

— Teste —
Desenvolvimento de Sistemas Software

LEI 2011/12

12/01/2012

Duração máxima: 2h00

Leia o exame com atenção e responda a cada grupo em folhas separadas!

Grupo I – Modelação de requisitos

Considere a seguinte proposta para o Use Case “Levantar Dinheiro” de uma máquina ATM.

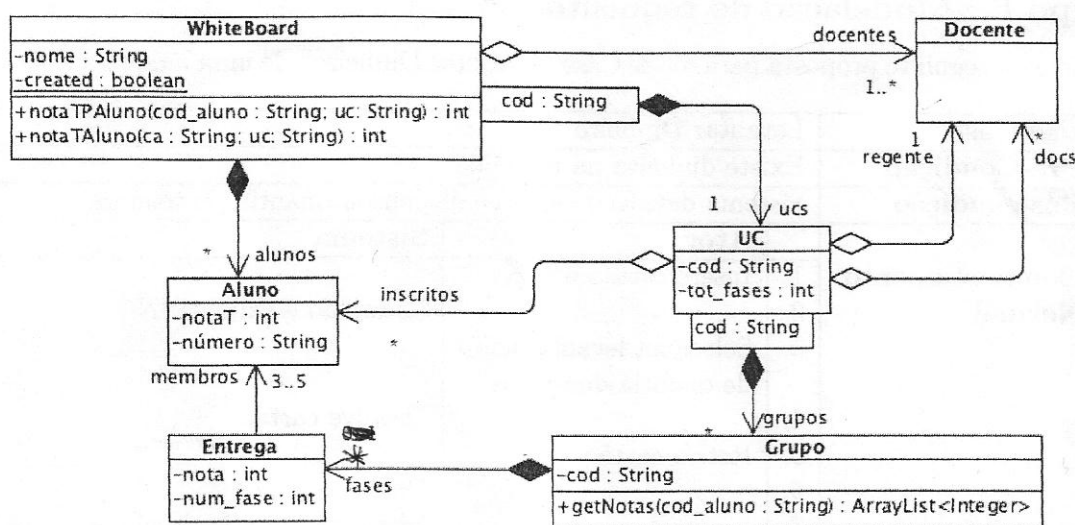
Use Case	Levantar Dinheiro		
Pré-Condição	Existe dinheiro na máquina		
Pós-condição	Sistema devolveu cartão e dispensou quantia pretendida		
Comportamento Normal		Ator	Sistema
	1.	Inserir cartão e PIN	Lê cartão e valida PIN
	2.		
	3.	Seleciona levantamento de quantia desejada	
	4.		Devolve cartão
	5.	Retira cartão	
	6.		Debita quantia Dispensa quantia desejada e imprime recibo
	7.		
	8.	Retira quantia	
Alternativa 1 (passo 7) [não existe papel]	9.		Fecha sessão
	1.		Dispensa quantia desejada
Exceção 1 (passo 7) [não existe quantia]	2.		Regressa a 8
	1.		Avisa que levantamento é impossível
	2.		Fecha sessão

1. Reescreva o Use Case sabendo que se pretendem adicionar as seguintes características:
 - (a) Se o cartão não for legível é devolvido e a sessão termina.
 - (b) Só é possível efectuar o levantamento se o PIN for válido — após três tentativas consecutivas com PIN inválido o cartão é retido (note que as tentativas não têm necessariamente que acontecer na mesma máquina).
 - (c) Quando não existe papel, deve ser dada ao utilizador a possibilidade de continuar, ou não, a operação.
2. Desenhe agora o **Diagrama de Sequência de Sistema** correspondente ao Use Case apresentado.

(Não se esqueça de responder a cada grupo em folhas separadas!)

Grupo II – Modelação de Arquitectura / Comportamento

Considere a proposta de arquitectura apresentada na figura:



3. Com base na arquitectura proposta (e sabendo que as associações qualificadas são implementadas com Map e as coleções com List), escreva o código Java para a classe WhiteBoard
4. Sabendo ainda que
 - (a) a nota prática é a média das notas em cada entrega do trabalho
 - (b) em cada UC, tot_fases diz quantas fases o trabalho teve

(c) entre fases os alunos podem ter mudado de grupo

desenhe um **Diagrama de Sequência** para o método `int notaTPAluno(cod_aluno: String, uc: String)` (da classe `WhiteBoard`) que calcula a nota prática de um aluno numa dada UC.

5. Sabendo que na janela da aplicação `WhiteBoard` relativa à consulta de alunos se pretende que:

- existam duas listas, uma de UCs (sempre com todas as UCs) e outra de alunos (inicialmente vazia);
- a lista de alunos deverá ser actualizada sempre que o utilizador escolher uma UC;
- caso a UC seleccionada tenha alunos inscritos, a lista de alunos apresenta-os, caso contrário, fica vazia;
- quando o utilizador selecciona um aluno na lista, ficam disponíveis duas opções: consultar e remover;
- caso opte por consultar, é apresentada uma nova janela com a informação desse aluno (e que, para continuar a utilizar a aplicação, o utilizador tem que fechar esta janela);
- caso opte por remover, o aluno é removido da UC e a lista actualizada.

e sabendo ainda que

- o método para obter a lista de UCs é: `List<UC> getUCs()`
- o método para obter os alunos de uma UC é: `List<Aluno> getAlunos(String uc)`
- o método para obter a informação de um aluno é: `Aluno getAluno(String cod_aluno)`
- o método para remover um aluno de uma UC é: `void delAluno(String cod_aluno, String uc)`

Desenhe um **Diagrama de Estados** que modele a sua proposta de interface para este aspecto do sistema. O diagrama deverá identificar os écrans da aplicação e seus estados relevantes, as transições entre eles e quando são invocados os métodos da lógica de negócio mencionados.