

PROJETO #1 ENGENHARIA DE SOFTWARE

TEST CASE OUTLINE

Version 3.3

24/11/19

Histórico de Versões

Version #	Implemented By	Revision Date	Approved By	Approval Date	Reason
1.0	Ana Henriques	18/11/19	Filipe	23/11/19	Âmbito
2.0	Ana Henriques	22/11/19	Filipe	23/11/19	Adicionado tópico 2
2.0.5	Filipe Oliveira	23/11/19	Rebeca	23/11/19	Preenchimento da descrição de alguns casos de teste
3.0	Rebeca Teixeira	23/11/19	Ana Henriques	23/11/19	Adicionadas tabelas de cada método ao ponto 2. Preenchimento do ponto 1.3. Reestruturação do documento
3.1	Ana Henriques	23/11/19	Rebeca	24/11/19	Adicionadas descrições de alguns teste cases Finalização do documento
3.2	Rebeca Teixeira	24/11/2019	Ana Henriques	24/11/19	Alteração das tabelas dos test cases para o componente Test Statistics.
3.3	Ana Henriques	24/11/19	Todo o grupo	24/11/19	Finalização do documento

Tabela de Conteúdos

1. Introdução	7
1.1. Identificador do documento.....	7
1.2. Âmbito	7
1.3. References	7
1.4 Glossário	7
2. Detalhes	8
• addQuestion()	8
• Test Case ID:1	9
• Test Case ID:2	9
• Test Case ID: 3,4,5.....	10
• Test Case ID:6	11
• Test Case ID:7	11
• removeQuestion() pela posição	13
• Test Case ID: 1,2,3,4	14
• Test Case ID: 5.....	14
• Test Case ID: 6.....	15
• removeQuestion()	17
• Test Case ID: 1.....	18
• Test Case ID:2	18
• Test Case ID: 3.....	19
• Test Case ID: 4.....	20
• Test Case ID: 5.....	20
• Test Case ID: 6.....	21
• numberQuestions()	22
• Test Case ID: 1.....	23
• Test Case ID: 2.....	23
• Test Case ID: 3.....	24
• Test Case ID: 4.....	25

• Test Case ID: 5.....	26
• isComplete()	27
• Test Case ID: 1.....	28
• Test Case ID: 2.....	28
• Test Case ID: 3.....	29
• Test Case ID: 4.....	30
• Test Case ID: 5.....	30
• Test Case ID: 6.....	31
• Test Case ID: 7.....	32
• correctAnswers()	33
• Test Case ID: 1.....	34
• Test Case ID: 2.....	35
• Test Case ID: 3.....	36
• Test Case ID: 4.....	37
• Test Case ID: 5.....	38
• Test Case ID: 6.....	39
• Test Case ID: 7.....	40
• Test Case ID: 8.....	40
• Test Case ID: 9.....	41
• Test Case ID: 10.....	42
• Test Case ID: 11.....	43
• incorrectAnswer()	45
• Test Case ID: 1.....	46
• Test Case ID: 2.....	46
• Test Case ID: 3.....	47
• Test Case ID: 4.....	48
• Test Case ID: 5.....	49
• Test Case ID: 6.....	50
• Test Case ID: 7.....	51
• Test Case ID: 8.....	52
• Test Case ID: 9.....	53
• Test Case ID: 10.....	53
• Test Case ID: 11.....	54

• correctAnswerPercentage()	56
• Test Case ID: 1.....	57
• Test Case ID: 2.....	57
• Test Case ID: 3.....	58
• Test Case ID: 4.....	59
• Test Case ID: 5.....	60
• Test Case ID: 6.....	61
• Test Case ID: 7.....	62
• Test Case ID: 8.....	63
• Test Case ID: 9.....	64
• Test Case ID: 10.....	64
• Test Case ID: 11.....	65
• incorrectAnswersPercentage()	67
• Test Case ID: 1.....	68
• Test Case ID: 2.....	68
• Test Case ID: 3.....	69
• Test Case ID: 4.....	70
• Test Case ID: 5.....	71
• Test Case ID: 6.....	72
• Test Case ID: 7.....	73
• Test Case ID: 8.....	74
• Test Case ID: 9.....	75
• Test Case ID: 10.....	75
• Test Case ID: 11.....	76
• QuestionMultipleChoice.answer()	78
• Test Case ID: 1.....	79
• Test Case ID: 2.....	79
• Test Case ID: 3.....	80
• Test Case ID: 4.....	81
• Test Case ID: 5.....	81
• Test Case ID: 6.....	82
• QuestionMultipleChoice.evaluateAnswer()	84
• Test Case ID: 1.....	85

• Test Case ID: 2.....	85
• QuestionNumeric.answer()	86
• Test Case ID: 1.....	87
• Test Case ID:2	87
• Test Case ID: 3.....	88
• Test Case ID: 4.....	89
• Test Case ID: 5.....	90
• Test Case ID: 6.....	90
• Test Case ID: 7.....	91
• Test Case ID: 8.....	92
• QuestionNumeric.evaluateAnswer()	93
• Test Case ID: 1.....	94
• Test Case ID: 2.....	95
• QuestionYesNo.answer()	96
• Test Case ID: 1.....	97
• Test Case ID: 2.....	98
• Test Case ID: 3.....	98
• Test Case ID: 4.....	99
• Test Case ID: 5.....	100
• Test Case ID: 6.....	101
• QuestionYesNo.evaluateAnswer()	102
• Test Case ID: 1.....	103
• Test Case ID: 2.....	103
• Test Case ID: 3.....	104

1. Introdução

1.1. Identificador do documento

TCO-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 componentes que definem os use cases disponíveis para o utilizador.

Este documento tem por base auxiliar na execução dos testes pois estão presentes todos os casos de teste.

1.3. References

Para uma melhor compreensão deste documento, mais especificamente a compreensão no que consiste cada EC (Equivalence Class) é recomendada a leitura do documento TCS-01.

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. Detalhes

- addQuestion()

Data Execução	Hora Execução	Tester
22/11/2019	12:02	Filipe

Pré-Condition		existência de uma coleção de perguntas instanciada					
Test Case ID	Use Case	Input	Valores do Inputs	Valid Equivalence Class and Bounds covered	Invalid Equivalence Class and Bounds covered	Resultado Esperado	Pass/Fail/Untested
1			NULL		EC6	Não adiciona	Fail
2			Question		EC6(BLB)	Não adiciona	Fail
3			QuestionYesNo	EC5		Adiciona	Pass
4			QuestionMultipleChoice	EC5		Adiciona	Pass
5			QuestionNumeric	EC5		Adiciona	Pass
6			Adicionar uma questão com a lista cheia		EC28	Retorna Exceção	Pass
7			Adicionar uma questão à última posição da lista	EC5, EC27		Adiciona	Pass

- Test Case ID:1

- ✓ *Pré-Condições*

- Existência de uma coleção de perguntas instanciada.

- ✓ *Procedimento*

- Criou-se uma questão à qual foi atribuída o valor null e foi feita uma tentativa de adicionar essa questão ao teste.

- ✓ *Output Esperado*

- False

- ✓ *Output Real*

- True

- ✓ *Observações Finais*

- O teste falhou, pois é possível adicionar ao teste uma questão declarada a null.

- Test Case ID:2

- ✓ *Pré-Condições*

- Existência de uma coleção de perguntas instanciada.

- ✓ *Procedimento*

- Criou-se uma questão do tipo Question e foi feita uma tentativa de adicionar essa questão ao teste.

- ✓ *Output Esperado*

- False

✓ *Output Real*

True

✓ *Observações Finais*

O teste falhou, pois é possível adicionar ao teste uma questão do tipo Question.

- Test Case ID: 3,4,5

✓ *Pré-Condições*

Existência de uma coleção de perguntas instanciada.

✓ *Procedimento*

3 - Criou-se uma questão do tipo QuestionYesNo e foi feita uma tentativa de adicionar essa questão ao teste.

4 - Criou-se uma questão do tipo QuestionMultipleChoice e foi feita uma tentativa de adicionar essa questão ao teste.

5 - Criou-se uma questão do tipo QuestionNumeric e foi feita uma tentativa de adicionar essa questão ao teste.

✓ *Output Esperado*

True

✓ *Output Real*

True

✓ *Observações Finais*

O teste passou, pois é possível adicionar ao teste esse tipo de questões.

- Test Case ID:6

- ✓ *Pré-Condições*

- Existência de uma coleção de perguntas instanciada.

- ✓ *Procedimento*

- Verificou-se se dava para adicionar uma questão a uma lista já cheia.

- ✓ *Output Esperado*

- ArrayOutOfBoundsException*

- ✓ *Output Real*

- ArrayOutOfBoundsException*

- ✓ *Observações Finais*

- O teste passou, pois, ao tentar adicionar a uma lista cheia, ela lança esse tipo de exceção.

- Test Case ID:7

- ✓ *Pré-Condições*

- Existência de uma coleção de perguntas instanciada.

- ✓ *Procedimento*

- Verificou-se se dava para adicionar uma questão à última posição da lista.

- ✓ *Output Esperado*

- True

✓ *Output Real*

True

✓ *Observações Finais*

O teste passou, pois conseguimos adicionar uma questão à última posição da lista.

- removeQuestion() pela posição

Data Execução	Hora Execução	Tester
22/11/2019	12:10	Filipe

Pré-Condition			existência de uma coleção de perguntas instanciada, pos >= 0 && pos < len(questions)				
Test Case ID	Use Case	Input	Valores do Inputs	Valid Equivalence Class and Bounds covered	Invalid Equivalence Class and Bounds covered	Resultado Esperado	Pass/Fail/Untested
1	removeQuestion()	pos	1	EC9		Remove questão	Pass
2			2	EC9		Remove questão	Pass
3			len(questions)-2	EC9		Remove questão	Pass
4			len(questions)-1	EC9		Remove questão	Pass
5			NULL		EC10	Não remove questão	Pass
6			Remover com uma lista vazia, input(pos=0)		EC28	Não remove questão	Pass

- Test Case ID: 1,2,3,4

- ✓ *Pré-Condições*

- Existência de uma coleção de perguntas instanciada, $pos \geq 0 \ \&\& \ pos < len(questions)$

- ✓ *Procedimento*

- 1 – Verificou-se se dava para remover uma pergunta na primeira posição da lista.

- 2 – Verificou-se se dava para remover uma pergunta na segunda posição da lista.

- 3 – Verificou-se se dava para remover uma pergunta na antepenúltima posição da lista.

- 4 – Verificou-se se dava para remover uma pergunta na penúltima posição da lista.

- ✓ *Output Esperado*

- True

- ✓ *Output Real*

- True

- ✓ *Observações Finais*

- O teste passou, pois conseguimos remover as perguntas nessas posições.

- Test Case ID: 5

- ✓ *Pré-Condições*

- Existência de uma coleção de perguntas instanciada, $pos \geq 0 \ \&\& \ pos < len(questions)$

✓ *Procedimento*

1 –Declarou-se uma questão a null;

2 –Questão foi adicionada à primeira posição da lista;

3 –Verificou-se se dava para remover essa questão

✓ *Output Esperado*

False

✓ *Output Real*

False

✓ *Observações Finais*

O teste passou, pois não é possível remover uma questão declarada a null.

- Test Case ID: 6

✓ *Pré-Condições*

Existência de uma coleção de perguntas instanciada, $pos \geq 0 \ \&\& \ pos < len(questions)$

✓ *Procedimento*

Tentou-se remover uma pergunta com a lista vazia.

✓ *Output Esperado*

False

✓ *Output Real*

False

✓ *Observações Finais*

O teste passou, pois não é possível remover questões de uma lista que está vazia.

- removeQuestion()

Data Execução	Hora Execução	Tester
22/11/2019	12:24	Filipe

Pré-Condition			existência de uma coleção de perguntas instanciada				
Test Case ID	Use Case	Input	Valores do Inputs	Valid Equivalence Class and Bounds covered	Invalid Equivalence Class and Bounds covered	Resultado Esperado	Pass/Fail/Untested
1	removeQuestion()	Question	Question e existe no array	EC23	EC6(BLB)	Não remove questão	Fail
2			Não existe no array	EC5	EC24	Não remove questão	Pass
3			QuestionMultipleChoice e existe no array	EC5, EC23, EC27		Remove questão	Pass
4			NULL		EC6	Não remove questão	Pass
5			Teste sem perguntas		EC28	Não remove questão	Pass
6			Apenas uma pergunta e é a que vai ser eliminada	EC5, EC23, EC27		Remove Questão	Pass

- Test Case ID: 1

- ✓ *Pré-Condições*

- Existência de uma coleção de perguntas instanciada

- ✓ *Procedimento*

- 1. Declarou-se uma questão do tipo Question;
 2. Verificou-se se dava para remover essa questão;

- ✓ *Output Esperado*

- False

- ✓ *Output Real*

- True

- ✓ *Observações Finais*

- O teste falhou, pois não deveria ser possível remover uma questão do tipo Question.

- Test Case ID:2

- ✓ *Pré-Condições*

- Existência de uma coleção de perguntas instanciada

- ✓ *Procedimento*

- 1. Declarou-se uma questão;
 2. Questão não foi adicionada à lista;
 3. Verificou-se se dava para remover essa questão;

✓ *Output Esperado*

False

✓ *Output Real*

False

✓ *Observações Finais*

O teste passou, pois não é possível remover uma questão que não existe na lista.

- Test Case ID: 3

✓ *Pré-Condições*

Existência de uma coleção de perguntas instanciada

✓ *Procedimento*

1. Declarou-se uma questão do tipo QuestionMultipleChoice;
2. Questão foi adicionada à lista;
3. Verificou-se se dava para remover essa questão;

✓ *Output Esperado*

True

✓ *Output Real*

True

✓ *Observações Finais*

O teste passou, pois é possível remover uma questão desse tipo. Não foi feito um teste para cada tipo de questão pois, nada nos leva a pensar que se remove uma questão do tipo Multiple Choice não irá remover os outros tipos de questão.

- Test Case ID: 4

- ✓ *Pré-Condições*

- Existência de uma coleção de perguntas instanciada

- ✓ *Procedimento*

- 1. Declarou-se uma questão a null;
 2. Questão foi adicionada à lista;
 3. Verificou-se se dava para remover essa questão;

- ✓ *Output Esperado*

- False

- ✓ *Output Real*

- False

- ✓ *Observações Finais*

- O teste passou, pois não é possível remover uma questão declarada a null.

- Test Case ID: 5

- ✓ *Pré-Condições*

- Existência de uma coleção de perguntas instanciada

- ✓ *Procedimento*

- 1. Declarou-se uma lista vazia;
 2. Verificou-se se dava para remover essa questão;

✓ *Output Esperado*

False

✓ *Output Real*

False

✓ *Observações Finais*

O teste passou, pois não é possível remover uma questão de uma lista vazia.

- Test Case ID: 6

✓ *Pré-Condições*

Existência de uma coleção de perguntas instanciada

✓ *Procedimento*

1. Declarou-se uma e uma só questão válida;
2. Questão foi adicionada à lista;
3. Verificou-se se dava para remover essa questão;

✓ *Output Esperado*

True

✓ *Output Real*

True

✓ *Observações Finais*

O teste passou, pois é possível remover uma questão do teste, mesmo sendo a única.

- numberQuestions()

Data Execução	Hora Execução	Tester
22/11/2019	12:46	Filipe

Pré-Condition			existência de uma coleção de perguntas instanciada				
Test Case ID	Use Case	Input	Valores do Inputs	Valid Equivalence Class and Bounds covered	Invalid Equivalenc e Class and Bounds covered	Resultado Esperado	Pass/Fail/Untested
1	numberQuestions()	Questions[]	Todas as posições são nulas		EC28	0	Pass
2			Primeira posição nula e as restantes com questões	EC27		9	Pass
3			Última posição nula e as restantes com questões	EC27		9	Pass
4			Todas as posições com questões	EC27 (UB)		10	Pass
5			Teste sem perguntas		EC28(BLB)	0	Pass

- Test Case ID: 1

- ✓ *Pré-Condições*

- Existência de uma coleção de perguntas instanciada

- ✓ *Procedimento*

- 1. Foi instanciada uma lista com todas as posições a null;
 2. Verificou-se se dava para fazer a contagem de questões desse teste;

- ✓ *Output Esperado*

- 0

- ✓ *Output Real*

- 0

- ✓ *Observações Finais*

- O teste passou, pois não é possível contar uma questão declarada a null.

- Test Case ID: 2

- ✓ *Pré-Condições*

- Existência de uma coleção de perguntas instanciada

- ✓ *Procedimento*

- 1. Declarou-se uma lista;
 2. Criaram-se 10 questões;
 3. Questão foi adicionada à primeira posição da lista e declarada a null;
 4. Nove questões foram adicionadas nas restantes posições;

5. Verificou-se se dava para contar o número de questões desse teste;

✓ *Output Esperado*

9

✓ *Output Real*

9

✓ *Observações Finais*

O teste passou, pois fez a contagem certa do número de perguntas que o teste continha.

- Test Case ID: 3

✓ *Pré-Condições*

Existência de uma coleção de perguntas instanciada

✓ *Procedimento*

1. Declarou-se uma lista;
2. Criaram-se 10 questões;
3. Nove questões foram adicionadas nas primeiras posições;
4. Questão foi adicionada à última posição da lista e declarada a null;
5. Verificou-se se dava para contar o número de questões desse teste;

✓ *Output Esperado*

9

✓ *Output Real*

9

✓ *Observações Finais*

O teste passou, pois fez a contagem certa do número de perguntas do teste.

- Test Case ID: 4

✓ *Pré-Condições*

Existência de uma coleção de perguntas instanciada

✓ *Procedimento*

1. Declarou-se uma lista;
2. Criaram-se 10 questões;
3. Foram adicionadas à lista;
4. Verificou-se se dava para contar o número de questões desse teste;

✓ *Output Esperado*

10

✓ *Output Real*

10

✓ *Observações Finais*

O teste passou, pois fez contagem certa do número de perguntas do teste.

- Test Case ID: 5

- ✓ *Pré-Condições*

- Existência de uma coleção de perguntas instanciada

- ✓ *Procedimento*

- 1. Declarou-se uma lista vazia;
 2. Verificou-se se dava para contar o número de questões desse teste;

- ✓ *Output Esperado*

- 0

- ✓ *Output Real*

- 0

- ✓ *Observações Finais*

- O teste passou, pois fez contagem certa do número de perguntas do teste.

- isComplete()

Data Execução	Hora Execução	Tester
22/11/2019	12:50	Filipe

Pré-Condition			existência de uma coleção de perguntas instanciada				
Test Case ID	Use Case	Input	Valores do Inputs	Valid Equivalence Class and Bounds covered	Invalid Equivalence Class and Bounds covered	Resultado Esperado	Pass/Fail/Un tested
1	isComplete()	Questions[]	Todas as posições são nulas		EC28	False	Fail
2			Primeira posição nula e as restantes com perguntas		EC28	False	Fail
3			Última posição nula e as restantes com perguntas		EC28	False	Fail
4			Todas as posições com questões Done	EC25		True	Pass
5			Todas as posições sem questões Done		EC26	False	Pass
6			Primeira questão Undone e restantes Done		EC26	False	Fail
7			Última questão Undone e restantes Done		EC26	False	Fail
8			Teste sem questões		EC28 (BLB)	Exceção	

- Test Case ID: 1

- ✓ *Pré-Condições*

- Existência de uma coleção de perguntas instanciada

- ✓ *Procedimento*

- 1. Encheu-se a lista com questões declaradas a null;
 2. Verificou-se se dava para completar o teste;

- ✓ *Output Esperado*

- False

- ✓ *Output Real*

- True

- ✓ *Observações Finais*

- O teste falhou, pois não deveria dar para completar o teste com perguntas a null.

- Test Case ID: 2

- ✓ *Pré-Condições*

- Existência de uma coleção de perguntas instanciada

- ✓ *Procedimento*

- 1. Declarou-se uma lista;
 2. Questão foi adicionada à primeira posição da lista e declarada a null;
 3. Preencheu-se as restantes posições com questões válidas e foram declaradas como feitas.

4. Verificou-se se dava para completar o teste;

✓ *Output Esperado*

False

✓ *Output Real*

True

✓ *Observações Finais*

O teste falhou, pois como a primeira pergunta é null o teste não deveria dar para ser completado.

- Test Case ID: 3

✓ *Pré-Condições*

Existência de uma coleção de perguntas instanciada

✓ *Procedimento*

1. Declarou-se uma lista;

2. Questão foi adicionada à última posição da lista e declarada a null;

3. Preencheu-se as restantes posições com questões válidas e foram declaradas como feitas.

4. Verificou-se se dava para completar o teste;

✓ *Output Esperado*

False

✓ *Output Real*

True

✓ *Observações Finais*

O teste falhou, pois como a última pergunta é null o teste não deveria dar para ser completado.

- Test Case ID: 4

✓ *Pré-Condições*

Existência de uma coleção de perguntas instanciada

✓ *Procedimento*

1. Preencheu-se as posições numa lista com questões válidas e foram declaradas como feitas.
2. Verificou-se se dava para completar o teste;

✓ *Output Esperado*

True

✓ *Output Real*

True

✓ *Observações Finais*

O teste passou, pois deu para completar o teste.

- Test Case ID: 5

✓ *Pré-Condições*

Existência de uma coleção de perguntas instanciada

✓ *Procedimento*

1. Preencheu-se as posições numa lista com questões válidas e foram declaradas como não feitas.
2. Verificou-se se dava para completar o teste;

✓ *Output Esperado*

False

✓ *Output Real*

False

✓ *Observações Finais*

O teste passou, pois não deu para completar o teste devido às questões estarem como não feitas.

- Test Case ID: 6

✓ *Pré-Condições*

Existência de uma coleção de perguntas instanciada

✓ *Procedimento*

1. Preencheu-se as posições numa lista com questões válidas e foram declaradas como feitas, sendo que a primeira foi declarada como não feita.
2. Verificou-se se dava para completar o teste;

✓ *Output Esperado*

False

✓ *Output Real*

True

✓ *Observações Finais*

O teste falhou, pois deu para completar o teste, sendo que existe uma pergunta declarada como não feita (primeira).

- Test Case ID: 7

✓ *Pré-Condições*

Existência de uma coleção de perguntas instanciada

✓ *Procedimento*

1. Preencheu-se as posições numa lista com questões válidas e foram declaradas como feitas, sendo que a última foi declarada como não feita.
2. Verificou-se se dava para completar o teste;

✓ *Output Esperado*

False

✓ *Output Real*

True

✓ *Observações Finais*

O teste falhou, pois deu para completar o teste, sendo que existia uma pergunta declarada como não feita (última).

- correctAnswers()

Data Execução	Hora Execução	Tester
22/11/2019	16:00	Rebeca Teixeira

Pré-Condition			existência de uma coleção de perguntas instanciada				
Test Case ID	Use Case	Input	Valores do Inputs	Valid Equivalence Class and Bounds covered	Invalid Equivalence Class and Bounds covered	Resultado Esperado	Pass/Fail/Un tested
1	correctAnswers()	Questions[]	Apenas uma pergunta e está errada	EC27		Array sem perguntas	Pass
2			Apenas uma pergunta e está certa	EC27		Array com uma pergunta	Pass
3			Primeira pergunta errada e as restantes certas	EC27 (BUB)		Array com 9 perguntas	Pass
4			Última pergunta errada e as restantes certas	EC27 (BUB)		Array com 9 perguntas	Pass
5			Primeira pergunta certa e as restantes erradas	EC27 (LB)		Array com uma pergunta	Pass
6			Última pergunta certa e as restantes erradas	EC27 (LB)		Array com uma pergunta	Pass
7			Todas as perguntas certas	EC27 (UB)		Array com 10 perguntas	Pass
8			Todas as perguntas erradas	EC27		Array sem perguntas	Pass
9			Array sem perguntas		EC28 (BLB)	Exceção	Fail
10			Array sem perguntas avaliadas		EC14 (BLB)	Exceção	Fail
11			Array sem perguntas Done (incompleto)		EC26 (BLB)	Exceção	Fail

- Test Case ID: 1
 - ✓ *Pré-Condições*

Existência de uma coleção de perguntas instanciada
 - ✓ *Procedimento*
 1. Instanciar um teste
 2. Instanciar uma pergunta
 3. Adicionar a pergunta ao teste
 4. Atribuir uma resposta incorreta a essa pergunta
 5. Avaliar a questão
 6. Adquirir as estatísticas do teste
 - ✓ *Output Esperado*

Array sem perguntas
 - ✓ *Output Real*

Array sem perguntas
 - ✓ *Observações Finais*

O método funciona corretamente para este caso de teste.

- Test Case ID: 2

- ✓ *Pré-Condições*

- Existência de uma coleção de perguntas instanciada

- ✓ *Procedimento*

- 1. Instanciar um teste
 2. Instanciar uma pergunta
 3. Adicionar a pergunta ao teste
 4. Atribuir uma resposta correta a essa pergunta
 5. Avaliar a questão
 6. Adquirir as estatísticas do teste

- ✓ *Output Esperado*

- Um array com apenas uma questão sendo ela aquela que introduzimos.

- ✓ *Output Real*

- Array com a questão que introduzimos

- ✓ *Observações Finais*

- O método funciona corretamente para este caso de teste.

- Test Case ID: 3

- ✓ *Pré-Condições*

- Existência de uma coleção de perguntas instanciada

- ✓ *Procedimento*

- 1. Instanciar um teste
 2. Instanciar uma pergunta
 3. Adicionar a primeira pergunta ao teste
 4. Atribuir uma resposta incorreta a essa pergunta
 5. Avaliar a questão
 6. Adicionar mais 9 perguntas com respostas corretas ao teste
 7. Avaliar perguntas
 8. Adquirir as estatísticas do teste

- ✓ *Output Esperado*

- Um array com 9 perguntas sendo elas as perguntas certas

- ✓ *Output Real*

- Array com 9 perguntas sendo elas as perguntas certas

- ✓ *Observações Finais*

- O método funciona corretamente para este caso de teste.

- Test Case ID: 4

- ✓ *Pré-Condições*

- Existência de uma coleção de perguntas instanciada

- ✓ *Procedimento*

- 1. Instanciar um teste
 2. Instanciar uma pergunta
 3. Adicionar 9 perguntas com respostas corretas ao teste
 4. Avaliar perguntas
 5. Adicionar a última pergunta ao teste
 6. Atribuir uma resposta incorreta a essa pergunta
 7. Avaliar a questão
 8. Adquirir as estatísticas do teste

- ✓ *Output Esperado*

- Um array com 9 perguntas sendo elas as perguntas certas

- ✓ *Output Real*

- Array com 9 perguntas sendo elas as perguntas certas

- ✓ *Observações Finais*

- O método funciona corretamente para este caso de teste.

- Test Case ID: 5

- ✓ *Pré-Condições*

- Existência de uma coleção de perguntas instanciada

- ✓ *Procedimento*

- 1. Instanciar um teste
 2. Instanciar uma pergunta
 3. Adicionar a primeira pergunta ao teste
 4. Atribuir uma resposta correta a essa pergunta
 5. Avaliar a questão
 6. Adicionar 9 perguntas com respostas incorretas ao teste
 7. Avaliar perguntas
 8. Adquirir as estatísticas do teste

- ✓ *Output Esperado*

- Um array com 1 pergunta sendo ela a pergunta certa

- ✓ *Output Real*

- Array com 1 pergunta sendo ela a pergunta certa

- ✓ *Observações Finais*

- O método funciona corretamente para este caso de teste.

- Test Case ID: 6

- ✓ *Pré-Condições*

- Existência de uma coleção de perguntas instanciada

- ✓ *Procedimento*

- 1. Instanciar um teste
 2. Instanciar uma pergunta
 3. Adicionar 9 perguntas com respostas incorretas ao teste
 4. Avaliar perguntas
 5. Adicionar a última pergunta ao teste
 6. Atribuir uma resposta correta a essa pergunta
 7. Avaliar a questão
 8. Adquirir as estatísticas do teste

- ✓ *Output Esperado*

- Um array com 1 pergunta sendo ela a pergunta certa

- ✓ *Output Real*

- Array com 1 pergunta sendo ela a pergunta certa

- ✓ *Observações Finais*

- O método funciona corretamente para este caso de teste.

- Test Case ID: 7

- ✓ *Pré-Condições*

- Existência de uma coleção de perguntas instanciada

- ✓ *Procedimento*

- 1. Instanciar um teste
 2. Instanciar uma pergunta
 3. Adicionar 10 perguntas com respostas corretas ao teste
 4. Avaliar perguntas
 5. Adquirir as estatísticas do teste

- ✓ *Output Esperado*

- Um array com 10 perguntas sendo elas as perguntas certas

- ✓ *Output Real*

- Array com 10 perguntas sendo elas as perguntas certas

- ✓ *Observações Finais*

- O método funciona corretamente para este caso de teste.

- Test Case ID: 8

- ✓ *Pré-Condições*

- Existência de uma coleção de perguntas instanciada

✓ *Procedimento*

1. Instanciar um teste
2. Instanciar uma pergunta
3. Adicionar 10 perguntas com respostas incorretas ao teste
4. Avaliar perguntas
5. Adquirir as estatísticas do teste

✓ *Output Esperado*

Um array sem perguntas

✓ *Output Real*

Array sem perguntas

✓ *Observações Finais*

O método funciona corretamente para este caso de teste.

- Test Case ID: 9

✓ *Pré-Condições*

Existência de uma coleção de perguntas instanciada

✓ *Procedimento*

1. Instanciar um teste
2. Adquirir as estatísticas do teste

✓ *Output Esperado*

Exceção que permita saber que não se pode devolver as respostas corretas pois o teste não contém questões

✓ *Output Real*

Array sem perguntas

✓ *Observações Finais*

O método não funciona corretamente para este caso de teste.

- Test Case ID: 10

✓ *Pré-Condições*

Existência de uma coleção de perguntas instanciada

✓ *Procedimento*

1. Instanciar um teste
2. Instanciar 10 perguntas
3. Adicionar as 10 perguntas ao teste
4. Atribuir uma resposta a cada pergunta
5. Marcar as 10 perguntas corretas como feitas
6. Adquirir as estatísticas do teste

✓ *Output Esperado*

Exceção que permita saber que não se pode devolver as respostas corretas pois as respostas não estão avaliadas

✓ *Output Real*

Array com 10 perguntas corretas

✓ *Observações Finais*

O método não funciona corretamente para este caso de teste.

- Test Case ID: 11

✓ *Pré-Condições*

Existência de uma coleção de perguntas instanciada

✓ *Procedimento*

1. Instanciar um teste
2. Instanciar 10 perguntas
3. Adicionar as 10 perguntas ao teste
4. Marcar as 10 perguntas como por fazer
5. Adquirir as estatísticas do teste

✓ *Output Esperado*

Exceção que permita saber que não se pode devolver as respostas corretas pois as respostas não estão terminadas, ou seja, o teste não está completo.

✓ *Output Real*

Array sem perguntas

✓ *Observações Finais*

O método não funciona corretamente para este caso de teste.

- incorrectAnswer()

Data Execução	Hora Execução	Tester
22/11/2019	17:52	Rebeca Teixeira

Pré-Condition			existência de uma coleção de perguntas instanciada				
Test Case ID	Use Case	Input	Valores do Inputs	Valid Equivalence Class and Bounds covered	Invalid Equivalence Class and Bounds covered	Resultado Esperado	Pass/Fail/Untested
1	incorrectAnswers()	Questions[]	Apenas uma pergunta e está errada	EC27		Array com uma pergunta	Pass
2			Apenas uma pergunta e está certa	EC27 (LB)		Array sem perguntas	Pass
3			Primeira pergunta errada e as restantes certas	EC27 (LB)		Array com uma pergunta	Pass
4			Última pergunta errada e as restantes certas	EC27 (BUB)		Array com uma pergunta	Pass
5			Primeira pergunta certa e as restantes erradas	EC27 (BUB)		Array com 9 perguntas	Pass
6			Última pergunta certa e as restantes erradas	EC27		Array com 9 perguntas	Pass
7			Todas as perguntas certas	EC27		Array sem perguntas	Pass
8			Todas as perguntas erradas	EC27 (UB)		Array com 10 perguntas	Pass
9			Array sem perguntas		EC28 (BLB)	Exceção	Fail
10			Array sem perguntas avaliadas		EC14 (BLB)	Exceção	Fail
11			Array sem perguntas Done (incompleto)		EC26 (BLB)	Exceção	Fail

- Test Case ID: 1

- ✓ *Pré-Condições*

- Existência de uma coleção de perguntas instanciada

- ✓ *Procedimento*

- 1. Instanciar um teste
 2. Instanciar uma pergunta
 3. Adicionar a pergunta ao teste
 4. Atribuir uma resposta incorreta a essa pergunta
 5. Avaliar a questão
 6. Adquirir as estatísticas do teste

- ✓ *Output Esperado*

- Array com uma pergunta sendo ela a pergunta incorreta

- ✓ *Output Real*

- Array com uma pergunta sendo ela a pergunta incorreta

- ✓ *Observações Finais*

- O método funciona corretamente para este caso de teste.

- Test Case ID: 2

- ✓ *Pré-Condições*

- Existência de uma coleção de perguntas instanciada

✓ *Procedimento*

1. Instanciar um teste
2. Instanciar uma pergunta
3. Adicionar a pergunta ao teste
4. Atribuir uma resposta correta a essa pergunta
5. Avaliar a questão
6. Adquirir as estatísticas do teste

✓ *Output Esperado*

Array sem questões

✓ *Output Real*

Array sem questões

✓ *Observações Finais*

O método funciona corretamente para este caso de teste.

- Test Case ID: 3

✓ *Pré-Condições*

Existência de uma coleção de perguntas instanciada

✓ *Procedimento*

1. Instanciar um teste
2. Instanciar uma pergunta

3. Adicionar a primeira pergunta ao teste
4. Atribuir uma resposta incorreta a essa pergunta
5. Avaliar a questão
6. Adicionar mais 9 perguntas com respostas corretas ao teste
7. Avaliar perguntas
8. Adquirir as estatísticas do teste

✓ *Output Esperado*

Um array com 1 pergunta sendo ela a pergunta incorreta

✓ *Output Real*

Um array com 1 pergunta sendo ela a pergunta incorreta

✓ *Observações Finais*

O método funciona corretamente para este caso de teste.

- Test Case ID: 4

✓ *Pré-Condições*

Existência de uma coleção de perguntas instanciada

✓ *Procedimento*

1. Instanciar um teste
2. Instanciar uma pergunta
3. Adicionar 9 perguntas com respostas corretas ao teste

4. Avaliar perguntas
5. Adicionar a última pergunta ao teste
6. Atribuir uma resposta incorreta a essa pergunta
7. Avaliar a questão
8. Adquirir as estatísticas do teste

✓ *Output Esperado*

Um array com 1 pergunta sendo ela a pergunta incorreta

✓ *Output Real*

Um array com 1 pergunta sendo ela a pergunta incorreta

✓ *Observações Finais*

O método funciona corretamente para este caso de teste.

- Test Case ID: 5

✓ *Pré-Condições*

Existência de uma coleção de perguntas instanciada

✓ *Procedimento*

1. Instanciar um teste
2. Instanciar uma pergunta
3. Adicionar a primeira pergunta ao teste
4. Atribuir uma resposta correta a essa pergunta

5. Avaliar a questão
6. Adicionar 9 perguntas com respostas incorretas ao teste
7. Avaliar perguntas
8. Adquirir as estatísticas do teste

✓ *Output Esperado*

Um array com 9 perguntas sendo elas as perguntas incorretas

✓ *Output Real*

Array com 9 perguntas sendo elas as perguntas incorretas

✓ *Observações Finais*

O método funciona corretamente para este caso de teste.

- Test Case ID: 6

✓ *Pré-Condições*

Existência de uma coleção de perguntas instanciada

✓ *Procedimento*

1. Instanciar um teste
2. Instanciar uma pergunta
3. Adicionar 9 perguntas com respostas incorretas ao teste
4. Avaliar perguntas
5. Adicionar a última pergunta ao teste

6. Atribuir uma resposta correta a essa pergunta

7. Avaliar a questão

8. Adquirir as estatísticas do teste

✓ *Output Esperado*

Um array com 9 perguntas sendo elas as perguntas incorretas

✓ *Output Real*

Array com 9 perguntas sendo elas as perguntas incorretas

✓ *Observações Finais*

O método funciona corretamente para este caso de teste.

- Test Case ID: 7

✓ *Pré-Condições*

Existência de uma coleção de perguntas instanciada

✓ *Procedimento*

1. Instanciar um teste

2. Instanciar uma pergunta

3. Adicionar 10 perguntas com respostas corretas ao teste

4. Avaliar perguntas

5. Adquirir as estatísticas do teste

✓ *Output Esperado*

Array sem perguntas

✓ *Output Real*

Array sem perguntas

✓ *Observações Finais*

O método funciona corretamente para este caso de teste.

- Test Case ID: 8

✓ *Pré-Condições*

Existência de uma coleção de perguntas instanciada

✓ *Procedimento*

1. Instanciar um teste
2. Instanciar uma pergunta
3. Adicionar 10 perguntas com respostas incorretas ao teste
4. Avaliar perguntas
5. Adquirir as estatísticas do teste

✓ *Output Esperado*

Array com 10 perguntas sendo elas as perguntas erradas

✓ *Output Real*

Array com 10 perguntas sendo elas as perguntas erradas

✓ *Observações Finais*

O método funciona corretamente para este caso de teste.

- Test Case ID: 9

✓ *Pré-Condições*

Existência de uma coleção de perguntas instanciada

✓ *Procedimento*

1. Instanciar um teste
2. Adquirir as estatísticas do teste

✓ *Output Esperado*

Exceção que permita saber que não se pode devolver as respostas incorretas pois o teste não contém questões

✓ *Output Real*

Array sem perguntas

✓ *Observações Finais*

O método não funciona corretamente para este caso de teste.

- Test Case ID: 10

✓ *Pré-Condições*

Existência de uma coleção de perguntas instanciada

✓ *Procedimento*

1. Instanciar um teste
2. Instanciar 10 perguntas
3. Adicionar as 10 perguntas ao teste
4. Atribuir uma resposta incorreta a cada pergunta
5. Marcar as 10 perguntas como feitas
6. Adquirir as estatísticas do teste

✓ *Output Esperado*

Exceção que permita saber que não se pode devolver as respostas incorretas pois as respostas não estão avaliadas

✓ *Output Real*

Array com 10 perguntas incorretas

✓ *Observações Finais*

O método não funciona corretamente para este caso de teste.

- Test Case ID: 11

✓ *Pré-Condições*

Existência de uma coleção de perguntas instanciada

✓ *Procedimento*

1. Instanciar um teste
2. Instanciar 10 perguntas

3. Adicionar as 10 perguntas ao teste
4. Marcar as 10 perguntas como por fazer
5. Adquirir as estatísticas do teste

✓ *Output Esperado*

Exceção que permita saber que não se pode devolver as respostas incorretas pois as respostas não estão terminadas, ou seja, o teste não está completo.

✓ *Output Real*

Array sem perguntas

✓ *Observações Finais*

O método não funciona corretamente para este caso de teste.

- correctAnswerPercentage()

Data Execução	Hora Execução	Tester
22/11/2019	19:00	Rebeca Teixeira

Pré-Condition			existência de uma coleção de perguntas instanciada				
Test Case ID	Use Case	Input	Valores do Inputs	Valid Equivalence Class and Bounds covered	Invalid Equivalence Class and Bounds covered	Resultado Esperado	Pass/Fail/Un tested
1	correctAnswersPercentage()	Questions[]	Apenas uma pergunta e está errada	EC27		0	Pass
2			Apenas uma pergunta e está certa	EC27 (LB)		1	Pass
3			Primeira pergunta errada e as restantes certas	EC27 (BUB)		0.9	Fail
4			Última pergunta errada e as restantes certas	EC27 (BUB)		0.9	Fail
5			Primeira pergunta certa e as restantes erradas	EC27 (LB)		0.1	Fail
6			Última pergunta certa e as restantes erradas	EC27 (LB)		0.1	Fail
7			Todas as perguntas certas	EC27 (UB)		1	Pass
8			Todas as perguntas erradas	EC27		0	Pass
9			Array sem perguntas		EC28 (BLB)	Exceção	Fail
10			Array sem perguntas avaliadas		EC14 (BLB)	Exceção	Fail
11			Array sem perguntas Done (incompleto)		EC26 (BLB)	Exceção	Fail

- Test Case ID: 1

- ✓ *Pré-Condições*

- Existência de uma coleção de perguntas instanciada

- ✓ *Procedimento*

- 1. Instanciar um teste
 2. Instanciar uma pergunta
 3. Adicionar a pergunta ao teste
 4. Atribuir uma resposta incorreta a essa pergunta
 5. Avaliar a questão
 6. Adquirir as estatísticas do teste

- ✓ *Output Esperado*

- 0

- ✓ *Output Real*

- 0

- ✓ *Observações Finais*

- O método funciona corretamente para este caso de teste.

- Test Case ID: 2

- ✓ *Pré-Condições*

- Existência de uma coleção de perguntas instanciada

✓ *Procedimento*

1. Instanciar um teste
2. Instanciar uma pergunta
3. Adicionar a pergunta ao teste
4. Atribuir uma resposta correta a essa pergunta
5. Avaliar a questão
6. Adquirir as estatísticas do teste

✓ *Output Esperado*

1

✓ *Output Real*

1

✓ *Observações Finais*

O método funciona corretamente para este caso de teste.

- Test Case ID: 3

✓ *Pré-Condições*

Existência de uma coleção de perguntas instanciada

✓ *Procedimento*

1. Instanciar um teste
2. Instanciar uma pergunta

3. Adicionar a primeira pergunta ao teste
4. Atribuir uma resposta incorreta a essa pergunta
5. Avaliar a questão
6. Adicionar mais 9 perguntas com respostas corretas ao teste
7. Avaliar perguntas
8. Adquirir as estatísticas do teste

✓ *Output Esperado*

0.9

✓ *Output Real*

0

✓ *Observações Finais*

O método não funciona corretamente para este caso de teste.

- Test Case ID: 4

✓ *Pré-Condições*

Existência de uma coleção de perguntas instanciada

✓ *Procedimento*

1. Instanciar um teste
2. Instanciar uma pergunta
3. Adicionar 9 perguntas com respostas corretas ao teste

4. Avaliar perguntas
5. Adicionar a última pergunta ao teste
6. Atribuir uma resposta incorreta a essa pergunta
7. Avaliar a questão
8. Adquirir as estatísticas do teste

✓ *Output Esperado*

0.9

✓ *Output Real*

0

✓ *Observações Finais*

O método não funciona corretamente para este caso de teste.

- Test Case ID: 5

✓ *Pré-Condições*

Existência de uma coleção de perguntas instanciada

✓ *Procedimento*

1. Instanciar um teste
2. Instanciar uma pergunta
3. Adicionar a primeira pergunta ao teste
4. Atribuir uma resposta correta a essa pergunta

5. Avaliar a questão
6. Adicionar 9 perguntas com respostas incorretas ao teste
7. Avaliar perguntas
8. Adquirir as estatísticas do teste

✓ *Output Esperado*

0.1

✓ *Output Real*

0

✓ *Observações Finais*

O método não funciona corretamente para este caso de teste.

- Test Case ID: 6

✓ *Pré-Condições*

Existência de uma coleção de perguntas instanciada

✓ *Procedimento*

1. Instanciar um teste
2. Instanciar uma pergunta
3. Adicionar 9 perguntas com respostas incorretas ao teste
4. Avaliar perguntas
5. Adicionar a última pergunta ao teste

6. Atribuir uma resposta correta a essa pergunta
7. Avaliar a questão
8. Adquirir as estatísticas do teste

✓ *Output Esperado*

0.1

✓ *Output Real*

0

✓ *Observações Finais*

O método não funciona corretamente para este caso de teste.

- Test Case ID: 7

✓ *Pré-Condições*

Existência de uma coleção de perguntas instanciada

✓ *Procedimento*

1. Instanciar um teste
2. Instanciar uma pergunta
3. Adicionar 10 perguntas com respostas corretas ao teste
4. Avaliar perguntas
5. Adquirir as estatísticas do teste

✓ *Output Esperado*

1

✓ *Output Real*

1

✓ *Observações Finais*

O método funciona corretamente para este caso de teste.

- Test Case ID: 8

✓ *Pré-Condições*

Existência de uma coleção de perguntas instanciada

✓ *Procedimento*

1. Instanciar um teste
2. Instanciar uma pergunta
3. Adicionar 10 perguntas com respostas incorretas ao teste
4. Avaliar perguntas
5. Adquirir as estatísticas do teste

✓ *Output Esperado*

0

✓ *Output Real*

0

✓ *Observações Finais*

O método funciona corretamente para este caso de teste.

- Test Case ID: 9

✓ *Pré-Condições*

Existência de uma coleção de perguntas instanciada

✓ *Procedimento*

1. Instanciar um teste
2. Adquirir as estatísticas do teste

✓ *Output Esperado*

Exceção que permita saber que não se pode devolver a percentagem de respostas corretas pois o teste não contém questões

✓ *Output Real*

0

✓ *Observações Finais*

O método não funciona corretamente para este caso de teste.

- Test Case ID: 10

✓ *Pré-Condições*

Existência de uma coleção de perguntas instanciada

✓ *Procedimento*

1. Instanciar um teste
2. Instanciar 10 perguntas
3. Adicionar as 10 perguntas ao teste
4. Atribuir uma resposta correta a cada pergunta
5. Marcar as 10 perguntas como feitas
6. Adquirir as estatísticas do teste

✓ *Output Esperado*

Exceção que permita saber que não se pode devolver a percentagem de respostas corretas pois as respostas não estão avaliadas

✓ *Output Real*

1

✓ *Observações Finais*

O método não funciona corretamente para este caso de teste.

- Test Case ID: 11

✓ *Pré-Condições*

Existência de uma coleção de perguntas instanciada

✓ *Procedimento*

1. Instanciar um teste
2. Instanciar 10 perguntas

3. Adicionar as 10 perguntas ao teste
4. Marcar as 10 perguntas como por fazer
5. Adquirir as estatísticas do teste

✓ *Output Esperado*

Exceção que permita saber que não se pode devolver a percentagem de respostas corretas pois as respostas não estão terminadas, ou seja, o teste não está completo.

✓ *Output Real*

0

✓ *Observações Finais*

O método não funciona corretamente para este caso de teste.

- incorrectAnswersPercentage()

Data Execução	Hora Execução	Tester
23/11/2019	14:31	Rebeca Teixeira

Pré-Condition			existência de uma coleção de perguntas instanciada				
Test Case ID	Use Case	Input	Valores do Inputs	Valid Equivalence Class and Bounds covered	Invalid Equivalence Class and Bounds covered	Resultado Esperado	Pass/Fail/Untested
1	incorrectAnswersPercentage()	Questions[]	Apenas uma pergunta e está errada	EC27		1	Pass
2			Apenas uma pergunta e está certa	EC27		0	Pass
3			Primeira pergunta errada e as restantes certas	EC27 (LB)		0.1	Fail
4			Última pergunta errada e as restantes certas	EC27 (LB)		0.1	Fail
5			Primeira pergunta certa e as restantes erradas	EC27 (BUB)		0.9	Fail
6			Última pergunta certa e as restantes erradas	EC27 (BUB)		0.9	Fail
7			Todas as perguntas certas	EC27		0	Pass
8			Todas as perguntas erradas	EC27 (UB)		1	Pass
9			Array sem perguntas		EC28 (BLB)	Exceção	Fail
10			Array sem perguntas avaliadas		EC14 (BLB)	Exceção	Fail
11			Array sem perguntas Done (incompleto)		EC26 (BLB)	Exceção	Fail

- Test Case ID: 1

- ✓ *Pré-Condições*

- Existência de uma coleção de perguntas instanciada

- ✓ *Procedimento*

- 1. Instanciar um teste
 2. Instanciar uma pergunta
 3. Adicionar a pergunta ao teste
 4. Atribuir uma resposta incorreta a essa pergunta
 5. Avaliar a questão
 6. Adquirir as estatísticas do teste

- ✓ *Output Esperado*

- 1

- ✓ *Output Real*

- 1

- ✓ *Observações Finais*

- O método funciona corretamente para este caso de teste.

- Test Case ID: 2

- ✓ *Pré-Condições*

- Existência de uma coleção de perguntas instanciada

✓ *Procedimento*

1. Instanciar um teste
2. Instanciar uma pergunta
3. Adicionar a pergunta ao teste
4. Atribuir uma resposta correta a essa pergunta
5. Avaliar a questão
6. Adquirir as estatísticas do teste

✓ *Output Esperado*

0

✓ *Output Real*

0

✓ *Observações Finais*

O método funciona corretamente para este caso de teste.

- Test Case ID: 3

✓ *Pré-Condições*

Existência de uma coleção de perguntas instanciada

✓ *Procedimento*

1. Instanciar um teste
2. Instanciar uma pergunta

3. Adicionar a primeira pergunta ao teste
4. Atribuir uma resposta incorreta a essa pergunta
5. Avaliar a questão
6. Adicionar mais 9 perguntas com respostas corretas ao teste
7. Avaliar perguntas
8. Adquirir as estatísticas do teste

✓ *Output Esperado*

0.1

✓ *Output Real*

0

✓ *Observações Finais*

O método não funciona corretamente para este caso de teste.

- Test Case ID: 4

✓ *Pré-Condições*

Existência de uma coleção de perguntas instanciada

✓ *Procedimento*

1. Instanciar um teste
2. Instanciar uma pergunta
3. Adicionar 9 perguntas com respostas corretas ao teste

4. Avaliar perguntas
5. Adicionar a última pergunta ao teste
6. Atribuir uma resposta incorreta a essa pergunta
7. Avaliar a questão
8. Adquirir as estatísticas do teste

✓ *Output Esperado*

0.1

✓ *Output Real*

0

✓ *Observações Finais*

O método não funciona corretamente para este caso de teste.

- Test Case ID: 5

✓ *Pré-Condições*

Existência de uma coleção de perguntas instanciada

✓ *Procedimento*

1. Instanciar um teste
2. Instanciar uma pergunta
3. Adicionar a primeira pergunta ao teste
4. Atribuir uma resposta correta a essa pergunta

5. Avaliar a questão
6. Adicionar 9 perguntas com respostas incorretas ao teste
7. Avaliar perguntas
8. Adquirir as estatísticas do teste

✓ *Output Esperado*

0.9

✓ *Output Real*

0

✓ *Observações Finais*

O método não funciona corretamente para este caso de teste.

- Test Case ID: 6

✓ *Pré-Condições*

Existência de uma coleção de perguntas instanciada

✓ *Procedimento*

1. Instanciar um teste
2. Instanciar uma pergunta
3. Adicionar 9 perguntas com respostas incorretas ao teste
4. Avaliar perguntas
5. Adicionar a última pergunta ao teste

6. Atribuir uma resposta correta a essa pergunta
7. Avaliar a questão
8. Adquirir as estatísticas do teste

✓ *Output Esperado*

0.9

✓ *Output Real*

0

✓ *Observações Finais*

O método não funciona corretamente para este caso de teste.

- Test Case ID: 7

✓ *Pré-Condições*

Existência de uma coleção de perguntas instanciada

✓ *Procedimento*

1. Instanciar um teste
2. Instanciar uma pergunta
3. Adicionar 10 perguntas com respostas corretas ao teste
4. Avaliar perguntas
5. Adquirir as estatísticas do teste

✓ *Output Esperado*

0

✓ *Output Real*

0

✓ *Observações Finais*

O método funciona corretamente para este caso de teste.

- Test Case ID: 8

✓ *Pré-Condições*

Existência de uma coleção de perguntas instanciada

✓ *Procedimento*

1. Instanciar um teste
2. Instanciar uma pergunta
3. Adicionar 10 perguntas com respostas incorretas ao teste
4. Avaliar perguntas
5. Adquirir as estatísticas do teste

✓ *Output Esperado*

1

✓ *Output Real*

1

✓ *Observações Finais*

O método funciona corretamente para este caso de teste.

- Test Case ID: 9

✓ *Pré-Condições*

Existência de uma coleção de perguntas instanciada

✓ *Procedimento*

1. Instanciar um teste
2. Adquirir as estatísticas do teste

✓ *Output Esperado*

Exceção que permita saber que não se pode devolver a percentagem de respostas incorretas pois o teste não contém questões

✓ *Output Real*

0

✓ *Observações Finais*

O método não funciona corretamente para este caso de teste.

- Test Case ID: 10

✓ *Pré-Condições*

Existência de uma coleção de perguntas instanciada

✓ *Procedimento*

1. Instanciar um teste
2. Instanciar 10 perguntas
3. Adicionar as 10 perguntas ao teste
4. Atribuir uma resposta incorreta a cada pergunta
5. Marcar as 10 perguntas como feitas
6. Adquirir as estatísticas do teste

✓ *Output Esperado*

Exceção que permita saber que não se pode devolver a percentagem de respostas incorretas pois as respostas não estão avaliadas

✓ *Output Real*

1

✓ *Observações Finais*

O método não funciona corretamente para este caso de teste.

- Test Case ID: 11

✓ *Pré-Condições*

Existência de uma coleção de perguntas instanciada

✓ *Procedimento*

1. Instanciar um teste
2. Instanciar 10 perguntas

3. Adicionar as 10 perguntas ao teste
4. Marcar as 10 perguntas como por fazer
5. Adquirir as estatísticas do teste

✓ *Output Esperado*

Exceção que permita saber que não se pode devolver a percentagem de respostas incorretas pois as respostas não estão terminadas, ou seja, o teste não está completo.

✓ *Output Real*

0

✓ *Observações Finais*

O método não funciona corretamente para este caso de teste.

- QuestionMultipleChoice.answer()

Data Execução	Hora Execução	Tester
22/11/19	18:35	Catarina

Pré-Condition			user_answer != null && len(correct_answer)>0				
Test Case ID	Use Case	Input	Valores do Inputs	Valid Equivalence Class and Bounds covered	Invalid Equivalence Class and Bounds covered	Resultado Esperado	Pass/Fail/Untested
1	QuestionMultipleChoice.answer()	User_answer	VC(101)		EC18(AUB)	Não guarda a user_answer	Fail
2			VC(0)		EC18(BLB)	Não guarda a user_answer	Fail
3			VC(1)	EC17		Guarda a user_answer	Pass
4			VC(100)	EC17		Guarda a user_answer	Pass
5			"abcd"	EC17		Guarda a user_answer	Pass
6			Colocar uma opção que não está nas opções		EC29	Não guarda a user_answer	Pass

- Test Case ID: 1

- ✓ *Pré-Condições*

- User_answer!=null && Len(correct_answer)>0

- ✓ *Procedimento*

- 1. Criar uma pergunta de escolha múltipla
 2. Atribuir uma resposta (answer) com 101 caracteres
 3. Verificar se a resposta é ou não aceite, isto é, verificamos se a pergunta fica done

- ✓ *Output Esperado*

- False

- ✓ *Output Real*

- True

- ✓ *Observações Finais*

- O teste falhou pois não devia ser possível aceitar uma resposta com 101 caracteres pois está fora dos limites

- Test Case ID: 2

- ✓ *Pré-Condições*

- User_answer!=null && Len(correct_answer)>0

- ✓ *Procedimento*

- 1. Criar uma pergunta de escolha múltipla
 2. Atribuir uma resposta (answer) nula

3. Verificar se a resposta é ou não aceite, isto é, verificamos se a pergunta fica done

✓ *Output Esperado*

False

✓ *Output Real*

True

✓ *Observações Finais*

O teste falhou pois não devia ser possível aceitar uma resposta nula

- Test Case ID: 3

✓ *Pré-Condições*

User_answer!=null && Len(correct_answer)>0

✓ *Procedimento*

1. Criar uma pergunta de escolha múltipla
2. Atribuir uma resposta que está nas opções
3. Verificar se a pergunta fica done

✓ *Output Esperado*

True

✓ *Output Real*

True

✓ *Observações Finais*

O teste passou porque a pergunta ficou done logo a resposta foi aceite

- Test Case ID: 4

✓ *Pré-Condições*

User_answer!=null && Len(correct_answer)>0

✓ *Procedimento*

1. Criar uma pergunta de escolha múltipla
2. Atribuir a essa pergunta uma resposta (answer) com 100 caracteres.
3. Verificar se a pergunta fica done

✓ *Output Esperado*

True

✓ *Output Real*

True

✓ *Observações Finais*

O teste passou porque a resposta foi aceite pois 100 caracteres está dentro dos limites

- Test Case ID: 5

✓ *Pré-Condições*

User_answer!=null && Len(correct_answer)>0

✓ *Procedimento*

1. Criar uma pergunta do tipo escolha múltipla
2. Associar a essa pergunta uma resposta, neste caso “abcd”
3. Verificar se a resposta é ou não aceite

✓ *Output Esperado*

True

✓ *Output Real*

True

✓ *Observações Finais*

O teste passou, pois, a resposta foi aceite

- Test Case ID: 6

✓ *Pré-Condições*

User_answer!=null && Len(correct_answer)>0

✓ *Procedimento*

1. Criar uma pergunta do tipo escolha múltipla
2. Atribuir a essa pergunta uma resposta do utilizador que não está nas opções
3. Verificar se a pergunta está done

✓ *Output Esperado*

False

✓ *Output Real*

True

✓ *Observações Finais*

O teste não passou porque a resposta foi aceite

- QuestionMultipleChoice.evaluateAnswer()

Data Execução	Hora Execução	Tester
22/11/2019	11:57	Catarina

Pré-Condition			correct_answer != null && len(correct_answer)>0				
Test Case ID	Use Case	Input	Valores do Inputs	Valid Equivalence Class and Bounds covered	Invalid Equivalence Class and Bounds covered	Resultado Esperado	Pass/Fail/Untested
1	QuestionMultipleChoice.evaluateAnswer()	user_answer	user_answer = correct answer	EC15		true	Pass
2			user_answer != correct answer		EC16	false	Pass

- Test Case ID: 1
 - ✓ *Pré-Condições*
correct_answer!=null && Len(correct_answer)>0
 - ✓ *Procedimento*
 1. Primeiro cria-se uma questão do tipo Múltipla escolha sendo que esta tem de ter um título, uma descrição, um conjunto de opções e por último uma resposta correta.
 2. Atribuímos a esta pergunta uma resposta de utilizador e colocamos a pergunta como done (setDone).
 3. Por último avaliamos se a questão está ou não correta.
 - ✓ *Output Esperado*
True
 - ✓ *Output Real*
True
 - ✓ *Observações Finais*
O teste passou logo a resposta do utilizador foi avaliada corretamente

- Test Case ID: 2
 - ✓ *Pré-Condições*
correct_answer!=null && Len(correct_answer)>0
 - ✓ *Procedimento*
 1. Primeiro cria-se uma questão do tipo Múltipla escolha sendo que esta tem de ter um título, uma descrição, um conjunto de opções e por último uma resposta correta.
 2. Atribuímos a esta pergunta uma resposta de utilizador e colocamos a pergunta como done (setDone).
 3. Por último avaliamos se a questão está ou não correta.
 - ✓ *Output Esperado*
False
 - ✓ *Output Real*
False
 - ✓ *Observações Finais*
O teste passou logo a resposta do utilizador foi avaliada corretamente

- QuestionNumeric.answer()

Data Execução	Hora Execução	Tester
22/11/2019	17:53	Catarina

Pré-Condition			user_answer != null && len(correct_answer)>0				
Test Case ID	Use Case	Input	Valores do Inputs	Valid Equivalence Class and Bounds covered	Invalid Equivalence Class and Bounds covered	Resultado Esperado	Pass/Fail/Untested
1	QuestinNumeric.answer()	user_answer	""		EC18(BLB)	Não guarda a user_answer	Fail
2			NULL		EC18	Não guarda a user_answer	Fail
3			VC(11)		EC18(AUB)	Não guarda a user_answer	Fail
4			VC(1) numérico	EC17		Guarda a user_answer	Pass
5			VC(100) numérico	EC17		Guarda a user_answer	Pass
6			"abcd"		EC22	Não guarda a user_answer	Fail
7			"22"	EC21		Guarda a user_answer	Pass
8			VC(100) Não é tudo números			Não guarda a user_answer	Fail

- Test Case ID: 1

- ✓ *Pré-Condições*

- user_answer!=null && Len(correct_answer)>0

- ✓ *Procedimento*

- 1. Criar uma pergunta do tipo numérico
 - 2. Atribuir uma resposta (answer) neste caso a resposta é "".
 - 3. Depois verificar se a resposta é ou não aceite, isto é, verificamos se a pergunta fica done.

- ✓ *Output Esperado*

- False

- ✓ *Output Real*

- True

- ✓ *Observações Finais*

- O teste falhou pois não devia ser possível aceitar a resposta ""

- Test Case ID:2

- ✓ *Pré-Condições*

- user_answer!=null && Len(correct_answer)>0

- ✓ *Procedimento*

- 1. Primeiramente cria-se uma pergunta do tipo numérico
 - 2. Atribui-se uma resposta (answer) nula

3. No fim verifica-se se a pergunta fica done, caso fique quer dizer que a resposta foi aceite.

✓ *Output Esperado*

False

✓ *Output Real*

True

✓ *Observações Finais*

O teste falhou pois não devia ser possível aceitar uma resposta nula

- Test Case ID: 3

✓ *Pré-Condições*

user_answer!=null && Len(correct_answer)>0

✓ *Procedimento*

1. Criar uma pergunta do tipo numérico
2. Atribuir uma resposta com 101 caracteres
3. Verificar se a resposta é ou não válida

✓ *Output Esperado*

False

✓ *Output Real*

True

✓ *Observações Finais*

O teste não passou porque a resposta foi aceite e não devia ser pois a resposta à pergunta tem 101 caracteres e no máximo só podiam ser aceites 100.

- Test Case ID: 4

✓ *Pré-Condições*

user_answer!=null && Len(correct_answer)>0

✓ *Procedimento*

1. Criar uma pergunta do tipo numérico
2. Atribuir como resposta a essa pergunta um caractere numérico
3. Verificar se a pergunta está done.

✓ *Output Esperado*

True

✓ *Output Real*

True

✓ *Observações Finais*

O teste passou porque a resposta foi aceite, logo a pergunta ficou done

- Test Case ID: 5

- ✓ *Pré-Condições*

- user_answer!=null && Len(correct_answer)>0

- ✓ *Procedimento*

- 1. Cria-se uma pergunta do tipo numérico
 - 2. Atribui-se a essa mesma pergunta uma resposta com 100 caracteres numéricos
 - 3. Verificamos se essa resposta foi aceite ou não verificando se a pergunta está done.

- ✓ *Output Esperado*

- True

- ✓ *Output Real*

- True

- ✓ *Observações Finais*

- O teste passou, pois, a resposta foi aceite

- Test Case ID: 6

- ✓ *Pré-Condições*

- user_answer!=null && Len(correct_answer)>0

- ✓ *Procedimento*

- 1. Criar uma pergunta do tipo numérico
 - 2. Atribuir uma resposta do tipo “abcd”.

3. Por último verificar se a pergunta está done.

✓ *Output Esperado*

False

✓ *Output Real*

True

✓ *Observações Finais*

O teste não passou porque a resposta foi aceite

- Test Case ID: 7

✓ *Pré-Condições*

user_answer!=null && Len(correct_answer)>0

✓ *Procedimento*

1. Primeiramente cria-se uma pergunta do tipo numérico

2. Atribuir a resposta “22”

3. Verificar se a pergunta está done.

✓ *Output Esperado*

True

✓ *Output Real*

True

✓ *Observações Finais*

O teste passou porque a resposta foi aceite

- Test Case ID: 8

✓ *Pré-Condições*

user_answer!=null && Len(correct_answer)>0

✓ *Procedimento*

1. Criar uma pergunta do tipo numérico
2. Atribuir uma resposta com 100 caracteres sendo alguns numéricos e outros não
3. Por último verificar se a pergunta está done.

✓ *Output Esperado*

False

✓ *Output Real*

True

✓ *Observações Finais*

O teste não passou porque a resposta foi aceite sendo que só devia ser possível aceitar números

- QuestionNumeric.evaluateAnswer()

DataExecução	Hora Execução	Tester
22/11/2019	12:03	Catarina

Pré-Condition			correct_answer != null && len(correct_answer)>0				
Test Case ID	Use Case	Input	Valores do Inputs	Valid Equivalence Class and Bounds covered	Invalid Equivalence Class and Bounds covered	Resultado Esperado	Pass/Fail/Untested
1	QuestionNumeric.evaluateAnswer()	user_answer	user_answer = correct answer	EC15		true	Pass
2			user_answer != correct answer		EC16	false	Pass

- Test Case ID: 1

- ✓ *Pré-Condições*

- `correct_answer!=null && Len(correct_answer)>0`

- ✓ *Procedimento*

- 1. Criar uma questão do tipo numérico (QuestionNumeric) sendo que esta tem de ter uma resposta correta associada, um título e uma descrição.
 - 2. Atribuir a esta pergunta uma resposta de utilizador.
 - 3. Por último avaliamos se a questão está ou não correta.

- ✓ *Output Esperado*

- True

- ✓ *Output Real*

- True

- ✓ *Observações Finais*

- O teste passou logo a resposta do utilizador foi avaliada corretamente

- Test Case ID: 2

- ✓ *Pré-Condições*

- `correct_answer!=null && Len(correct_answer)>0`

- ✓ *Procedimento*

- 1. Criar uma questão do tipo numérico (QuestionNumeric) sendo que esta tem de ter uma resposta correta associada, um título e uma descrição.
 2. Depois atribuímos à pergunta uma resposta de utilizador.
 3. Por último avaliamos se a questão está ou não correta.

- ✓ *Output Esperado*

- False

- ✓ *Output Real*

- False

- ✓ *Observações Finais*

- O teste passou logo a resposta do utilizador foi avaliada corretamente

- QuestionYesNo.answer()

Dta Execução	Hora Execução	Tester
22/11/2019	17:39	Catarina

Pré-Condition			user_answer != null && len(correct_answer)>0				
Test Cas ID	Use Case	Input	Valores do Inputs	Valid Equivalence Class and Bounds covered	Invalid Equivalence Class and Bounds covered	Resultado Esperado	Pass/Fail/Untested
1	QuestinYesNo.answer()	user_answer	""		EC18(BLB)	Não guarda a user_answer	Fail
2			NULL		EC18	Não guarda a user_answer	Fail
3			VC(4)		EC18(AUB)	Não guarda a user_answer	Fail
4			VC(1)		EC18(BLB)	Não guarda a user_answer	Fail
5			YES	EC17		Guarda a user_answer	Pass
6			NO	EC17		Guarda a user_answer	Pass

- Test Case ID: 1

- ✓ *Pré-Condições*

- user_answer!=null && Len(correct_answer)>0

- ✓ *Procedimento*

- 1. Criar uma pergunta do tipo yesNo
 - 2. Atribuir uma resposta (answer).
 - 3. Depois verificar se a resposta é ou não aceite, isto é, verificamos se a pergunta fica done.

- ✓ *Output Esperado*

- False

- ✓ *Output Real*

- True

- ✓ *Observações Finais*

- O teste falhou pois não devia ser possível aceitar a resposta ""

- Test Case ID: 2

- ✓ *Pré-Condições*

- user_answer!=null && Len(correct_answer)>0

- ✓ *Procedimento*

- 1. Primeiramente cria-se uma pergunta do tipo YesNo
 - 2. Atribui-se uma resposta (answer) nula.
 - 3. No fim verifica-se se a pergunta fica done, caso fique done quer dizer que a resposta foi aceite.

- ✓ *Output Esperado*

- False

- ✓ *Output Real*

- True

- ✓ *Observações Finais*

- O teste falhou pois não devia ser possível aceitar uma resposta nula

- Test Case ID: 3

- ✓ *Pré-Condições*

- user_answer!=null &&Len(correct_answer)>0

- ✓ *Procedimento*

- 1. Criar uma pergunta do tipo YesNo
 - 2. Atribuir uma resposta com 4 caracteres.

3. No final verificar se a resposta é ou não válida

✓ *Output Esperado*

False

✓ *Output Real*

True

✓ *Observações Finais*

O teste não passou porque a resposta foi aceite e não devia ser pois a resposta no máximo só pode ter 3 caracteres que corresponde à resposta yes

- Test Case ID: 4

✓ *Pré-Condições*

user_answer!=null &&Len(correct_answer)>0

✓ *Procedimento*

1. Criar uma pergunta do tipo YesNo
2. Atribuir como resposta a essa pergunta um único caracter.
3. No final verificar se a pergunta está done.

✓ *Output Esperado*

False

✓ *Output Real*

True

✓ *Observações Finais*

O teste não passou porque a resposta foi aceite e não devia ter sido pois nenhuma das respostas possíveis, isto é, Yes ou No tem apenas um único carácter.

- Test Case ID: 5

✓ *Pré-Condições*

user_answer!=null && Len(correct_answer)>0

✓ *Procedimento*

1. Em primeiro lugar cria-se uma pergunta do tipo YesNo
2. Depois atribui-se a essa mesma pergunta a resposta Yes
3. Verificamos se esta resposta foi aceite ou não verificando se a pergunta está done.

✓ *Output Esperado*

True

✓ *Output Real*

True

✓ *Observações Finais*

O teste passou, pois, a resposta foi aceite

- Test Case ID: 6

- ✓ *Pré-Condições*

- user_answer!=null && Len(correct_answer)>0

- ✓ *Procedimento*

- 1. Primeiramente cria-se uma pergunta do tipo YesNo
 - 2. Atribui-se a resposta No.
 - 3. Por último verifica-se se a pergunta está done.

- ✓ *Output Esperado*

- True

- ✓ *Output Real*

- True

- ✓ *Observações Finais*

- O teste passou porque a resposta foi aceite

- QuestionYesNo.evaluateAnswer()

Data Execução	Hora Execução	Tester
2212019	13:16	Catarina

Pré-Condition			correct_answer != null && len(correct_answer)>0				
Test Case ID	Use Case	Input	Valores do Inputs	Valid Equivalence Class and Bounds covered	Invalid Equivalence Class and Bounds covered	Resultado Esperado	Pass/Fail/Untested
1	QuestionYesNo.evaluateAnswer()	user_answer	user_answer = correct answer	EC15		true	Pass
2			user_answer != correct answer		EC16	false	Pass
3			User_answer="Yes" Correct_answer="yes"	EC16		true	Fail

- Test Case ID: 1

- ✓ *Pré-Condições*

- correct_answer!=null && Len(correct_answer)>0

- ✓ *Procedimento*

- 1. Primeiro cria-se uma questão do tipo YesNo (QuestionYesNo) sendo que esta tem de ter um título e uma descrição.
 - 2. Depois atribuímos a esta pergunta uma resposta correta e uma resposta do utilizador.
 - 3. Por último avaliamos se a questão está ou não correta.

- ✓ *Output Esperado*

- True

- ✓ *Output Real*

- True

- ✓ *Observações Finais*

- O teste passou logo a resposta do utilizador foi avaliada corretamente

- Test Case ID: 2

- ✓ *Pré-Condições*

- correct_answer!=null && Len(correct_answer)>0

- ✓ *Procedimento*

- 1. Primeiro cria-se uma questão do tipo YesNo (QuestionYesNo) sendo que esta tem de ter um título e uma descrição.
 - 2. Depois atribuímos a esta pergunta uma resposta correta e uma resposta do utilizador.

3. Por último avaliamos se a questão está ou não correta.

✓ *Output Esperado*

False

✓ *Output Real*

False

✓ *Observações Finais*

O teste passou logo a resposta do utilizador foi avaliada corretamente

- Test Case ID: 3

✓ *Pré-Condições*

`correct_answer!=null && Len(correct_answer)>0`

✓ *Procedimento*

1. Primeiro cria-se uma questão do tipo YesNo (QuestionYesNo) sendo que esta tem de ter um título e uma descrição.
2. Depois atribuímos a esta pergunta uma resposta correta neste caso “Yes”
3. Atribuir “yes” como resposta do utilizador
4. Por último avaliamos se a questão está ou não correta.

✓ *Output Esperado*

True

✓ *Output Real*

False

✓ *Observações Finais*

O teste não passou porque o programa é *case sensitive*, isto é, faz distinção entre letras maiúsculas e minúsculas. Não foi testado para os outros métodos este caso de teste pois nada nos leva a pensar que nos outros métodos seria diferente.