



LEIC-A, LEIC-T, LETI, MEIC-T, MEIC-A

## Engenharia de Software

2º Semestre – 2014/2015

### Enunciado da Primeira Parte do Projecto

#### 1. Primeira Parte do Projecto ES

Este enunciado descreve o trabalho a realizar pelos alunos para a primeira parte do projecto. Para a primeira parte do projecto pretende-se:

- Concretizar as entidades do domínio do problema apresentado no enunciado geral utilizando o mecanismo de persistência de objectos exposto nas aulas de laboratório. Para isso, deve ser definido um ficheiro dml que descreva a estrutura das entidades do domínio. Esta modelação e concretização parcial deverá considerar todas as entidades do domínio descritas (excluindo as relacionadas com o conceito de intervalo e função de intervalo ) incluindo todos os atributos existentes, as associações entre estas entidades e as possíveis relações de herança que existam.

**Nota:** Por razões de simplificação, não é necessário considerar ainda nem a modelação nem a concretização das entidades que deverão suportar interacção com os serviços externos, nomeadamente a autenticação dos utilizadores e o armazenamento fiável de documentos. No entanto, e por forma a preparar já a solução para a utilização do serviço externo SD-STORE, todos os grupos deverão concretizar as operações de importação e exportação do conteúdo de uma folha de cálculo.

##### 1.1. Regras de Negócio

Além da definição das entidades e relações do domínio da aplicação descritas no enunciado geral do projecto conjunto de Engenharia de Software e Sistemas Distribuídos, cada grupo deverá concretizar a seguinte funcionalidade no domínio da aplicação:

- Concretização das operações de importação e exportação de uma folha de cálculo para XML. Estas operações são descritas com detalhe na secção 1.1.1.
- Dado um *username*, aceder ao utilizador da aplicação com esse nome. Esta funcionalidade deve estar concretizada na classe que contém a informação sobre os utilizadores registados na aplicação.

- Dado uma cadeia de caracteres *str* e um utilizador, devolver a lista de folhas de cálculo cujo nome é igual a *str* e que foram criadas pelo utilizador em causa. Esta funcionalidade deve estar concretizada na classe do domínio que representa os utilizadores.
- Remover uma folha de cálculo do estado persistente da aplicação. Esta remoção implica também remover do estado persistente todos os objectos relacionados com o conteúdo actual da folha de cálculo.

A primeira parte do projecto não requer a concretização da lógica de negócio responsável por determinar o valor de uma da célula da folha de cálculo, apenas é necessário a definição das entidades e a sua correcta inicialização. No que diz respeito às funções intervalo e à entidade intervalo, não é necessário sequer concretizar já estas classes.

Só as entidades *Utilizador* e *Documento* é que realmente necessitam de um identificador único. Não é necessário associar um atributo único para as restantes entidades do domínio da aplicação.

### 1.1.1 Importação e Exportação de uma Folha de Cálculo

O formato utilizado nas operações de importação e exportação para representar o conteúdo de uma folha de cálculo é o XML. Assim, o serviço externo SD-STORE irá guardar documentos XML que representam o conteúdo de folhas de cálculo. Para transformar objectos do domínio em XML e vice-versa deve-se utilizar a biblioteca JDOM (<http://www.jdom.org/>). Esta biblioteca permite ler, escrever e modificar documentos XML.

A operação de exportação corresponderá a converter o conteúdo de uma folha de cálculo num documento XML. A estrutura do documento XML a criar deverá corresponder tanto quanto possível ao grafo de objectos formado por uma folha de cálculo. Assim, a criação do documento XML deverá seguir a seguinte descrição:

- Cada objecto referenciado numa folha de cálculo é representado por uma instância da classe *Element* da biblioteca JDOM;
- A raiz do documento XML a criar é representada por uma instância de *Element* correspondente ao objecto que representa a folha de cálculo a converter;
- Os atributos de cada objecto a converter são guardados como atributos do elemento XML que representa o objecto utilizando o método *setAttribute* da classe *Element*.
- Cada relação de um para um presente num objecto a converter é representada por uma instância de *Element* que irá descrever o objecto referenciado na relação. Esta instância de *Element* deverá ser adicionada como um sub-elemento do elemento XML que representa o objecto a converter. A adição deverá ser realizada utilizando o método *addContent* de *Element*.
- Cada relação de um para muitos de um objecto a converter deve ser representada por uma instância de *Element*. Este elemento XML deve ser guardado como um sub-elemento do elemento que representa o objecto a converter. Cada objecto pertencente à relação deve ser convertido numa instância de *Element* e adicionado como sub-elemento do elemento XML que representa a associação.

Por forma a simplificar o processo de importação e exportação, não é necessário guardar a informação sobre os utilizadores que podem aceder à folha de cálculo, mas deve ser guardado o identificador do utilizador que criou a folha de cálculo. A operação de importação

deverá receber um documento XML (representado por uma instância de `org.jdom2.Document`) e o identificador de um utilizador e apenas realizará a importação caso o identificador do dono da folha de cálculo guardado no documento XML seja igual ao recebido como argumento. A folha criada numa operação de importação com sucesso fica apenas com um utilizador com direitos de escrita, o dono da folha. A operação de importação não substitui nenhuma folha de cálculo que exista no estado persistente da aplicação. Esta operação cria uma nova folha de cálculo (com um novo identificador) cujo conteúdo corresponde ao representado pelo documento XML.

Na página da cadeira relativa ao guia de laboratório da segunda aula foi disponibilizado uma versão da aplicação PhoneBook que concretiza as operações de importação e exportação para XML com uma funcionalidade semelhante à pretendida para este projecto.

**Nota:** Para utilizar a biblioteca JDOM no seu projecto Maven é necessário indicar que o seu projecto depende agora desta biblioteca. Assim, deve editar a secção `<dependencies>` do ficheiro POM e acrescentar a seguinte dependência:

```
<dependency>
  <groupId>org.jdom</groupId>
  <artifactId>jdom2</artifactId>
  <version>2.0.5</version>
</dependency>
```

## 1.2. Estado Inicial da Base de Dados

Para se poder realizar a visualização da primeira entrega, é necessário ter um cenário de teste inserido na base de dados. O cenário de teste é o seguinte:

1. Utilizador com o username *pf*, palavra chave *sub* e nome *Paul Door*.
2. Utilizador com o username *ra*, palavra chave *cor* e nome *Step Rabbit*.
3. Uma folha de cálculo com 300 linhas e 20 colunas, com o nome *Notas ES* e criada pelo utilizador *pf*.
4. Esta folha deverá ter o seguinte conteúdo:
  - Literal 5 na posição (3, 4)
  - Referência para a célula (5, 6) na posição (1, 1)
  - Função = *ADD*(2, 3; 4) na posição (5, 6)
  - Função = *DIV*(1; 1, 3; 4) na posição (2, 2)
  - As restantes posições desta folha de cálculo não estão preenchidas.

Os alunos, devem concretizar a classe `pt.tecnico.bubbledocs.BubbleApplication` que deverá ter o método `main` com a seguinte funcionalidade:

- Inserir os dados relativos ao cenário de teste na base de dados caso o estado persistente da aplicação não esteja inicializado.
- Escrever a informação sobre todos os utilizadores registados na aplicação.

- Escrever o nomes de todas as folhas de cálculo dos utilizadores *pf* e *ra*.
- Aceder às folhas de cálculo do utilizador *pf*. Utilizando a funcionalidade de exportação, converte cada folha de cálculo para o formato XML e escreve no terminal o resultado desta conversão.
- Remover do estado persistente da aplicação a folha de cálculo *Notas ES* do utilizador *pf*.
- Escrever os nomes e identificadores de todas as folhas de cálculo do utilizador *pf*.
- Utilizar a funcionalidade de importação para criar uma folha de cálculo semelhante à que foi exportada anteriormente e que foi removida agora.
- Escrever os nomes e identificadores de todas as folhas de cálculo do utilizador *pf*.
- Aceder às folhas de cálculo do utilizador *pf*. Utilizando a funcionalidade de exportação, converte cada folha de cálculo para o formato XML e escreve no terminal o resultado desta conversão.

A execução deste método deve ficar atribuída ao objectivo *exec:java* do projecto Maven.

**Notas:**

A listagem de todos os utilizadores deve incluir o valor dos seus atributos (username, nome e password).

A listagem de uma folha de cálculo em formato XML deve ser feita utilizando a classe `XMLOutputter` presente na biblioteca `JDOM`.

Cada uma das acções descritas deve ser feita numa transacção independente.

As excepções que possam ocorrer durante a execução das acções descritas no método `main` devem ser tratadas correctamente.

## 2. Realização da Primeira Parte do Projecto

O corpo docente de Engenharia de Software prevê que a realização desta parte do projecto exigirá cerca de 5 horas de trabalho a cada aluno do grupo. Nesta previsão, o corpo docente assume que os alunos já perceberam o funcionamento da Fénix Framework antes da realização do trabalho e que existe um planeamento do projecto por forma a perceber o trabalho a realizar e a distribuir o trabalho pelos vários elementos do grupo.

Na realização do projecto, aconselha-se que cada grupo siga o seguinte o planeamento:

1. Cada elemento do grupo deve realizar o diagrama de classes UML do domínio do projecto e representar este diagrama utilizando a linguagem DML.
2. O grupo deve reunir-se e comparar os diferentes diagramas de classes dos vários elementos do grupo e concluir qual deve ser o diagrama de classes do domínio do projecto e o respectivo ficheiro DML que representa este domínio.
3. Criar um projecto Maven para a aplicação a desenvolver. Este projecto deverá ter uma estrutura semelhante ao descrito para a aplicação *PhoneBook* apresentada no segundo laboratório da cadeira. O ficheiro DML criado no ponto anterior deve ser colocado no directório correcto (*src/main/dml*).

4. Colocar o ficheiro *fenix-framework-jvstm-obj.properties* no directório *src/main/resources*. Este ficheiro deve definir os dados a utilizar na ligação ao sistema de bases de dados (ver secção 3).
5. Compilar o projecto por forma a ver se está tudo correcto.
6. Distribuir as regras de negócio, definição do estado inicial do sistema e funcionalidade do método *main* da classe *pt.tecnico.bubbledocs.BubbleApplication* pelos vários elementos do grupo.
7. Testar o projecto.

Note-se que este planeamento apenas tem a indicação das tarefas a realizar. É da responsabilidade do grupo fazer a distribuição temporal das tarefas pelo tempo disponível para realizar o projecto. Esta distribuição também corresponde a uma tarefa e deve ser a primeira tarefa a realizar pelo grupo e deve ser feita o mais cedo possível.

### 3. Entrega da Primeira Parte Projecto

O prazo de entrega da primeira parte do projecto é até ao início da **quarta** aula de laboratório do grupo de **11 a 17 de Março de 2015**. O projecto a realizar deve apresentar uma estrutura de directórios semelhante à da aplicação *PhoneBook* apresentada durante as aulas de laboratório.

O código produzido deve ser guardado no repositório Git de cada grupo. Cada grupo, após ter concretizado esta parte do projecto e ter guardado no seu repositório o código respectivo, deverá criar a *tag* *R\_1*. Esta *tag* representará a versão do código produzido para esta parte do projecto que os alunos querem submeter a avaliação.

Para facilitar a execução do código entregue, os grupos **têm** que utilizar os seguintes dados para a definição da ligação à base de dados:

- **username:** *bubble*
- **password:** *bubbl3*
- **base de dados:** *bubbledb*

#### 3.1. Penalizações

Projectos que guardem ficheiros desnecessários no repositório terão uma penalização na nota de 2 a 4 valores. Consideram-se desnecessários os ficheiros *.class* gerados na compilação das classes Java, os ficheiros *\_Base.java* automaticamente gerados na compilação da DML, ou ficheiros *.jar*. Para isso, deverão criar o ficheiro *.gitignore* no directório base do projecto. Este ficheiro deverá indicar que o directório *target* não deve ser colocado no sistema de controlo de versões Git.