conecções de Galois (CG).

- Significado de um modelo VDM-SL. Semântica relacional de um par **pre-/post-**. Relações em compreensão. Relações simples finitas e sua representação em VDM-SL ("mappings").

- Significado de um *invariante* em VDM-SL. Prova de preservação de um invariante. Noção de *precondição mais fraca* que garante uma propriedade. Regras para combinação de especificações que satisfazem propriedades.

- A *integridade-referencial* como uma classe de invariantes sobre relações simples e finitas em bases de dados. Diagramas Entidades-Relações (ER) e sua semântica *pointfree* baseada na preordem de definição de relações.

Resultados de Aprendizagem Específicos

- Dominar a noção de modelo abstracto de um problema
- Criar, rever, analisar, animar e testar modelos abstractos de problemas e sistemas
- Classificar problemas e seus modelos.
- Aplicar o cálculo relacional e a transformada *pointfree* da lógica clássica à construção, transformação e cálculo de modelos formais de software.

5.3.2 Cálculo de Sistemas de Informação (CSI)

Descrição. Esto módulo pretende transportar, para o plano prático e aplicado, a bagagem em modelação formal adquirida no módulo de *Métodos Formais* que a precede, à qual acrescenta a valência de refinamento e implementação por cálculo. O objectivo é mostrar como, de modelos abstractos, puramente declarativos, como os que se estudaram a conceber nesse módulo, se podem inferir aplicações reais, implementadas em arquitectura *cliente-servidor*, com garantia de correcção por cálculo.

No plano operacional, aborda-se a derivação de máquinas de estados abstractas (ASM, vulg. *objectos*) que prestam serviços através de uma API pública a partir dos modelos declarativos estudados em *Métodos Formais*, por factorização do seu estado interno. A esta derivação está associada a passagem da notação VDM-SL para a sua extensão VDM++, que é orientada ao objecto.

Passa-se de seguida ao estudo do refinamento de tais ASMs, que se processa aos dois níveis do serviço que prestam: *dataware* e *middleware*. Ao primeiro nível, é dada particular ênfase a um cálculo de estruturas de dados que decorre do cálculo de relações binárias estudado anteriormente e que, em particular, substitui o cálculo de normalização de bases de dados relacionais. O uso do referido cálculo é apoiado por uma ferramenta (VooDooM) desenvolvida localmente e que calcula modelos de dados concretos em SQL a partir de modelos abstractos em VDM-SL. Ao segundo nível, estuda-se como calcular o código dos métodos (eg. em C/C++, Java, etc) a partir dos respectivos modelos abstractos e com base na ordem de refinamento que introduz definição e determinismo.

No plano prático (e através de um *trabalho laboratorial* de grupo) o módulo articula com PI (Projecto Integrado) na medida em que se pede aos alunos que desenvolvam um protótipo em VDMTools de uma ASM em arquitectura *cliente-servidor*, parte da qual fica em fase de protótipo e a outra é implementada numa plataforma de desenvolvimento, em articulação com Processos e Arquitectura de Software (PAS). Tal prototipagem beneficia, por sua vez, das técnicas estudadas no semestre anterior em *Análise e Teste de Software* (ATS).

Programa resumido

- **Cálculo de refinamento formal de dados.** Princípio da abstracção dos dados. Relações de abstracção e de representação. Inequações de refinamento. Teorema de desrecursivação genérica. Relações de pertença estrutural e acessibilidade estrutural.Implementação de tipos indutivos polinomais em linguagens com gestão de memória dinâmica: Introdução de apontadores em linguagens tipo C/C++. Representação orientada a objectos.

- **Cálculo de refinamento algorítmico.** A *eficiência* como principal motivação para o refinamento algorítmico. Fases do refinamento algorítmico: *simulação*, redução do não-determinismo, mudança de estrutura de dados virtual. Lei de refinamento funcional. Lei de refinamento relacional. Leis de fusão "vertical" e "horizontal". Lei de refinamento de simultâneo de dados e algoritmos. Cálculo de ciclos `while` como hilomorfismos.

- **Prototipagem rápida e integração com outras tecnologias.** Introdução ao VDM++. Integração de protótipos reactivos em arquitecturas cliente-servidor. Emulação de serviços intermédios ("middleware") e de serviços de dados ("dataware").

Resultados de Aprendizagem Específicos

- Distinguir as noções de abstracção e representação, especificação e implementação
- Transformar especificações de sistemas em implementações
- Calcular esquemas de bases de dados a partir de modelos abstractos

5.3.3 Processos e Arquitecturas de Software (PAS)

Descrição. Este módulo constitui uma introdução ao estudo da *arquitectura* dos sistemas na dupla perspectiva da *estrutura das suas interações* e do *comportamento emergente*. Como os restantes módulos nesta UCE, também este adopta um ponto de vista científico-pedagógico que procura aliar o rigor e poder de cálculo matematicamente fundamentado a uma componente laboratorial que, na forma de projecto horizontal de módulo, integra aspectos metodológicos e de realização tecnológica (sobre, por exemplo, plataformas orientadas a componentes, serviços-web, linguagens de coordenação e plataformas específicas sobre e.g. JAVA ou C#).

A primeira parte do curso é uma introdução à especificação e cálculo do *comportamento* dos sistemas a partir de uma caracterização matemática das noções de *interacção*, como elemento base da computação, e de *processo*, como padrão de interações. Será colocada uma enfâse particular no estudo de processos capazes de auto-reconfigurarem dinamicamente as suas interligações e possibilidades de interação. Esta abordagem será situada no contexo da *programação reactiva*, onde surgiu, e do fenónomo da *computação global* na análise do qual é um instrumento fundamental. De um ponto de vista técnico esta primeira componente é baseada no *cálculo-*π e nos métodos coindutivos, traçando-se através deles uma ponte com o cálculo de *funcionalidade* e *estruturas de informação* discutidos noutros módulos desta UCE.

A segunda parte do curso, capitalizando nos fundamentos estudados, incide sobre a sua aplicação à especificação, análise e transformação de *arquitecturas de software*.

Esta segunda componente, inicia-se com uma caracterização da *disciplina de arquitectura de software*, referindo os standards aplicáveis e traçando a história do conceito em Engenharia de Software. Em particular será enfatizada a diferença entre arquitecturas baseadas na interconecção estática (*compile-time*) de módulos, orientada à macro-estrutura da aplicação e ao controlo do fluxo de recursos entre módulos, e arquitecturas baseadas na orquestração dinâmica (*run-time*) de componentes e serviços autónomos, distribuídos ao longo de redes de computação global e altamente heterogéneos. A ênfase é, neste último caso, colocada no controlo do fluxo de interacções (não raro baseadas em comunicação anónima e de localização variável) e na integração de entidades heterogéneas cujas ligações são estabelecidas e revistas dinamicamente, sem interrupção de serviço. Procura-se motivar os alunos para os desafios colocados pela passagem, paralela à registada na prática económica, de uma visão que enfatizava o *software* como *produto* para uma outra que o encara essencialmente como um *serviço*.

É nesse contexto que o curso vai caracterizar diversos estilos arquitecturais, sublinhado o modo como tanto conceitos base (e.g., objecto, contrato, interface, componente, serviço, connector, *glue code*, padrão, configuração, etc.) como opções metodológicas emergem dos desafios e complexidade dos problemas reais emergentes quer na concepção quer na re-engenharia de software. Discutem-se, em particular,

- arquitecturas orientadas ao *objecto*
- arquitecturas orientadas à *componente*
- arquitecturas orientadas ao *serviço*

Em cada caso discute-se o vocabulário associado, estilos de interação, composicionalidade, modelos formais, e tecnologias associadas. São especificamente abordados abordando especificamente os *problemas*

- da orquestração e interacção de componentes (objectos, serviços, recursos),
- da mobilidade e reconfiguração dinâmica dessas conecções,
- da planificação, documentação, análise e evolução arquitectural.

Programa

1. Componentes, Serviços e Processos
    - Motivação: estrutura *vs* comportamento; prescrição *vs* observação; indução *vs* coindução.
    - Interacção e cálculo de processos.
    - Componentes e Serviços como processos interactivos.
    - Mobilidade e reconfiguração dinâmica de software. Introdução ao π-*calculus*.

- Estudo de caso e sua análise laboratorial (com recurso ao `mwb`)

2. Projecto e Cálculo de Arquitecturas de Software
   - Motivação e história do conceito em Engenharia de Software.
   - Modelação de arquitecturas; ADL (*Architecture Description Languages*); Estilos e padrões arquitecturais; metodologias.
   - Arquitecturas Orientadas ao Objecto. Composição de objectos.
   - Arquitecturas Orientadas à Componente. Coordenação de componentes.
   - Arquitecturas Orientadas ao Serviço. Orquestração de serviços.
   - Documentação, projecto e análise de arquitecturas de software.
   - Estudos de caso.

Resultados de Aprendizagem

- Identificar as noções de interacção, processo, componente (objecto, serviço, recurso) e distinguir as diferentes formas de composição e coordenação de software.
- Reconhecer e classificar diferentes padrões de interacção e organização arquitectural do software.
- Criar, analisar, testar e transformar modelos de processos e arquitecturas de software.

5.3.4 Análise e Teste de Software (ATS)

Descrição. Neste módulo, convidam-se os alunos a trocar a visão *construtiva* do arquitecto de *software* pela visão do *avaliador*, ou gestor de qualidade. Ensinam-se as competências necessárias para esse efeito a nível da ciência da análise e do teste de *software*, a vários níveis de granularidade, isto é, desde o teste unitário de rotinas, funções ou serviços, até ao teste integrado de aplicações.

É importante referir que este módulo não apresenta análise e teste de programas como um simples aglomerado de técnicas *ad-hoc*. Em lugar disso, é apresentada uma abordagem coerente em que a análise e o teste estão ligados entre si, de forma complementar às técnicas de construção de programas. Nesta visão, por exemplo, as baterias de teste são vistas como contrapartidas *extensionais* das propriedades *intencionais* que podem ser derivadas dos modelos abstractos de problemas (especificações). Consequentemente, a geração *model-driven* de testes emerge em paralelo com a engenharia reversa e refinamento por cálculo de modelos em programas. Este enquadramento teórico da análise e do teste equipa os alunos com conhecimentos sólidos numa área que é apreendida pela maioria dos programadores de uma forma intuitiva e muito pouco sistemática.

A articulação do módulo com Projecto Integrado (PI) é, assim, a dual da adoptada pelos outros dentro da UCE: em lugar de construir novo *software* aplicacional, os alunos empregarão uma gama de métodos e ferramentas para avaliarem e testarem *software* já existente, por exemplo, *software* de domínio público disponível.

Programa Resumido

1. Teoria geral da análise de programas. Interpretação abstracta. Técnicas de *slicing*.
2. Teste à unidade, teste funcional, análise de cobertura de uma bateria de testes.
3. Teste orientado ao modelo, geração automática de testes, teste por injecção de falhas
4. Métricas para avaliação de *software*, normas de codificação, estilo e documentação
5. Compreensão de programas, engenharia reversa
6. Verificação, análise de segurança, avaliação de risco.
7. Análise de complexidade e de *performance*.

Resultados de Aprendizagem Específicos

- Derivar conjuntos de teste a partir de modelos de software.
- Planear e proceder ao teste sistemático de software.
- Identificar e seleccionar técnicas de análise e teste de software em função do contexto tecnológico.
- Avaliar a qualidade de um projecto de software.

5.3.5 Projecto Integrado (PI)

Descrição. Esto módulo é de índole exclusivamente prático, correndo em sessões de laboratório ou de acompanhamento de projectos de forma integrada com os outros módulos da UCE. No início do ano, é definido um projecto integrador (possivelmente proposto no âmbito de uma articulação com a indústria), que os alunos implementarão em sucessivos níveis de sofisticação, à medida que adquirem conhecimentos nos outros módulos. Tipicamente, o projecto começa pela especificação de um modelo formal abstracto do problema proposto, que é de imediato animado em laboratório e sujeito a testes de coerência (invariantes, pós-condições, etc). Mais tarde, é calculada a sua implementação e feita a sua codificação segundo uma arquitectura (eg. cliente-servidor) entretando estudada, tudo acompanhado de um processo de avaliação e teste sistemático.

Este trabalho não exclui o desenvolvimento, em paralelo, de ferramentas de apoio, desenvolvidas pelos alunos e incorporadas no património do grupo de investigação em que os docentes se integram (eg. *UMinho Haskell libraries*, *Camila Revival*, etc). A interoperabilidade com outras technologias (*foreign functions*, *data mappings*) é também componente integrante desta formação, já que, por via de regra, os projectos propostos serão evoluções de projectos já parcialmente implementados (eg. vindos da industria) ou de outros propostos no mesmo ano lectivo por UCEs tecnologicamente distintas.

Resultados de Aprendizagem Específicos

- Planear, executar e avaliar projectos de modelação e desenvolvimento em Engenharia de Software.
- Aplicar ferramentas conceptuais de modelação e cálculo ao projecto de software.
- Integrar diferentes tecnologias e paradigmas computacionais.
- Desenvolver capacidades de trabalho em equipa, comunicação e gestão de projectos de software.

6

EL
Engenharia de Linguagens

Sumário

A necessidade de processar com eficácia linguagens (formais ou naturais) levou à criação de formalismos para especificação das linguagens e respectivo tratamento e introduziu na programação alguns conceitos que se revelaram relevantes e poderosos no sentido de permitir um desenvolvimento rápido e seguro dos programas responsáveis pelo reconhecimento e transformação das frases dessas linguagens (os textos-fonte). Desses conceitos destacamos: a abordagem sintáctico-semântica *aos problemas (que também costumamos designar por* abordagem gramatical*); o uso de* árvores de sintaxe abstractas decoradas com atributos *para representação interna do conhecimento; e a* geração automática, ou síntese, de programas. *Nesta UCE pretende-se ensinar essas bases e transportar os métodos, técnicas e ferramentas para três campos de aplicação prática mais vastos:* engenharia documental*, para anotar e processar estruturalmente documentos, de modo a permitir construir, gerir e usar bibliotecas, arquivos e museus virtuais;* engenharia de software*, fornecendo meios de compreender e transformar programas, tarefas essas que estão na base da manutenção de software;* engenharia do conhecimento*, fornecendo meios de explorar documentos (textos em língua natural e documentos anotados) e extrair deles informação necessária para a sua classificação, sumarização, tradução, etc.*

Coordenação: *Pedro Rangel Henriques, José João Almeida, José Carlos Ramalho, Jorge Gustavo Rocha, Alberto Simões, Maria João Varanda (IPB), José Paulo Leal (FCUP)*

6.1 Objectivos

Pretende-se, nesta unidade modular de 30 ECTS, ensinar gramáticas de atributos como base para a especificação formal da sintaxe e semântica das linguagens e para a construção sistemática dos respectivos processadores. Num sentido mais lato, pretende-se ensinar as gramáticas como me-

todologia de especificação no universo da análise e especificação do modelo de dados de diversas aplicações.
Leccionam-se, também, técnicas expeditas de representação de conhecimento — adequadas à complexidade do conhecimento que tem de ser armazenado e manipulado neste contexto e adequadas à eficiência requerida no âmbito do processamento de linguagens — bem como, linguagens de scripting que fornecem um meio eficaz para manusear strings (textos) e operar com as ditas representações complexas.
Com estas bases, será então possível explorar as técnicas e ferramentas robustas e ágeis, usadas para processar linguagens formais ou naturais, atribuindo aos alunos competências para:

- armazenar e manipular com eficácia o conhecimento associado à representação de objectos complexos, como o significado de um texto, ou os fluxos de dados e de controlo de um programa.
- desenvolver interfaces gráficas que permitam visualizar objectos complexos, descrevendo o conhecimento que se extrai dos textos ou dos programas.
- construir sistemas para análise de programas.
- construir sistemas para transformação (optimização, cálculo parcial, refactoring,...), ou tradução de programas.
- construir sistemas para testar programas e suportar um estilo de programação orientado ao teste.
- construir processadores agéis para Linguagens de Domínio Específico (DSL).
- explorar textos e documentos, extraindo informação para: classificação/criação de indices, síntese/sumarização, população de BDs ou outros repositórios de conhecimento como dicionários, ontologias ou enciclopédias.
- anotar e processar estruturalmente documentos, singularmente ou agrupados em repositórios.
- representar e transportar dados de modo a assegurar a interoperabilidade e o intercâmbio entre sistemas de informação.

6.2 Caracterização Global

6.2.1 Regime e Escolaridade

Esta UCE está organizada em 4 módulos (com 2h teóricas e 2h téorico-práticas cada) mais Projecto Integrador (com 8h práticas), o que corresponderá a uma escolaridade semanal de: 8T + 8TP + 8P.
A UCE apresentada neste documento poderá ser oferecida em regime semestral ou anual (com

metade da escolaridade semanal acima apresentada) de acordo com o público a que se destine. O regime anual será preferencialmente adoptado para a formação inicial, durante a qual num mesmo ano um aluno poderá realizar duas UCE (que serão leccionadas em paralelo em cada semestre), enquanto o regime semestral, mais intensivo mas concentrado no tempo, parece ser mais adequado para oferta em formação contínua ou quando exigido pela estrutura de Mestrados Erasmus-Mundos que venha a integrar.

6.2.2 Áreas Científicas

Os 30 ECTS do módulo correspondem a valências nas áreas científicas de **especificação e processamento de linguagens**, **processamento estruturado de documentos** e da **engenharia de software** (análise e especificação, geração automática, manutenção, transformação e compreensão de programas).

Área Científica geral: Informática (30 ECTS)

Áreas Científicas (classificação ACM): A distribuição dos 30 ECTS da UCE pelas áreas científicas classificadas no ACM Computing Curricula (1998), é apresentada a seguir:

- Grammars and Other Rewriting Systems (5 ECTS)
  Path: ACMCCS98/Theory of Computation/Mathematical logic and formal languages/Grammars and Other Rewriting Systems/
- Processors (5 ECTS)
  Path: ACMCCS98/Software/Programming languages/Processors/
- Natural Language Processing (5 ECTS)
  Path: ACMCCS98/Computing Methodologies/Artificial intelligence/Natural Language Processing
- Program transformation (5 ECTS)
  Path: ACMCCS98/Computing Methodologies/Artificial intelligence/Automatic Programming/Program transformation/
- Information interfaces and presentation (5 ECTS)
  Path: ACMCCS98/Information Systems/Information interfaces and presentation/
- Document and text processing (5 ECTS)
  Path: ACMCCS98/Computing Methodologies/Document and text processing/

6.2.3 Requisitos

Para frequentar esta UCE o aluno deverá ter conhecimentos de programação (algoritmia e estruturas de dados). As restantes competências irá adquiri-las ao longo da frequência da UCE.

6.2.4 Resultados de Aprendizagem

Depois da breve introdução da secção anterior é fácil concluir o aluno frequentador desta UCE terá um leque bastante alargado de competências o que lhe permitirá actuar num série de contextos diversos.

As competências a adquirir ao longo da UCE podem dividir-se em três grupos: tecnológicas, específicas e genéricas; que se enumeram nas secções seguintes.

Competências Tecnológicas

Toda a aprendizagem da Engenharia de Linguagens requer o domínio de um conjunto alargado de tecnologias. Estas competências devem ser adquiridas numa fase inicial pois tudo o resto dependerá delas.

Assim, as competências tecnológicas que um aluno deverá adquirir são as seguintes:

1. Modelação de dados estruturados e semiestruturados baseada em gramáticas: Gramáticas Independentes de Contexto, Gramáticas de Atributos, XML Schema e RelaxNG.

2. Programação baseada em regras: autómatos reactivos, programação por eventos, programação estratégica (uso do Strafunski).

3. Especificação e Processamento de Linguagens de Anotação: XML, XPath e XQuery.

4. Utilização de ferramentas de geração de compiladores, dos casos mais simples como o lex/yacc, ou o Halex/Happy, aos mais evoluídos como o SGen ou o LRC, passando pelos intermédios e clássicos como o LISA, CoCO/R, AntLR, PCCTS, Eli.

5. Programação funcional com XML/XSLT.

6. Programação Web com HTML, JavaScript, CGI Script, PHP, Perl, ou similares.

7. Utilização de ferramentas para extracção automática, manipulação e navegação em ontologias, como o Metamorphosis, ou o Omnigator.

8. Criação de serviços Web a partir de aplicações locais: SOAP, WSDL, ...

9. Utilização de ferramentas para compreensão de programas, como o CodeSurfer, SHriMP, WARE/WANDA, PBS.

10. Utilização de ferramentas de edição especializada como editores estruturados: XMLSpy, Oxygen, Eclipse.

Competências Específicas

No fim da UCE, um aluno deverá ter adquirido as competências específicas que lhe permitem desempenhar as tarefas chave no processo de desenvolvimento dentro das áreas científicas características desta unidade:

1. Ver a Engenharia de Linguagens como uma área multidisciplinar que junta factores de engenharia, tecnológicos e pedagógicos.

2. Compreender a filosofia de desenvolvimento baseada na separação sintáctico-semântica, ou seja, da estrutura/conteúdo do problema, e no uso de uma representação interna do significado para travessia sucessiva a fim de a transformar na solução desejada.

3. Compreender o contexto das aplicações e sistemas baseados em regras.

4. Modelar um sistema: levantamento dos requisitos, especificação dos requisitos do utilizador usando casos de uso e cenários; especificação dos requisitos da informação; especificação dos atributos de apresentação.

5. Especificar a interface.

6. Prototipar incrementalmente e iterativamente: programação; integração e teste de módulos; testes de consistência.

7. Reflectir, avaliar, explicar e justificar soluções de Engenharia de Linguagens.

Competências Genéricas

Independentemente das áreas científicas específicas de cada UCE, um futuro mestre em engenharia terá de adquirir competências relacionadas com a capacidade genérica de *projectar e desenvolver* em equipa *sistemas* complexos, nomeadamente:

1. Planeamento e Gestão de Projectos.

2. Trabalho em equipa.

3. Reutilização de materiais.

4. Pesquisa, relato e apresentação, *escrita* ou *oral*, do processo de desenvolvimento e dos resultados obtidos.

6.3 Estrutura Curricular

6.3.1 Estrutura Base

A presente UCE organiza-se em torno de 4 módulos temáticos, correspondendo cada um a 5 ECTS, articulados entre si por um projecto laboratorial, denominado projecto integrado, que garante a experimentação e aplicação prática dos resultados da aprendizagem.

Os módulos referidos são:

- `EG` Engenharia Gramatical
- `PLN` Scripting no Processamento de Linguagem Natural
- `ATS` Análise e Transformação de Software
- `PED` Processamento Estruturado de Documentos

6.3.2 Métodos de Ensino

De uma forma geral, esta UCE seguirá aquilo que se designa por *ensino orientado ao projecto*. Isto é, a matéria leccionada nos módulos TP será organizada e apresentada tomando por base o trabalho a resolver no contexto do Projecto Integrador.
Em particular, praticar-se-á:

- Ensino teórico-prático (exposição teórica das matérias, imediatamente seguida de exercícios de consolidação no papel e no computador), nos 4 módulos;
- Ensino experimental, no Projecto Integrador.

6.3.3 Avaliação

A avaliação tem duas componentes obrigatórias (contribuindo cada uma com 50% para a nota final):

- Nota teórica, obtida através de 1 Exame escrito único, integrando as matérias dos 4 módulos;
- Nota prática, obtida no trabalho realizado no âmbito do Projecto Integrador.

6.3.4 Conteúdos Programáticos

ENGENHARIA GRAMATICAL (EG)

Programa resumido.

- O paradigma da Programação baseada em Gramáticas (PG).
- Conceitos sobre Gramáticas de Atributos (GA); definição formal.
- Desenvolvimento modular/incremental de GAs (aproximação OO e AO).
- Processamento de Linguagens baseado em GAs (tradução dirigida pela semântica).
- Geração de Programas (processadores de Linguagens, protótipos rápidos) a partir de GAs.
- Representação de Conhecimento:
  - Dicionários (*hash-tables*) e Árvores Generalizadas
  - Frames, Grafos Conceptuais, Redes Semânticas (abordagens clássicas em IA)
  - Ontologias: conceito, representação e manipulação

Resultados de Aprendizagem Específicos:

1. Capacidade para desenvolver especificações da sintaxe/semântica de linguagens e problemas em geral com gramáticas de atributos.
2. Capacidade para gerar programas (protótipos) usando ferramentas automáticas baseadas em gramáticas de atributos.
3. Capacidade para gerar ou utilizar ambientes de desenvolvimento estruturais e orientados à semântica.
4. Capacidade para representar, armazenar e manipular Conhecimento com eficiência.

SCRIPTING NO PROCESSAMENTO DE LINGUAGEM NATURAL (PLN)

Programa resumido.

- Linguagens de scripting:
  - características, objectivos e conceitos.
  - introdução a uma linguagem de Scripting (e.g. Perl).
- Processadores de linguagens regulares e programação orientada à expressão regular.
- Design Patterns no processamento de linguagens.
- Linguagens baseadas em regras:

  - reescrita textual e DSLs baseadas em regras de reescrita.
  - sistemas de produção.
- Processamento estrutural de árvores e DSLs baseadas nestes processadores.
- Processamento de Linguagem Natural: Análise Morfo-sintáctica de textos em língua natural:
  - Modelos morfológicos.
  - Gramáticas lógicas e DCGs.
  - Parsing robusto.
- Processamento de Linguagem Natural: semântica e pragmática.
- Dicionários (multi-fonte) e Thesaurus.
- Extracção de Conhecimento a partir de Textos.
- Sumarização e Classificação.
- Tradução automática.

Resultados de Aprendizagem Específicos:

1. Ser capaz de escrever scripts para automatização de uma variedade de tarefas e transformações.
2. Ser capaz de resolver problemas usando transformações via Expressões Regulares.
3. Ser capaz de compreender as vantagens e o funcionamento de sistemas guiados por regras de produção (condição-reacção).
4. Ser capaz de construir Linguagens de Domínio Específico (DSLs) concretas.
5. Ser capaz de construir e usar corpora.
6. Ser capaz de extrair informação diversa a partir de corpora.
7. Ser capaz de construir dicionários electrónicos.
8. Ser capaz de construir pequenos protótipos para modelar linguagem natural.

ANÁLISE E TRANSFORMAÇÃO DE SOFTWARE (ATS)

Programa resumido.

- Parsing generalizado e Regras de Desambiguidade:
  - Parsing LL ou LR Generalizado (estudo de sistemas como o SGLR e o HaGLR).
- Aspectos quantitativos de Linguagens de Programação:
  - Métricas para Linguagens de Programação.
  - Métricas para análise de programas (e.g., detecção de código morto, clones, etc).
- Transformação de programas:
  - Programação estratégica.
  - Técnicas de *Slicing*.
  - Especialização de programas e cálculo parcial.
- Teste de Software.
- Paradigmas para visualização de dados e de conhecimento:
  - visualização vs navegação.
  - visualização/navegação em árvores, grafos, hiper-cubos, petri-nets.
  - visualização/navegação em ontologias.
  - animação de algoritmos.
  - modelos de visualização, visualização com ponto de vista móvel.
  - Exploração de Ambientes de Trabalho usando os paradigmas explicitados nos items acima.

Resultados de Aprendizagem Específicos:

1. Capacidade de construir front-ends poderosos para a análise de linguagens de programação (ambíguas ou não).
2. Capacidade de desenvolver software como uma tarefa de transformar programas e/ou especificações em implementações eficientes.
3. Capacidade de utilizar métricas e técnicas de transformação de programas para optimizar programas (e.g. cálculo parcial, detecção de código morto), efectuar debugging de programas (e.g. slicing), melhorar a estrutura dos programas (e.g. refactoring).
4. Capacidade para de definir testes para software e testar automaticamente programas em diferentes linguagens de programação.
5. Capacidade para criar representações visuais adequadas à compreensão clara do conhecimento complexo detido.

PROCESSAMENTO ESTRUTURADO DE DOCUMENTOS (PED)

Programa resumido.

- Introdução e um pouco de história: SGML, HTML, XML, SML.
- Documentação Estruturada e Anotação.
- Linguagens de Anotação para a Web: XML, HTML, WML, WSDL, SVG.
- Documentos XML: estrutura e conceitos.
- Ciclo de vida documental.
- Desenvolvimento de DTDs e Schemas.
- Transformação e Manipulação de Documentos:
  - Árvore Documental Abstracta.
  - XPath.
  - Modelos de processamento: DOM e SAX.
  - XSL.
  - XQuery.
- XML e Bases de Dados: armazenamento de dados estruturados e semi-estruturados.
- Integração e intercâmbio de informação entre sistemas.
- Web Semântica:
  - Ontologias para recursos Web: OWL, Topic Maps, RDF.
  - Web Services.
  - Navegação Semântica.
  - Linguagens de Query para Ontologias.
- Publicação Electrónica: a norma XSLFO.
- Estudos de alguns casos de estudo: Docbook, WebServices, OWL, MathML,...

Resultados de Aprendizagem Específicos:

1. Conhecer o ciclo de vida documental dos documentos estruturados. Saber identificar as várias fases e as tecnologias a utilizar em cada uma.

2. Ser capaz de especificar uma linguagem de anotação para um conjunto de requisitos.
3. Ser capaz de implementar transformações de documentos para diversos fins: extracção de conhecimento, publicação na Web, intercâmbio de informação, ...
4. Conhecer e utilizar soluções de armazenamento para documentos anotados.
5. Ser capaz de definir as camadas necessárias para integrar e realizar o intercâmbio de informação entre sistemas de informação distintos.
6. Ser capaz de implementar um projecto de publicação electrónica recorrendo a normas internacionais abertas: XML, XSLFO e XSL.
7. Ser capaz de programar a geração automática de sítios Web a partir de um repositório de documentos XML.
8. Ser capaz de utilizar linguagens de anotação e respectivas ferramentas desenvolvidas por outrém.

PROJECTO INTEGRADO (PI)

Descrição.
O projecto terá de integrar, por um lado a construção de processadores de linguagens com base em gramáticas de atributos para criar um sistema de análise e teste de programas. Esse ambiente poderá ter facilidade para realizar algumas operações de transformação de programs pré-definidas ou permitir a definição e adição fácil de novos operadores.
Por outro lado deve incluir a manipulação de documentos (textos não-estruturados ou semi-estruturados) para extracção de conhecimento. que possa ser útil para integrar no ambiente de análise e teste.
No desenvolvimento do projecto deve ser usada a programação suportada por sistemas de scripting e devem ser usados ambientes estruturados e incrementais de apoio à programação.

A título de exemplo e para deixar bem clara e concreta a proposta, uma instância de projecto poderia ser *a criação de um ambiente para Compreensão de Programas (CP), com todas as questões de análise e visualização que esse ambiente levanta, em que adicionalmente seja necessário explorar a documentação associada à aplicação que se pretende mostrar e compreender com vista a extrair dessa documentação conhecimento sobre os domínios do problema e do programa*.

Programa resumido.

- Desenvolvimento de um projecto único abrangendo as áreas leccionadas nos 4 módulos TP.

Resultados de Aprendizagem Específicos:

1. Ser capaz de utilizar as tecnologias leccionadas nas várias disciplinas do módulo para solucionar os problemas propostos[1].
2. Ser capaz de trabalhar em equipa para a realização de um projecto.
3. Ser capaz de discutir o trabalho desenvolvido, criticá-lo e apresentá-lo a terceiros.

[1] Ver detalhes das capacidades a atingir em cada um nas subsecções precendentes.

7

CPD
Computação Paralela Distribuída

Sumário

O presente documento descreve uma UCE30 (="Unidade Curricular de Especialidade"), creditada com 30 ECTS, para formação avançada ao nível do 2º ciclo de estudos universitários, no espírito do processo de Bolonha, na sequência da oferta formativa em Informática na Universidade do Minho.

Esta UCE30 é fruto da colaboração científica entre o Grupo de Engenharia de Computadores do Dep. Informática e o Grupo de Análise Numérica do Dep. de Matemática, na área da computação paralela, existente desde o início da década de 90, que conduziu com sucesso várias teses de doutoramento.

A computação paralela e distribuída evoluiu na última década para um modelo integrador de componentes de custo reduzido e de elevada capacidade, nomeadamente ao nível da arquitectura dos processadores multi-core e das tecnologias de interligação. Teve como consequência a proliferação de clusters computacionais para a resolução de problemas computacionalmente intensivos, e a sua interligação em grelha inter-institucional (grid) como forma de se aceder a um maior poder computacional a uma fracção do custo de um supercomputador da geração anterior.

O domínio das tecnologias de CPD para a utilização, programação e administração eficiente destes novos instrumentos da sociedade de conhecimento é um factor chave para o desenvolvimento do Minho, numa busca de excelência em tecnologias de informação.

Coordenação: *Alberto Proença, António Pina, João Sobral, Rui Ralha (DMat)*

7.1 Contexto e Objectivos

O aumento em complexidade e dimensão dos problemas e o volume de dados a tratar, em domínios tão diversos que vão da física à engenharia ou da biologia às ciências sociais, impõe a

utilização de ferramentas de elevado custo computacional, normalmente associadas à utilização de supercomputadores, sistemas com espaço disponível para armazenamento de dados, memória e velocidade de processamento cada vez maiores. Frequentemente tais requisitos inviabilizam o desenvolvimento de projectos relacionados por instituições públicas, laboratórios e empresas que não disponham de muitos recursos, devido aos altos custos de aquisição, de manutenção e de actualização dos equipamentos necessários.

A crescente disponibilidade de processadores com paralelismo interno, de nós de computação de elevado desempenho e de capacidade de armazenamento de dados, e aliada às recentes tecnologias de interligação de alta velocidade, apresenta-se como uma alternativa economicamente viável para solucionar este problema. Uma nova visão do que tradicionalmente se designava por computação paralela e distribuída (CPD) tem vindo a impor-se: a actual CPD em ambiente de rede local está concentrada em clusters computacionais Beowulf, substituindo os computadores vectoriais e os MPP, e quando o ambiente de rede ultrapassa os limites institucionais encontramo-nos perante uma grelha computacional (grid).

Os clusters concentram num espaço reduzido o elevado poder computacional dos actuais processadores que suportam formas elaboradas de paralelismo na execução de código, permitindo uma concretização eficiente e económica de um sistema de computação de elevado desempenho, enquanto a grelha computacional possibilita a utilização da Internet para a partilha e agregação de recursos na resolução de problemas específicos. Aos utilizadores é oferecida uma maior flexibilidade em termos de configuração, actualização e fornecedores, o que favorece a longevidade do sistema e confiança no investimento realizado.

Adicionalmente, os clusters Beowulf integram uma extensa gama de software de sistema que fornecem soluções de baixo custo ou a custo zero para a maior parte das necessidade de funcionamento do sistema. Como resultado, os clusters têm vindo a afirmar-se em quase todos os segmentos da computação paralela e distribuída, sendo os nós por excelência de uma grelha computacional alargada.

A computação usando clusters interligados em grelha envolve quatro áreas distintas mas interrelacionadas, a saber:

- a estrutura do sistema físico computacional com a dupla finalidade de execução eficiente de aplicações e da gestão de recursos;
- a identificação e utilização de algoritmos, modelos, bibliotecas e ferramentas de apoio à programação paralela;
- a análise dos algoritmos e métodos numéricos escaláveis mais comuns e respectiva complexidade;
- o ambiente de administração e de gestão de recursos computacionais, quer em ambiente de cluster, quer em grelha.

A estrutura do sistema computacional inclui todos os aspectos do equipamento e componentes dos nós e as suas capacidades, com destaque para as evoluções nas relações CPU-memória,

as placas de rede dedicadas e os comutadores, as tecnologias de interconexão e a topologia que determina a organização global do sistema.

O domínio dos algoritmos paralelos fornece os modelos e abordagens usados para explorar os níveis intrínsecos de paralelismo dos problemas no contexto dos recursos disponíveis e dos requisitos efectivos de desempenho. Os algoritmos numéricos terão um tratamento mais aprofundado, bem como a análise da sua complexidade, tendo em vista uma melhor preparação no âmbito da computação científica. As bibliotecas de programação distribuída e as ferramentas de suporte determinam os paradigmas usados pelos utilizadores para coordenar os recursos distribuídos de computação e executar os componentes que constituem as aplicações paralelas.

Finalmente, o ambiente de administração e gestão de recursos é constituído pela bateria de software de sistema e utensílios que governam todas as fases da operação do sistema de operação desde a instalação, configuração e iniciação, através dos quais os administradores monitorizam o estado do sistema, fazem o diagnóstico de falhas e a manutenção.

7.2 Caracterização Global

7.2.1 Regime e Escolaridade

A UCE em CPD está estruturada para funcionamento preferencial em regime anual, podendo, em caso excepcional (Erasmus-Mundus, por exemplo), ser adaptado a um regime semestral.

A UCE em CPD está organizada em 4 módulos temáticos de índole teórica e metodológica com 5 ECTS cada (2 por semestre), complementada por um projecto integrador de duração anual.

O planeamento do tempo de contacto estudante-docente foi efectuado para uma distribuição equitativa das componentes de exposição participada de conceitos e exemplos (120h), de diálogo e treino da capacidade de resolução de problemas (120h), e de experimentação laboratorial (120h), prevendo-se uma maior incidência das primeiras componentes no início de cada semestre, e da parte experimental no fim de cada semestre.

7.2.2 Áreas Científicas

Áreas Científicas: Informática (23 ECTS), Matemática (7 ECTS)

Áreas Científicas (classificação ACM): [1]

- (ACMCCS98 / Computer Systems Organization / PROCESSOR ARCHITECTURES / Parallel Architectures / , ACMCCS98 / Computer Systems Organization / PERFORMANCE OF SYSTEMS / Measurement techniques / , ACMCCS98 / Software / SOFTWARE ENGINEERING / Metrics /) - 7 ECTS
- (ACMCCS98 / Software / PROGRAMMING TECHNIQUES / Concurrent Programming / , ACMCCS98 / Software / SOFTWARE ENGINEERING / Coding Tools and Techniques /) - 8 ECTS

[1] The ACM Computing Classification System [1998 Version], em http://www.acm.org/class/1998/

- (ACMCCS98 / Mathematics of Computing / NUMERICAL ANALYSIS / , ACMCCS98 / Mathematics of Computing / PROBABILITY AND STATISTICS / , ACMCCS98 / ANALYSIS OF ALGORITHMS AND PROBLEM COMPLEXITY / Numerical Algorithms and Problems) - 7 ECTS
- (ACMCCS98 / Software / OPERATING SYSTEMS / , ACMCCS98 / MANAGEMENT OF COMPUTING AND INFORMATION SYSTEMS / Installation Management / , ACMCCS98 / MANAGEMENT OF COMPUTING AND INFORMATION SYSTEMS / Software Management /) - 8 ECTS

7.2.3 Requisitos

- Experiência em Unix e em programação imperativa, de preferência em C.
- Conhecimentos de arquitectura de computadores, de análise numérica e de sistemas operativos a nível de 1º ciclo.

7.2.4 Resultados de Aprendizagem

Um estudante que complete com sucesso a UCE em CPD deverá ser capaz de demonstrar que adquiriu as seguintes competências:

- Caracterizar e avaliar qualitativa e quantitativamente a arquitectura de sistemas de computação paralelos/distribuídos e respectivo desempenho na execução de aplicações.
- Projectar, desenvolver, implementar e optimizar aplicações paralelas e distribuídas, utilizando paradigmas de computação paralela.
- Caracterizar e desenvolver implementações sequenciais e paralelas de algoritmos e métodos numéricos.
- Planear e instalar clusters e grelhas computacionais, e avaliar a aplicar as técnicas mais adequadas de administração e gestão dos recursos.
- Planear, executar e divulgar os resultados de projectos de avaliação e optimização de desempenho de sistemas de computação na execução de tarefas.
- Desenvolver de forma integrada e em equipa, a função de concepção e projecto em Engenharia da Computação, e respectiva comunicação escrita e oral dos resultados.

7.3 Estrutura Curricular

7.3.1 Estrutura Base

A UCE em CPD está organizada em 4 módulos temáticos, de 5 ECTS cada, os quais se articulam entre si através de um módulo integrador que tem como principal objectivo a concretização de

um projecto (no sentido lato), passando pelas fases de procura e síntese de informação, auto-aprendizagem de conceitos complementares aos adquiridos nos módulos temáticos, treino de aptidões experimentais na aplicação dos conhecimentos adquiridos, e análise, interpretação e avaliação de resultados experimentais, comunicação escrita e defesa oral.

Os módulos temáticos que constituem a UCE em CPD são:

SCD Sistemas de Computação e Desempenho

PCP Paradigmas de Computação Paralela

AMN Algoritmos e Métodos Numéricos

PAC Planeamento e Administração de Clusters

7.3.2 Métodos de Ensino

- Exposição de conceitos e análise/discussão participada de casos de estudo.
- Trabalho de grupo na resolução de exercícios de aplicação prática de conceitos, com eventual recurso a ferramentas informáticas.
- Trabalho de projecto em grupo na integração e aplicação laboratorial de conceitos adquiridos nos módulos temáticos e por auto-aprendizagem, sob a estreita coordenação de toda a equipa docente.

7.3.3 Avaliação

O processo de avaliação é essencialmente constituído por 2 componentes: uma que tem como objectivo avaliar a capacidade de assimilação e compreensão dos conceitos adquiridos (as competências intelectuais), e outro para ajuizar e quantificar as competências práticas resultantes da integração e aplicação laboratorial dos conhecimentos na resolução concreta de problemas. A primeira componente é avaliada em exame final, enquanto a segunda é avaliada por um relatório final e consequente defesa oral dos resultados obtidos.

A classificação final será a média ponderada destas duas componentes.

7.3.4 Conteúdos Programáticos

MÓDULO SISTEMAS DE COMPUTAÇÃO E DESEMPENHO(SCD)

Descrição.

A aquisição dos conhecimentos básicos sobre o funcionamento e arquitectura dos sistemas informáticos é tema de 1º ciclo de qualquer curso de informática. A compreensão da complexidade crescente dos mecanismos de aceleração que têm vindo a ser introduzidos ao nível da arquitectura dos processadores e respectiva hierarquia de memória, complementados com as tecnologias de interligação destes pares CPU-memória, é cada vez mais crucial para a programação e utilização eficiente dos recursos computacionais. A recente evolução dos processadores com pipeline, superescalaridade, multi-threading e multi-core veio chamar de novo a atenção para o impacto que os modelos de desenvolvimento de software podem ter no seu desempenho. Uma apresentação crítica desta evolução e numa perspectiva do desempenho na execução de aplicações, pode contribuir para esse objectivo.

A execução eficiente de código requer a utilização criteriosa de métricas e a análise de técnicas optimização de codificação e compilação de código, quer as independentes do sistema de computação, quer as que dele podem depender. Este módulo deve assim assegurar que as questões de profiling, métricas, benchmarking e de análise e optimização de programas são devidamente discutidas.

Os sistemas de computação de elevado desempenho são constituídos por nós de computação interligados por diversas tecnologias, os quais se agrupam em clusters e em grelhas computacionais. Uma apresentação dos conceitos por detrás destas tecnologias contribui para uma melhor adaptação profissional aos sistemas vindouros.

Programa resumido.

- Evolução recente da arquitectura de processadores: pipeline, superescalaridade, multi-threading, multi-core. Hierarquia de memória em ambiente partilhado.
- Avaliação de desempenho na execução de aplicações: métricas, profiling, análise e afinação de código (destaque para ciclos iterativos, manuseamento de matrizes e chamada de funções), benchmarking.
- Evolução do paralelismo nas arquitecturas: arquitecturas vectoriais, arquitecturas paralelas com memória partilhada e memória distribuída, arquitecturas distribuídas em rede. Análise de desempenho em arquitecturas paralelas.

Resultados de Aprendizagem Específicos:

Um estudante que complete com sucesso a parte do exame referente a este módulo deverá ser capaz de demonstrar que adquiriu as seguintes competências:

- Identificar e caracterizar a introdução dos diversos tipos de paralelismo na arquitectura interna dos processadores e no seu acesso à memória.
- Seleccionar as métricas, métodos e técnicas de avaliação de desempenho de programas em execução, para configurações da arquitectura de um sistema computacional.
- Avaliar quantitativamente o desempenho de um sistema de computação na execução de aplicações informáticas.

- Descrever e caracterizar os principais componentes de um sistema de computação em cluster ou em grelha.
- Avaliar criticamente a arquitectura dos sistemas de computação paralelos e distribuídos.

MÓDULO PARADIGMAS DE COMPUTAÇÃO PARALELA (PCP)

Descrição.

Com este módulo pretende-se introduzir os conceitos básicos do desenvolvimento de aplicações que tirem partido de arquitecturas paralelas. São desenvolvidas competências necessárias para o projecto e desenvolvimento de aplicações que executem numa gama alargada de arquitecturas, incluindo arquitecturas multi-core¸ Clusters de máquinas e grelhas computacionais (grid).

O módulo incide fortemente nos paradigmas base da computação paralela, designadamente em modelos de programação baseados em fios de execução, passagem de mensagens e objectos distribuídos, e nas metodologias de desenvolvimento de aplicações paralelas. O módulo inclui ainda uma introdução aos tipos básicos de algoritmos paralelos (pipeline, farming, etc.), à medição e optimização do desempenho das aplicações em sistemas de memória distribuída, incluindo a análise e optimização do rácio computação / comunicação e o escalonamento de recursos ao nível das aplicações.

Programa resumido.

- Modelos de programação: fios de execução, passagem de mensagens, objectos distribuídos, workflows.
- Metodologias de desenvolvimento de aplicações paralelas: partição, comunicação, agregação e mapeamento de tarefas.
- Análise de algoritmos paralelos típicos: pipelining, farming, heartbeat e divide & conquer.
- Medição e optimização do desempenho de aplicações em sistemas de memória distribuída.
- Escalonamento de recursos ao nível da aplicação.

Resultados de Aprendizagem Específicos:

Um estudante que complete com sucesso a parte do exame referente a este módulo deverá ser capaz de demonstrar que adquiriu as seguintes competências:

- Desenvolver aplicações paralelas capazes de executar numa gama alargada de arquitecturas.
- Implementar aplicações utilizando os tipos mais comuns de algoritmos paralelos.
- Medir e optimizar o desempenho de aplicações em sistemas de memória distribuída.

- Implementar técnicas de gestão de recursos ao nível da aplicação.

MÓDULO ALGORITMOS E MÉTODOS NUMÉRICOS (AMN)

Descrição.

Esta UCE pretende dotar os alunos de competências na área dos métodos numéricos e aplicações. A selecção dos métodos numéricos a expor e a debater com os estudantes é orientada por critérios de abrangência, procurando-se abarcar o tratamento de problemas de natureza muito variada; um objectivo central é o de mostrar, através de aplicações concretas ("case-studies") que requerem algoritmos de elevada complexidade computacional, a importância do desenvolvimento de algoritmos paralelos. Na concepção destes algoritmos privilegia-se o modelo de passagem de mensagens e, por esta razão, a distribuição de dados e as necessidades de comunicação entre diferentes nós de computação são factores determinantes para a eficiência dos códigos desenvolvidos.

Programa resumido.

- Métodos de Monte Carlo: geradores sequenciais e paralelos de números aleatórios, aplicações do método de Monte Carlo ("case studies").
- Multiplicação de matrizes: implementações sequenciais, implementações paralelas no modelo "message-passing".
- Sistemas de equações lineares: método de eliminação de Gauss, métodos iterativos, análise da convergência, implementações sequenciais e paralelas.
- Método das diferenças finitas: resolução numérica das equações diferenciais que ocorrem em certos problemas "clássicos"(vibração de uma corda, difusão de calor).
- Transformada rápida de Fourier: análise de Fourier (introdução), as transformadas DDT (discreta) e FFT, algoritmos sequenciais e paralelos.

Resultados de Aprendizagem Específicos:
Um estudante que complete com sucesso a parte do exame referente a este módulo deverá ser capaz de demonstrar que adquiriu as seguintes competências:

- Conhecer a um nível básico a teoria matemática e numérica que serve de suporte aos métodos estudados.
- Identificar, a diferentes níveis (complexidade, robustez numérica, etc.) pontos fortes e fracos dos métodos.
- Desenvolver implementações sequenciais e discutir os resultados numéricos obtidos.

- Desenvolver implementações paralelas no paradigma de passagem de mensagens.
- Identificar, nos algoritmos paralelos, eventuais problemas de balanceamento de carga e/ou elevados custos de comunicação entre os processadores.
- Usar bibliotecas numéricas (BLAS, Lapack, Scalapack).

MÓDULO PLANEAMENTO E ADMINISTRAÇÃO DE CLUSTERS (PAC)

Descrição.

O módulo de Planeamento e Administração de Clusters, tem como objectivo geral oferecer aos estudantes uma formação sólida no planeamento e administração de Clusters e à computação em Grid que beneficia da existência de um cluster HPC instalado no Dep. Informática (SEARCH), que irá ser usada ao longo do tempo como plataforma para experimentação e validação dos temas apresentados.

O módulo acompanha a emergência dos clusters, como um novo ponto de referência, em termos da relação preço/rendimento, para a construção de sistema de computação de elevado desempenho (HPC), apesar de muitos dos temas que aborda poderem ser aplicados em sistemas de clusters de elevada disponibilidade (HPA). O programa inclui os aspectos mais relevantes do planeamento sistema, das escolhas de equipamento e da instalação do sistema operativo e do ambiente de administração e gestão de recursos. Uma vez que a existe uma grande variedade de de opções em cada uma das áreas dos software para clustering o curso inclui a discussão dos prós e contras dos mais importantes projectos de software livre e a escolha configuração e uso dos mais adaptados às tarefas de administração de recursos.

Programa resumido.

- Introdução aos clusters: arquitectura (terminologia, tecnologias, limitações); equipamentos (componentes individuais e de rede, coordenação de recursos descentralizados).
- Linux para clusters Beowulf: características; instalação e configuração de serviços; segurança.
- Planeamento e construção de clusters: missão, arquitectura e suporte lógico; sistemas de ficheiros paralelo; tecnologias de interligação; clonagem e automatização de instalações.
- Gestão de clusters: modelos de execução de trabalhos, monitorização e administração de utilizadores e recursos, políticas de escalonamento e contabilização, segurança de dados, análise de desempenho e afinação.
- Construção de grids: arquitectura, protocolo e serviços; modelos de utilização e segurança; interfaces de programação.

- Estudo de Caso (o cluster SEARCH na Universidade do Minho, com Rocks, Ganglia, OpenPBS, Maui, GPFS).

Resultados de Aprendizagem Específicos:
Um estudante que complete com sucesso a parte do exame referente a este módulo deverá ser capaz de demonstrar que adquiriu as seguintes competências:

- Identificar os diferentes tipos de arquitecturas de cluster e grids e discutir as respectivas vantagens e limitações.
- Analisar e avaliar os requisitos de hardware e software com vista ao planeamento e instalação do equipamento.
- Identificar os requisitos dos utilizadores necessários para seleccionar o software a instalar e a manter.
- Identificar e caracterizar políticas de gestão e escalonamento de trabalhos e avaliar os resultados da sua aplicação.
- Discutir e avaliar o desempenho efectivo de programas paralelos em clusters e grids.

MÓDULO PROJECTO INTEGRADO (PI)

Descrição.

O Projecto Integrado tem, como o próprio nome sugere, a função de aliar a componente experimental laboratorial da UCE em CPD, com uma integração com o conjunto de conhecimentos e aptidões intelectuais adquiridos durante a frequência nos módulos temáticos.

O módulo do Projecto Integrado, que se desenrola durante os 2 semestres, contempla 4 fases de desenvolvimento de aplicações de complexidade crescente: uma fase introdutória de treino de competências e de ambiente de trabalho, 2 fases em ambiente de computação paralela em cluster, e uma fase em grelha computacional.

Numa fase inicial são desenvolvidas aptidões de utilização de um ambiente de trabalho em cluster e respectivo suporte ao desenvolvimento de aplicações sequenciais e paralelas, complementado com utilização de bibliotecas numéricas.

Numa segunda fase é desenvolvida uma aplicação simples funcionado em sistemas de memória partilhada. Esta fase incide sobre a análise e optimização de código em sistemas uniprocessador (single-core e multi-core) e multi-processador de memória partilhada.

Numa terceira fase a aplicação é adaptada para sistemas de memória distribuída, tirando partido do cluster computacional existente. Nesta fase os alunos irão projectar a aplicação seguindo as metodologias leccionadas nos módulos do primeiro semestre, efectuando a partição da aplicação em vários módulos, que serão executados de forma distribuída. A aplicação alvo terá complexidades que deverão ser interpretadas à luz dos conhecimentos adquiridos com os algoritmos numéricos do módulo temático específico. Os alunos deverão ter a capacidade de

prever analiticamente o desempenho da aplicação e confirmar, através da experimentação, o desempenho real e justificar eventuais discrepâncias. Adicionalmente, em face da experimentação, a aplicação deverá ser ajustada por forma a optimizar o desempenho.

Na última fase a aplicação é adaptada para grelhas computacionais, funcionando em múltiplos domínios administrativos.

Programa resumido.

- Iniciação ao ambiente de trabalho em cluster computacional e respectivas ferramentas de desenvolvimento de software e de apoio à execução.
- Iniciação ao método científico em práticas laboratoriais experimentais e em trabalho de grupo.
- Análise experimental de execução de código e consequente avaliação e optimização do desempenho da execução, em ambiente paralelo (multi-threading, por passagem de mensagens e híbrido).
- Resolução de problemas computacionais de média complexidade, com requisitos de métodos numéricos.

Resultados de Aprendizagem Específicos:

Um estudante que complete com sucesso a parte do exame referente a este módulo deverá ser capaz de demonstrar que adquiriu as seguintes competências:

- Planear, executar e divulgar os resultados de projectos de avaliação e optimização de desempenho de sistemas de computação na execução de tarefas.
- Desenvolver de forma integrada e em equipa, a função de concepção e projecto em Engenharia da Computação, e respectiva comunicação escrita e discussão oral dos resultados.

8

SI
Sistemas Inteligentes

Sumário

O presente documento descreve uma Unidade Curricular de Especialidade (UCE), creditada com 30 ECTS, oferecida pelo Grupo de Inteligência Artificial do Departamento de Informática da Escola de Engenharia da Universidade do Minho, para o 2º ciclo da oferta formativa em Informática, na sequência da re-estruturação das licenciaturas no contexto do Processo de Bolonha.

Coordenação: *Paulo Novais, Cesar Analide, Luís Brito (PhD-Indústria), Manuel Santos (DSI), Fernando Ribeiro (DEI)*

8.1 Objectivos

A Inteligência Artificial (IA) é por um lado uma ciência que procura estudar e compreender o fenómeno da cognição, e por outro lado um ramo da engenharia, na medida em que procura construir instrumentos de uso universal. A IA é inteligência como computação, tenta simular o pensamento dos peritos e os fenómenos cognitivos que daí decorrem, i.e., procura um corpo de explicações algorítmicas dos seus processos cognitivos. É isto o que distingue a IA dos outros campos de saber; a IA coloca a ênfase na elaboração de teorias e modelos da inteligência como programas.

Com esta unidade pretende-se dar uma formação na área científica da IA. As disciplinas que a compõem garantem que o aluno possuirá conhecimentos ao nível das tecnologias, algoritmos e metodologias de resolução de problemas mais comuns nesta área, sendo capaz de as contextualizar, aplicar e analisar, i.e., pretende-se estabelecer as competências básicas definidoras de um perfil de Mestre em Informática da Universidade do Minho.

A unidade terá ainda uma forte componente prática. Neste âmbito, será incluído um projecto integrador das matérias versadas nas diferentes disciplinas (e.g., na Área das Ciências da Saúde, do Direito, do e-Business, da Robótica Inteligente).

8.2 Caracterização Global

8.2.1 Regime e Escolaridade

A UCE apresentada neste documento poderá ser oferecida em regime semestral ou anual de acordo com o público a que se destine. O regime anual será preferencialmente adoptado para a formação inicial, durante a qual num mesmo ano um aluno poderá realizar duas UCE, enquanto o regime semestral, mais intensivo, mas concentrado no tempo, parece ser o mais adequado para a oferta em formação contínua ou quando exigido pela estrutura de Mestrados Erasmus-Mundos que venha a integrar. Atendendo à sua articulação em 4 módulos e um projecto integrador, conforme se detalha a seguir, a escolaridade proposta é a seguinte: 8T + 8TP + 8P (sendo as 8P correspondentes à componente de projecto integrador).

8.2.2 Áreas Científicas

Áreas Científicas: Informática (30 ECTS).

Áreas Científicas (classificação ACM): É a seguinte a distribuição dos 30 ECTS da UCE pelas áreas científicas classificadas no ACM Computing Curricula (1998):

I. Computing Methodologies -- 10
I.2 Artificial Intelligence
I.2.1 Applications and Expert Systems
I.2.3 Deduction and Theorem Proving
I.2.4 Knowledge Representation Formalisms and Methods
I.2.6 Learning
I.2.8 Problem Solving, Control Methods, and Search
I.2.9 Robotics
I.2.11 Distributed Artificial Intelligence

H. Information Systems -- 10
H.2 Database Management
H.2.5 Heterogeneous Databases
H.2.8 Database Applications

K. Computing Milieux -- 10
K.4 Computers and Society
K.4.4 Electronic Commerce

8.2.3 Requisitos

1. Licenciatura do 1º ciclo em TIEs, o que pressupõe experiência em programação, incluindo, preferencialmente, alguma familiaridade com a programação declarativa (e.g., programação em lógica).

2. Conhecimentos de lógica e matemática discreta a nível de primeiro ciclo.

8.2.4 Resultados de Aprendizagem

1. Avaliar se um dado sistema inteligente é o mais apropriado para a solução de um dado problema em particular.
2. Compreender como usar a informação disponível para implementar sistemas de aprendizagem, selecção de modelos e teste.
3. Compreender as vantagens e/ou desvantagens dos sistemas inteligentes estudados no curso, e decidir qual é o mais apropriado para corporizar uma dada funcionalidade(s).
4. Aplicar entidades virtuais, construídas em termos das redes neuronais artificiais, programação genética e evolucionária, sistemas simbólicos avaliando o seu desempenho.
5. Compreender a relação entre a complexidade de um modelo e o seu desempenho, utilizando esta informação na definição de uma estratégia para optimizar os sistemas existentes.
6. Desenvolver de forma integrada a função de concepção e projecto em Engenharia.

8.3 Estrutura Curricular

8.3.1 Estrutura Base

A presente UCE organiza-se em torno de 4 (quatro) módulos temáticos, correspondendo cada um a 5 ECTS, articulados entre si por um "bus" laboratorial, denominado *projecto integrado*, que garante a experimentação e aplicação prática dos resultados da aprendizagem. Esta estrutura é apresentada na Figura 1. Os módulos referidos são:

BAEC Bases de Dados, Aprendizagem e Extracção de Conhecimento
CI Computação Inteligente
AI Agentes Inteligentes
SA Sistemas Autónomos

8.3.2 Métodos de Ensino

1. Exposição de conceitos e análise de casos de estudo.
2. Trabalho de grupo na resolução de exercícios.
3. Trabalho de projecto em grupo sob a orientação da equipa docente afecta à UCE.

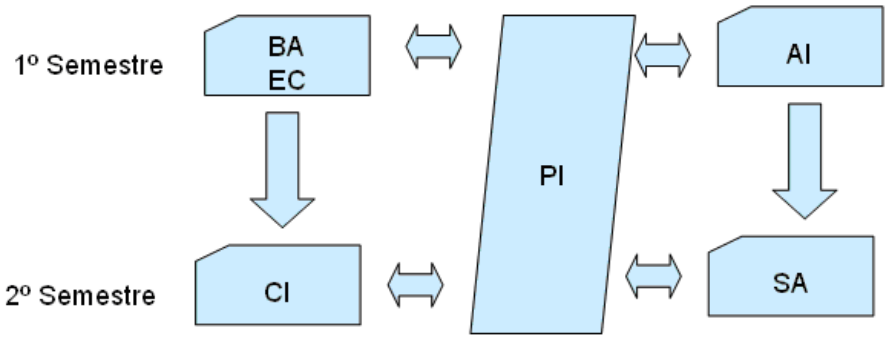

Figura 8.1: Estrutura Curricular da UCE30 - Sistemas Inteligentes

8.3.3 Avaliação

O processo de avaliação visa aferir não apenas os conhecimentos e competências mínimas adquiridas, mas também a capacidade revelada na sua integração e aplicação no projecto de engenharia de software. Assim, esta é referida através de um exame final, a que se soma a avaliação contínua em prática laboratorial e o resultado (ferramentas, relatório, apresentação) do projecto integrador.

8.3.4 Conteúdos Programáticos

Os conteúdos programáticos desta UCE são agrupados em 4 (quatro) módulos temáticos e 1 (um) de projecto, conforme a seguir se detalha.

BASES DE DADOS, APRENDIZAGEM E EXTRACÇÃO DE CONHECIMENTO

Descrição

Actualmente, o uso de inferência (e.g., probabilística) para o desenvolvimento de sistemas inteligentes tem sido objecto de atenção pela comunidade científica. Em boa verdade se pode dizer que um dos objectivos mais perseguidos pelos investigadores passa pelo desenvolvimento de técnicas de inferência eficientes para o uso em sistemas inteligentes. No entanto, o seu uso pressupõe a disponibilidade de modelos de conhecimento válidos. Surgem assim as técnicas de "Data Mining" e a descoberta de conhecimento baseado na tecnologia de Redes Bayesianas, Redes Neuronais Artificiais, Programação Genética e Evolucionária, "Enxames de Partículas", para a extracção de conhecimento a partir de Bases de Dados. O processo de construção automática de modelos de conhecimento com base no enunciado em epígrafe, parte de dois princípios básicos: (i) o algoritmo deve considerar sómente as variáveis mais relevantes para a tomada de decisão, i.e., deve delimitar ao máximo o número de variáveis sem perda de informação; (ii) o

algoritmo deve realizar esta delimitação automaticamente, sem a intervenção do utilizador (e.g., o médico não elege um diagnóstico baseado em miríades de sintomas, mas sim a partir dos sintomas com maior relevância).

Programa resumido

1. Aprendizagem automática supervisionada: Extracção de conhecimento de dados e aprendizagem automática.

2. Construção automática de árvores de decisão.

3. Construção automática de regras. Algoritmo de cobertura. Avaliação de árvores de decisão / regras. Validação cruzada. Testes de significância.

4. Classificação versus regressão.

5. Transformação de variáveis contínuas em variáveis categóricas. Combinação de atributos através de indução construtiva.

6. Aprendizagem não-supervisionada. Metodologias de agrupamento. Sistemas de KDD (e.g., Clementine) e aplicações.

7. Integração de Conhecimento: Métodos para selecção e combinação de modelos.

8. Combinação paralela e "stacking". Combinação de previsões versus integração de conhecimento.

Resultados de Aprendizagem

1. Identificar, descrever e definir os principais conceitos relacionados com os sistemas de “data warehousing”, processamento analítico de dados, mineração de dados, extracção de conhecimento e aprendizagem máquina.

2. Procurar utilizar, classificar e avaliar as aplicações existentes ou a desenvolver, que dêm corpo aos sistemas acima equacionados.

3. Seleccionar as metodologias apropriadas e aplicar software disponível na resolução de problemas reais, quer ao nível da análise de dados, mineração de dados, extracção de conhecimento e tomada de decisão.

4. Conhecer e ser capaz de implementar os principais algoritmos relacionados com as técnicas de mineração de dados, extracção de conhecimento e formas de aprendizagem, referidas em epígrafe.

COMPUTAÇÃO INTELIGENTE

Descrição

Neste módulo estudar-se-ão os métodos de construção de programas capazes de exibir um comportamento inteligente na realização de tarefas complexas, para as quais a computação convencional ainda não consegue resultados satisfatórios. Em particular, uma redobrada atenção sera devotada à computaccão simbólicas..

Programa resumido

1. Inteligência Evolutiva
2. Inteligência Colectiva
3. Computação Natural

Resultados da Aprendizagem

1. Identificar, descrever e definir os principais conceitos relacionados com Computação Evolutiva (e.g. Entidades Virtuais), Inteligência Colectiva (e.g., "Enxame de Partículas") e Computação Natural (e.g., Redes Neuronais Artificias, Programação Genética e Evolucionária).
2. Seleccionar as metodologias de resolução de problemas, assim como os paradigmas computacionais que melhor se lhes adequam.

AGENTES INTELIGENTES

Descrição

Um agente é uma entidade computacional com um comportamento autónomo que lhe permite decidir sobre as suas próprias acções. Os agentes têm uma existência própria, independente da existência de terceiros. Usualmente, cada agente possui um conjunto de características comportamentais que definem a sua competência, um conjunto de objectivos, e a autonomia necessária para utilizar as suas capacidades comportamentais. A decisão sobre qual a acção a desencadear é determinada pelo agente, tendo em consideração as mudanças que ocorrem no ambiente em que actua e o desejo (intrínseco) de alcançar os seus fins.

Programa resumido

1. Agentes deliberativos, reactivos, híbridos e BDI
2. Agentes Móveis

3. Agentes de Interface e de Pesquisa
4. Arquitecturas de agentes

Resultados da Aprendizagem

1. Identificar, descrever e definir os principais conceitos relacionados com Agentes
2. Identificar situações que beneficiem da utilização de sistemas baseados em Agentes.
3. Conhecer e ser capaz de utilizar as metodologias de resolução de problemas e os paradigmas computacionais associados à utilização de agentes.

SISTEMAS AUTÓNOMOS

Descrição

Este módulo tem por objectivo apresentar ao aluno os conceitos que fundamentam uma solução alternativa, embora não exclusiva, à teoria de controle clássica; i.e., adquirir conhecimentos no actual estado da arte no domínio da robótica autónoma e das suas técnicas e metodologias de desenvolvimento e de resolução de problemas. Desta forma, as ferramentas matemáticas necessárias à análise de sistemas dinâmicos são apresentadas juntamente com os aspectos históricos que motivaram o desenvolvimento e emprego destas ferramentas. Assim, o aluno consegue visualizar a necessidade e a importância de utilizar-se tais conceitos para analisar o comportamento de tais sistemas.

Programa resumido

1. Sistemas Robóticos Autónomos
2. Arquitecturas para Sistemas Autónomos
3. Sistemas MultiAgente Robóticos
4. Sensores e Actuadores
5. Percepção, Mobilidade e Navegação
6. Coordenação e Interacção
7. Simulação

Resultados da Aprendizagem

1. Identificar, descrever e definir os principais conceitos relacionados com Sistemas Autónomos e Agentes Robóticos.
2. Realizá-los através da utilização do conceito de Agente.

3. Conhecer e saber utilizar o conhecimento proveniente de sensores.
4. Conhecer e saber utilizar os actuadores que potenciam a mobilidade e a navegação.
5. Estudar e desenvolver métodos de coordenação e interacção com o meio.
6. Utilizar a prototipação e a análise de complexidade para a sua aplicação na resolução de problemas.

PROJECTO INTEGRADO (PI)

Descrição

Este módulo é de índole exclusivamente prática, correndo em sessões de laboratório ou de acompanhamento de projectos, de forma integrada com os outros módulos da UCE. No início do ano, é definido um projecto integrador (possivelmente proposto no âmbito de uma articulação com a indústria ou serviços), que os alunos implementarão em sucessivos patamares de sofisticação, e/ou complexidade à medida que os seus conhecimentos aumentam. Tipicamente, o projecto começa pela especificação de um modelo formal abstracto do problema proposto, que é de imediato animado em laboratório e sujeito a testes de coerência (e.g., invariantes, pós-condições). Passa-se então à implementação, em termos da obtenção de código de acordo com uma arquitectura pré-definida (e.g., cliente-servidor), entretando acompanhada por processo de avaliação e teste sistemático. A interoperabilidade com outras tecnologias (e.g., "foreign functions", "data mappings") é também componente integrante desta formação, já que, por via de regra, os projectos propostos serão evoluções de outros já parcialmente implementados (e.g., vindos da indústria) ou que tenham sido subscritos, no mesmo ano lectivo, por outras UCEs.

Resultados de Apreendizagem

1. Planear, executar e avaliar projectos de modelação e desenvolvimento em Engenharia de Software.
2. Aplicar ferramentas conceptuais de modelação e cálculo ao projecto de software.
3. Integrar diferentes tecnologias e paradigmas computacionais.
4. Desenvolver capacidades de trabalho em equipa, comunicação e gestão de projectos de software.

Parcerias

1. APPIA - Associação Portuguesa Para a Inteligência Artificial
2. Hospital Geral de Santo António, EPE, Porto
3. Centro Hospitalar do Vale de Sousa, EPE, Porto

9

EC
Engenharia do Conhecimento

Sumário

O presente documento descreve uma Unidade Curricular de Especialidade (UCE), creditada com 30 ECTS, oferecida pelo Grupo de Inteligência Artificial do Departamento de Informática da Escola de Engenharia da Universidade do Minho, para o 2º ciclo da oferta formativa em Informática, na sequência da re-estruturação das licenciaturas no contexto do Processo de Bolonha.

Coordenação: *José Neves, José Machado, Victor Alves, António Abelha.*

9.1 Objectivos

A área de Engenharia de Conhecimento tem a ver com os sistemas baseados em conhecimento e suas aplicações. Aqui faz-se investigação fundamental de modelos de representação de conhecimento e formas de raciocínio, estabelecimento de métodos de comparação, tanto do ponto de vista formal como experimental, desenvolvimento de sistemas baseados em conhecimento e estudo das relações entre sistemas e o processo ensino/aprendizagem. Sendo-se mais específico, nesta área são abordados temas que vão dos Sistemas Cognitivos Construtivistas, i.e. desde o estudo de sistemas orientados ao conhecimento adoptando uma abordagem construtivista inspirada no trabalho de Piaget, aos Modelos Conexionistas, em que se trata da representação de conhecimento de forma distribuída. Pretende-se investigar o poder expressivo de tais modelos e os processos cognitivos que podem ser desenvolvidos, que leve em linha de conta as funções neuronais não lineares e a plasticidade neurónica observada nas estruturas biológicas. Pretende-se, por outro lado, tratar de tópicos avançados em áreas fundamentais de fronteira entre modelos de representação de conhecimento, formas de raciocínio e os sistemas simbólicos, com base na Programação em Lógica, sem descurar as áreas aplicadas que se têm revelado como centrais (e.g. na Medicina, no Direito, no Comércio Inteligente).

Com esta unidade pretende-se, por conseguinte, introduzir uma vasta gama de métodos e técnicas de resolução de problemas que têm vindo a ser não só utilizadas como forma de corporizar o que foi enunciado em epígrafe, mas também objecto de atenção em projectos de investigação, e que se referem a um certo domínio de aplicação, i.e. pretende-se estabelecer as competências básicas consideradas definidoras do perfil de Mestre em Informática da Universidade do Minho.

9.2 Caracterização Global

9.2.1 Regime e Escolaridade

A UCE apresentada neste documento poderá ser oferecida em regime semestral ou anual de acordo com o público a que se destine. O regime anual será preferencialmente adoptado para a formação inicial, durante a qual num mesmo ano um aluno poderá realizar duas UCE, enquanto o regime semestral, mais intensivo mas concentrado no tempo, parece ser mais adequado para a oferta em formação contínua ou quando exigido pela estrutura de Mestrados Erasmus-Mundos que venha a integrar. Atendendo à sua articulação em 4 módulos e um projecto integrador, conforme se detalha a seguir, a escolaridade proposta é a seguinte: 8T + 8TP + 8P (sendo as 8P correspondentes à componente de projecto integrador)

9.2.2 Áreas Científicas

Áreas Científicas: Informática (30 ECTS).

Áreas Científicas (classificação ACM): É a seguinte a distribuição dos 30 ECTS da UCE pelas áreas científicas classificadas no ACM Computing Curricula (1998):

I Computing Methodologies -- 10
I.2 Artificial Intelligence
I.2.3 Deduction and Theorem Proving
I.2.4 Knowledge Representation Formalisms and Methods
I.2.6 Learning
I.2.8 Problem Solving, Control Methods, and Search
I.2.11 Distributed Artificial Intelligence
H Information Systems -- 10
H.2 Database Management
H.2.4 Systems
H.2.5 Heterogeneous Databases
H.3 Information Storage and Retrieval
H.3.3 Information Search and Retrieval
H.3.5 Online Information Services
K Computing Milieux -- 10
K.3 Computers and Education
K.3.1 Computer Uses in Education

9.2.3 Requisitos

1. Experiência em programação, incluindo, preferencialmente, alguma familiaridade com a programação declarativa (e.g. programação em lógica).

2. Conhecimentos de lógica e matemática discreta a nível de primeiro ciclo.

9.2.4 Resultados de Aprendizagem

1. Distinguir entre os métodos computacionais, em geral, e as tecnologias baseadas em conhecimento, em particular.

2. Compreender não só as diferenças entre os potenciais sistemas de representação de conhecimento, mas também ser capaz de organizar os aspectos mais relevantes dos objectos e relações entre estes que ocorrem num dado universo de discurso, de acordo com o sistema de representação de conhecimento seleccionado.

3. Relacionar o sistema de representação de conhecimento com diferentes formas de raciocínio, bem como em optar entre os formalismos que melhor potenciem este relacionamento.

4. Compreender a relação entre causalidade e inferência.

5. Compreender a relação entre a complexidade de um modelo e o seu desempenho, utilizando esta informação na definição de uma estratégia para optimizar os sistemas existentes.

6. Desenvolver de forma integrada a função de concepção e projecto em Engenharia.

9.3 Estrutura Curricular

9.3.1 Estrutura Base

A presente UCE organiza-se em torno de 4 (quatro) módulos temáticos, correspondendo cada um a 5 ECTS, articulados entre si por um "bus" laboratorial, denominado *projecto integrado*, que garante a experimentação e aplicação prática dos resultados da aprendizagem. Esta estrutura é apresentada na Figura 1. Os módulos referidos são:

WD Web e Dados
SAI Sistemas Adaptativos e Ambientes de Aprendizagem Inteligentes
SMA Sistemas MultiAgente
DCN Design e Computação Natural

9.3.2 Métodos de Ensino

1. Exposição de conceitos e análise de casos de estudo.
2. Trabalho de grupo na resolução de exercícios e pequenos casos de estudo.
3. Trabalho de projecto em grupo sob a orientação da equipa docente afecta à UCE.

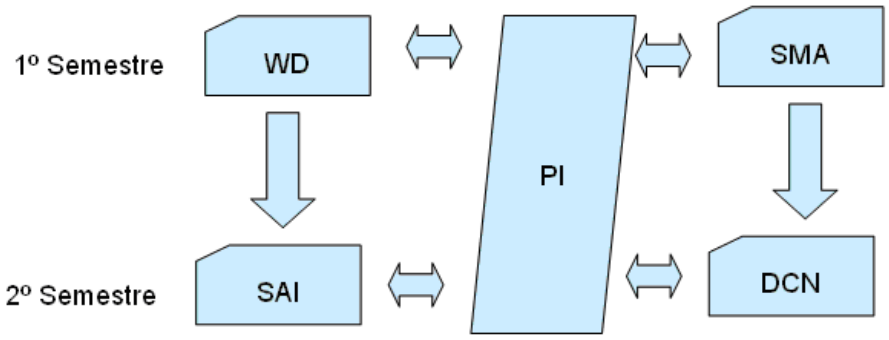

Figura 9.1: Estrutura Curricular da UCE30 - Engenharia do Conhecimento

9.3.3 Avaliação

O processo de avaliação visa aferir não apenas os conhecimentos e competências mínimas adquiridas, mas também a capacidade revelada na sua integração e aplicação no projecto de engenharia de software. Assim, é constituída por um exame final único em média pesada com a avaliação contínua em prática laboratorial e o resultado (ferramentas, relatório, apresentação) do projecto integrador.

9.3.4 Conteúdos Programáticos

Os conteúdos programáticos desta UCE são agrupados em 4 (quatro) módulos temáticos e 1 (um) de projecto, conforme a seguir se detalha.

WEB E DADOS

Descrição

Existe hoje um largo leque de tecnologias de apoio à decisão por meio de descoberta (semi-)automática em grandes Bases de Dados (BDs) de "insights" para o funcionamento de organizações. Porém, o processo de descoberta de conhecimento em BDs permanece extremamente

laborioso devido à falta de integração dessas várias tecnologias, que apesar de se complementarem, foram desenvolvidas por comunidades diferentes (e.g. Bases de Dados x Inteligência Artificial, Academia x Indústria) e em diferentes linguagens. Neste módulo visa-se o desenvolvimento de um ambiente abrangente de descoberta de conhecimento em BDs para apoio à decisão, integrando de maneira transparente e eficiente componentes de "Data Warehousing", OLAP (processamento de consultas analíticas em Bases de Dados multidimensionais), Mineração de Dados, Sistemas de Classificação, Redes Neuronais Artificiais, Programação Genética e Evolucionária.

Programa resumido

1. Bases de Dados Inteligentes
2. Extracção e Integração de Informação
3. "Semantic web"
4. "Web Mining" e Pesquisa

Resultados de Aprendizagem

1. Compreender e dominar as técnicas de aprendizagem, "data mining", "text mining" e bases de dados por forma a extrair conhecimento que seja relevante para a resolução de problemas a partir de dados na Web.
2. Identificar situações que beneficiem da utilização deste conhecimento (e.g. "business intelligence").
3. Seleccionar as metodologias de resolução de problemas, assim como os paradigmas computacionais que melhor se lhes adequam, com aplicação em áreas como o Comércio Electrónico, e-CRM, Sistemas de Informação na Web.

SISTEMAS ADAPTATIVOS E AMBIENTES DE APRENDIZAGEM INTELIGENTES

Descrição

O desenvolvimento de ambientes virtuais de aprendizagem tem a sua origem na segunda metade da década de noventa, quando as primeiras ferramentas e os primeiros ambientes de ensino baseados na Web foram desenvolvidos e utilizados em cursos à distância. Estas ferramentas integram serviços de comunicações disponíveis na Internet - tais como o Correio Electrónico e Salas de Discussão (Chat), com mecanismos de gestão de cursos e sistemas de envio de arquivos (upload). Todos estes recursos teriam como elemento unificador as tecnologias utilizadas na Web, tais como HTML, CGI e Java. Por outro lado a consolidação destas ferramentas de apoio à aprendizagem à distância ocorrida nos últimos anos, deveu-se à democratização do processo de *ensino à distância* nos meios académicos e da *formação à distância*, esta de cunho empresarial

(vulgo *e-learning*). Por conseguinte, o nível de conhecimento das partes (agentes) deve ser levado em consideração, para que se possa ter a possibilidade de se trabalhar dentro da perspectiva de *livre no tempo*, no *ritmo*, e no *espaço*, que são características necessárias para se concretizar uma interação via Web. Isto sem descurar as capacidades cognitivas que caracterizam a necessidade de um espaço de avaliação e auto-avaliação do próprio agente sobre a sua capacidade de construção de conceitos, as quais devem também ser objecto de consideração, atendendo a que as necessidades individuais de aprendizagem são extremamente variáveis entre agentes. Com este módulo procurar-se-á ultrapassar os senãos que acabam de ser enunciados.

Programa resumido

1. Representação de Conhecimento
2. Resolução de Problemas
3. Planeamento e Geração de Planos
4. Raciocínio Probabilístico
5. Modelação Cognitiva

Resultados de Aprendizagem

1. Saber projectar interfaces homem-máquina que suportem de uma forma efectiva o processo de interação.
2. Saber como representar o conhecimento necessário de forma que o sistema identifique as tarefas a serem executadas e o utilizador.
3. Saber utilizar este conhecimento por forma a conciliar as intervenções por parte do sistema às necessidades do utilizador.

SISTEMAS MULTIAGENTE

Descrição

Os Sistemas MultiAgente (SMA) formam uma subárea da Inteligência Artificial Distribuída e concentram-se no estudo de agentes autónomos em ambientes distribuídos. O termo autónomo designa o facto de que os agentes têm uma existência própria, independente da existência de outros agentes. No entanto, os SMA disponibilizam a base para os Sistemas de Processamento Distribuído, na presunção da benevolência e na perseguição de objectivos comuns, com um intento centralizador na concepção de sistemas. A ideia central num SMA é que um comportamento global inteligente pode ser alcançado a partir do comportamento individual das partes.

Programa resumido

1. Descrição formal de um Sistema Multiagente
2. Sociedades de Agentes

3. Interacção, Comunicação, Colaboração e Cooperação entre Agentes
4. Aplicação dos Sistema Multiagente na resolução de problemas

Resultados de Aprendizagem

1. Identificar, descrever e definir os principais conceitos relacionados com Sistemas MultiAgente.
2. Identificar situações que beneficiem da utilização de sistemas baseados em Sistemas Multiagente.
3. Conhecer e ser capaz de utilizar as metodologias de resolução de problemas e os paradigmas computacionais associados à utilização de Sistemas Multiagente.

DESIGN E COMPUTAÇÃO NATURAL

Descrição

Vários trabalhos e aplicações combinando a teoria dos conjuntos e a computação evolutiva têm vindo a ser propostos. Em particular, um grande número de trabalhos explora o uso da programação genética e evolucionária no projecto e optimização de sistemas e hardware adaptativo. Por outro lado a escolha de alguns dos parâmetros associados ao algoritmo evolutivo têm um papel fundamental no desempenho da evolução na sua tarefa de procura de uma solução. Entretanto, esta escolha é uma tarefa complicada e a adaptação aparece como uma boa alternativa. O objectivo geral do módulo tem a ver com a modelação de sistemas de hardware adaptativos que se alicerçam na teoria de conjuntos através de algoritmos baseados em computação evolutiva.

Programa resumido

1. "Design" em engenharia e optimização
2. Autómatos celulares
3. Vida artificial, teoria dos conjuntos e computação evolutiva
4. Hardware e evolução
5. Evolução interactiva

Resultados de Aprendizagem

1. Demonstrar não só um bom domínio das diferentes técnicas de "design" inspiradas na compreensão de processos similares que ocorrem na natureza, mas também como utilizá-las na resolução de problemas.
2. Compreender e utilizar as vantagens e desvantagens que estas técnicas oferecem, face às de uso corrente.

PROJECTO INTEGRADO

Descrição

Este módulo é de índole exclusivamente prática, correndo em sessões de laboratório ou de acompanhamento de projectos, de forma integrada com os outros módulos da UCE. No início do ano, é definido um projecto integrador (possivelmente proposto no âmbito de uma articulação com a indústria ou serviços), que os alunos implementarão em sucessivos patamares de sofisticação, à medida que os seus conhecimentos aumentam. Tipicamente, o projecto começa pela especificação de um modelo formal abstracto do problema proposto, que é de imediato animado em laboratório e sujeito a testes de coerência (e.g. invariantes, pós-condições). Mais tarde, passa-se à implementação, em termos da obtenção de código de acordo com uma arquitectura pré-definida (e.g. cliente-servidor), entretando acompanhada por processo de avaliação e teste sistemático. A interoperabilidade com outras tecnologias (e.g. "foreign functions", "data mappings") é também componente integrante desta formação, já que, por via de regra, os projectos propostos serão evoluções de outros já parcialmente implementados (e.g. vindos da indústria) ou que tenham sido subscritos, no mesmo ano lectivo, por UCEs tecnologicamente distintas.

Resultados de Apreendizagem

1. Planear, executar e avaliar projectos de modelação e desenvolvimento em Engenharia de Software.
2. Aplicar ferramentas conceptuais de modelação e cálculo ao projecto de software.
3. Integrar diferentes tecnologias e paradigmas computacionais.
4. Desenvolver capacidades de trabalho em equipa, comunicação e gestão de projectos de software.

Parcerias

1. APPIA - Associação Portuguesa Para a Inteligência Artificial
2. Hospital Geral de Santo António, EPE, Porto
3. Centro Hospitalar do Vale de Sousa, EPE, Porto

10

PV
Programação e Verificação

Sumário

O presente documento descreve uma Unidade Curricular de Especialidade (30 ECTS), oferecida conjuntamente pelo grupo de Lógica e Métodos Formais do Departamento de Informática e pelo Departamento de Matemática, no 2º ciclo da oferta formativa em Informática da Universidade do Minho, no espírito do Processo de Bolonha.

Esta UCE30 atribui competências de base em três temas essenciais na Ciência da Computação: a Lógica e Semântica da Programação*, as* Linguagens de Programação*, e os* Algoritmos *e sua Complexidade. Ao nível prático este estudo permite o desenvolvimento de componentes de sistemas informáticos com ênfase na sua* eficiência *e na* verificação formal *das suas propriedades, utilizando um conjunto de ferramentas sofisticadas para assistir quer no processo de desenvolvimento quer no de verificação.*

Coordenação: *José Bacelar Almeida, José Barros, Maria João Frade, João Saraiva, Jorge Sousa Pinto*

Colaboração (DMAT): *Pedro Patrício, Luís Pinto, José Espírito Santo.*

10.1 Objectivos

No quadro actual, em que a segurança dos sistemas assume um papel determinante, torna-se fulcral a capacidade de produzir código *certificado*: o produto final de um projecto de software deixa de consistir exclusivamente em aplicações executáveis, sendo necessariamente acompanhado de certificados que asseguram que essas aplicações possuem propriedades desejadas, sejam elas *funcionais* ou de *segurança*. Por exemplo, pode ser imperioso que um programa cumpra uma determinada política que o impeça de aceder a recursos que lhe são vedados. A *Verificação Formal* é a actividade que pretende estabelecer que um sistema se comporta, efectivamente, de acordo com a sua *especificação*, ou que o seu comportamento apresenta certas propriedades, ou ainda que não apresenta defeitos.

Por outro lado, a emergência de novas áreas de aplicação como a bioinformática e a criptografia relança o papel fundamental do estudo dos algoritmos e das suas propriedades nas ciências da computação. A presente UCE responde precisamente a estes imperativos cruciais. Pretende-se estabelecer competências a dois níveis:

- Ao nível fundamental, pretende-se construír uma base sólida de conhecimentos, indispensáveis para a abordagem de alguns dos problemas mais sofisticados na área da Informática. Estes conhecimentos organizam-se em três áreas essenciais da Ciência da Computação: a *Lógica e Semântica* da Programação; os princípios das *Linguagens* de Programação; e o estudo dos *Algoritmos* e sua complexidade.

- Ao nível aplicado, esta UCE atribui competências na área da Verificação Formal de sistemas informáticos e na utilização de ferramentas de verificação (baseadas em modelos ou em inferência lógica), bem como na utilização de técnicas avançadas no domínio das linguagens de programação (nomeadamente as funcionais, que se prestam particularmente ao raciocínio sobre programas).

10.2 Caracterização Global

10.2.1 Regime e Escolaridade

A UCE apresentada neste documento poderá ser oferecida em regime semestral ou anual de acordo com o público a que se destine. O regime anual será preferencialmente adoptado para a formação inicial, durante a qual num mesmo ano um aluno poderá realizar duas UCE, enquanto o regime semestral, mais intensivo mas concentrado no tempo, parece ser mais adequado para oferta em formação contínua ou quando exigido pela estrutura de Mestrados Erasmus-Mundos que venha a integrar.

Atendendo à sua articulação em 3 módulos e um Seminário/Projecto , conforme se detalha a seguir, a escolaridade proposta para o funcionamento em regime semestral (concentrado) é a seguinte: 9T + 6TP + 3P/SE. Em regime anual, a escolaridade será 4.5T + 3TP + 1.5P/SE.

10.2.2 Áreas Científicas

Áreas Científicas: Informática (30 ECTS)

Áreas Científicas (classificação ACM): A distribuição dos 30 ECTS da UCE pelas áreas científicas classificadas no ACM *Computing Curricula* (ACMCCS98) é a seguinte.

- /Theory of Computation/MATHEMATICAL LOGIC AND FORMAL LANGUAGES/Mathematical Logic/ – 5 ECTS

- /Theory of Computation/COMPUTATION BY ABSTRACT DEVICES/ – 5 ECTS

- /Theory of Computation/LOGICS AND MEANINGS OF PROGRAMS/Semantics of Programming Languages/ +
  /Theory of Computation/LOGICS AND MEANINGS OF PROGRAMS/Specifying and Verifying and Reasoning about Programs/ – 5 ECTS
- /Software/SOFTWARE ENGINEERING/Software/Program Verification/ – 5 ECTS
- /Software/PROGRAMMING LANGUAGES/ +
  /Software/PROGRAMMING TECHNIQUES/Applicative (Functional) Programming/ – 5 ECTS
- /Theory of Computation/ANALYSIS OF ALGORITHMS AND PROBLEM COMPLEXITY/ – 5 ECTS

10.2.3 Requisitos

Obrigatórios:

- Conhecimentos básicos, ao nível de 1o. ciclo, sobre programação, estruturas de dados e algoritmos.
- Conhecimentos básicos, ao nível de 1o. ciclo, sobre linguagens de programação imperativas.
- Noções elementares de Lógica e Matemática Discreta.

Recomendados:

- Familiaridade com programação declarativa (funcional ou lógica) e/ou orientada aos objectos.
- Conhecimentos sobre Semântica das Linguagens de Programação.

10.2.4 Resultados de Aprendizagem

- Compreender os fundamentos da Teoria de Tipos, suas implicações nos sistemas de tipos de linguagens, e sua interpretação lógica; exprimir propriedades lógicas em sistemas de prova assistida, e conduzir demonstrações nesses sistemas.
- Conhecer as principais classes de complexidade e as relações elementares entre estas.
- Especificar, exprimir, e verificar a validade de propriedades (relativas à correcção ou outras) de sistemas de *software*, com recurso a ferramentas de prova assistida e verificadores de modelos.
- Aplicar técnicas avançadas de programação, nomeadamente funcional, para a resolução de problemas concretos.

- Reconhecer características dos problemas em áreas de aplicação diversas; utilizar algoritmos clássicos dessas áreas; analisar, comparar, e desenhar algoritmos.

- Desenvolver projectos integrados de desenvolvimento e verificação de *software*

10.3 Estrutura Curricular

10.3.1 Estrutura Base

A presente UCE organiza-se em torno de 3 módulos temáticos, articulados entre si por um *Seminário e Projecto Integrado* que garante, por um lado, complementar os temas abordados nos diversos módulos com palestras pontuais de professores e investigadores convidados; por outro lado permite a experimentação e aplicação prática dos resultados da aprendizagem no contexto de um projecto integrado. Os módulos referidos são

FP Fundamentos da Programação

VS Verificação de *Software*

PA Programação Avançada

10.3.2 Métodos de Ensino

- Exposição de conceitos e análise de casos de estudo em aulas teóricas.

- Resolução de exercícios, mini-projectos, e pequenos estudos de caso, possivelmente com recurso a ferramentas informáticas específicas.

- Trabalho de projecto em grupo com orientação directa da equipa docente afecta à UCE, integrador de matérias dos vários módulos da unidade.

10.3.3 Avaliação

A avaliação visa avaliar integralmente as competências adquiridas, e a capacidade de as colocar em prática de forma integrada. Será efectuada com base nas seguintes componentes:

- Avaliação contínua: trabalhos escritos e/ou exposições em aula efectuadas pelos alunos.

- Avaliação de projecto: aplicações, relatórios, exposição.

- Exame final (único).

10.3.4 Conteúdos Programáticos

Os conteúdos programáticos desta UCE são agrupados em três módulos temáticos e um Seminário e Projecto, conforme a seguir se detalha.

FUNDAMENTOS DA PROGRAMAÇÃO

Descrição.

Este módulo aborda tópicos fundamentais em duas vertentes das Ciências da Computação: a *Lógica da Programação* e a *Complexidade Algorítmica*.

A Teoria de Tipos não só encontra aplicação prática no desenvolvimento de sistemas de tipos para linguagens de programação, como se tem revelado uma ferramenta fundamental nos sistemas de prova assistida. Este módulo introduz a Teoria de Tipos, dando especial enfoque à sua ligação à lógica e aos sistemas de prova assistida. A ligação entre a teoria de tipos e a lógica é feita pelo *isomorfimo de Curry-Howard* que estabelece uma relação precisa entre a lógica intuicionista e o λ-calculus.

Os sistemas formais apresentados serão transpostos para o plano prático através da utilização do sistema Coq (http://coq.inria.fr/), um sistema de prova assistida baseado no Cálculo de Construções Indutivas, de ampla utilização quer nos meios académicos, quer já em certos meios industriais.

Um aspecto central no estudo dos algoritmos diz respeito à avaliação dos recursos necessários à sua execução. Em particular, interessa estudar o tempo e a memória requerida durante a execução, e como esses valores escalam à medida que se aumenta o tamanho dos dados de entrada. A Teoria da Complexidade é o ramo das ciências da computação que estuda esses aspectos, classificando os vários problemas computacionais com base no comportamento dos algoritmos que os resolvem.

Programa resumido.

- Lógica da Programação
    1. Lógica construtiva.
    2. Lambda-calculus com tipos simples.
    3. Princípio proposições-como-tipos.
    4. Tipos dependentes e Polimorfismo.
    5. Cálculo das Construções.
    6. Tipos indutivos.
    7. Introdução aos sistemas de demonstração assistida.
- Complexidade
    1. Classes P, NP, PSPACE, LOG e NLOG.

2. Reduções e problemas completos.
3. Classes de complexidade para computação paralela.
4. Classes de complexidade para computação probabilística.
5. Problemas demonstravelmente intratáveis.

Resultados de Aprendizagem Específicos:

- Compreender a relação existente entre lógicas construtivas e teorias de tipos.
- Explorar a expressividade de uma linguagem com tipos dependentes, polimórficos ou indutivos na elaboração de especificações.
- Compreender o paradigma da programação em Teoria de Tipos e a inerente garantia de correcção total.
- Utilizar sistemas de demonstração assistida para conduzir demonstrações formais simples.
- Conhecer as principais classes de complexidade e as relações elementares entre estas.
- Fazer reduções entre problemas e dar exemplos de problemas completos.
- Relacionar as classes de complexidade para computação paralela e computação probabilística com as classes P e NP, respectivamente.
- Dar exemplos de problemas demonstravelmente intratáveis.

VERIFICAÇÃO DE *Software*

Descrição.

A verificação de *software* recorre a um conjunto de ferramentas que permitem modelar e raciocinar sobre programas e sistemas, e propõe uma abordagem rigorosa ao problema de se garantir que um sistema se comporta de acordo com uma especificação, um conceito central em toda a área de Engenharia de Software.

A verificação de programas é um caso particular da verificação formal, em que se procura verificar directamente o *código* de um programa, e não um modelo abstracto. A verificação de programas recorre a um conjunto de formalismos (relacionados com a *semântica* das linguagens de programação, como por exemplo a Lógica de Hoare) e ferramentas que permitem raciocinar sobre programas.

Neste módulo são introduzidos os conceitos e resultados fundamentais na área da Verificação, sendo utilizada para este fim uma família de ferramentas fulcrais na área do desenvolvimento de *software*: os Verificadores de Modelos. Explora-se em seguida a utilização de sistemas de prova

assistida no sentido de se obter *certificados* de correcção e/ou segurança para a execução de programas.

Programa resumido.

1. Correcção de Software: especificação; correcção parcial e total.
2. Modelação de sistemas em autómatos.
3. Especificação de propriedades comportamentais em lógica temporal.
4. Técnicas de verificação de modelos.
5. Utilização de uma ferramenta de verificação de modelos.
6. Noção de correcção para programas funcionais e imperativos.
7. Verificação da correcção de programas funcionais: extracção do conteúdo computacional de uma prova de correcção; utilização de um programa na estruturação da prova de correcção.
8. Especificação de programas imperativos com base numa Semântica Axiomática.
9. Construção de algoritmos a partir de especificações: correcção implícita.
10. Técnicas de programação genéricas; fortalecimento de invariantes. Desenvolvimento de programas eficientes. Algoritmos sobre vectores. Estudo de casos práticos.
11. Utilização de *Sistemas de Prova Assistida* na Especificação, Verificação e Certificação de Programas.

Resultados de Aprendizagem Específicos:

- Modelar sistemas reactivos em autómatos.
- Exprimir propriedades comportamentais em lógica temporal e avaliar as sua validade com recurso a ferramentas de verificação de modelos.
- Especificar e desenvolver programas utilizando uma linguagem de comandos com guardas.
- Exprimir propriedades relativas à correcção de programas em sistemas de prova assistida.
- Verificar a correcção de programas funcionais e imperativos em sistemas de prova assistida.

PROGRAMAÇÃO AVANÇADA

Descrição.

O objectivo deste módulo é complementar conhecimentos básicos nas áreas das *Linguagens de Programação* e dos *Algoritmos*. O módulo será organizado em pequenas unidades correspondentes a diferentes áreas de aplicação, podendo estas ser modificadas em diferentes instâncias do módulo, que deverá contar com a participação de especialistas nas várias áreas, e ocasionalmente com a exposição de casos de estudo por parte dos alunos.

Serão estudados aspectos teóricos e práticos do desenvolvimento e utilização de uma linguagem de programação. Desenvolvimentos recentes na programação funcional que melhoram o desempenho dos programadores serão apresentados e estudados, tais como, sistemas de tipos e módulos, programação genérica, e linguagens de domínio especifico embebidas. Será dado ênfase à resolução de problemas de programação através da reutilização de bibliotecas de software.

O estudo dos algoritmos será efectuado ilustrando a sua utilização em situações úteis e aplicações de interesse real. Um domínio particularmente importante de aplicação de técnicas e algoritmos oriundos da álgebra e de outras áreas da matemática discreta é a teoria de códigos, que estuda o desenho de códigos de transmissão com habilidade de detecção e correcção de erros. A utilização destes códigos é um ingrediente fundamental no estabelecimento de canais de comunicação fiáveis em meios com ruído.

Programa resumido.

- Linguagens de Programação
  1. Programação com *monads* e estado
  2. Sistemas de tipos avançados e sua utilização para o suporte de características avançadas das linguagens de programação; programação genérica
  3. *Debugging* e teste de programas funcionais
  4. Concepção e implementação de linguagens de domínio específico (DSLs)
- Algoritmos
  1. Análise probabilística de algoritmos; algoritmos probabilísticos
  2. Estratégias algorítmicas: algoritmos *greedy*; programação dinâmica. Algoritmos aproximados
  3. Apresentação de problemas nas seguintes áreas de aplicação, e estudo de algoritmos para a sua resolução: Compressão; Triangulação; Concordância de Padrões e de Sequências (com aplicação em Biologia Computacional); Indexação e Pesquisa; Criptografia e Segurança.
  4. Introdução ao Problema Principal da Teoria de Códigos.

5. Limites nos códigos.
6. Códigos lineares; códigos cíclicos; códigos de Hadamard e Reed-Muller; códigos de convolução.

Resultados de Aprendizagem Específicos:

- Reconhecer características dos problemas nas áreas de aplicação estudadas, e utilizar algoritmos clássicos úteis nessas áreas
- Analisar, comparar, e desenhar novos algoritmos
- Desenvolver projectos de *software* utilizando extensões, bibliotecas, e *toolkits* de uma linguagem de programação para a resolução de problemas concretos
- Descrever, produzir, parametrizar e aplicar diversos tipos de códigos e algoritmos relacionados.

SEMINÁRIO E PROJECTO

Descrição.

Este módulo é de índole predominantemente prático, decorrendo em sessões de laboratório ou de acompanhamento de projectos, de forma integrada com os outros módulos da UCE. No início do período lectivo é definido um projecto que os alunos implementam progressivamente, à medida que adquirem conhecimentos nos outros módulos.

O projecto poderá ter duas componentes: numa primeira procurar-se-á desenvolver um projecto de *software* aplicando as competências específicas sobe linguagens de programação e algoritmos adquiridas nesta UCE.

A segunda componente será de verificação formal, do sistema desenvolvido na primeira componente, ou alternativamente de um sistema externo considerado interessante (por exemplo recolhido na indústria, desenvolvido numa outra UCE30 frequentada pelo estudante, ou ainda vindo de um projecto de investigação activo nesta área).

Esta componente passará sempre pelas fases seguintes:

- desenvolvimento de um modelo para o sistema;
- especificação das propriedades funcionais ou de segurança que se pretende verificar, e sua expressão sobre o modelo elaborado;
- verificação das propriedades sobre o modelo.

Alternativamente, o projecto poderá ser mais conceptual (orientado para a investigação de base), e abordar temas como os aspectos teóricos e práticos dos próprios modelos, plataformas e ferramentas de verificação.

O módulo incluirá também uma componente de seminário, com

- exposição periódica por parte dos alunos, versando temas escolhidos que poderão ser alvo de estudo autónomo, ou com a apresentação de resultados intermédios do projecto desenvolvido;
- um conjunto de palestras convidadas sobre temas de interesse no contexto da UCE, como por exemplo as Linguagens de Programação Orientadas aos Objectos; as máquinas abstractas e seus *bytecodes*, etc.

Resultados de Aprendizagem Específicos:

- Estudar e investigar de forma autónoma artigos científicos.
- Organizar e apresentar comunicações científicas.
- Seleccionar, aplicar, e combinar as diferentes técnicas e abordagens estudadas na unidade curricular.
- Trabalhar em equipa; gerir e executar projectos de programação e verificação formal.

11

CSSI
Criptografia e Segurança de Sistemas de Informação

Sumário

O presente documento descreve uma UCE30 (=Unidade Curricular de Especialidade), creditada com 30 ECTS, oferecida pelos grupos de Lógica e Métodos Formais e de Sistemas Distribuídos do Departamento de Informática no 2^o ciclo da reestruturação curricular da oferta formativa em Informática da Universidade do Minho, no espírito do Processo de Bolonha.

Esta UCE30 resulta de mais de 10 anos de experiência no ensino e investigação fundamental e aplicada nas áreas da criptografia, segurança de sistemas infor-máticos, segurança da informação, e sistemas confiáveis. Pretende responder aos desafios que nos dias de hoje decorrem não só da ubiquidade do suporte informático na sociedade, mas também da globalização da prestação de serviços em informática.

O sucesso deste modelo de formação depende antes mais da receptividade por parte do seu principal mercado alvo a indústria. Espera-se que esta indústria, em particular a de âmbito local e regional, se associe à academia e assim feche o ciclo entre produção e absorção de recursos humanos de qualidade, promovendo o Minho como região de conhecimento e exigência.

Coordenação: *José Manuel Valença, Manuel Bernardo Barbosa, José Bacelar Almeida, Victor Francisco Fonte.*

11.1 Objectivos

Este módulo tem como alvo a *segurança da informação* e *confiabilidade dos sistemas informáticos*.[1] As tecnologias de suporte são a criptografia e a implementação e administração de sistemas informáticos distribuídos e confiáveis. O contexto integrador é dado pela normas e legislação nacionais e internacionais para a segurança da informação.

11.2 Caracterização Global

11.2.1 Regime e Escolaridade

A UCE apresentada neste documento poderá ser oferecida em regime semestral ou anual de acordo com o público a que se destine. O regime anual será preferencialmente adoptado para a formacção inicial, durante a qual num mes-mo ano um aluno poderá realizar duas UCE, enquanto o regime semestral, mais intensivo mas concentrado no tempo, parece ser mais adequado para oferta em formação contínua quando exigido pela estrutura de Mestrados Eras-mus-Mundos que venha a integrar. Atendendo à sua articulação em 5 módulos e um projecto integrador, conforme se detalha a seguir, a escolaridade proposta é a seguinte: 8T + 8TP + 6P +2S (sendo as 6P correspondentes à componente de projecto integrador, e as 2S correspondentes à componente de seminário).

11.2.2 Áreas Científicas

Áreas Científicas: Informática (30 ECTS)

Áreas Científicas (classificação ACM): [2]

- ANALYSIS OF ALGORITHMS AND PROBLEM COMPLEXITY
- RELIABILITY AND FAULT-TOLERANCE
- DATA ENCRYPTION
- SECURITY AND PROTECTION

11.2.3 Requisitos

Conhecimentos elementares de matemática e programação exigíveis a qualquer aluno com formação de primeiro ciclo na área das TICs.

[1] No sentido INFOSEC e COMPSEC (exemplos em cordis.europa.eu/infosec/ e www.compseconline.com/).

[2] A classificação ACM de 1998 está desactualizada relativamente ao panorama actual desta área disciplinar. As áreas indicadas são as que, nesse documento, mais se aproximam dos temas abordados neste módulo.

11.2.4 Resultados de Aprendizagem

- Identificar os problemas da teoria de números mais relevantes à criptografia moderna, e discutir o conceito de *problema difícil* neste contexto.
- Explicar os objectivos fundamentais da criptografia moderna, reconhecer as primitivas criptográficas que lhes estão associadas e explicar o funcionamento interno das técnicas criptográficas mais relevantes.
- Gerir sistemas de certificação digital e PKI e utilizar aplicações correntes/comerciais da criptografia.
- Compreender e explorar o paradigma transaccional e a replicação por software no desenvolvimento de sistemas confiáveis.
- Conhecer e dominar as diversas vertentes da administração de sistemas informáticos como forma de assegurar a sua segurança e correcção.
- Conhecer, seleccionar e aplicar técnicas de desenvolvimento de aplicações seguras.

11.3 Estrutura Curricular

11.3.1 Estrutura Base

A presente UCE organiza-se em torno de 4 *módulos temáticos*, 2 *módulos complementares*, e um *projecto integrado*.

Módulos Temáticos

Designação	ECTS	Escolaridade
(CSI) Criptografia e Segurança da Informação	5	2T+1TP
(PSC) Paradigmas de Sistemas Confiáveis	5	2T+1TP
(TC) Técnicas Criptográficas	5	2T+1TP
(GSI) Gestão da Segurança da Informação	2	2S

Módulos Complementares

Designação	ECTS	Escolaridade
(TNC) Teoria dos Números Computational	5	2T+1TP
(SSI) Segurança de Sistemas Informáticos	5	2T+1TP

Componente Laboratorial

Designação	ECTS	Escolaridade
(PI) Projecto Integrado	8	3P

A escolaridade associada a esta UCE compreende os módulos temáticos, de frequência obrigatória, e um módulo complementar. Os módulos serão articulados entre si por uma componente laboratorial, denominada *projecto integrado*, que garante a experimentação e aplicação prática dos resultados da aprendizagem.

11.3.2 Métodos de Ensino

- Exposição de conceitos e análise de casos de estudo.
- Trabalho de grupo em exercícios e pequenos estudos de caso, em certos casos com recurso a ferramentas informáticas específicas.
- Trabalho de projecto em grupo com orientação directa da equipa docente afecta à UCE.

11.3.3 Avaliação

O processo de avaliação visa aferir não apenas os conhecimentos e competên-cias mínimas mas também a capacidade revelada na sua integração e aplicação no projecto de engenharia. Assim, é constituída por um exame final único em média pesada com a avaliação contínua laboratorial e o resultado (ferramentas, relatório, apresentação) do projecto integrador.

11.3.4 Contéudos Programáticos

MÓDULO CRIPTOGRAFIA E SEGURANÇA DA INFORMAÇÃO (CSI)

Programa resumido.

- Conceitos fundamentais da criptografia moderna.
- Introdução às primitivas, esquemas e protocolos criptográficos mais relevantes como blocos construtivos de funcionalidades de segurança em sistemas informáticos.
- Certificação digital e a utilização de Infrastruturas de Chave Pública.
- Aplicações correntes/comerciais da criptografia.

Resultados de Aprendizagem Específicos:

- Explicar os objectivos fundamentais da criptografia moderna (confidencialidade, integridade, autenticação e não repúdio) e reconhecer as primitivas criptográficas que lhes estão associadas.
- Efectuar análise de requisitos de segurança no âmbito do desenvolvimento / avaliação de sistemas informáticos.
- Seleccionar e integrar os protocolos criptográficos relevantes à segurança de sistemas concretos.

- Discutir a utilização de Certificados de Chave Pública e descrever os componentes centrais de uma Public Key Infrastructure.
- Manipular aplicações correntes/comerciais da criptografia.

MÓDULO PARADIGMAS DE SISTEMAS CONFIÁVEIS (PSC)

Programa resumido.

- Modelos de faltas
- Coordenação distribuída
- Paradigma transaccional
- Comunicação em grupo fiável
- Replicação por software

Resultados de Aprendizagem Específicos:

- Compreender as características e compromissos na modelação de faltas.
- Compreender e explorar as garantias oferecidas pelos sistemas transaccionais e de comunicação em grupo.
- Desenvolver aplicações tolerantes a faltas baseadas em replicação por software.

MÓDULO TÉCNICAS CRIPTOGRÁFICAS (TC)

Programa resumido.

- Noções fundamentais da criptografia: computabilidade, conhecimento e aleatoriedade, confiança e provas probabilísticas.
- Teoria matemática básica: estruturas algébricas relevantes e teoria dos números.
- Primitivas criptográficas: cifras e sua criptoanálise, sistemas de chave pública, sistemas orientados à identidade.
- Esquemas criptográficos: modelos e provas de segurança.

Resultados de Aprendizagem Específicos:

- Explicar o funcionamento interno das técnicas criptográficas mais relevantes.
- Discutir os modelos de segurança teóricos aceites para cada técnica criptográfica e explicar o conceito de *prova/redução de segurança*.
- Inferir o nível de segurança computacional expectável para uma técnica criptográfica com base numa redução de segurança dentro de um modelo adequado.

MÓDULO GESTÃO DA SEGURANÇA DA INFORMAÇÃO (GSI)

Programa resumido.

- Aspectos legais da segurança da informação.
- Privacidade e protecção de dados pessoais.
- Gestão da confiança.
- Auditoria e acreditação de segurança.

Resultados de Aprendizagem Específicos:

- Reconhecer as normas e legislação nacionais e internacionais na área da segurança que são relevantes numa determinada área aplicacional.
- Explicar os conceitos e procedimentos de acreditação e auditoria de segurança para pessoas, equipamentos e instalações.

MÓDULO TEORIA DE NÚMEROS COMPUTACIONAL (TNC)

Programa resumido.

- Aspectos da teoria dos números sobre os quais assentam as técnicas criptográficas modernas; teoria dos números elementares, congruências, estruturas algébricas finitas, teoria dos números algébricos.
- Problemas difíceis que servem como base ao estabelecimento de níveis mínimos de segurança e definição de tamanhos de chaves criptográficas: factorização, logaritmo discreto e outros.

- Curvas Elípticas e Hiperelípticas.

Resultados de Aprendizagem Específicos:

- Identificar os problemas da teoria de números mais relevantes à criptografia moderna.
- Manipular os conceitos matemáticos necessários à compreensão destes problemas.
- Explicar e implementar os algoritmos mais relevantes na área da teoria dos números computacional.
- Discutir o conceito de *problema difícil* no contexto da criptografia e teoria dos números computacional modernas.

MÓDULO SEGURANÇA DE SISTEMAS INFORMÁTICOS (SSI)

Programa resumido.

- Tópicos de programação segura com ênfase no dimensionamento de privilégios, protecção de dados, "stack" e "sandbox" de execução.
- Técnicas, ferramentas e boas práticas de administração na operação segura de sistemas informáticos.
- Perímetros de segurança como elementos estruturantes na no comparticionamento de acesso e protecção dos recursos do sistema.
- Redundância, "fail-over", salvaguarda e reposição de dados e serviços como técnicas para melhoria da disponibilidade e segurança de operação de sistemas informáticos.
- Análise forense aplicada à detecção e avaliação de situações de comprometimento de segurança de sistemas.

Resultados de Aprendizagem Específicos:

- Reconhecer as técnicas de programação responsáveis pelas mais frequentes situações de vulnerabilidade de segurança de sistemas informáticos e aplicar boas práticas de desenvolvimento conducentes a uma maior segurança.
- Reconhecer as principais competências e boas práticas necessárias a uma eficaz administração de sistemas no domínio da segurança.
- Identificar e estimar o risco associado a potenciais vulnerabilidades das várias componentes de um sistema informático, bem como aplicar medidas correctivas ou mitigadoras.

- Definir e implementar perímetros de segurança adequados ao risco associado às várias componentes do sistema.
- Desenvolver procedimentos conducentes a uma maior disponibilidade de dados e serviços.
- Aplicar técnicas de análise forense a situações de comprometimento da segurança de um sistema informático.

12

SD
Sistemas Distribuídos

Sumário

O presente documento descreve uma UCE30 (="Unidade Curricular de Especialidade"), creditada com 30 ECTS, oferecida pelo Grupo de Sistemas Distribuídos do Departamento de Informática no 2º ciclo da re-estruturação curricular da oferta formativa em Informática da Universidade do Minho, no espírito do Processo de Bolonha.

Esta UCE30 resulta de mais de uma década de experiência no ensino de sistemas distribuídos e numa sequência consolidada de projectos de investigação com financiamento nacional e europeu na área. Tem como alvo uma formação em sistemas distribuídos clássicos, com ênfase nas tecnologias orientadas por objectos, sistemas transaccionais, e tolerância a faltas, mas aborda também áreas emergentes como sistemas móveis, peer-to-peer *e* grid*. O perfil de formação proposto alia uma sólida componente fundamental em modelos e algoritmos a uma componente tecnológica que segue de perto as necessidades da indústria. Esta dualidade permite uma profunda compreensão e relacionamento dos problemas e soluções.*

Este modelo de formação tem como mercado alvo a indústria de software *e serviços, não só na região Minho, mas a nível nacional e internacional. Espera-se então que a crescente importância da fiabilidade, disponibilidade e escala atingíveis apenas com sistemas distribuídos se traduza numa procura crescente de competências nesta área.*

Duas outras UCE em Engenharia de Aplicações e em Segurança de Informação, permitem complementar a sólida formação em Sistemas Distribuídos com um perfil directamente orientado à industria de software *e serviços, bem como aprofundar os aspectos relacionados com segurança e confiabilidade.*

Coordenação: *Rui Oliveira, José Orlando Pereira, Carlos Baquero Moreno, Paulo Sérgio Almeida.*

12.1 Contexto e Objectivos

Num Sistema Distribuído, um conjunto de sistemas informáticos trabalha interligado por uma rede de comunicação de dados obtendo vantagens em termos de acesso a recursos remotos e coordenação de tarefas, mas também na disponibilidade e robustez em aplicações críticas. A área de Sistemas Distribuídos assume assim uma tripla importância. Em primeiro lugar, pela necessidade de construir sistemas que acompanham a sempre crescente distribuição geográfica inerente às tarefas a realizar por sistemas informáticos. Por outro lado, pela possibilidade oferecida de aumentar a confiabilidade de sistemas de forma económica. Finalmente, pela actual adopção de técnicas originárias de sistemas distribuídos na estruturação de aplicações informáticas em geral, frequentemente ocultas sob a forma de *middleware*.

São numerosos os exemplos de novas aplicações e mesmo completas áreas de negócio possibilitadas pelos desenvolvimentos em Sistemas Distribuídos. Um exemplo é o motor de busca *Google*, que tirando partido de um grande conjunto de servidores obtém uma capacidade de armazenamento e processamento em grande escala. Outro exemplo é a plataforma tecnológica da *Amazon.com*, que além de suportar a operação da livraria à escala planetária, é disponibilizada como serviço a terceiros. As aplicações *peer-to-peer* de partilha de ficheiros são outro exemplo, de como uma aplicação distribuída pode ter um profundo impacto em áreas de negócio e no quotidiano. Finalmente, as vantagens de disponibilidade e desempenho reconhecidas à estruturação de aplicações em camadas, em particular da plataforma J2EE, devem-se em grande parte à aplicação de técnicas de Sistemas Distribuídos.

O objectivo desta Unidade Curricular é pois proporcionar uma formação especializada e avançada, nas vertentes teórica, tecnológica e experimental. Na sua vertente teórica, são sistematizados os principais modelos e problemas fundamentais em Sistemas Distribuídos, bem como a sua resolução algorítmica e a sensibilização para aspectos de correcção, escala, fiabilidade e desempenho. Na sua vertente tecnológica, apresenta uma componente importante de estudo de casos de tecnologias da actualidade, representativas de cada uma dos principais paradigmas mas com ênfase na sua aplicabilidade imediata. Na sua componente experimental, procura-se a resolução de problemas complexos relacionando múltiplos conceitos e tecnologias.

Esta Unidade Curricular aproveita a experiência de mais de uma década do Grupo de Sistemas Distribuídos (GSD) do Departamento de Informática no ensino de Sistemas Distribuídos em várias disciplinas dos 4º e 5º ano da Lic. Sistemas e Informática, bem como nos cursos de Mestrado em Informática e de Mestrado em Sistemas Móveis na Universidade do Minho. Fazendo parte de um curso de 2º Ciclo, esta Unidade Curricular encontra-se também enraizada em trabalho de investigação desenvolvido numa sequência consolidada de projectos com financiamento nacional e europeu, com ênfase na tolerância à faltas, bases de dados distribuídas e sistemas móveis.

12.2 Caracterização Global

12.2.1 Regime e Escolaridade

A UCE apresentada neste documento poderá ser oferecida em regime semestral ou anual de acordo com o público a que se destine. O regime anual será preferencialmente adoptado para a formação inicial, durante a qual num mesmo ano um aluno poderá realizar duas UCE. Atendendo ao desenvolvimento em paralelo de dois temas programáticos e ao desenvolvimento da componente de projecto considera-se uma escolaridade semanal com duas sessões de duas horas teóricas seguidas de uma hora de carácter prático laboratorial, sendo o acompanhamento presencial de projecto feito em quatro horas de apoio tutorial. A escolaridade ao longo de um ano conduz a uma carga presencial semanal de 4T + 2PL + 4OT.

12.2.2 Áreas Científicas

Áreas Científicas: Informática (30 ECTS)

Áreas Científicas (classificação ACM):

- C. Computer Systems Organization / C.2 COMPUTER-COMMUNICATION NETWORKS / C.2.4 Distributed Systems
- C. Computer Systems Organization / C.4 PERFORMANCE OF SYSTEMS
- D. Software / D.1 PROGRAMMING TECHNIQUES / D.1.3 Concurrent Programming
- D. Software / D.1 PROGRAMMING TECHNIQUES / D.1.5 Object-oriented Programming
- D. Software / D.2 SOFTWARE ENGINEERING / D.2.12 Interoperability
- D. Software / D.4 OPERATING SYSTEMS
- H. Information Systems / H.2 DATABASE MANAGEMENT / H.2.4 Systems

12.2.3 Requisitos

Conhecimentos elementares de programação concorrente e distribuída exigíveis a qualquer aluno com formação de primeiro ciclo na área das TICs.

12.2.4 Resultados de Aprendizagem

- Saber analisar problemas na área dos SD compreendendo os pressupostos e compromissos do modelo e solução pretendidos.

- Saber identificar as potencialidades e limitações da infra-estrutura de computação e comunicação disponível por forma a seleccionar a gama de modelos aplicáveis.
- Conhecer e explorar algoritmos fundamentais e diversos paradigmas para o desenvolvimento de sistemas e aplicações confiáveis, eficientes e escaláveis.
- Explorar os mecanismos transaccionais e técnicas de replicação por *software* para a construção de sistemas e aplicações fiáveis e de elevada disponibilidade.
- Conceber soluções distribuídas capazes de operar em ambientes móveis e/ou de larga escala, com eventual grande variabilidade dos recursos disponíveis.
- Conceber, desenvolver e validar aplicações sobre middleware de objectos distribuídos.
- Desenvolver de forma integrada a função de concepção e projecto em Engenharia.

12.3 Estrutura Curricular

12.3.1 Estrutura Base

A presente UCE organiza-se em torno de 6 temas, correspondendo a 20 ECTS, articulados entre si por um projecto laboratorial integrado, correspondendo a 10 ECTS, que garante a experimentação e aplicação prática dos resultados da aprendizagem. Os temas referidos são:

FSD Fundamentos de Sistemas Distribuídos

OD Objectos Distribuídos

ST Sistemas Distribuídos Transaccionais

SM Sistemas Móveis

TF Tolerância a Faltas

SGE Sistemas em Grande Escala

12.3.2 Métodos de Ensino

A organização da Unidade prevê em cada momento o decorrer em paralelo de dois temas programáticos (abordados em aulas teóricas e laboratoriais) e do projecto integrado. Para cada tema, haverá uma exposição nas aulas teóricas, que envolverá também o estudo de problemas de pequena dimensão. A resolução de alguns destes problemas será efectuada nas aulas laboratoriais, utilizando tecnologias concretas (*middleware*) para a sua implementação. A aplicação dos conhecimentos a problemas de maiores dimensões será deixada para o laboratório integrado onde o aluno terá oportunidade de desenvolver aplicações relativamente realistas envolvendo uma interdisciplinaridade relativamente aos vários temas abordados na Unidade Curricular.

12.3.3 Avaliação

A avaliação da Unidade Curricular tem em conta duas componentes. A primeira é um exame final, contendo perguntas de desenvolvimento e exercícios envolvendo a resolução de pequenos problemas relativamente abstractos. O exame pretende avaliar sobretudo os conhecimentos teóricos. A segunda componente tem em conta o trabalho desenvolvido pelos alunos no projecto integrado, com o fim de avaliar a aplicação do conhecimento à resolução de problemas reais, a capacidade de identificação e relacionamento dos diferentes conhecimentos necessários, e a capacidade de desenvolver aplicações utilizando tecnologias, nomeadamente *middleware*, relativamente sofisticadas.

12.3.4 Conteúdos Programáticos

FUNDAMENTOS DE SISTEMAS DISTRIBUÍDOS (FSD)

Descrição.
É apropriado modelar um sistema distribuído, seja ele confinado a um sistema local restrito ou se expanda à escala da Internet, como um conjunto de entidades de processamento autónomas interligadas por canais de comunicação para troca de mensagens. Sobre esta base, complementada com pressupostos de refinamento aos sistemas particulares considerados, constam da literatura da área um conjunto de algoritmos que, pelos problemas abordados, pelas técnicas empregues ou ainda pelos resultados limite que identificam, são blocos fundamentais no estudo e programação de sistemas distribuídos.

Nesta Unidade Curricular são apresentados os modelos e pressupostos dos sistemas distribuídos que permitem estudar um conjunto de algoritmos distribuídos clássicos e simultaneamente configuram o modelo que serve de base aos refinamento das Unidades Curriculares que se seguem.

Programa resumido.

- Modelos: redes síncronas e assíncronas; computação por eventos; causalidade; tempo lógico; cortes e predicados globais.
- Medidas de complexidade: tempo e comunicação.
- Algoritmos em redes síncronas: eleição, árvore de cobertura com custo mínimo.
- Algoritmos em redes assíncronas: eleição, detecção de terminação, determinação de instantâneos globais, alocação de recursos.

OBJECTOS DISTRIBUÍDOS (OD)

Descrição.

A programação por objectos, um paradigma de programação com grande sucesso, estendeu-se ao uso na programação de sistemas distribuídos. Os sistemas de objectos distribuídos constituem um dos paradigmas importantes usados na concepção de aplicações distribuídas.

Neste paradigma são baseados instâncias importantes de *middleware* como Java RMI e CORBA. Estes podem ser usados directamente na construção de aplicações mas essencialmente como infra-estrutura em servidores aplicacionais (e.g. J2EE), que oferecem ao programador modelos de componentes. O aparecimento de servidores aplicacionais de código aberto, como JBoss, e a correspondente necessidade de manutenção e extensão tornam importante o domínio deste paradigma.

Nesta Unidade Curricular, o tema de Objectos Distribuídos é abordado não só com uma componente teórica mas também com um estudo de caso bastante detalhado do *middleware* mais relevante. Como resultado, o aluno deverá ser capaz de desenvolver aplicações distribuídas segundo o paradigma de objectos distribuídos, utilizando *middleware* apropriado como CORBA.

Programa resumido.

- Programação concorrente com objectos: suporte para concorrência e distribuição; objectos activos e passivos; controlo de concorrência intra-objecto e inter-objecto; exploração da imutabilidade; programação concorrente em Java.
- Sistemas de objectos distribuídos: efeitos da distribuição na identidade, estado, passagem de parâmetros, reciclagem de objectos, despacho de invocações e controlo de concorrência; mobilidade de código.
- Estudo de caso – CORBA: arquitectura CORBA; o IDL do OMG; interface do ORB; o *Portable Object Adapter*; políticas de associação entre OIDs e *servants* e de activação de objectos; interfaces dinâmicas (DII e DSI); repositório de interfaces e implementações; padrões de desenho em CORBA; serviços CORBA.

SISTEMAS DISTRIBUÍDOS TRANSACCIONAIS (SDT)

Descrição.

A utilização de mecanismos transaccionais permite melhorar a robustez, desempenho e modularidade de sistemas distribuídos. Em particular, a garantia de atomicidade de operações favorece a recuperação rápida de falhas. A coordenação de diferentes componentes através de mecanismos transaccionais simplifica também a construção de aplicações distribuídas complexas e a optimização do seu desempenho.

Os mecanismos transaccionais encontram-se na prática no *middleware* de acordo com a norma *JMS* de filas de mensagens e publicação/subscrição e as a norma *XA/Transactions* de composição de sistemas transaccionais, em particular, nos servidores aplicacionais *J2EE*.

Este tema é abordado nesta Unidade Curricular de forma a garantir que o aluno é compreende e é capaz de explicar os pressupostos, mecanismos e vantagens da aplicações do modelo transaccional em sistemas distribuídos. Este conhecimento é aplicado em tecnologias concretas de *middleware* transaccional no desenvolvimento de aplicações distribuídas.

Programa resumido.

- Modelo transaccional em sistemas distribuídos: pressupostos; propriedades ACID; arquitectura de monitores transaccionais.
- Invocações remotas transaccionais: gestores de recursos; contexto transaccional.
- Compromisso atómico: algoritmos 2-PC e 3-PC; concretização na norma *XA/Transactions*.
- Filas de mensagens transaccionais: implementação; aplicações de *workflow* distribuído.
- Publicação/subscrição transaccional: modelos de subscrição; composição de serviços.

SISTEMAS MÓVEIS (SM)

Descrição.

O uso de dispositivos e meios de comunicação móveis como base de construção de soluções distribuídas exige o uso de mecanismos específicos por forma a ultrapassar as limitações presentes num meio com mobilidade. Estes mecanismos permitem, nomeadamente, uma adequação a canais de comunicação com alta latência e a cenários de conectividade intermitente onde se permite a evolução de um estado global com base em operação optimista.

Estes mecanismos encontram-se na prática associados a soluções de Web móvel onde se optimiza o uso da ligação disponível e da memória local ao dispositivo; no suporte à replicação de sistemas de ficheiros e na sincronização de unidades de armazenamento móveis; no desenho de soluções com acesso móvel a bases de dados.

O tratamento deste tema na Unidade Curricular permite garantir que os alunos compreendem as limitações inerentes aos ambientes móveis, sendo capazes de desenhar soluções eficazes e correctas nestes ambientes. Devem ser capazes de ajustar as soluções a ambientes de conectividade fraca (ex: canais GPRS/UMTS) ou a conectividade ocasional com uma infra-estrutura fixa ou entre unidades móveis.

Programa resumido.

- Modelos de mobilidade: condicionalismos na mobilidade, escala física, conectividade e energia.

- Transferência de estado: fontes únicas de escrita; sincronização unidireccional (rsync, xdelta); simplificação de dados; *caching* e *prefetch*.
- Controlo de divergência: modelação de ordenação; vectores versão e outras representações de histórias causais.
- Reconciliação: preservação de intenções (unison); garantias de convergência; fusão de *logs*.
- Replicação optimista: limitação de divergência; reservas; propagação de operações vs. de estado; transformação de operações.

TOLERÂNCIA A FALTAS (TF)

Descrição.

A capacidade de assegurar a continuidade da correcta operação dos sistemas informáticos apesar da ocorrência de faltas é vital para um número crescente de actividades comerciais, sociais e de administração pública. A tolerância a faltas é geralmente obtida através da replicação de componentes críticos sendo apelativo, por questões económicas ou de distribuição geográfica, efectuar a sua coordenação por *software*. A redundância no sistema permite ainda que, na ausência de faltas e apesar da complexidade acrescida, o desempenho do sistema possa ser beneficiado pelo aumento do nível de concorrência.

A replicação por *software* é instrumental no desenvolvimento de aplicações confiáveis baseadas em objectos distribuídos (FT-CORBA) ou em sistemas de gestão de bases de dados (MySQL Cluster, Oracle RAC, Continuent Uni/Cluster). Na construção de sistemas resistentes a situações catastróficas representa um solução de flexibilidade e competitividade ímpar.

Nesta Unidade Curricular são ensinados os conceitos e entraves à confiabilidade das aplicações, abordados problemas de coordenação distribuída na presença de faltas e são analisadas com detalhes as técnicas e ferramentas preponderantes na construção de sistemas e aplicações tolerantes a faltas. O conhecimento transmitido é sedimentado com o desenvolvimento de aplicações replicadas.

Programa resumido.

- Confiabilidade: conceitos e pressupostos.
- Modelos: taxonomia de faltas; sincronia parcial; detecção de faltas.
- Coordenação distribuída: acordo e eleição.
- Comunicação fiável em grupos: difusão fiável e ordenada, serviços de filiação, sincronia virtual.

- Replicação por *software*: replicação de serviços e bases de dados; recuperação em situação de desastre.

SISTEMAS EM GRANDE ESCALA (SGE)

Descrição.

A investigação científica em Sistemas Distribuídos tem dado uma importância crescente ao tratamento de sistemas em grande escala. Consideram-se assim um elevado número de sistemas interligados, a utilização de redes geograficamente distribuídas e o tratamento de elevados volumes de informação e de operações sobre essa informação.

Para além da sua relevância científica, estes temas assumem relevância prática e têm impacto directo no quotidiano de grande número de utilizadores, pela sua aplicação a sistemas *peer-to-peer*, redes de sensores autónomos e sistemas de computação em *grid*, entre outros.

Este tema é abordado nesta Unidade Curricular de forma a garantir que o aluno conhece os modelos principais da computação em grande escala e o seu impacto na solubilidade de um conjunto de problemas. Para cada uma deles são discutidas e avaliadas diversas soluções, com ênfase na discussão dos compromissos entre desempenho e escala.

Programa resumido.

- Modelos: *peer-to-peer*; redes de sensores; *grid*.
- Indexação e pesquisa: abordagens não-estruturadas; tabelas de *hash* distribuídas (DHT).
- Publicação/subscrição: subscrição e encaminhamento baseado em conteúdo; difusão epidémica.
- Replicação de dados: replicação massiva; colocação e aprovisionamento de réplicas em *grid*.
- Agregação: detecção de predicados globais; aplicações a redes de sensores.

13

EA
Engenharia de Aplicações

Sumário

O presente documento descreve uma UCE30 (="Unidade Curricular de Especialidade"), creditada com 30 ECTS, oferecida conjuntamente pelos Grupos de Sistemas Distribuídos (GSD) e de Sistemas Interactivos e Multimédia (SIM) do Departamento de Informática no 2º ciclo da re-estruturação curricular da oferta formativa em Informática da Universidade do Minho, no espírito do Processo de Bolonha.

Esta UCE30 vai de encontro a uma alteração significativa da prestação de serviços informáticos em torno de sistemas multi-tier*, onde as aplicações genéricas são substituidas pela integração e parametrização de componentes em torno de* application servers*, e centros de dados, frequentemente em regime de* outsourcing*, que substituem colecções de sistemas* ad hoc*. Responde a este desafio com a formação no perfil de Engenharia da Aplicações, que combina um conhecimento dos mecanismos e compromissos fundamentais a todos os niveis do sistema.*

Este modelo de formação tem como mercado alvo a indústria de software *e serviços, não só na região Minho, mas a nível nacional e internacional. Espera-se então que a crescente adopção dos sistemas* multi-tier *se traduza numa forte aceitação no mercado de trabalho.*

A UCE proposta e complementa por duas outras em Sistemas Distribuídos e em Segurança de Informação que permitem aprofundar a teoria e tecnologia subjacentes a sistemas multi-tier *bem como os aspectos relacionados com segurança e confiabilidade.*

Coordenação: *Francisco Soares de Moura, António Luís Sousa, António Nestor Ribeiro, José Creissac Campos.*

13.1 Contexto e Objectivos

Tem-se assistido a uma alteração na arquitectura e infraestrutura das aplicações informáticas, com uma popularidade crescente do alojamento conjunto de múltiplas aplicações *multi-tier* em

centros de dados com administração profissional. Em vez de propostas comerciais verticalmente integradas, incluindo *hardware* dedicado e licenças para pacotes de *software* genérico, é cada vez mais comum a oferta de *software* resultante da integração de múltiplos componentes em redor de um *application server*. Diferentes aplicações convivem então num único centro de dados integrado, dentro da própria organização ou, cada vez mais, em regime de *outsourcing* no próprio prestador de serviços. Este movimento tem um reflexo claro na estutura e objectivos das empresas prestadores de serviços informáticos, sendo claro também a nível nacional e em particular na região Minho.

Neste contexto, surge a Engenharia de Aplicações como perfil de relevância crescente tanto nas instituições prestadoras como nas consumidoras de serviços informáticos. Por um lado, o Engenheiro de Aplicações deve ser capaz de participar no processo de desenvolvimento de aplicações *multi-camada* a partir de componentes, com ênfase na parametrização da lógica de negócio e da interface do utilizador. Por outro lado, deve ser capaz de participar no planeamento, instalação e operação da infraestrutura de centros de dados em grande escala. Na integração com sucesso e optimização da fiabilidade e desempenho do sistema como um todo, o Engenheiro de Aplicações distingue-se pela sua compreensão integrada dos compromissos fundamentais e das tecnologias disponíveis nas várias camadas.

Esta Unidade Curricular aproveita a experiência existente no Departamento de Informática no ensino de administração de sistemas, programação orientada por objectos e sistemas interactivos nos cursos de Lic. Sistemas e Informática, de Mestrado em Informática e Mestrado em Sistemas Móveis na Universidade do Minho. Esta Unidade Curricular encontra-se também enraizada em projectos de investigação na área de sistemas interactivos e na experiência obtida em Prestações de Serviço Especializados à Comunidade (PSEC) no ambito de planeamento de administração de centros de dados e parques informáticos, bem como no desenvolvimento de *software*.

13.2 Caracterização Global

13.2.1 Regime e Escolaridade

UCE apresentada neste documento poderá ser oferecida em regime semestral ou anual de acordo com o público a que se destine. O regime anual será preferencialmente adoptado para a formacção inicial, durante a qual num mesmo ano um aluno poderá realizar duas UCE. Atendendo ao desenvolvimento em paralelo de dois temas programáticos e ao desenvolvimento da componente de projecto considera-se uma escolaridade semanal com duas sessões de duas horas teóricas seguidas de uma hora de carácter prático laboratorial, sendo o acompanhamento presencial de projecto feito em quatro horas de apoio tutorial. A escolaridade ao longo de um ano conduz a uma carga presencial semanal de 4T + 2PL + 4OT.

13.2.2 Áreas Científicas

Áreas Científicas: Informática (30 ECTS)

Áreas Científicas (classificação ACM):

- D. Software/D.2 SOFTWARE ENGINEERING/D.2.2 Design Tools and Techniques
- D. Software/D.2 SOFTWARE ENGINEERING/D.2.3 Coding Tools and Techniques
- D. Software/D.2 SOFTWARE ENGINEERING/D.2.11 Software Architectures
- D. Software/D.2 SOFTWARE ENGINEERING/D.2.12 Interoperability
- D. Software/D.4 OPERATING SYSTEMS
- H. Information Systems/H.2 DATABASE MANAGEMENT/H.2.4 Systems
- H. Information Systems/H.5 INFORMATION INTERFACES AND PRESENTATION (e.g., HCI)/H.5.2 User Interfaces
- K. Computing Milieux/K.6 MANAGEMENT OF COMPUTING AND INFORMATION SYSTEMS

13.2.3 Requisitos

Conhecimentos elementares de programação e sistemas operativos exigíveis a qualquer aluno com formação de primeiro ciclo na área das TICs.

13.2.4 Resultados de Aprendizagem

- Compreender e explicar os compromissos entre custo, desempenho, e confiabilidade na concepção de centros de dados e a sua tradução em tecnologias concretas.
- Aplicar mecanismos de redundância, virtualização e administração centralizada no desenvolvimento e operação de centros de dados.
- Analisar e conhecer os principais patterns estruturais e de comportamento utilizados para o desenvolvimento de sistemas de software complexo e de grande escala, tendo em conta as especificidades arquitecturais das aplicações multi-camada.
- Saber desenvolver camadas computacionais que permitam evolução controlada e independente das camadas de apresentação e dados e que permitam a disponibilização de serviços como mecanismo de integração.
- Identificar as principais características dos servidores aplicacionais por forma a escolher o modelo de programação pretendido e saber utilizar tecnologia orientada a serviços como mecanismo de criação de arquitecturas de software parametrizáveis.
- Compreender e explorar diferentes técnicas de desenvolvimento de camadas interactivas, concebendo interfaces com o utilizador com consideração por aspectos de usabilidade.

- Saber desenvolver camadas de apresentação que permitam evolução controlada e independente das camadas de lógica e de dados.
- Desenvolver de forma integrada a função de concepção e projecto em Engenharia.

13.3 Estrutura Curricular

13.3.1 Estrutura Base

A presente UCE organiza-se em torno de 4 módulos temáticos, correspondendo a 20 ECTS, articulados entre si por um projecto laboratorial integrado, que garante a experimentação e aplicação prática dos resultados da aprendizagem. Os módulos referidos são:

ICD Infraestrutura de Centros de Dados

ABD Administração de Bases de Dados

AA Arquitectura de Aplicações

SI Sistemas Interactivos

13.3.2 Métodos de Ensino

A organização da Unidade prevê em cada momento o decorrer em paralelo de dois temas programáticos (abordados em aulas teóricas e laboratoriais) e do projecto integrado. Para cada tema, haverá uma exposição nas aulas teóricas, que envolverá também o estudo de problemas de pequena dimensão. A resolução de alguns destes problemas será efectuada nas aulas laboratoriais. A aplicação dos conhecimentos a problemas de maiores dimensões será deixada para o laboratório integrado onde o aluno terá oportunidade de desenvolver aplicações relativamente realistas envolvendo uma inter-disciplinaridade relativamente aos vários temas abordados na Unidade Curricular. Serão também promovidos os contactos com a indústria através da realização de seminários e visitas de estudo.

13.3.3 Avaliação

A avaliação da Unidade Curricular tem em conta duas componentes. A primeira é um exame final, contendo perguntas de desenvolvimento e exercícios envolvendo a resolução de pequenos problemas relativamente abstractos. O exame pretende avaliar sobretudo os conhecimentos teóricos. A segunda componente tem em conta o trabalho desenvolvido pelos alunos no projecto integrado, com o fim de avaliar a aplicação do conhecimento à resolução de problemas reais, a capacidade de identificação e relacionamento dos diferentes conhecimentos necessários, e a capacidade de desenvolver aplicações, de as instalar, e as configurar e optimizar para o melhor desempenho.

13.3.4 Contéudos Programáticos

INFRAESTRUTURA DE CENTROS DE DADOS (ICD)

Descrição.

A crescente dependência de multiplas actividades de aplicaçoes informáticas conduz a uma maior profissionalização das infraestruturas de centros de dados. Além das instituições de grande dimensão, são também relevantes os centros de dados usados por prestadores de serviços para alojar aplicações de multiplos clientes em regime de *outsourcing*. Cada vez mais, a administração de centros de dados se baseia em mecanismos de virtualização e a redundância de dados e serviços como forma de optimizar custo, desempenho e fiabilidade.

Em concreto, a gestão de armazenamento vai além do tradicional RAID com a total virtualização de volumes sobre armazenamento em rede. Por outro lado, a utilização de produtos como VMWare ou Xen permite a virtualização de servidores, e a utilização de *application switches*, popularizados pela marca Alteon, a virtualização de serviços.

Este tema é abordado nesta Unidade Curricular de forma a que o aluno compreenda e seja capaz de explicar os compromissoes fundamentais entre custo, desempenho e fiabilidade e o seu impacto na selecção de tecnologias. Este conhecimento é aplicado numa componente prática que permite um conhecimento directos de tecnologias representativas e da sua articulação, com ênfase em soluções *open source*.

Programa resumido.

- Infraestrutura de armazenamento de dados: armazenamento em rede; RAID; gestão de volumes lógicos; sistemas de ficheiros; mecanismos e policas de salvaguarda.
- Infraestrutura de serviços: serviços redundantes; virtualização de servidores.
- Monitorização e gestão centralizada: gestão de parques informáticos; análise de desempenho; monitorização.

ADMINISTRAÇÃO DE BASES DE DADOS (ABD)

Descrição.

A camada de dados numa aplicação *multi-tier* é tipicamente assegurada pela combinação de uma base de dados relacional com *middleware* de acesso orientado por objectos. Esta camada tem uma importância fundamental na aplicação como um topo, em particular no seu desepenho com grandes volumes de dados e de transacções, bem como na sua robustez e disponibilidade.

A concretização desta camada recorre a uma diversidade cada vez maior de produtos. Além dos tradicionais servidores de bases de dados comerciais, produtos *open source* com o PostgreSQL ou MySQL são cada vez mais. O *middleware* de acesso a dados é concretizado no contexto do *application server*, por exemplo, o *Hibernate* em J2EE.

Este tema é abordado nesta Unidade Curricular de forma a que o aluno seja capaz de compreender e explicar os mecanismos fundamentais de implementação de uma base de dados (i.e. controlo de concorrência, gestão de *buffers*, *logs* e recuperação) bem como os compromissoes fundamentais na sua optimização. Este conhecimento é aplicado numa componente prática que permite um conhecimento directos de tecnologias representativas e da sua articulação, com ênfase em soluções *open source*.

Programa resumido.

- Conceitos de implementação de bases de dados: Mecanismos de controlo de concorrência; gestão de buffers; logging e recuperação.
- Gestão de serviços de bases de dados: Planeamento; monitorização; replicação; salvaguarda.
- Optimização de serviços de acesso a dados: Middleware orientado a objectos de acesso a bases de dados; optimização.

ARQUITECTURAS APLICACIONAIS (AA)

Descrição.

A utilização de práticas bem fundadas para a definição da camada aplicacional apresenta-se como determinante na qualidade intrinseca do sistema de software final. A correcta utilização das soluções padronizadas existentes e comprovadas, numa lógica de construção orientada à interconexão de componentes estanques e bem-definidos, é uma mais valia no aumento da qualidade e garantia de evolução da aplicação. A capacidade de construção de aplicações complexas e de larga escala implica uma correcta definição e programação dos serviços existentes por forma a incorporar as necessárias integrações aplicacionais a montante e a jusante do sistema. A capacidade de desenvolver uma aplicação com o respeito pela independência de camadas é um requisito chave para a correcta operação do sistema em período de execução, bem como um factor de automatização do desenvolvimento do mesmo.

O desenvolvimento de uma aplicação multi-camada em contexto de um servidor aplicacional, seja este *Java*, *.Net* ou outro, implica a aquisição de conhecimentos especializados na programação por objectos, nomeadamente nas frameworks e ambientes de exploração definidos pelo *J2EE*. A construção de serviços, com a concretização aplicacional como *Web Services*, permite a disponibilização de um nível de *middleware* que facilmente pode incorporar novas funcionalidades.

Os mecanismos de comunicação e sincronização com as camadas de dados e de apresentação são também importantes e determinam as estratégias de *caching* e lógicas de sessão a desenvolver.

Este tema é abordado nesta Unidade Curricular de forma a garantir que o aluno compreende a assimila as necessidades e pressupostos base para a construção de arquitecturas aplicacionais para sistemas multi-camada e é capaz de explicar as vantagens que a abordagem orientada aos objectos, sustentada num contexto de servidor aplicacional, proporciona. Este conhecimento é concretizado num conjunto de *ferramentas* e técnicas de que o aluno passa a dispôr por forma a aplicá-las na concretização e operação de sistemas de software.

Programa resumido.

- Definição Arquitectural do Sistema de Software: Patterns estruturais e de comportamento; Aspectos avançados de programação orientada aos objectos; Arquitecturas orientadas a serviços; Modelos de programação orientados à construção de componentes reutilizáveis; Manutenção evolutiva de arquitecturas orientadas aos objectos.

- Tecnologias de Programação Multi-Camada: Servidores aplicacionais como contexto aplicacional; Estratégias de desenvolvimento dos mecanismos de independência multi-camada; Programação concorrente; Mecanismos de caching e de sessão; Serviços como técnicas de integração multi-aplicação.

SISTEMAS INTERACTIVOS (SI)

Descrição.

A criação de camadas de interacção com o utilizador independentes das restantes camadas das aplicações, permite o desenvolvimento de sistemas capazes de melhor se adaptarem à evolução quer dos requisitos, quer da tecnologia, bem como a disponibilização das aplicações em ambientes cada vez mais heterogéneos e distribuídos. Neste contexto, o desenvolvimento da camada de interface tem vindo a representar uma cada vez maior fatia dos custos de desenvolvimento das aplicações, existindo estudos que apontam para números superiores a 50% do esforço de desenvolvimento dedicado à interface com o utilizador.

A programação da componente gráfica das interfaces tem vindo a ser facilitada por IDEs com capacidades de programação visual cada vez mais poderosas. A definição das arquitecturas e lógicas comportamentais de suporte às interfaces, no entanto, necessitam de recorrer a padrões e tecnologias apropriados (cf., padrão *Observer-Observable*, mecanismos de *publish/subscribe*, *UI controls*). No caso particular de aplicações disponibilizadas via web, passou-se de uma paradigma de programação síncrono para um paradigma assíncrono (cf. AJAX) que permite o desenvolvimento de interfaces mais complexas e com patamares de usabilidade mais elevados. O desenvolvimento de interfaces para a web pressupõe também a criação de camadas de objectos de apresentação (*Presentation Objects*) como mecanismo de abstracção.

Os diferentes aspectos identificados anteriormente serão abordados nesta Unidade Curricular de forma a garantir que o aluno compreende a implicação de factores de usabilidade no desenvolvimento de interfaces com o utilizador, e é capaz de aplicar técnicas de desenvolvimento de camadas interactivas por forma a desenvolver interfaces que tenham esses factores de usabilidade em consideração e que permitam uma evolução controlada e independente das camadas de lógica e de dados.

Programa resumido.

- Conceitos de IHC: noções básicas de IHC (modelo de interacção de Norman); desenho de interfaces (metáforas, estilos de interacção, guidelines).
- Modelação: prototipagem de interfaces (protótipos de baixa e alta fidelidade); arquitecturas software para sistemas interactivos; padrões de desenho (Observer-Observable, Publish-Subscribe, ...); independência do dispositivo/adaptação ao dispositivo.
- Desenvolvimento: programação orientada a eventos; tecnologias de programação de interfaces (toolkits, AJAX); programação de interfaces síncronas vs. assíncronas; programação de interface para sistemas mono-utilizador vs. multi- utilizador.

14

CG
Computação Gráfica

Sumário

O presente documento descreve uma UCE30 ("Unidade Curricular de Especialidade"), creditada com 30 ECTS, oferecida por um grupo de docentes da Escola de Engenharia, envolvidos na área da Computação Gráfica, no 2o ciclo da re-estruturação curricular da oferta formativa em Informática da Universidade do Minho, no espírito do Processo de Bolonha.

A UCE30 em Computação Gráfica tem por objectivo formar indivíduos com conhecimentos sólidos de base e competências avançadas na área em questão. Estes formandos deverão ser capazes de responder, com conhecimento e qualidade, às actuais e futuras necessidades relacionadas com esta área de conhecimento, verificadas em inúmeros sectores da sociedade e com uma tendência clara para aumentar nos próximos anos. Estes técnicos deverão ser competentes para avaliar/conceber soluções e desenvolver software aplicacional tendo por base técnicas de Computação Gráfica e de Realidade Virtual. Neste sentido, pretende-se que adquiram uma preparação sólida, vocacionada para a inovação mas de grande flexibilidade técnica.

A unidade curricular integra aspectos como as tecnologias avançadas de programação, a geometria computacional, a visão por computador, as técnicas de iluminação e síntese de imagem, a construção de mundos virtuais, e de áreas aplicacionais concretas tais como informação geográfica, arqueologia virtual, arquitectura, jogos, etc.

A equipa proponente acumula vários anos de experiência na investigação, aplicação e ensino de Computação Gráfica, resultando esta UCE directamente desta experiênca.

Coordenação: *António Ramires Fernandes, Luís Paulo Santos, Adérito Marcos, Manuel João Ferreira.*

14.1 Objectivos

A Computação gráfica é uma área na qual a Universidade do Minho se tem empenhado com grande intensidade, em particular desde a instalação do Centro de Computação Gráfica, integrado na reputada organização internacional INI-Graphics Net, até à criação de um mestrado vertical totalmente dedicado à Computação Gráfica e Ambientes Virtuais.

Esta UCE reflecte a experiência adquirida neste campo por um conjunto de docentes de vários departamentos da Escola de Engenharia, no sentido de promover a formação em Computação Gráfica dentro de um mestrado horizontal. Sendo necessáriamente menos abrangente que o mestrado vertical acima mencionado, vem no entanto permitir um contacto efectivo com a àrea, em termos de segundo ciclo, contacto esse que fornece um conjunto de competências que se julga adequado para desenvolver uma actividade profissional que, não sendo centrada na computação gráfica, tire partido das técnicas e metodologias aqui leccionadas.

A metodologia utilizada na construção desta UCE pretende, para cada um dos seus módulos constituintes, apresentar os conceitos teóricos subjacentes às matérias leccionadas, mas sempre complementada com uma forte ligação ao campo aplicacional. Neste sentido esta UCE abrange um conjunto de matérias e permite o desenvolvimento de um leque de competências que representam um espectro alargado da área, e, simultaneamente, formam um corpo de conhecimento coerente, sendo devidamente articuladas ao longo da Unidade Curricular.

A riqueza desta unidade curricular vem não só da sua abrangência, mas também da forma como as matérias e competências são articuladas ao longo dos diversos módulos, estabelecendo ligações fortes entre o 2D e o 3D, entre o processamento e a síntese de imagens, entre a aplicação de tecnologia de ponta, a utilização de equipamentos de valor elevado e a utilização de recursos comuns como computadores pessoais. Garante desta articulação entre os vários módulos, além da experiência dos docentes na leccionação conjunta no Mestrado em Computação Gráfica e Ambientes Virtuais, é o Projecto Integrado, "barramento"laboratorial correspondente a 10 ECTS, que decorre ao longo de todo o período lectivo e que, coordenado por todo o corpo docente, assegura a integração e aplicação de todos os conhecimentos e competências trabalhadas nos vários módulos em projectos práticos e abrangentes de dimensão e complexidade média/alta.

14.2 Caracterização Global

14.2.1 Regime e Escolaridade

A UCE apresentada neste documento é oferecida em regime anual, durante o qual num mesmo ano um aluno poderá realizar duas UCEs. Atendendo à sua articulação em 4 módulos e um projecto integrador, conforme se detalha a seguir, a escolaridade proposta, por semestre, é a seguinte: 4T + 4TP + 4P (sendo as 4P correspondentes à componente de projecto integrador).

14.2.2 Áreas Científicas

Áreas Científicas: Informática (30 ECTS)

Áreas Científicas (classificação ACM): É a seguinte a distribuição dos 30 ECTS da UCE pelas áreas cientificas classificadas no ACM Computing Curricula (1998), apresentadas por ordem lexicográfica:

- Computing Methodologies/COMPUTER GRAPHICS 22
- Computing Methodologies/IMAGE PROCESSING AND COMPUTER VISION 8

14.2.3 Requisitos

Ao longo de todo o curso serão utilizadas intensamente linguagens como C++, APIs como OpenGL, VRML, e bibliotecas de programação diversas. Não sendo um curso orientado para o ensino da programação, exigem-se assim bons conhecimentos prévios da mesma, dado que o objecto fundamental do curso é a Computação Gráfica. Resumidamente os requisitos são os seguintes:

- Experiência em programação, incluindo, preferencialmente, alguma familiaridade com a programação orientada aos objectos em C++.
- Conhecimentos básicos em computação gráfica ao nível do primeiro ciclo.

14.2.4 Resultados de Aprendizagem

- Identificar, classificar e utilizar técnicas de processamento de imagem e extracção de conhecimento a partir de imagens
- Identificar, classificar e aplicar os principais algoritmos e técnicas, básicas e avançadas, de desenho e desenvolvimento de sistemas e aplicações de realidade virtual e aumentada;
- Modelar geometricamente, através de técnicas algoritmicas
- Implementar de forma optimizada recorrendo ao CPU e GPU aplicações gráficas
- Analisar, classificar e implementar modelos e algoritmos de iluminação foto-realista
- Conceber e avaliar soluções e arquitecturas de aplicações de computação gráfica por forma a obter um elevado nível de qualidade e/ou desempenho de acordo com os requisitos do problema.

14.3 Estrutura Curricular

14.3.1 Estrutura Base

A presente UCE organiza-se em torno de 4 módulos temáticos, correspondendo cada um a 5 ECTS, articulados entre si por um 'bus' laboratorial, denominado projecto integrado, que garante a experimentação, a aplicação prática e a integração dos resultados da aprendizagem. Esta estrutura é apresentada na Fig. **??**. Os módulos referidos são os seguintes:

- Visão por Computador (VC) - 5 ECTS
- Realidade Virtual e Aumentada (RVA) - 5 ECTS
- Modelação e Visualização (MV) - 5 ECTS
- Iluminação e Foto-Realismo (IFR) - 5 ECTS
- Projecto Integrado (PI) - 10 ECTS

14.3.2 Métodos de Ensino

- Exposição de conceitos e análise de situações concretas e/ou casos de estudo
- Trabalho de grupo em exercícios e pequenos casos de estudo, com recurso a ferramentas gráficas
- Trabalho de projecto em grupo com orientação directa da equipa docente afecta à UCE.

14.3.3 Avaliação

O processo de avaliação visa aferir não apenas os conhecimentos e competências mínimas adquiridas, mas também a capacidade revelada na sua integração e aplicação no projecto de computação gráfica. Assim, é constituída por um exame final único em média pesada com a avaliação contínua em prática laboratorial e o resultado (ferramentas, relatório, apresentação) do projecto integrador.

14.3.4 Contéudos Programáticos

Os conteúdos programáticos desta UCE são agrupados em 4 módulos temáticos e um projecto integrador, conforme a seguir se detalha

VISÃO POR COMPUTADOR

Descrição.

Este módulo fornece informação de base científica e tecnológica acerca do processamento e análise de imagem, tendo em vista, em específico, a aplicação de visão por computador nos domínios da interacção e da computação gráfica.

Começa por introduzir os conceitos fundamentais de aquisição e formação de imagem, seguindo-se a aquisição de competências relacionadas com o desenvolvimento e implementação de técnicas processamento para o melhoramento e restauração de imagem. No âmbito do tema de análise de imagem são abordadas diversas técnicas de análise de regiões, textura e cor, encaminhando o processo de aprendizagem no sentido das abordagens de tipo "Human Motion Capture".

Programa resumido.

- Formação de Imagem e Aquisição: Técnicas de Iluminação, Óptica, Sistemas de Aquisição, Imagem Digital (Resolução espacial, Resolução tonal, Conectividade e Métricas).
- Representação de Imagem: Espaços de cor, Transformação de espaços de cor.
- Calibração dos sistemas de visão / Restauração de imagem: Correcção de distorções, Correcção de focagem, Calibração e correspondência dimensional, Eliminação de ruído repetitivo, Correcção de intensidade da resposta do sensor.
- Melhoramento de Imagem: Operações sobre imagens (Ponto-a-Ponto, Operações Locais, Operações Globais), Remoção de ruído (Combinação de imagens, Filtros espaciais, Filtros no domínio das frequências), Segmentação (Histograma, Binarização, Extracção de Regiões), Operações Morfológicas.
- Análise de Imagem: Codificação e representação de regiões (rle, chain code, Aproximações poligonais), Análise dimensional.
- Análise de Textura: Transformada de wavelets. Análise fractal. Mapas de interacção de pares de pixels.
- Análise de cor: Percepção da cor pelo sistema de Visão Humano. Medição de cor. Variação de cor. Calibração dos sistemas de visão por computador.
- Aquisição 3D: Técnicas de luz estruturada. Técnicas por Múltiplas-imagens.
- Análise de movimento: Detecção e Classificação do objecto em movimento, Seguimento (tracking), Reconhecimento do movimento.
- Apresentação e Discussão de Casos.

Resultados de Aprendizagem Específicos:

- Avaliar as diferentes soluções tecnológicas ao nível da: iluminação, óptica, hardware de aquisição de imagem (cartas de aquisição e câmaras) e hardware de processamento de imagem.
- Implementar e avaliar técnicas de melhoramento de imagem.
- Conhecer e avaliar técnicas de análise de imagem (textura e cor).
- Implementar algoritmos de análise de movimento.
- Desenvolver e aplicar técnicas de calibração.

REALIDADE VIRTUAL E AUMENTADA

Descrição.

Este módulo tem por finalidade proporcionar aos alunos os conhecimentos e práticas fundamentais acerca dos princípios, conceitos, modelos e principais técnicas relacionadas com a Realidade Virtual e Aumentada. Na abordagem dos conteúdos seleccionados privilegiar-se-á predominantemente a dimensão teórico-prática, com o objectivo de facultar a compreensão de conhecimentos e desenvolver capacidades e habilidades para a concepção, desenho e implementação de aplicações e sistemas de realidade mista.

Programa resumido.

- Introdução: Definições fundamentais; Taxonomia da integração do Real e Virtual; Modelos, Dispositivos de Entrada e Saída para Realidade Aumentada (RA) e Realidade Mista (RM);
- Modelos de Registo e Rendering: Registo baseado na Visão; Seguimento em Realidade Aumentada para Ambientes Naturais; Realidade Mista baseada em Visão por Vídeo See-through; Modelação Fotométrica; Solução Ray-based para comunicação espacial aumentada; Modelação de um mundo virtual a partir do real;
- Aumento Multi-sensorial: Percepção auditiva de profundidade; Som 3D; Modelo de Force feedback; Interfaces tangíveis;
- Comunicação e colaboração: Telepresença aumentada; Colaboração em RM; Tecnologias para a comunicação multimédia em RA e RM;
- Desenho de sistemas e cenários de aplicação : Considerações gerais de desenho; Localização de objectos virtuais; Qualidade versus Tempo-real; Cenários de estudo: RA/RM aplicada à engenharia, herança cultural, turismo, arqueologia.
- Práticas de Implementação.

Resultados de Aprendizagem Específicos:

- Explicar a importância da realidade virtual e aumentada no desenho e implementação de sistemas e aplicações interactivas total ou parcialmente imersivas;
- Identificar os princípios, principais modelos e técnicas relacionadas com a realidade virtual e aumentada, tendo em vista a criação de aplicações e sistemas interactivos imersivos;
- Identificar, classificar e aplicar os principais algoritmos e técnicas, básicas e avançadas, de desenho e desenvolvimento de sistemas e aplicações de realidade virtual e aumentada;

- Identificar, analisar, categorizar e avaliar sistemas e tecnologia existentes; integrar estes em soluções de realidade virtual e aumentada.

MODELAÇÃO E VISUALIZAÇÃO

Descrição.

Este módulo pretende introduzir a criação de modelos com recurso a algoritmos computacionais. A geração de modelos geométricos é assim realizada de uma forma procedimental ao invés de ser criada em software de modelação 3D. Para a geração de modelos geométricos serão explorados algoritmos que em alguns casos tiveram a sua génese em áreas que não a computação gráfica, mas que no entanto se adequam perfeitamente à geração de modelos gráficos com características que por vezes se assemelham à natureza que nos rodeia.

Em seguida o módulo incidirá na visualização de modelos geométricos em situações em que a interactividade é essencial recorrendo a técnicas avançadas de computação gráfica tanto ao nível de um ponto de vista de software, como do aproveitamento das capacidades do hardware gráfico.

No plano prático, (e através de um trabalho laboratorial de grupo) o módulo articula com PI (Projecto Integrado) e com o módulo de Iluminação e Foto-Realismo, na medida em que se pede aos alunos que gerem e visualizem modelos complexos de forma interactiva e utilizando modelos de iluminação que permitam a interactividade, em paralelo com criação de imagens de alta qualidade (foto-realista), embora sem interactividade, dos mesmos modelos.

Programa resumido.

- **Modelação.** Os fractais e suas propriedades. Sistemas de reescrita, em particular L-systems e suas aplicações em geração de plantas, e estruturas arquitectónicas. Algoritmos de geração de terrenos artificias. Algoritmos de geração de texturas.
- **Visualização.** Níveis de detalhe em modelos geométricos. Utilização de mecanismos das APIs gráficas como Display Lists e Vertex Buffers. Programação de shaders. Criação de efeitos gráficos em múltiplos passos em tempo real.

Resultados de Aprendizagem Específicos:

- Definir geometria através de algoritmos
- Utilizar as capacidades das APIs e do Hardware gráfico para desenhar aplicações interactivas com qualidade visual.

ILUMINAÇÃO E FOTO-REALISMO

Descrição.

Este módulo introduz a teoria subjacente à iluminação global e apresenta/discute os modelos e algoritmos de iluminação global baseados nos fenómenos físicos de transporte e reflexão da luz. O formando é assim alertado para a necessidade de conformidade com a realidade nos métodos de síntese de imagens de alta qualidade que possam ser usadas para fins preditivos.

Os algoritmos conhecidos para obter soluções aproximadas da equação de *rendering* são apresentados e discutidos, bem como os raciocínios que os suportam e as suas limitações na simulação de determinados fenómenos de transporte de luz. A complexidade destes algoritmos é estudada, assim como as consequências em tempo de execução dos mesmos.

A componente prática do módulo consiste na utilização de bibliotecas de rotinas que permitem, além da prototipagem rápida de modelos de iluminação global bem conhecidos, a experimentação de novos modelos de iluminação concebidos pelo formando.

A articulação com os módulos de "Realidade Virtual e Aumentada"e "Modelação e Visualização"é conseguida através da utilização dos mesmos modelos virtuais e comparação dos resultados (imagens) e tempos de execução associados a diferentes modelos de iluminação. Esta articulação é ainda complementada com a realização de projectos laboratoriais multidisciplinares no âmbito de "Projecto Integrado".

Programa resumido.

- Modelos de iluminação locais e globais, empíricos e baseados na física (Phong, Cook-Torrance, Ward);
- Radiometria e Fotometria;
- Mecanismos de transporte de luz, a BRDF e a equação de rendering;
- Algoritmos de iluminação global: Ray tracing (Clássico, distribuído e Monte Carlo), radiosidade, photon mapping.

Resultados de Aprendizagem Específicos:

- Explicar a equação de rendering e discutir o significado de cada um dos seus factores;
- Relacionar os vários métodos de iluminação global com o modelo geral sustentado pela equação de rendering;
- Inferir quais os fenómenos de iluminação modelados pelos vários métodos de iluminação global;
- Seleccionar as técnicas de iluminação global mais indicadas para cada problema;
- Projectar soluções para novos problemas de iluminação por recombinação de soluções conhecidas;

- Reconhecer as limitações funcionais e/ou de desempenho associadas a cada algoritmo de iluminação global.

15

BIO
Bioinformática

Sumário

Coordenação: *Miguel Rocha e , Rui Mendes (DI)*
Eugénio Ferreira e Isabel Rocha (DEB),
Margarida Casal (Dep. Biologia), Pedro Oliveira (DPS)

15.1 Objectivos

Esta unidade curricular pretende dar uma formação inicial no campo da Bioinformática, dotando os alunos que a completem de conhecimentos ao nível dos princípios biológicos básicos subjacentes, dos principais problemas que se colocam nesta área e das abordagens estatísticas e algorítmicas mais comuns para a sua solução.

Os módulos diversos que compõem este módulo garantem que o aluno que o complete possuirá conhecimentos ao nível das principais tecnologias e algoritmos da Bioinformática, sendo capaz de as contextualizar, aplicar e analisar. Uma ênfase especial será dedicada aos métodos estatísticos e ao papel da Estatística como base teórica importante para a construção dos algoritmos e modelos subjacentes. A extracção de conhecimento útil a partir de bases de dados biológicas de grande dimensão e complexidade será uma das aplicações privilegiadas.

O módulo terá uma forte componente prática, promovendo o desenvolvimento e a implementação de algoritmos para os principais problemas da Bioinformática, bem como a aplicação e a avaliação de software disponível. Neste âmbito, será incluído um projecto integrador das diferentes disciplinas.

15.2 Caracterização global

15.2.1 Requisitos

Licenciatura do 1º ciclo em Engenharia Informática, Ciências da Computação, Tecnologias e Sistemas da Informação e áreas afins.

15.2.2 Regime e escolaridade

Esta UCE será oferecida preferencialmente em regime anual, durante a qual num mesmo ano um aluno poderá realizar duas UCE. Esta UCE será leccionada também no âmbito do Mestrado em Bioinformática, recentemente criado.

Atendendo à sua articulação em 5 módulos e um projecto integrador, conforme se detalha a seguir, a escolaridade proposta é a seguinte: 10T + 7TP + 3P.

15.2.3 Áreas Científicas (ACM)

J.3 Computer Applications – LIFE AND MEDICAL SCIENCES – Biology and Genetics – 15 ECTS
F.2.2 Theory of Computation – ANALYSIS OF ALGORITHMS AND PROBLEM COMPLEXITY – Nonnumerical Algorithms and Problems – 3 ECTS
H.2.8 Information Systems – DATABASE MANAGEMENT – Database Applications – 3 ECTS
I.2.6 Computing Methodologies – ARTIFICIAL INTELLIGENCE – Learning – 3 ECTS
I.2.8 Computing Methodologies – ARTIFICIAL INTELLIGENCE – Problem Solving, Control Methods, and Search – 3 ECTS
I.6.5 Computing Methodologies – SIMULATION AND MODELING – Model Development – 3 ECTS

15.2.4 Resultados de aprendizagem

- Identificar os principais problemas na área da Bioinformática e saber seleccionar os algoritmos e as técnicas computacionais mais adequadas, sabendo procurar, utilizar, classificar e avaliar aplicações de software disponível na área da Bioinformática e de Extracção de Conhecimento.

- Analisar os resultados e o interpretar o significado biológico de ferramentas de Bioinformática e análise dados biológicos.

- Conhecer e ser capaz de desenhar e implementar os principais algoritmos relacionados com a Bioinformática e as técnicas de Aprendizagem Máquina e Mineração de Dados.

- Identificar, definir e descrever estrutural e funcionalmente as principais biomoléculas, vias metabólicas, processos ao nível intra-celular incluindo a síntese proteica.

- Identificar e descrever os mecanismos de crescimento microbiano e de mistura e transferência de massa e de calor em reactores biológicos bem como os princípios da Engenharia Metabólica.

- Formular e testar hipóteses de acordo com o método estatístico mais adequado no contexto da análise de dados biológicos.

15.3 Estrutura curricular

15.3.1 Módulos internos

- Algoritmos e Tecnologias da Bioinformática
- Biologia Molecular e Celular
- Engenharia Bioquímica
- Métodos Estatísticos para a Bioinformática
- Extracção de Conhecimento em Bases de Dados Biológicas
- Projecto de Bioinformática

15.3.2 Métodos de ensino

- Aulas teóricas com exposição de conceitos, apresentação de exemplos e análise de casos de estudo;
- Trabalho em grupo na resolução de exercícios e abordagem de estudos de caso, usando ferramentas de programação e em certos casos com recurso a ferramentas informáticas específicas;
- Trabalho de projecto com orientação directa da equipa docente afecta à UCE.

15.3.3 Avaliação

O processo de avaliação visa aferir não apenas os conhecimentos e competências mínimas adquiridas, mas também a capacidade revelada na sua integração e aplicação a problemas reais e casos de estudo específicos. A avaliação é constituída por um exame final único, que faz média pesada com a avaliação contínua dos diversos módulos (que podem incluir o desempenho nas aulas, a prática laboratorial, trabalhos individuais ou em grupo), bem como o resultado (ferramentas, relatório, apresentação) do projecto integrador.

15.3.4 Conteúdos programáticos

Algoritmos e tecnologias da Bioinformática

- Regime, escolaridade e ECTS: 2 T + 2 TP, 5 ECTS (Semestre 1)
- Programa resumido: Introdução aos conceitos da Bioinformática; Bases de dados biológicas; Algoritmos de alinhamento de sequências; Procura de padrões em sequências; Análise filogenética; Clustering de microarrays; Classificação de proteínas e previsão da estrutura.
- Resultados de aprendizagem específicos:
  - Identificar, descrever e definir os principais conceitos na área da Bioinformática;
  - Identificar e descrever os principais problemas que se colocam ao nível da Bioinformática;
  - Selecionar, aplicar e avaliar ferramentas de Bioinformática na resolução de problemas;
  - Escolher as classes de algoritmos apropriadas para a resolução dos problemas básicos da Bioinformática;
  - Implementar algoritmos básicos de Bioinformática nas diversas linguagens de programação.

Biologia Molecular e Celular

- Regime, escolaridade e ECTS: 2 T + 1 TP, 5 ECTS (Semestre 1)
- Programa resumido: Armazenamento e transporte de energia nas células. Glicólise, ciclo de Krebs fosforilação oxidativa. Transportadores de electrões em sistemas biológicos. Mecanismo quimio-osmótico de formação de ATP. Via das pentoses-fosfato. Formação de NADPH e interconversão hexoses/pentoses. Organização dos sistemas vivos. Diferenciação celular. Principais métodos de estudo da célula: métodos citológicos e bioquímicos. Biomembranas: estrutura e função, sistemas de transporte. Núcleo: ultra-estrutura, composição, organização e funções. Replicação do DNA. Transcrição, tradução e as moléculas mRNA, tRNA e rRNA. Citosol: composição e principais características. Compartimentação endomembranar. Retículo endoplasmático: funções na síntese lipídica e proteica. Ultraestrutura e função do Complexo de Golgi. Lisossomas. Peroxisomas. Mitocôndria. Citoesqueleto.
- Resultados de aprendizagem específicos:
  - Descrever as principais vias metabólicas celulares: glicólise, ciclo de Krebs e a via das pentoses fosfato
  - Descrever funcional e estruturalmente as várias estruturas celulares, nomeadamente, membranas celulares, núcleo, ribossoma, cromossoma, retículo endoplasmático, aparelho de Golgi, lisossoma, peroxissoma, mitocôndria e citosqueleto

  - Relacionar os diferentes mecanismos de transporte de substâncias para o interior das células e de tráfego intracelular: difusão, osmose, difusão facilitada, transporte activo, endocitose, exocitose e secreção de proteínas
  - Descrever as etapas da síntese de proteínas, identificando o papel do DNA, do mRNA, do tRNA, do rRNA e dos ribosomas nos processos de transcrição e de tradução
  - Aplicar conceptualmente as metodologias experimentais básicas de biologia molecular e celular na resolução de problemas biológicos na investigação científica

Engenharia Bioquímica

- Regime, escolaridade e ECTS: 2 T + 1 TP, 5 ECTS (Semestre 1)
- Programa resumido: Problemas Fundamentais da Engenharia Bioquímica: papel dos bioreactores em Biotecnologia; organismos e meios de cultura industriais. Cinética do Crescimento Microbiano: parâmetros de crescimento; análise de dados de crescimento; equação de Monod e outras equações descritivas do crescimento microbiano; taxas específicas de consumo e produção. Variáveis de Estado: biomassa, substrato, produtos, determinação Experimental. Modos de Operação de Reactores Biológicos: cultura em contínuo / quimiostato, cultura em modo semi-contínuo, cultura em descontínuo. Estequiometria das Reacções Microbianas: equação estequiométrica geral, necessidades em carbono e energia.
- Resultados de aprendizagem específicos:
  - Identificar e descrever os mecanismos de crescimento microbiano
  - Identificar os mecanismos de mistura e transferência de massa e de calor em reactores biológicos
  - Identificar o modo de operação de reactores biológicos
  - Identificar os princípios da Engenharia Metabólica

Métodos estatísticos para a Bioinformática

- Regime, escolaridade e ECTS: 2 T + 1 TP, 5 ECTS (Semestre 2)
- Programa resumido: Distribuições de Probabilidade. Inferência Estatística. Planeamento de Experiências. Regressão linear. Regressão Logística. Análise Discriminante. Análise de Sobrevivência. Processos estocásticos e modelos de Markov; Métodos estatísticos para a análise de sequências; Aplicações da estatística à análise filogenética.
- Resultados de aprendizagem específicos:
  - Identificar, descrever e definir os principais conceitos da teoria de probalidades, da inferência estatística e dos processos estocásticos;

  - Planear experiências e recolher dados de acordo com um plano de amostragem.
  - Formular e testar hipóteses de acordo com o método estatístico mais adequado no contexto da análise de dados biológicos;
  - Aplicar os conceitos estatísticos, fazendo uso de software disponível, à análise de sequências biológicas e à análise filogenética;
  - Aplicar processos estocásticos à modelação

Extracção de conhecimento em bases de dados biológicas

- Regime, escolaridade e ECTS: 2 T + 2 TP, 5 ECTS (Semestre 2)
- Programa resumido: Sistemas de suporte à decisão; Processamento analítico de dados; Data Warehousing; Mineração de Dados e Aprendizagem Máquina: Modelos e algoritmos de classificação e regressão; Indução de árvores de decisão e regras de classificação; Aprendizagem baseada em instãncias.
- Resultados de aprendizagem específicos:
  - Identificar, descrever e definir os principais conceitos relacionados com os sistemas de suporte à decisão, processamento analítico de dados, data warehousing e mineração de dados;
  - Seleccionar as metodologias apropriadas e aplicar software disponível na resolução de problemas reais ao nível da análise de dados e tomada de decisão;
  - Conhecer e ser capaz de implementar os principais algoritmos relacionados com técnicas de Mineração de Dados e Aprendizagem Máquina;
  - Aplicar as metodologias de extracção de conhecimento no caso específico das bases de dados biológicas.

Projecto de Bioinformática

- Regime, escolaridade e ECTS: 3 P, 5 ECTS (Semestre 2)
- Programa resumido: Desenvolvimento de um projecto integrador, fazendo a especificação, o desenho e a implementação de aplicações na área da Bioinformática, comparando-as criticamente com aplicações existentes no mesmo âmbito.
- Resultados de aprendizagem específicos:
  - Conhecer e descrever os principais algoritmos da Bioinformática;
  - Desenvolver e implementar aplicações de Bioinformática para os principais algoritmos da Bioinformática;
  - Pesquisar, aplicar e integrar módulos de software disponível ao nível de repositórios livres;
  - Integrar o conhecimento das disciplinas anteriores na resolução de problemas específicos.

16

SSD
Sistemas de Suporte à Decisão

Sumário

Coordenação: *Orlando Belo, Luís Amaral (DSI), Paulo Azevedo, Maribel Fernandes (DSI)*

16.1 Contextualização

Nos dias de hoje, as organizações para serem competitivas e capazes de reagirem de forma adequada aos mais diversos eventos que ocorrem nos seus domínios de trabalho constantes mudanças e novos requisitos do mercado em que se inserem, impacto dos progressos tecnológicos, emergência de novos processos de gestão organizacionais, novos modelos de organização e exploração de dados, mudanças de atitude, tácticas e estratégias nos processos de negócio, etc. têm a necessidade de lidarem frequentemente com grandes volumes de informação, normalmente armazenados em sistemas de dados distribuídos, avaliarem essa mesma informação e, acima de tudo, serem capazes de com ela tomarem decisões correctas. Dias após dia, no desenvolvimento das suas tarefas quotidianas, estas organizações vão gerando grandes volumes de informação que, de uma forma ou de outra, vão armazenando nos repositórios de dados dos seus sistemas operacionais, muitas vezes com uma organização algo deficitária e precária. Desta forma, o desenvolvimento de processos de exploração e utilização dos dados, contidos nas fontes de informação das organizações, vão sendo cada vez mais de difícil execução, provocando, a curto prazo, a saturação do sistema e dos seus próprios utilizadores. Tal provoca, naturalmente, atrasos significativos em eventuais processos de tomada de decisão. A heterogeneidade das fontes de informação de uma organização associada com a (frequente) incapacidade dos seus sistemas seleccionarem e fornecerem informação adequada, em tempo útil e em formatos ajustados, fazem com que as organizações sejam muitas vezes confrontadas com resultados pouco credíveis (e às vezes algo inesperados) quando questionam os sistemas sobre uma mesma matéria, mas utilizando caminhos de interrogação alternativos. O efeito da diversidade e heterogeneidade das

fontes de informação pode ser atenuado se, com base em critérios bem definidos de selecção, extracção e integração, a informação necessária aos processos de tomada de decisão, contida nessas fontes, fosse conciliada num único sistema de dados homogéneos. Para que isso seja possível é necessário planear previamente a definição e implementação de um sistema de armazenamento especial pelo menos com características especiais ao nível do desenho dos seus sistemas de armazenamento e processos de integração de dados. A área tecnológica dos Sistemas de Data Warehousing tem vindo a tratar deste tipo de problemas e situações que começam, desde há alguns anos para cá a nível nacional, a serem muito frequentes numa organização. A implementação de um sistema de data warehousing no seio de uma organização disponibiliza um meio eficiente para o armazenamento consistente de dados sem ruído, organizados segundo os diversos eixos de decisão organizacionais, e os serviços adequados para a manipulação e exploração desses dados de forma simples e flexível.

16.2 Motivação e Objectivos

Tendo em conta que, nos últimos anos, temos vindo a assistir, a nível nacional, à emergência de várias iniciativas e promoção de projectos, assim como de inúmeras outras actividades de desenvolvimento de trabalhos no domínio dos sistemas de suporte à decisão, e que foram detectadas faltas de pessoal devidamente qualificado para conduzir essas iniciativas e trabalhos especializados, achamos que este é momento adequado para lançar um novo projecto de ensino - A Unidade Curricular de Ensino (UCE) em Sistemas de Suporte à Decisão (SSD) -, especificamente relacionado com esse domínio, em particular na área dos Sistemas de Data Warehousing e Processamento Analítico. Esta UCE pretende, assim, transmitir aos alunos o conhecimento necessário para planearem, justificarem, desenharem, implementarem e gerirem um projecto para um sistema de suporte à decisão, com particular incidência na implementação de sistemas de data warehousing, bem como fornecer-lhes os conhecimentos necessários para serem capazes de explorarem efectivamente os data warehouses desenvolvidos ou as estruturas multidimensionais de dados hipercubos criadas como suporte para a optimização dos processos de satisfação de resultados. A actual UCE visa completar e aprofundar os conhecimentos de jovens licenciados, profissionais de sistemas de informação e das tecnologias da informação em geral, analistas, consultores e gestores de empresas no domínio dos sistemas de suporte à decisão.

16.3 Público Alvo

A UCE em SSD destina-se essencialmente a jovens licenciados, a profissionais de sistemas de informação e a quadros de gestão de empresas que pretendam aprofundar os seus conhecimentos no domínio dos sistemas de suporte à decisão e, mais especificamente, no domínio dos sistemas de data warehousing e processamento analítico. O curso garantir-lhes-á valências efectivas nas principais áreas de formação dos sistemas de data warehousing. Em termos gerais, no final da leccionação desta UCE, os alunos terão a perícia e o conhecimento necessários para definir, justificar e implementar um projecto de um sistema de data warehousing de raiz, bem como saber

como optimizar o seu desempenho nas diversas áreas de intervenção do sistema e explorar, através do emprego de técnicas avançadas de extracção de conhecimento, os seus vastos repositórios de dados temporais.

16.4 Linhas de Orientação

A UCE em SSD foi organizada, essencialmente, de acordo com quatro linhas de orientação distintas, nomeadamente: Administração e Exploração Avançada de Bases de Dados (BD), Sistemas de Data Warehousing (SDW); Sistemas de Processamento Analítico (SPA); e Sistemas de Extracção de Conhecimento (SEC). Cada uma destas linhas corresponde a uma área especializada de estudos, a um perfil de formação específico, sendo composta por um conjunto de tarefas e trabalhos que asseguram a formação dos alunos nas suas várias vertentes científicas e técnicas. Em termos gerais, a primeira linha disponibiliza uma formação complementar à adquirida pelos alunos durante a sua licenciatura, garantindo-lhes os conhecimentos para administrarem e explorarem de forma efectiva qualquer sistema operacional de uma organização. A segunda linha de orientação, Sistemas de Data Warehousing, forma os alunos na análise, planeamento, desenho, projecto, implementação e exploração de sistemas de data warehousing. A linha de Sistemas de Processamento Analítico disponibiliza, essencialmente, o conhecimento e as técnicas necessárias para a modelação de bases de dados mutltidimensionais e processamento analítico de dados. Por fim, a linha de Sistemas de Extracção de Conhecimento disponibiliza as metodologias, modelos e técnicas para a extracção de conhecimento em sistemas que integrem repositórios de dados de grande dimensão, contendo informação pertinente para a aplicação de sistemas de mineração de dados. Pensamos assim cobrir de forma efectiva as quatro vertentes fundamentais no estudo de sistemas reais para suporte à decisão. No sentido de providenciar uma forma para a combinação de perícias e conhecimento entre as diversas linhas enunciadas, foi criado um módulo de trabalho prático para a realização de projectos interdisciplinares, com o objectivo de providenciar aos alunos do curso a oportunidade de desenvolverem projectos que integrem os conhecimentos adquiridos em todas as linhas referidas. Desta forma, garante-se não só a formação específica em cada uma delas, mas também uma abordagem mais abrangente, cobrindo todos os aspectos de desenvolvimento e implementação de um sistema de data warehousing e de processamento analítico, e a potencial exploração dos seus repositórios de dados por técnicas avançadas de descoberta de conhecimento.

16.5 Resultados da Aprendizagem

- A administração avançada de sistemas de gestão de bases de dados.
- A auditoria e optimização de sistemas de bases de dados.
- O projecto e implementação de sistemas de data warehousing.
- O projecto e implementação de aplicações para a extracção, transformação e integração de dados em sistemas de data warehousing.

- O projecto e implementação de aplicações para sistemas de bases de dados multidimensionais.
- O projecto e o desenvolvimento sistemas de descoberta de conhecimento.
- A gestão, o acompanhamento e a avaliação de projectos de implementação de armazéns de dados.

16.6 Estrutura Curricular

A estrutura curricular da UCE em SSD é constituída pelos seguintes módulos e respectivos programas:

16.6.1 M1: Administração e Exploração Avançada de Bases de Dados, 5 ECTS.

Administração avançada de sistemas de gestão de bases de dados. Reavaliação de projectos de sistemas de bases de dados. Plataformas computacionais para sistemas de bases de dados. Monitorização de eventos. Planos de segurança e recuperação de bases de dados. Programação embebida em sistemas de gestão de bases de dados. Análise e Optimização de Interrogações. Fundamentos do processamento de interrogações. Decomposição de interrogações. Sistemas para a indexação de dados. Antecipação de resultados. Aproximações heurísticas à optimização de interrogações. Estimativa de custos computacionais para operações de processamento e optimização de interrogações. Analisadores de interrogações.

16.6.2 M2: Sistemas de Data Warehousing, 5 ECTS.

Ambiente e estrutura funcional de um data warehouse. Ciclo de vida e desenvolvimento incremental de um data warehouse. Granularidade, particionamento e estruturação da informação num data warehouse. Projecto de um data warehouse. Modelos de dados para um data warehouse. Projecto e implementação de sistemas de extracção, transporte, transformação e integração de dados. Aplicação de técnicas de workflowing em processo de povoamento de sistemas de data warehousing. Ferramentas de modelação. Sistemas de wrapping. Data warehouse departamentais. Gestão e administração de data warehouses. Implementação de data warehouses em plataformas Cliente/Servidor. Data warehouses distribuídos.

16.6.3 M3: Processamento Analítico de Dados OLAP, 5 ECTS.

Fundamentos do processamento analítico de dados. Estruturas de armazenamento de dados para sistemas analíticos. Algoritmos e estruturas de dados para o processamento analítico de dados. Reestruturação dinâmica de cubos. Materialização de vistas. Processamento e materialização distribuída de estruturas multidimensionais de dados. Administração de sistemas de gestão de

bases de dados multidimensionais. Elaboração de projectos de sistemas de bases de dados. Descrição de dados multidimensional. Manipulação de dados multidimensional. Definição de critérios de acesso a dados. Manipulação de vistas multidimensionais. Programação de sistemas de interrogação para sistemas de bases de dados multidimensionais.

16.6.4 M4: Mineração de Dados OLAP: 5 ECTS.

Introdução aos sistemas de descoberta de conhecimento. Ciclo de vida da descoberta de conhecimento. Introdução à aprendizagem máquina e mineração de dados. Preparação e preprocessamento de dados. Linguagens e arquitecturas para mineração de dados. Descrição de conceitos. Técnicas e modelos de classificação e de predição. Mineração de regras de associação. Mineração de sequências. Geração e análise de clusters. Mineração de tipos de dados complexos.

16.6.5 M5: Projecto Multidisciplinar, 10 ECTS.

Parte I - Projecto e implementação de um Sistema de Data Warehousing. Desenvolvimento de um projecto de aplicação prática dos conhecimentos adquiridos nos módulos de Administração e Exploração Avançada de Bases de Dados e Sistemas de Armazéns de Dados Data Warehousing.

Parte II - Projecto de sistemas analíticos de dados com integração de componentes de mineração de dados. Desenvolvimento de um projecto de aplicação prática dos conhecimentos adquiridos nos módulos de Processamento Analítico de Dados e Mineração de Dados.