

ANÁLISE E CONCEÇÃO DE *SOFTWARE*

PROJETO INTEGRADO

MEI – DI/UM - 2012/2013

PROJETO DE ENSINO

Pretende-se que este projeto funcione como base única de referência para a apresentação de todas as questões relacionadas com a análise e desenvolvimento de sistemas *software* tendo por base o processo RUP, a linguagem UML, alguns métodos de utilização e transformação sistemática de alguns modelos UML, até ao desenvolvimento do código final da aplicação.

Assim, a partir de um caderno de requisitos inicial, naturalmente vago e possivelmente incompleto, pretende-se realizar toda a modelação correspondente à análise e captura de requisitos, passar para a fase de conceção de alto nível e em seguida realizar a conceção detalhada e a implementação, sempre tendo em atenção o facto de que todo o processo é iterativo e, de alguma forma, segue um método e um calendário que será proposto.

A análise e captura de requisitos serão baseadas em técnicas específicas lecionadas nos módulos Análise e Modelação de Requisitos (AMR), bem como em modelos lecionados em Arquiteturas de Software (AqS). Alguns exemplos são o **modelo de domínio**, **os diagramas de casos de uso**, **a descrição dos casos de uso** e, caso seja necessário, **os diagramas de atividade**.

Numa primeira fase, toda a modelação será realizada ao nível do **sistema**. A partir dos casos de uso serão desenvolvidos, de forma metodológica, **os diagramas de sequência** ao nível do sistema. Nesta fase serão realizados os primeiros testes baseados na noção de **cenários dos casos de uso**.

O início da fase de conceção consistirá fundamentalmente na aplicação de várias estratégias que nos permitam realizar o refinamento do designado **sistema**, identificando subsistemas internos e externos. Para os subsistemas internos serão desenvolvidas **classes** que, numa metodologia OO, representam a estrutura e o comportamento necessários à satisfação das funcionalidades capturadas nos casos de uso, as quais foram previamente detalhadas em termos de fluxo nos diagramas de sequência. Nesta fase, as classes poderão ser agrupadas em **packages** caso apresentem a coesão funcional que o justifique.

Uma análise detalhada permitirá completar estas classes com os seus atributos e métodos, resultando no **diagrama de classes, interfaces e *packages*** que representarão a estrutura estática do projeto nesta fase.

Nesta fase, serão também realizados testes que, tendo por base os **diagramas de sequência refinados** entretanto elaborados, irão permitir concluir sobre o grau de completude da conceção das classes concebidas.

As classes resultantes da conceção final serão posteriormente implementadas em JAVA, devendo ser estudadas várias possibilidades de **arquitetura de implementação final** (cf. interfaces com o utilizador, utilização de componentes, etc).

Contributos dos Módulos de ACS para o Projeto

1. Análise e Modelação de Requisitos (AMR)

Os conteúdos abordados no módulo de AMR serão úteis em várias tarefas do projeto. Como regra, o levantamento de requisitos deve ser executado com recurso a uma gama diversificada de técnicas. Para além de outras técnicas de levantamento de requisitos que podem ser úteis, sugere-se que cada equipa:

- Faça uma análise de sistemas semelhantes já disponíveis no mercado,
- Caracterize detalhadamente algumas (~3) *personas* para as quais o produto possa estar especialmente dirigido,
- Contacte alguns especialistas do domínio (e.g., agente de viagens, funcionário de empresa de transportes, etc) que possam aportar algum requisito para a aplicação.

Adicionalmente, as equipas devem associar a cada requisito uma prioridade e usar esta informação durante o projeto para tomar decisões de implementação. Deve ainda haver o cuidado em escrever os requisitos num formato padronizado (e.g., **Volere**) e segundo as recomendações que vão ser descritas nas aulas de AMR.

2. Arquiteturas de *Software* (AqS)

O módulo de Arquiteturas de *Software* contribuirá para o projeto integrado com técnicas e ferramentas que cobrem as várias fases do projeto desde a conceção até à instalação, designadamente:

- 1) Captura de requisitos funcionais e modelação usando UML;
- 2) Transformação sistemática de modelos UML;
- 3) Estudo e conhecimento da plataforma J2EE e referências a .NET;
- 3) Utilização de padrões de conceção ("*design patterns*") comuns e úteis;

- 4) Estudo de Java Servlets, JSP e JSF, bem como a integração destas tecnologias num modelo MVC de implementação;
- 5) Caso se justifique, modelação e verificação de modelos de partes da aplicação usando Alloy.

3. Análise e Transformação de Software (ATS)

4. Engenharia de Aplicações Web (EAW)

Do projeto a desenvolver deverá constar a implementação de uma interface que permita aceder ao sistema através da Web. Espera-se, em particular, que sejam desenvolvidas interfaces para os *browsers* mais utilizados (e.g., IE/Firefox/Chrome) em ambiente PC mas também versões desenhadas e implementadas à medida de dispositivos mais simples (tipicamente através de versões 'mobile' das anteriores). [Para atingir tal objetivo, cada grupo terá de desenvolver o Coarse Design da sua aplicação Web.](#) Espera-se que neste sentido possam ser úteis os conteúdos do módulo de Engenharia de Aplicações Web (EAW).

Resultados de Aprendizagem

- Adquirir competências que permitam o trabalho em grupo.
- Análise de um problema e levantamento dos respetivos requisitos.
- Elaboração de uma solução concetual completa para um problema.
- Implementação de uma determinada solução, dada a sua especificação.
- Capacidade para desenvolver aplicações Web para diferentes plataformas.
- Adquirir competências na utilização de tecnologias JAVA para a Web.

DESCRIÇÃO DO PROJETO

EuroTripsFinder

Pretende-se desenvolver e implementar um sistema acessível via Web, utilizando um *browser* ou um cliente móvel, que disponibilize informações sobre a possibilidade de efetuar uma determinada viagem pela Europa, sujeita a condicionalismos temporais, geográficos, financeiros, ou outros. No contexto deste projeto, a uma “viagem possível” irá corresponder um conjunto de possíveis percursos encontrados pelo sistema, em que cada percurso é constituído por uma sequência de etapas, e cada etapa é uma ligação entre duas cidades/estações.

A informação disponibilizada pressupõe que as viagens serão efetuadas em meios de transporte coletivos, tais como comboio, autocarro, barco, ou avião. Contudo, pode assumir-se que o comboio será o “caso de estudo” a tratar em detalhe.

Cada possível percurso, e cada etapa desse percurso, estão sujeitas a condicionalismos que o sistema deve ser capaz de tratar:

- Condicionalismos temporais que definem as datas pretendidas para a partida e a chegada, e o tempo de estadia na cidade destino da etapa/percurso.
- Condicionalismos geográficos, que definem a cidade de partida e a cidade de destino.
- Condicionalismos financeiros que especificam o custo máximo a gastar no percurso, ou mesmo em cada etapa do percurso.
- Adicionalmente, poderão ainda ser especificados condicionalismos sobre o tipo de meio de transporte pretendido para o percurso como um todo, ou para cada etapa em particular.

O utilizador deve poder indicar o número máximo de cidades/estações a considerar na procura de percursos possíveis.

Respeitando os condicionalismos impostos à viagem que se quer planear, o sistema deve:

- Fornecer os percursos possíveis. O resultado deverá incluir toda a informação associada a cada percurso: a sequência de etapas e a descrição de cada etapa (cidades, datas, custo em Euro, tipo de meio de transporte, etc). Convém recordar que os percursos poderão conter descontinuidades temporais, impostas pelos condicionalismos do viajante ou pela ausência de transportes nos horários pretendidos;

- Na impossibilidade de fornecer um ou mais percurso que respeitem todos os condicionalismos, deve procurar fornecer alternativas aproximadas. Se isto não for exequível, o sistema deve sempre indicar quais os aspetos da viagem para os quais não encontra solução e pedir ao utilizador que ajuste ou reajuste os seus condicionalismos impostos à viagem.

O sistema deve ainda permitir que:

- O utilizador peça informações sobre cada estação/cidade, por exemplo, quais as atividades que decorrem na data em que o percurso passe por lá;
- O utilizador se registre e desse modo possa guardar, consultar e gerir os percursos de viagem que o sistema lhe forneceu;
- A seleção de estações/cidades e a visualização de percursos seja efetuada sobre um mapa geo-referenciado (Google maps, OpenStreetMaps, etc).

Devem ser exploradas possibilidades de negócio que visem rentabilizar o sistema, o qual é por natureza de acesso gratuito e se destina a grandes massas:

- Cada estação/cidade pode ter a si associada publicidade paga relativa a empresas ou produtos relacionados com o local;
- O sistema também pode aceitar publicidade global (não localizada).

Extras valorativos:

- Para que o sistema possa ser convenientemente testado e tenha interesse prático, tem que possuir informação válida e em quantidade significativa. Para atingir esse objetivo, deve ser tentada a possibilidade de acesso a informação (horários e preços) de alguma(s) empresa(s) de transportes (por exemplo, a CP ou a Rede Nacional de Expressos).
- Seria interessante que o utilizador possa descarregar os percursos, fornecidos pelo sistema, para um dispositivo móvel (tipicamente um *smartphone*). Esses percursos poderiam depois ser visualizados nesse dispositivo móvel, antes e durante a possível viagem.

Nota Final:

Numa primeira abordagem ao projeto, poderão ser assumidas duas “simplificações”:

- a) Considerar apenas dois meios de transporte, como por exemplo: comboio-autocarro ou comboio-avião.
- b) Considerar apenas informação referente à península Ibérica.

Execução do Projeto e Entregas

FASE 1: Início

- **Duração:** 7/8 semanas
- **Tarefas:**
 - Elaborar o **modelo de domínio**, incluindo um **modelo de negócio simples** (a **29.out entregar DOC_MDom**)
 - **Levantamento de requisitos**, explicando a forma como foram obtidos (análise do enunciado do projeto, diálogo com *stakeholders*, *personas*, entrevistas, análise de produtos semelhantes, etc.) (a **19.nov entregar DOC_Req**)
 - Elaboração do **diagrama de casos de uso** (1/2 semanas)

FASE 2: Elaboração

- **Duração:** 14/15 semanas
- **Tarefas:**
 - Refinar o modelo de domínio e negócio (ao longo de toda a fase)
 - Refinar os requisitos (ao longo de toda a fase)
 - **Descrição textual dos casos de uso** (a **17.dez entregar DOC_UC** contendo todos os diagramas e a descrição textual de apenas 5 casos de uso, considerados relevantes e representativos do problema)
 - Elaboração dos **diagramas de atividade** (a **14.jan entregar DOC_DA**, contendo apenas os diagramas relativos aos 5 casos de uso selecionados na entrega de 17/dez)
 - Elaboração do **diagrama de classes**, contemplando o modelo de negócio pensado até esta fase, e estruturado com *packages* e interfaces (a **4.fev entregar DOC_DC**)
 - **Interface com o utilizador:** desenhar a interface com o utilizador para o sistema (e cliente móvel), utilizando para isso esboços do tipo *mockup* (a **25.fev entregar DOC_UI**).
 - **Relatório intermédio:** o grupo deverá entregar um relatório intermédio a documentar o trabalho desenvolvido até à altura. Este documento deverá ser estruturado e ser mais do que uma mera compilação dos documentos anteriores (a **4.mar**)

- Elaboração de **Interaction Overview Diagrams (IODs)** (a **8.abr** entregar DOC_IOD, contendo apenas os IODs correspondentes aos 5 casos de uso selecionados na entrega de 17/dez)

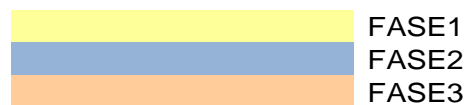
FASE 3: Construção

- **Duração: 12 semanas**
- **Tarefas:**
 - Afinar os modelos UC, DS e DC
 - **Modelo de dados persistentes:** nesta fase será necessário entregar o modelo dos dados persistentes, ou seja, aqueles que serão armazenados em base de dados (a **22.abr** entregar DOC_MD)
 - Definir a **arquitetura de implementação** do sistema, incluindo o cliente móvel se optarem por o implementar, justificando as opções tomadas (a **13.mai** entregar DOC_ArqS).
 - **Protótipo 1:** o grupo terá que mostrar um primeiro protótipo do sistema em projeto (a **3.jun**)
 - Concluir a **implementação do sistema** e do **cliente móvel**. A **1.jul** deverão ser apresentados os protótipos finais.
 - Efetuar **testes** ao sistema antes da apresentação final.
 - **Relatório final:** a última tarefa do grupo será entregar um relatório final descrevendo todas as tarefas realizadas e os resultados obtidos no projeto (a **8.jul**).

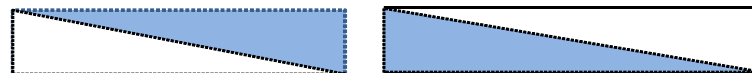
NOTA 1: O relatório deve ser estruturado para que os capítulos principais não ultrapassem 75 páginas, sendo a informação mais técnica e volumosa (UC, DSS,...) remetida para anexos.

NOTA 2: Todos os resultados (deliverables) do projeto (relatório, ficheiros com os modelos UML, código desenvolvido e as instruções para instalação do protótipo) devem ser entregues no final num CD/DVD ou num ficheiro compactado acessível eletronicamente.

	SEMANA 1	SEMANA 2	SEMANA 3	SEMANA 4	SEMANA 5
OUT					<i>Modelo de Domínio (e negócio)</i>
NOV				<i>Documento Requisitos</i>	
DEZ				Diagrama de UC e descrições textuais / FÉRIAS	FÉRIAS
JAN			<i>Diagramas de Atividade</i>		
FEV		<i>Diagrama de Classes</i>			<i>Interface com Utilizador</i>
MAR		<i>Relatório Intermédio e Apresentação 1</i>			FÉRIAS
ABR		<i>Interaction Overview Diagrams</i>		<i>Modelo de Dados persistentes</i>	
MAI			<i>Arquitectura de Implementação</i>		
JUN		<i>Demonstração 1 do protótipo</i>			
JUL	<i>Protótipos Finais e Apresentação 2</i>	<i>Relatório Final</i>			



Semana repartida por 2 meses



Apresentações

1 – **Apresentação intermédia:** Nesta fase o grupo fará uma apresentação sobre todo o trabalho desenvolvido até à altura. A apresentação será pública e o Departamento será convidado a assistir.

Data: 4 de Março de 2013

2 – **Apresentação final:** uma vez o projeto concluído, o grupo terá de fazer uma apresentação do seu trabalho. Mais uma vez, este evento será aberto ao público.

Data: 1 de Julho de 2013

NOTA FINAL

Toda a calendarização do projeto, decisões fundamentais de projeto, técnicas, ferramentas e tecnologias a integrar neste projeto deverão ser validadas e concertadas com os *stakeholders* do projeto, os docentes de ACS representados pelo gestor do projeto o Prof. António Esteves.
