

PROJETO #1 ENGENHARIA DE SOFTWARE

TEST DESIGN SPECIFICATION

Version 1.5
24/11/2019

Histórico de Versões

Version #	Implemented By	Revision Date	Approved By	Approval Date	Reason
1.0	Catarina	22/11/19	Rebaca Teixeira	23/11/19	Preenchimento ponto 2. Introdução e colocação das tabelas do ponto 3.
1.1	Rebeca Teixeira	23/11/2019	Filipe	23/11/19	Reestruturação do documento. Preenchimento do ponto 4.
1.2	Filipe	23/11/2019	Catarina	23/11/19	Completar a identificação de testes
1.3	Catarina	23/11/2019	Rebeca	24/11/2019	Reestruturação do documento
1.4	Catarina	24/11/19	Rebeca	24/11/2019	Colocação das tabelas BVA Preenchimento do âmbito
1.5	Rebeca	24/11/2019	Todo o grupo	24/11/2019	Reestruturação do documento e preenchimento dos últimos dois pontos

Tabela de Conteúdos

1. Introdução.....	5
1.1. Identificador do documento.....	5
1.2. Âmbito.....	5
1.3. Referências	5
1.4. Glossário	5
2. Features/Itens a testar.....	6
2.1. Diagrama Use Case Test.....	6
Adicionar questão.....	7
Obter questão.....	7
Obter estatísticas do teste.....	7
Verificar se o teste está completo.....	8
Carregar questões de um ficheiro JSON	8
Número de questões de um teste	8
Remover questão de um teste pela posição.....	9
Remover a primeira ocorrência de uma questão	9
Salvar resultados	9
2.2 Diagrama Use Case TestStatistics	10
Calcular o número de respostas corretas.....	11
Ver percentagem de respostas corretas	11
Calcular o número de respostas incorretas.....	12
Ver percentagem de respostas incorretas	12
Ver respostas corretas.....	13
Ver respostas incorretas.....	13
Calcular o tempo de resposta para cada questão	13
Ver o número total de questões	14
Calcular o desvio padrão do tempo de resposta	14
2.3 Diagrama UseCase questionMultipleChoice.....	15
Definir a resposta de uma pergunta	16
Avaliar a resposta de uma questão	16
Obter a resposta correta de uma questão.....	16
Obter as opções de uma questão	17
Obter a resposta do utilizador a uma questão.....	17
Definir as opções de uma pergunta	17
Definir a resposta de um utilizador a uma pergunta	17
Definir a resposta correta de uma questão	18
2.4 Diagrama UseCase NumericQuestion	19
Definir a resposta de uma questão.....	20
Avaliar a resposta de uma pergunta	20
Obter a resposta correta de uma questão.....	20
Obter a resposta do utilizador a uma questão.....	21
Definir a resposta correta de uma questão	21
Definir a resposta do utilizador de uma questão.....	21
2.5 Diagrama UseCase QuestionYesNo.....	22
Definir a resposta de uma questão.....	23
Avaliar a resposta de uma pergunta	23
Obter a resposta correta de uma questão.....	23
Obter a resposta do utilizador a uma questão.....	23
Definir a resposta correta de uma questão	24
Definir a resposta do utilizador de uma questão.....	24
2.6 Itens a testar.....	25

3. Detalhes da abordagem aos testes	27
Classes de equivalência	27
• Test.AddQuestion()	28
• Test.RemoveQuestion()	28
• Test.RemoveQuestion() (Remover questão pela posição da mesma)	29
• Test.numberQuestions()	29
• Test.isComplete()	30
• TestStatistics.correctAnswers()	30
• TestStatistics.incorrectAnswers()	31
• TestStatistics.correctAnswerPercentage()	32
• TestStatistics.incorrectAnswerPercentage()	32
• QuestionMultipleChoice.answer()	33
• QuestionMultipleChoice.evaluateAnswer()	33
• QuestionNumeric.answer()	34
• QuestionNumeric.evaluateAnswer()	34
• QuestionYesNo.answer()	35
• QuestionYesNo.evaluateAnswer()	35
4. Identificação dos Testes.....	37
5. Critérios de passagem ou falha das features	37

1. Introdução

1.1. Identificador do documento

TCS-01

1.2. Âmbito

No Âmbito da disciplina de Engenharia de software foi-nos solicitado que elaborássemos um conjunto de casos de teste seguindo uma perspetiva black-box, usando as técnicas ECP e BVA, de modo a verificar o comportamento o programa, fornecido pelos docentes da unidade curricular, e os compoentes que definem os use cases disponíveis para o utilizador.

O sistema deve permitir adicionar e remover questões, definir uma resposta, ver as respostas certas e erradas e também deve permitir ver a percentagem de respostas erradas e corretas. Os componentes *Test*, *TestStatistics*, *QuestionNumeric*, *QuestionMultipleChoice* e *QuestionYesNo* serão alvos de teste.

1.3. Referências

É recomendada a leitura do documento TCO-01 onde estão especificados todos os casos de teste.

O código dos testes pode ser consultado no seguinte repositório:
<https://gitlab.estg.ipp.pt/8170274/1stprojectesii>

1.4. Glossário

ECP - Equivalence Class Partitioning

BVA - Boundary Value Analysis

BLB – Below Lower Bound

LB – Lower Bound

ALB – Above Lower Bound

BUB – Below Upper Bound

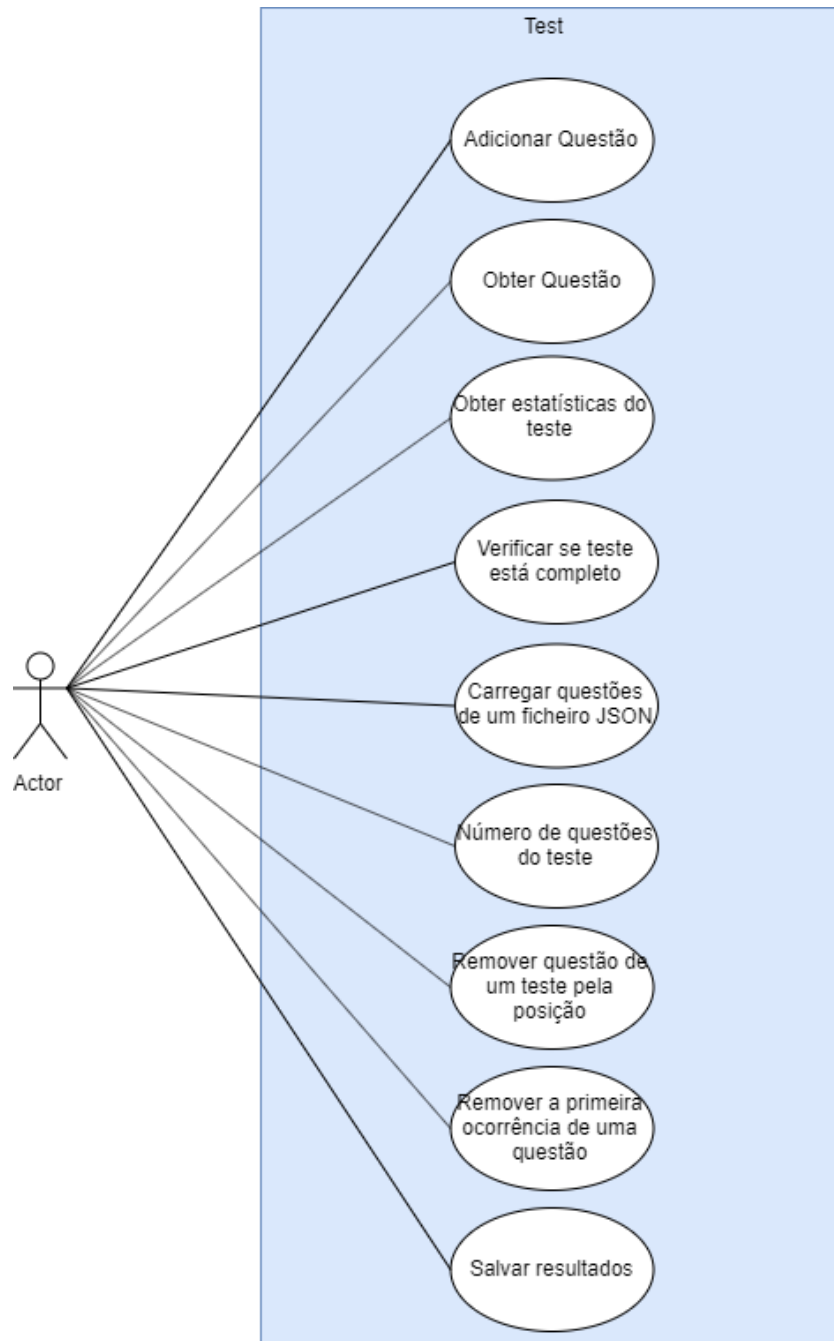
UB – Upper Bound

AUB – Above Upper Bound

2. Features/Itens a testar

Nesta secção irão ser apresentadas as diversas funcionalidades do programa.

2.1. Diagrama Use Case Test



Adicionar questão

Use Case Name	Adicionar questão
Short Description	O professor tem a possibilidade de adicionar questões a um teste
Actors	Professor
Pre-conditions	Existência de uma coleção de perguntas instanciadas
Typical Flow	1. Adicionar a questão num ficheiro previamente criado 2. Carregar as questões para o programa
Alternative Flow	N.A
Use Case Extensions	N.A

Obter questão

Use Case Name	Obter questão
Short Description	O utilizador tem a possibilidade de ver uma questão de um teste
Actors	Professor
Pre-conditions	Existência de uma coleção de perguntas instanciadas

Obter estatísticas do teste

Use Case Name	Obter estatísticas do teste
Short Description	O professo pode consultar as estatísticas de um teste
Actors	Professor
Pre-conditions	Existência de uma coleção de perguntas instanciadas
Typical Flow	1. O utilizador efetua o teste
Alternative Flow	N.A
Use Case Extensions	N.A
Post-Conditions	Quando o teste estiver concluído aparecerá todas as estatísticas.

Verificar se o teste está completo

Use Case Name	Verificar se o teste está completo
Short Description	O professor tem a possibilidade de verificar se todas as questões de um teste estão realizadas (done)
Actors	Professor
Pre-conditions	Existência de uma coleção de perguntas instanciadas
Typical Flow	1. Inicia o teste
Alternative Flow	N.A
Use Case Extensions	N.A
Post-Conditions	Caso o utilizador tente dar Finish sem todas as perguntas respondidas, isto não é possível.

Carregar questões de um ficheiro JSON

Use Case Name	Carregar questões de um ficheiro JSON
Short Description	O professor tem a possibilidade de carregar todas as questões de um teste de um ficheiro de texto
Actors	Professor
Pre-conditions	O ficheiro tem de conter perguntas e tem que ser do formato JSON.
Typical Flow	1. Iniciar o teste
Alternative Flow	N.A
Use Case Extensions	N.A
Post-Conditions	O ficheiro json será carregado.

Número de questões de um teste

Use Case Name	Número de questões de um teste
Short Description	O professor tem a possibilidade de consultar o número total de questões de um teste
Actors	Professor
Pre-conditions	Existência de uma coleção de perguntas instanciadas

Remover questão de um teste pela posição

Use Case Name	Remover questões de um teste pela posição
Short Description	O professor pode remover uma questão de um teste pela posição específica que esta ocupa
Actors	Professor
Pre-conditions	Existência de uma coleção de perguntas instanciadas

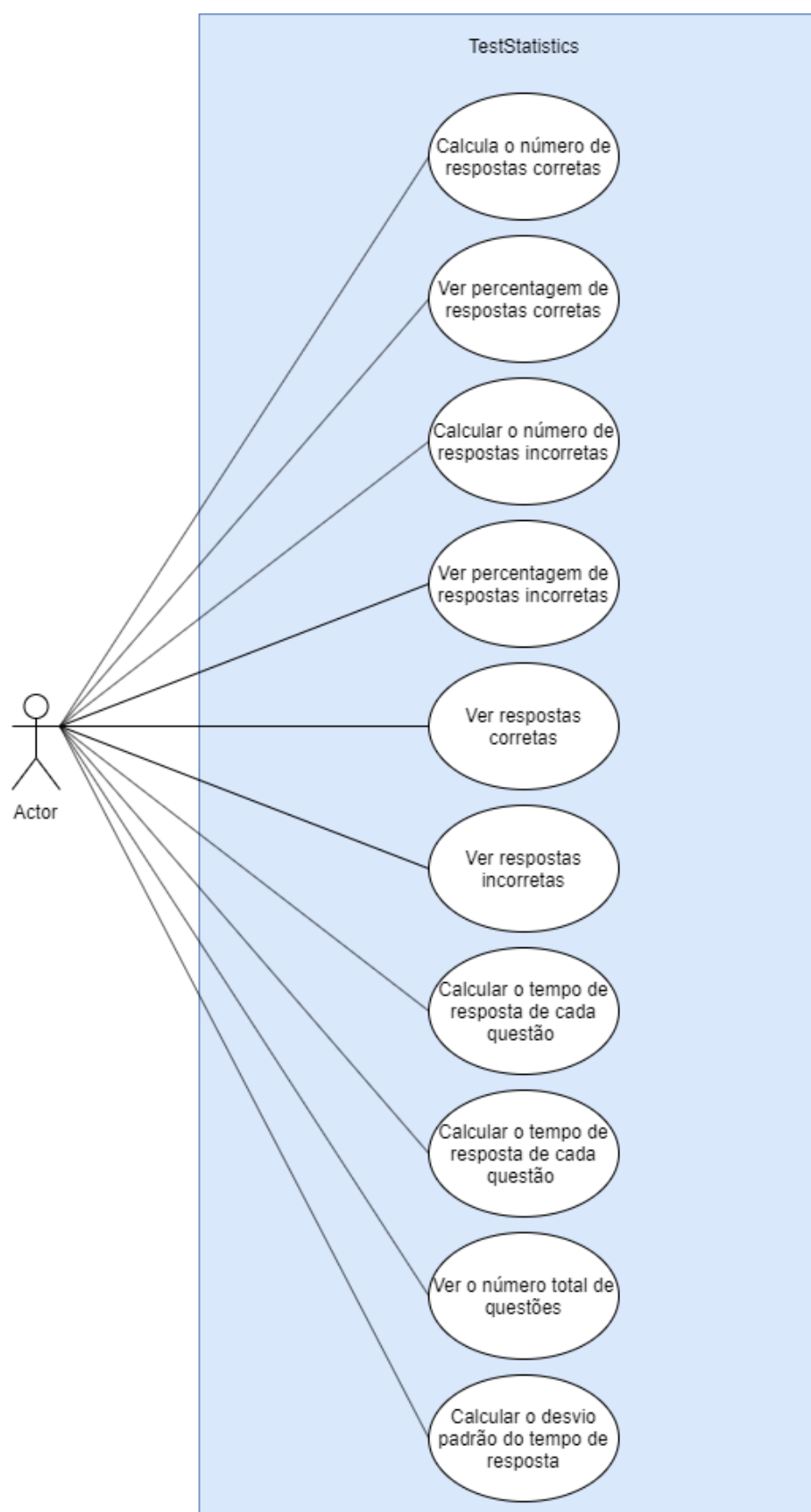
Remover a primeira ocorrência de uma questão

Use Case Name	Remover a primeira ocorrência de uma questão
Short Description	O professor pode remover a primeira ocorrência de uma pergunta da estrutura de dados usada para armazenar as perguntas
Actors	Professor
Pre-conditions	Existência de uma coleção de perguntas instanciadas

Salvar resultados

Use Case Name	Salvar resultados
Short Description	O professor pode guardar os resultados
Actors	Professor, Utilizador
Pre-conditions	Existência de uma coleção de perguntas instanciadas
Typical Flow	<ol style="list-style-type: none">1. Iniciar o teste2. Concluir o teste
Alternative Flow	N.A
Use Case Extensions	N.A
Post-Conditions	O utilizador terá a possibilidade de guardar os resultados

2.2 Diagrama Use Case TestStatistics



Calcular o número de respostas corretas

Use Case Name	Calcular o número de respostas corretas
Short Description	Os utilizadores têm a possibilidade de consultar o número de respostas corretas
Actors	Professor e alunos
Pre-conditions	Existência de um teste completo com todas as questões avaliadas
Typical Flow	<ol style="list-style-type: none">1. Iniciar o teste2. Concluir o teste
Alternative Flow	N.A
Use Case Extensions	N.A
Post-Conditions	O utilizador no final do teste poderá ver o número de respostas corretas

Ver percentagem de respostas corretas

Use Case Name	Ver percentagem de respostas corretas
Short Description	Os utilizadores têm a possibilidade de consultar a percentagem de respostas corretas
Actors	Professor e alunos
Pre-conditions	Existência de um teste completo com todas as questões avaliadas
Typical Flow	<ol style="list-style-type: none">1. Iniciar o teste2. Concluir o teste
Alternative Flow	N.A
Use Case Extensions	
Post-Conditions	O utilizador no final do teste poderá ver a percentagem de perguntas corretas

Calcular o número de respostas incorretas

Use Case Name	Calcular o número de respostas incorretas
Short Description	Os utilizadores têm a possibilidade de consultar o número de respostas incorretas
Actors	Professor e alunos
Pre-conditions	Existência de um teste completo com todas as questões avaliadas
Typical Flow	<ol style="list-style-type: none">1. Iniciar o teste2. Concluir o teste
Alternative Flow	N.A
Use Case Extensions	N.A
Post-Conditions	O utilizador no final do teste poderá ver o número de respostas incorretas

Ver percentagem de respostas incorretas

Use Case Name	Ver percentagem respostas incorretas
Short Description	Os utilizadores têm a possibilidade de consultar a percentagem do número de respostas incorretas
Actors	Professor e alunos
Pre-conditions	Existência de um teste completo com todas as questões avaliadas
Typical Flow	<ol style="list-style-type: none">1. Iniciar o teste2. Concluir o teste
Alternative Flow	N.A
Use Case Extensions	N.A
Post-Conditions	O utilizador no final do teste poderá ver a percentagem de perguntas incorretas

Ver respostas corretas

Use Case Name	Ver respostas corretas
Short Description	Os utilizadores têm a possibilidade de consultar as respostas corretas
Actors	Professor e alunos
Pre-conditions	Existência de um teste completo com todas as questões avaliadas
Typical Flow	<ol style="list-style-type: none">1. Iniciar o teste2. Concluir o teste
Alternative Flow	N.A
Use Case Extensions	N.A
Post-Conditions	O utilizador poderá ver as perguntas que acertou

Ver respostas incorretas

Use Case Name	Ver respostas incorretas
Short Description	Os utilizadores têm a possibilidade de consultar as respostas incorretas
Actors	Professor e alunos
Pre-conditions	Existência de um teste completo com todas as questões avaliadas
Typical Flow	<ol style="list-style-type: none">1. Iniciar o teste2. Concluir o teste
Alternative Flow	N.A
Use Case Extensions	N.A
Post-Conditions	O utilizador poderá ver as perguntas que errou

Calcular o tempo de resposta para cada questão

Use Case Name	Calcular o tempo de resposta para cada questão
Short Description	Os utilizadores podem saber quanto tempo demoraram a responder a uma determinada questão
Actors	Professor e alunos
Pre-conditions	Existência de um teste completo com todas as questões avaliadas
Typical Flow	<ol style="list-style-type: none">1. Iniciar o teste2. Concluir o teste

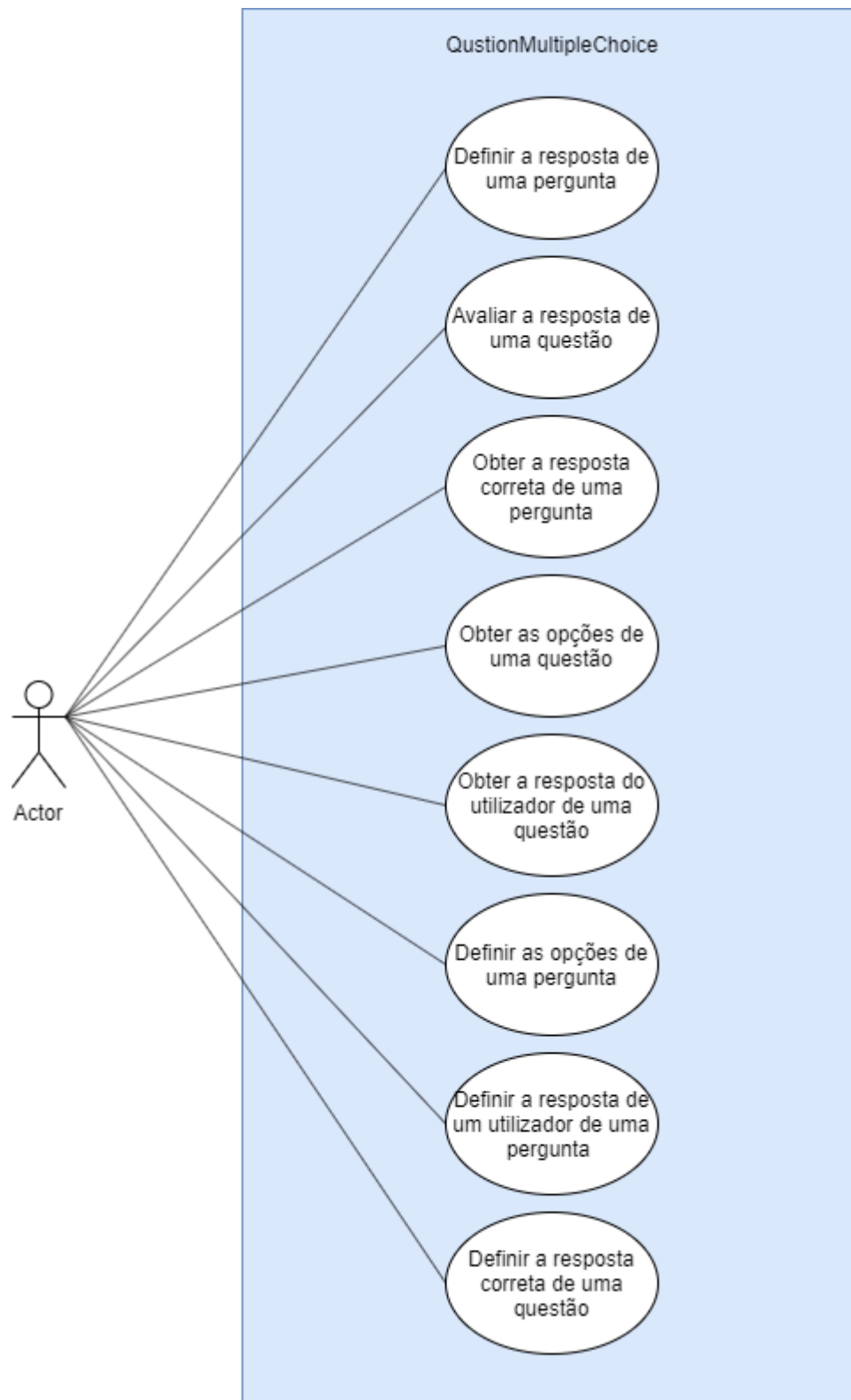
Ver o número total de questões

Use Case Name	Ver o número total de questões
Short Description	Os utilizadores têm a possibilidade de consultar o número total de questões
Actors	Professor e alunos
Pre-conditions	Existência de um teste com perguntas não nulas

Calcular o desvio padrão do tempo de resposta

Use Case Name	Calcular o desvio padrão do tempo de resposta
Short Description	O utilizador pode consultar o desvio padrão do tempo de resposta
Actors	Professor e alunos
Pre-conditions	Existência de um teste completo
Typical Flow	<ol style="list-style-type: none">1. Iniciar o teste2. Concluir o teste
Alternative Flow	N.A
Use Case Extensions	NA
Post-Conditions	No final do teste o utilizador pode ver o desvio padrão do tempo de resposta

2.3 Diagrama UseCase questionMultipleChoice



Definir a resposta de uma pergunta

Use Case Name	Definir a resposta de uma pergunta
Short Description	O utilizador pode definir a resposta a uma determinada pergunta
Actors	Professor e alunos
Pre-conditions	Existência de uma coleção de perguntas
Typical Flow	1. Iniciar o teste
Alternative Flow	N.A
Use Case Extensions	N.A
Post-Conditions	O utilizador pode definir uma resposta para uma pergunta

Avaliar a resposta de uma questão

Use Case Name	Avaliar a resposta de uma questão
Short Description	A resposta do aluno é avaliada
Actors	Professor
Pre-conditions	Existência de uma resposta

Obter a resposta correta de uma questão

Use Case Name	Obter a resposta correta de uma questão
Short Description	A resposta correta de uma determinada questão pode ser consultada sendo que o aluno só pode consultar no final do teste
Actors	Professor, Aluno
Pre-conditions	Existência de uma resposta correta associada a cada pergunta
Typical Flow	1. Iniciar o teste
Alternative Flow	N.A
Use Case Extensions	N.A
Post-Conditions	O utilizador depois de responder à questão pode ver a resposta correta à mesma

Obter as opções de uma questão

Use Case Name	Obter as opções de uma questão
Short Description	O utilizador pode consultar as opções associadas a determinada pergunta
Actors	Professor e alunos
Pre-conditions	Cada pergunta tem de ter um conjunto de opções associado
Typical Flow	1. Iniciar o teste
Alternative Flow	N.A
Use Case Extensions	N.A
Post-Conditions	O utilizador tem acesso às opções de uma questão

Obter a resposta do utilizador a uma questão

Use Case Name	Obter a resposta do utilizador a uma questão
Short Description	O utilizador pode consultar as opções associadas a determinada pergunta
Actors	Professor e alunos
Pre-conditions	Cada pergunta tem de ter um conjunto de opções associado
Typical Flow	1. Iniciar o teste

Definir as opções de uma pergunta

Use Case Name	Definir as opções de uma pergunta
Short Description	O professor tem de definir as opções que estão associadas a cada pergunta
Actors	Professor
Pre-conditions	Existência de uma coleção de perguntas

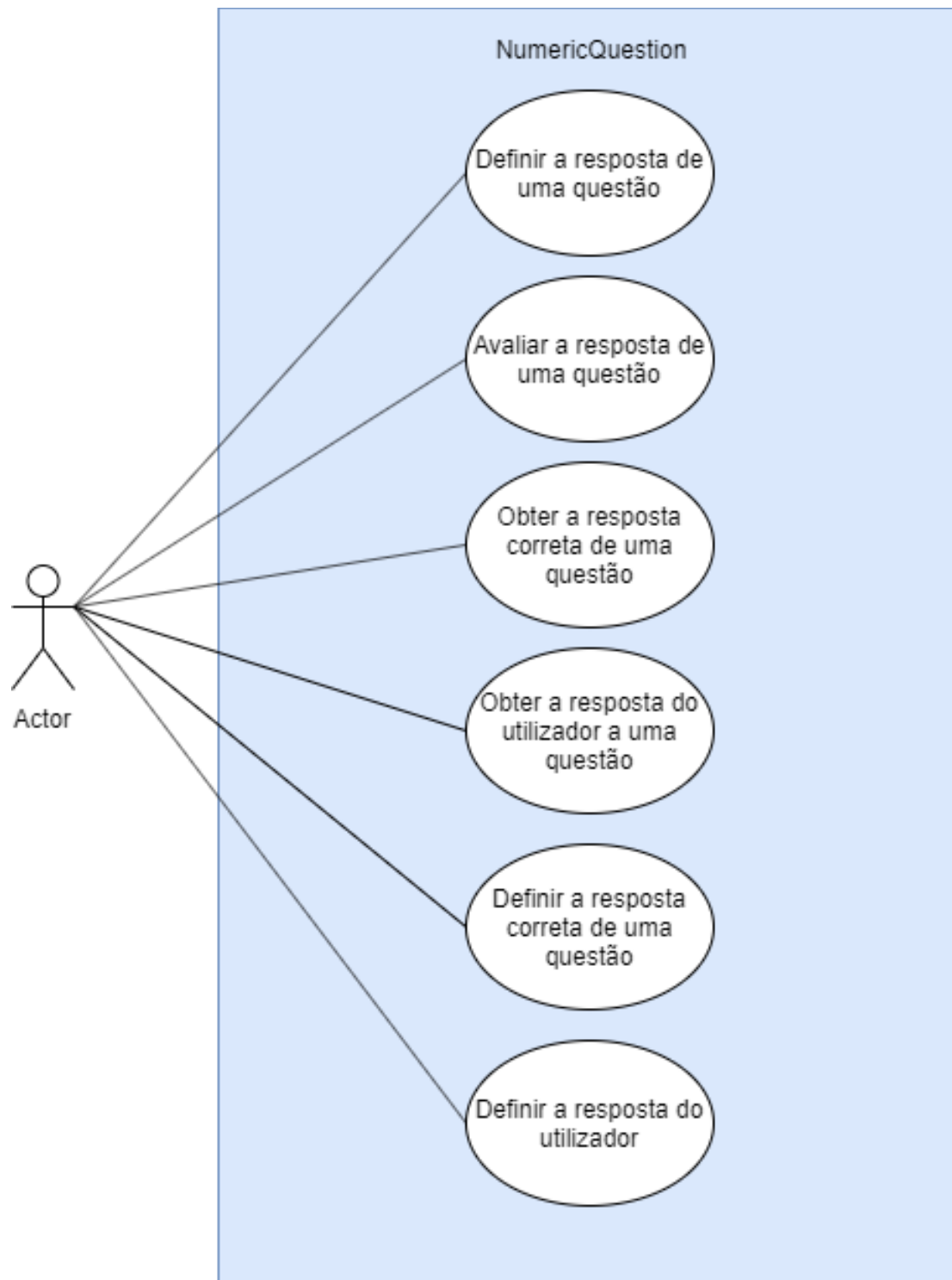
Definir a resposta de um utilizador a uma pergunta

Use Case Name	Definir a resposta de um utilizador a uma pergunta
Short Description	A resposta do utilizador é associada a uma pergunta
Actors	Professor, Aluno
Pre-conditions	Existência de uma coleção de perguntas
Typical Flow	1. Iniciar o teste
Alternative Flow	N.A
Use Case Extensions	N.A
Post-Conditions	A resposta fica definida quando o utilizador introduz uma resposta

Definir a resposta correta de uma questão

Use Case Name	Definir a resposta correta de uma questão
Short Description	O professor tem de associar uma resposta correta para cada questão
Actors	Professor
Pre-conditions	Existência de uma coleção de perguntas

2.4 Diagrama UseCase NumericQuestion



Definir a resposta de uma questão

Use Case Name	Definir a resposta de uma questão
Short Description	O utilizador pode definir a resposta a uma determinada pergunta
Actors	Professor e alunos
Pre-conditions	Existência de uma coleção de perguntas
Typical Flow	1. Iniciar o teste
Alternative Flow	N.A
Use Case Extensions	N.A
Post-Conditions	A resposta fica definida quando o utilizador responde a uma questão

Avaliar a resposta de uma pergunta

Use Case Name	Avaliar a resposta de uma pergunta
Short Description	A resposta do aluno é avaliada
Actors	Professor
Pre-conditions	Existência de uma resposta

Obter a resposta correta de uma questão

Use Case Name	Obter a resposta correta de uma questão
Short Description	A resposta correta de uma determinada questão pode ser consultada sendo que o aluno só pode consultar no final do teste
Actors	Professor, Aluno
Pre-conditions	Existência de uma resposta correta associada a cada pergunta
Typical Flow	1. Iniciar o teste
Alternative Flow	N.A
Use Case Extensions	N.A
Post-Conditions	O utilizador pode ver a resposta correta quando responde a uma questão

Obter a resposta do utilizador a uma questão

Use Case Name	Obter a resposta do utilizador a uma questão
Short Description	O utilizador pode consultar a resposta dada pelo aluno a uma determinada questão
Actors	Professor e alunos
Pre-conditions	Existência de uma resposta do utilizador associada a uma pergunta

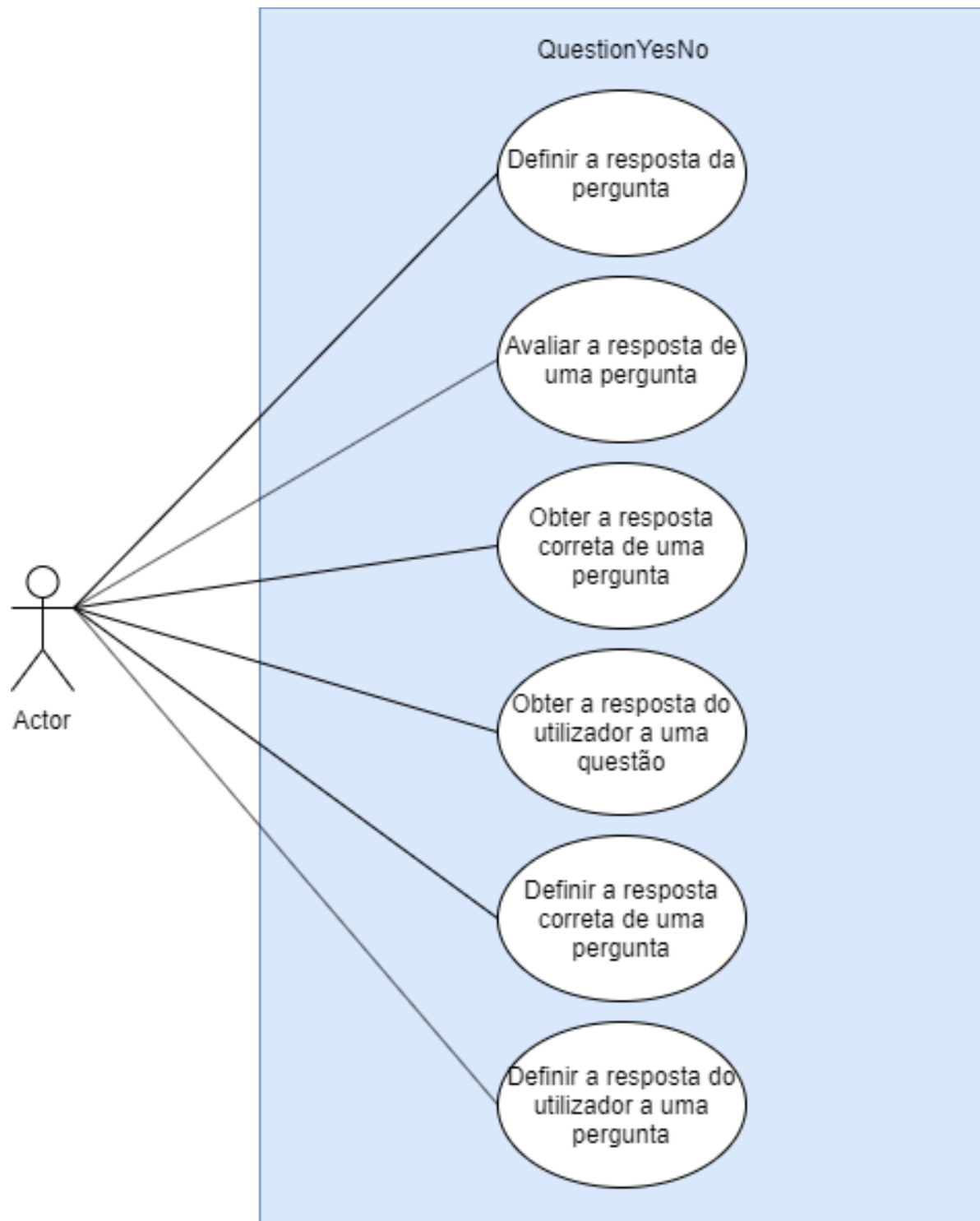
Definir a resposta correta de uma questão

Use Case Name	Definir a resposta correta de uma questão
Short Description	O professor tem de associar uma resposta correta para cada questão
Actors	Professor
Pre-conditions	Existência de uma coleção de perguntas

Definir a resposta do utilizador de uma questão

Use Case Name	Definir a resposta de um utilizador a uma questão
Short Description	A resposta do utilizador é associada a uma pergunta
Actors	Professor, Aluno

2.5 Diagrama UseCase QuestionYesNo



Definir a resposta de uma questão

Use Case Name	Definir a resposta de uma questão
Short Description	O utilizador pode definir a resposta a uma determinada pergunta
Actors	Professor e alunos
Pre-conditions	Existência de uma coleção de perguntas
Typical Flow	<ol style="list-style-type: none">1. Iniciar o teste2. Selecionar a resposta que se pretende escolher

Avaliar a resposta de uma pergunta

Use Case Name	Avaliar a resposta de uma pergunta
Short Description	A resposta do aluno é avaliada
Actors	Professor
Pre-conditions	Existência de uma resposta
Typical Flow	<ol style="list-style-type: none">1. Terminar teste2. Comparar resposta do utilizador com a resposta correta

Obter a resposta correta de uma questão

Use Case Name	Obter a resposta correta de uma questão
Short Description	A resposta correta de uma determinada questão pode ser consultada sendo que o aluno só pode consultar no final do teste
Actors	Professor, Aluno
Pre-conditions	Existência de uma resposta correta associada a cada pergunta
Typical Flow	<ol style="list-style-type: none">1. Terminar o teste2. Selecionar a questão que pretende ver

Obter a resposta do utilizador a uma questão

Use Case Name	Obter a resposta do utilizador a uma questão
Short Description	O utilizador pode consultar a resposta dada pelo aluno a uma determinada questão
Actors	Professor e alunos
Pre-conditions	Existência de uma resposta do utilizador associada a uma pergunta
Typical Flow	<ol style="list-style-type: none">1. Abrir teste realizado pelo aluno2. Selecionar a pergunta que pretende ver

Definir a resposta correta de uma questão

Use Case Name	Definir a resposta correta de uma questão
Short Description	O professor tem de associar uma resposta correta para cada questão
Actors	Professor
Pre-conditions	Existência de uma coleção de perguntas
Typical Flow	<ol style="list-style-type: none">1. Abrir o teste realizado pelo aluno2. Atribuir a resposta correta à questão

Definir a resposta do utilizador de uma questão

Use Case Name	Definir a resposta de um utilizador a uma questão
Short Description	A resposta do utilizador é associada a uma pergunta
Actors	Professor, Aluno
Pre-conditions	Existência de uma coleção de perguntas
Typical Flow	<ol style="list-style-type: none">1. Iniciar teste2. Escolher uma resposta

2.6 Itens a testar

Item a testar	Descrição	UseCase	Responsabilidade
Test.addQuestion()	Adicionar uma questão a um teste	Adicionar questão	Filipe
Test.removeQuestion()	Remover uma questão de um teste pela sua posição	Remover questão pela posição	Filipe
Test.removeQuestion()	Remover a primeira ocorrência de um teste	Remover a primeira ocorrência de um teste	Filipe
Test.numberQuestions()	Saber o número de questões de um teste	Ver o número total de questões	Filipe
Test.isComplete()	Saber se o teste está com todas as questões <i>done</i>	Verificar se teste está completo	Filipe
TestStatistics.correctAnswers()	Saber quais as respostas corretas	Ver respostas corretas	Rebeca
TestStatistics.incorrectAnswers()	Saber quais as respostas incorretas	Ver respostas incorretas	Rebeca
TestStatistics.correctAnswerPercentage()	Saber a percentagem de respostas corretas	Ver percentagem de respostas certas	Rebeca
TestStatistics.incorrectAnswerPercentage()	Saber a percentagem de respostas incorretas	Ver percentagem de respostas incorretas	Rebeca
QuestionMultipleChoice.answer()	Definir a resposta de uma questão de múltipla escolha	Definir a resposta de uma pergunta	Catarina
QuestionMultipleChoice.evaluateAnswer()	Avaliar a resposta de uma questão de múltipla escolha	Avaliar a resposta de uma questão	Catarina

QuestionNumeric.answer()	Definir a resposta de uma questão numérica	Definir a resposta de uma questão	Catarina
---------------------------------	--------------------------------------------	-----------------------------------	----------

QuestionNumeric.evaluateAnswer()	Avaliar a resposta de uma questão numérica	Avaliar a resposta de uma questão	Catarina
-----------------------------------------	--------------------------------------------	-----------------------------------	----------

QuestionYesNo.answer()	Definir a resposta de uma questão sim ou não	Definir a resposta da pergunta	Catarina
-------------------------------	----------------------------------------------	--------------------------------	----------

QuestionYesNo.evaluateAnswer()	Avaliar a resposta de uma questão sim ou não	Avaliar a resposta de uma pergunta	Catarina
---------------------------------------	----------------------------------------------	------------------------------------	----------

3. Detalhes da abordagem aos testes

Para testar os diversos componentes vamos usar testes unitários. Neste caso iremos usar uma perspectiva *black-box* usando as técnicas *Equivalence Class Partitioning* (ECP) e *Boundary Value Analysis* (BVA) - *Input Domain-Based Techniques*.

A técnica ECP consiste na divisão do domínio de inputs em classes de equivalência, as classes de equivalência tanto podem ser válidas como inválidas. Na técnica BVA serão analisados os valores limite, tanto inferiores como superiores das diferentes classes de equivalência (classes de equivalência válidas ou inválidas).

Com base nas classes de equivalência e nos valores limite serão elaborados os casos de teste estes, serão executados no IntelliJ e os seus resultados serão avaliados através da inspeção visual no IDE.

Em seguida serão apresentadas as classes de equivalência, as tabelas ECP e BVA:

Classes de equivalência

EC1	1 input
EC2	!1 input
EC3	Question
EC4	!Question
EC5	Question != NULL
EC6	Question == NULL ou inválida
EC7	int
EC8	!int
EC9	pos != NULL
EC10	pos == NULL
EC11	String
EC12	!String
EC13	Teste com todas as questões avaliadas
EC14	Teste sem todas as questões avaliadas
EC15	user_answer é igual à correct_answer
EC16	user_answer não é igual à correct_answer
EC17	user_answer != NULL
EC18	user_answer ser inválida ou NULL
EC19	correct_answer != NULL
EC20	correct_answer inválida ou NULL
EC21	user_answer só pode ter números
EC22	user_answer não contém só números
EC23	question existe
EC24	question não existe
EC25	Array com todas as questões Done
EC26	Array sem todas as questões Done

EC27	Array de questões válidas
EC28	Array de questões inválidas

- Test.AddQuestion()

→ ECP

Pré-Condition	existência de uma coleção de perguntas instanciada	
Use Case	Test.addQuestion()	
nº inputs	EC1	EC2
input range	EC3	EC4
input values	EC5	EC6

→ BVA

Test.addQuestion()				
	Invalid	Valid		Invalid
nº input	BLB = 0	LB = 1	UB = 1	AUB = 2
	$[-\infty, 1[$	1		$]1, +\infty]$
Input Values	Question			
		Question, NULL	[QuestionYesNo, QuestionMultipleChoice, QuestionNumeric]	

- Test.RemoveQuestion()

→ ECP

Pré-Condition	existência de uma coleção de perguntas instanciada	
Use Case	Test.removeQuestion()	
nº inputs	EC1	EC2
input range	EC3	EC4
input values	EC5, EC23	EC6, EC24

→ BVA

Test.removeQuestion()				
		Invalid	Valid	Invalid
Input Values	Question	BLB = 0	LB = 1 UB = 1	AUB = 2
		$[-\infty, 1[$	1	$]1, +\infty]$
Input Values	Question	Question, NULL	[QuestionYesNo, QuestionMultipleChoice, QuestionNumeric]	

- Test.RemoveQuestion() (Remover questão pela posição da mesma)

→ ECP

Pré-Condition	$pos > 0 \ \&\& \ pos < \text{len}(\text{questions})$	
Use Case	Test.removeQuestion()	
nº inputs	EC1	EC2
input range	EC7	EC8
input values	EC9	EC10

→ BVA

Test.removeQuestion()				
		Invalid	Valid	Invalid
Input Values	Question	BLB = 0	LB = 1 UB = 1	AUB = 2
		$[-\infty, 1[$	1	$]1, +\infty]$
Input Values	pos	BLB=0 LB=1 ALB=2 BUB=len(questions)-2 UB=len(questions)-1	AUB=len(questions)	
		$[-\infty, 0]$	$[1, \text{len}(\text{questions})-1]$	$[\text{len}(\text{questions}), +\infty]$

- Test.numberQuestions()

→ ECP

Pré-Condition	existência de uma coleção de perguntas instanciada	
Use Case	Test.numberQuestions()	
nº inputs	EC1	EC2
input range		
input values	EC27	EC28

→ BVA

Test.numberQuestions()				
Input Values	Questions[]	nº input	Invalid	Valid
			Invalid	Invalid
		size	BLB = 0 LB = 1 UB = 1 AUB = 2	
			[-∞,1[1]1, +∞]	
Input Values	Questions[]	size	BLB = size = 0 LB = 1 ALB=2 BUB=99 UB = 100 AUB = size = 101	
			[-∞,0] [1,100] [101, +∞]	

- Test.isComplete()

→ ECP

Pré-Condition	existência de uma coleção de perguntas instanciada	
Use Case	Test.isComplete()	
nº inputs	EC1	EC2
input range		
input values	EC25, EC27	EC26, EC28

→ BVA

Test.isComplete()				
Input Values	Questions[]	nº input	Invalid	Valid
			Invalid	Invalid
		size	BLB = 0 LB = 1 UB = 1 AUB = 2	
			[-∞,1[1]1, +∞]	
		done	BLB = size = 0 LB = 1 ALB=2 BUB=99 UB = 100 AUB = size = 101	
			[-∞,0] [1,100] [101, +∞]	
Input Values	Questions[]	done	NULL, not done	All Done

- TestStatisticscorrectAnswers()

→ ECP

Pré-Condition	existência de uma coleção de perguntas instanciada	
Use Case	TestStatistics.correctAnswers()	
nº inputs	EC1	EC2
input range		
input values	EC13, EC25, EC27	EC14, EC26, EC28

→ BVA

		TestStatistics.correctAnswers()								
			Invalid	Valid	Invalid					
		nº input	BLB = 0	LB = 1	UB = 1	AUB = 2				
			$[-\infty,1[$	1		$]1, +\infty]$				
Input Value s	Questions[]	size	BLB = size = 0	LB = 1	ALB=2	BUB=99	UB = 100	AUB = size = 101		
			$[-\infty,0]$	$[1,100]$				$[101, +\infty]$		
		done								
			NULL, not done	All Done						
				evaluate d						
					NULL, not evaluated	All Evaluated				

- TestStatistics.incorrectAnswers()

→ ECP

Pré-Condition	existência de uma coleção de perguntas instanciada	
Use Case	TestStatistics.incorrectAnswers()	
nº inputs	EC1	EC2
input range		
input values	EC13, EC25, EC27	EC14, EC26, EC28

→ BVA

		TestStatistics.incorrectAnswers()							
			Invalid	Valid		Invalid			
nº input			BLB = 0	LB = 1	UB = 1	AUB = 2			
			$[-\infty,1[$	1		$]1, +\infty]$			
Input Values	Questions []	size	BLB = size = 0	LB = 1	ALB=2	BUB=99	UB = 100	AUB = size = 101	
			$[-\infty,0]$	[1,100]			$[101, +\infty]$		
		done							
			NULL, not done	All Done					
		evaluated							
			NULL, not evaluated	All Evaluated					

- TestStatistics.correctAnswerPercentage()

→ ECP

Pré-Condition	existência de uma coleção de perguntas instanciada	
Use Case	TestStatistics.correctAnswerPercentage()	
nº inputs	EC1	EC2
input range		
input values	EC13, EC25, EC27	EC14, EC26, EC28

→ BVA

TestStatistics.correctAnswersPercentage()				
		Invalid	Valid	Invalid
Input Values	nº input	BLB = 0	LB = 1 UB = 1	AUB = 2
		$[-\infty, 1[$	1	$]1, +\infty]$
	size	BLB = size = 0	LB = 1 ALB=2 BUB=99 UB = 100	AUB = size = 101
		$[-\infty, 0]$	$[1, 100]$	$[101, +\infty]$
	done	NULL, not done	All Done	
	evaluated	NULL, not evaluated	All Evaluated	

- TestStatistics.incorrectAnswerPercentage()

→ ECP

Pré-Condition	existência de uma coleção de perguntas instanciada	
Use Case	TestStatistics.incorrectAnswerPercentage()	
nº inputs	EC1	EC2
input range		
input values	EC13, EC25, EC27	EC14, EC26, EC28

→ BVA

TestStatistics.incorrectAnswersPercentage()				
		Invalid	Valid	Invalid
Input Values	n° input	BLB = 0	LB = 1 UB = 1	AUB = 2
		$[-\infty, 1[$	1	$]1, +\infty]$
	size	BLB = size = 0	LB = 1 ALB=2 BUB=99 UB = 100	AUB = size = 101
		$[-\infty, 0]$	[1,100]	[101, +∞]
	done	NULL, not done	All Done	
	evaluated	NULL, not evaluated	All Evaluated	

- QuestionMultipleChoice.answer()

→ ECP

Pré-Condition	user_answer != null && len(correct_answer)>0	
Use Case	QuestionMultipleChoice.answer()	
n° inputs	EC1	EC2
input range	EC11	EC12
input values	EC19	EC20

→ BVA

QuestionMultipleChoice.answer()				
	Invalid	Valid		Invalid
n° input	BLB = 0	LB = 1	UB = 1	AUB = 2
	$[-\infty, 1[$	1		$[2, +\infty]$
Input Values	user_answer	BLB = 0	LB=VC(1) ALB=VC(2) BUB=VC(99) UB=VC(100)	AUB=VC(101)
	"" , NULL, 0	[VC(1),VC(100)]		VC(101)

- QuestionMultipleChoice.evaluateAnswer()

→ ECP

Pré-Condition	correct_answer != null && len(correct_answer)>0	
Use Case	QuestionMultipleChoice.evaluateAnswer()	
n° inputs	EC1	EC2
input range	EC11	EC12
input values	EC15,EC17	EC16,EC18

→ BVA

QuestionMultipleChoice.evaluateAnswer()				
	Invalid	Valid		Invalid
n° input	BLB = 0 [-∞,1[LB = 1	UB = 1	AUB = 2 [2,+∞]
Input Values	user_answer	BLB = 0 "",NULL,0	LB=VC(1) ALB=VC(2) BUB=VC(99) UB=VC(100) [VC(1),VC(100)]	AUB=VC(101) VC(101)

- QuestionNumeric.answer()

→ ECP

Pré-Condition	user_answer != null && len(correct_answer)>0	
Use Case	QuestionNumeric.answer()	
n° inputs	EC1	EC2
input range	EC11	EC12
input values	EC19,EC21	EC20,EC22

→ BVA

QuestionNumeric.answer()				
	Invalid	Valid		Invalid
n° input	BLB = 0 [-∞,1[LB = 1	UB = 1	AUB = 2 [2,+∞]
Input Values	user_answer	BLB = 0 "",NULL,0	LB=VC(1) ALB=VC(2) BUB=VC(99) UB=VC(100) [VC(1),VC(100)]	AUB=VC(101) VC(101)

- QuestionNumeric.evaluateAnswer()

→ ECP

Pré-Condition	correct_answer != null && len(correct_answer)>0	
Use Case	QuestionNumeric.evaluateAnswer()	
n° inputs	EC1	EC2
input range	EC11	EC12
input values	EC15,EC17,EC21	EC16,EC18,EC22

→ BVA

QuestionNumeric.evaluateAnswer()				
	Invalid	Valid		Invalid
n° input	BLB = 0 [-∞,1[LB = 1	UB = 1	AUB = 2 [2,+∞]
Input Values	user_answer	BLB = 0 "",NULL,0	LB=VC(1) ALB=VC(2) BUB=VC(99) UB=VC(100) [VC(1),VC(100)]	AUB=VC(101) VC(101)

- QuestionYesNo.answer()

→ ECP

Pré-Condition	user_answer != null && len(correct_answer)>0	
Use Case	QuestionYesNo.answer()	
n° inputs	EC1	EC2
input range	EC11	EC12
input values	EC19	EC20

→ BVA

QuestionYesNo.answer()				
	Invalid	Valid		Invalid
n° input	BLB = 0 [-∞,1[LB = 1	UB = 1	AUB = 2 [2,+∞]
Input Values	user_answer	BLB = 0 "",NULL,0	LB=VC(1) ALB=VC(2) BUB=VC(99) UB=VC(100) [VC(1),VC(100)]	AUB=VC(101) VC(101)

- QuestionYesNo.evaluateAnswer()

→ ECP

Pré-Condition	correct_answer != null && len(correct_answer)>0	
Use Case	QuestionYesNo.evaluateAnswer()	
n° inputs	EC1	EC2
input range	EC11	EC12
input values	EC15,EC17	EC16,EC18

→ BVA

QuestionYesNo.evaluateAnswer()				
	Invalid	Valid		Invalid
n° input	BLB = 0	LB = 1	UB = 1	AUB = 2
	$[-\infty, 1[$	1		$[2, +\infty]$
Input Values	user_answer	BLB = 0	LB=VC(1) ALB=VC(2) BUB=VC(99) UB=VC(100)	AUB=VC(101)
		"" ,NULL,0	[VC(1),VC(100)]	VC(101)

4. Identificação dos Testes

Os casos de teste estão detalhadamente especificados no documento TCO-01 (Test Case Outline) na secção 2.

5. Critérios de passagem ou falha das features

Assertions são utilizadas para verificar uma condição que deverá ser avaliada de forma a que o teste possa continuar a executar. Caso uma *assertion* falhe, o teste irá ser suspenso e uma falha é reportada. Com isto o nosso critério de passagem/falha são as *assertions*. Para cada método de teste temos um resultado esperado, e caso este não seja o adquirido o teste falha, sendo reportado no TCO-01 as falhas e observações por parte dos *testers*.