

TEST WORKSCHEDULE – FP (BLACKBOX TEST #2)

TEST CASE

Version 1.0

20/11/2023

Histórico de Versões

Version #	Implemented By	Revision Date	Approved By	Approval Date	Reason
1.0	Maria Dias	20/11/2023	All team	20/11/2023	Criação do documento.
1.0	Maria Dias	21/11/2023	All team	21/11/2023	Classes de ECP e BVA.
1.0	Maria Dias	21/11/2023	All team	21/11/2023	Casos de testes sobre o método de verificar se estabelece o número de employees necessários
1.0	Maria Dias	21/11/2023	All team	21/11/2023	Casos de testes sobre o método de verificar se escalona employee para trabalhar
1.0	Maria Dias	21/11/2023	All team	21/11/2023	Casos de testes sobre o método de verificar se retorna os employees que foram escalonados
1.0	Maria Dias	21/11/2023	All team	21/11/2023	Casos de testes sobre o método de verificar se retorna o tempo mais próximo de employees necessários ainda não foram escalonados

Tabela de Conteúdos

1. Introdução	4
1.1. Identificador do documento	4
1.2. Âmbito	4
1.3. References.....	4
1.4 Glossário	4
2. Detalhes.....	4
• <i>Verificar se estabelece o número de employees necessários.....</i>	4
• <i>Verificar se escalona employee para trabalhar</i>	5
• <i>Verificar se retorna os employees que foram escalonados</i>	5
• <i>Verificar se retorna o tempo mais próximo de employees necessários ainda não foram escalonados</i>	6
2.1. Casos de teste.....	6
• <i>Verificar se estabelece o número de employees necessários.....</i>	6
• <i>Verificar se escalona employee para trabalhar</i>	7
• <i>Verificar se retorna os employees que foram escalonados</i>	7
• <i>Verificar se retorna o tempo mais próximo de employees necessários ainda não foram escalonados</i>	8

1. Introdução

1.1. Identificador do documento

TestCaseWorkSchedule

1.2. Âmbito

Este documento tem como objetivo especificar o intervalo de testes a serem implementados, as técnicas utilizadas e a documentação desses mesmos testes de forma a garantir a qualidade de software.

1.3. References

Documentação fornecida pela unidade curricular de Engenharia de Software II.

1.4 Glossário

Testes caixa Preta	É um tipo de testes de <i>software</i> , onde o foco é sobre o comportamento do sistema, não considerando a implementação, mas sim o resultado das operações. Com este tipo de testes é verificado se o sistema tem o comportamento esperado.
ECP (<i>Equivalence Class Partitioning</i>)	É uma técnica para a escolha de testes com o intuito de reduzir o número de testes necessários, fazendo com que haja uma seleção sobre o domínio em classes tanto dos <i>inputs</i> como <i>outputs</i> , onde assim é possível abranger todo o domínio válido e inválido com um número reduzido de testes.
BVA (<i>Boundary Value Analysis</i>)	É uma técnica para a escolha de testes, onde é focado os limites dos testes nos limites e valores especiais (p.ex. nulo ou 0), neste tipo de testes não é abrangido os valores intermédios. É uma estratégia que tenta contrariar o facto de que uma boa parte das anomalias de <i>software</i> ocorrem em valores de fronteira.

2. Detalhes

Este documento irá apenas ser responsável por toda a documentação relativa aos testes realizados sobre a classe WorkSchedule. A leitura e compreensão deste documento pressupõe o conhecimento do projeto e o seu âmbito geral.

• Verificar se estabelece o número de employees necessários

ECP (Equivalence Class Partition) Classe: WorkSchedule Método: public void setRequiredNumber(int nemployee, int starttime, int endtime)					
Pressuposto	Verificar se estabelece o número de employees necessários para todas as horas entre o intervalo starttime a endtime.				
Pré-Condições	Ter uma instância de WorkSchedule no sistema.				
Casos de Uso	Critérios	Classe Válida	[ECP]	Classe Inválida	[ECP]
Verificar se estabelece o número de employees necessários.	Nº argumentos de entrada	3	[ECP1]	!= 3	[ECP2]
	Tipo argumento de entrada	int nemployee, int starttime, int endtime		!= (int nemployee, int starttime, int endtime)	
	Critérios de validade	starttime >= 0 && endtime < size && starttime <= endtime && nemployee >= 0		starttime < 0 endtime >= size starttime > endtime nemployee < 0	
	Resultado esperado	atualização bem-sucedida da tabela para as horas especificadas com requiredNumber correto		"IndexOutOfBoundsException"	
	Exemplo	setRequiredNumber(5, 3, 6)		setRequiredNumber(1, -2, 4)	

BVA (Boundary Value Analysis) Classe: WorkSchedule Método: public void setRequiredNumber(int nemployee, int starttime, int endtime)		
Inputs	min	max
nemployee	0	MAX_INT

starttime	0	Size - 1
endtime	0	size - 1

- Verificar se escalona employee para trabalhar

ECP (Equivalence Class Partition) Classe: WorkSchedule Método: public boolean addWorkingPeriod(String employee, int starttime, int endtime)					
Pressuposto	Verificar se escalona employee para trabalhar durante as horas entre starttime a endtime.				
Pré-Condições	Ter uma instância de WorkSchedule no sistema. Definir o método setRequiredNumber no sistema.				
Casos de Uso	Critérios	Classe Válida	[ECP]	Classe Inválida	[ECP]
Verificar se escalona employee para trabalhar.	Nº argumentos de entrada	3	[ECP3]	!= 3	[ECP4]
	Tipo argumento de entrada	Stringf employee, int starttime, int endtime		!= (String employee, int starttime, int endtime)	
	Critérios de validade	employee != null && employee != "" && starttime >= 0 && starttime <= endtime && endtime < size		employee = null employee = "" starttime < 0 starttime > endtime endtime > size	
	Resultado esperado	True (adição do periodo de trabalho for bem-sucedida)		False (nenhum efeito no calendário de trabalho)	
	Exemplo	addWorkingPeriod("John", 3, 6)		addWorkingPeriod("David", 2, 8)	

BVA (Boundary Value Analysis) Classe: WorkSchedule Método: public boolean addWorkingPeriod (String employee, int starttime, int endtime)		
Inputs	min	max
employee	Non-null string	MAX_STRING
starttime	0	Size - 1
endtime	0	Size - 1

- Verificar se retorna os employees que foram escalonados

ECP (Equivalence Class Partition) Classe: WorkSchedule Método: public String[] workingEmployees(int starttime, int endtime)					
Pressuposto	Verificar se retorna os nomes dos employees que, até ao momento, foram escalonados para trabalhar naquela hora.				
Pré-Condições	Ter uma instância de WorkSchedule no sistema. Definir o método setRequiredNumber no sistema. Definir o método addWorkingPeriod no sistema.				
Casos de Uso	Critérios	Classe Válida	[ECP]	Classe Inválida	[ECP]
Verificar se escalona employee para trabalhar.	Nº argumentos de entrada	2	[ECP5]	!= 2	[ECP6]
	Tipo argumento de entrada	int starttime, int endtime		!= (int starttime, int endtime)	
	Critérios de validade	starttime >= 0 && starttime <= endtime && endtime < size		starttime < 0 starttime > endtime endtime >= size	
	Resultado esperado	["John Doe", "Alice Dias"]		[]	
	Exemplo	workingEmployees(3, 6)		workingEmployees(2, 8)	

BVA (Boundary Value Analysis) Classe: WorkSchedule Método: public String[] workingEmployees(int starttime, int endtime)		
Inputs	min	max
starttime	0	Size - 1
endtime	0	Size - 1

- Verificar se retorna o tempo mais próximo de employees necessários ainda não foram escalonados

ECP (Equivalence Class Partition) Classe: WorkSchedule Método: public int nextIncomplete(int currenttime)					
Pressuposto	Verificar se retorna o tempo mais próximo a contar o tempo/hora atual para o qual a totalidade de employees necessários ainda não foi escalonada.				
Pré-Condições	Ter uma instância de WorkSchedule(size) no sistema.				
Casos de Uso	Critérios	Classe Válida	[ECP]	Classe Inválida	[ECP]
Verificar se retorna o tempo mais próximo de employees necessários ainda não foram escalonados	Nº argumentos de entrada	1	[ECP7]	!= 1	[ECP8]
	Tipo argumento de entrada	int currenttime		!= (int currenttime)	
	Critérios de validade	currenttime >= 0 && currenttime < size		currenttime < 0 currenttime > size	
	Resultado esperado	3		IndexOutOfBoundsException	
	Exemplo	nextIncomplete(3)		nextIncomplete(-2)	

BVA (Boundary Value Analysis) Classe: WorkSchedule Método: public int nextIncomplete(int currenttime)			
Inputs	min	max	
currenttime	0	Size - 1	

2.1. Casos de teste

- Verificar se estabelece o número de employees necessários

Test Case ID	testSetRequiredNumber_Valid	Dependências	
--------------	-----------------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
21/11/2023	21:25	Maria Dias	Pass	

Pré-condições / estado inicial				
Ter uma instância WorkSchedule no sistema.				
Procedimento / Passos do user para esta função				
1. Executar o método construtor. 2. Executar o método setRequiredNumber.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
[ECP1] Verificar se estabelece o número de employees necessários	int nemployee, int starttime, int endtime	5, 3, 6	Atualização bem-sucedida da tabela para as horas especificadas com requiredNumber correto	Pass

Test Case ID	testSetRequiredNumber_ReturnIndexOutOfBoundsException_Valid	Dependências	
--------------	---	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
21/11/2023	21:30	Maria Dias	Pass	

Pré-condições / estado inicial				
Ter uma instância WorkSchedule no sistema.				
Procedimento / Passos do user para esta função				

1. Executar o método construtor. 2. Executar o método setRequiredNumber.				
Requisito/Use Case/Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
[ECP2] Verificar se estabelece o número de employees necessários	int nemployee, int starttime, int endtime	1, -2, 4	IndexOutOfBoundsException	Pass

• Verificar se escalona employee para trabalhar

Test Case ID	testAddWorkingPeriod_ReturnTrue_Valid	Dependências			
Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha	
21/11/2023	21:30	Maria Dias	Pass		
Pré-condições / estado inicial					
Ter uma instância WorkSchedule no sistema.					
Procedimento / Passos do user para esta função					
1. Executar o método construtor. 2. Executar o método addWorkingPeriod. 3. Executar o método setRequiredNumber.					
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested	
[ECP3] Verificar se escalona employee para trabalhar	String employee, int starttime, int endtime	"John", 3, 6	True	Pass	

Test Case ID	testAddWorkingPeriod_ReturnFalse_Valid	Dependências			
Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha	
21/11/2023	21:30	Maria Dias	Pass		
Pré-condições / estado inicial					
Ter uma instância WorkSchedule no sistema.					
Procedimento / Passos do user para esta função					
4. Executar o método construtor. 1. Executar o método addWorkingPeriod. 2. Executar o método setRequiredNumber.					
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested	
[ECP4] Verificar se escalona employee para trabalhar	String employee, int starttime, int endtime	"David", 2, 8	False	Pass	

• Verificar se retorna os employees que foram escalonados

Test Case ID	testWorkingEmployees_ReturnList_Valid	Dependências			
Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha	

21/11/2023	21:32	Maria Dias	Pass	
------------	-------	------------	------	--

Pré-condições / estado inicial				
Ter uma instância WorkSchedule no sistema.				
Procedimento / Passos do user para esta função				
1. Executar o método construtor. 2. Executar o método setRequiredNumber. 3. Executar o método addWorkingPeriod. 4. Executar o método workingEmployees.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
[ECP5] Verificar se retorna os employees que foram escalonados	int starttime, int endtime	3, 6	["John Doe", "Alice Dias"]	Pass

Test Case ID	testWorkingEmployees_ReturnEmptyList_Valid	Dependências	
--------------	--	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
21/11/2023	21:33	Maria Dias	Pass	

Pré-condições / estado inicial				
Ter uma instância WorkSchedule no sistema.				
Procedimento / Passos do user para esta função				
1. Executar o método construtor. 2. Executar o método setRequiredNumber. 3. Executar o método addWorkingPeriod. 4. Executar o método workingEmployees.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
[ECP6] Verificar se retorna os employees que foram escalonados	int starttime, int endtime	2, 8	[]	Pass

- Verificar se retorna o tempo mais próximo de employees necessários ainda não foram escalonados

Test Case ID	testNextIncomplete_ReturnTime_Valid	Dependências	
--------------	-------------------------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
21/11/2023	21:35	Maria Dias	Pass	

Pré-condições / estado inicial				
Ter uma instância WorkSchedule no sistema.				
Procedimento / Passos do user para esta função				
1. Executar o método construtor. 2. Executar o método setRequiredNumber. 3. Executar o método nextIncomplete.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
[ECP7] Verificar se retorna o tempo mais próximo de employees	int currenttime	3	3	Pass

necessários ainda não foram escalonados				
---	--	--	--	--

Test Case ID	testNextIncomplete_ReturnIndexOutOfBoundsException_Valid	Dependências	
---------------------	--	---------------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
21/11/2023	21:35	Maria Dias	Pass	

Pré-condições / estado inicial				
Ter uma instância WorkSchedule no sistema.				
Procedimento / Passos do user para esta função				
1. Executar o método construtor. 2. Executar o método setRequiredNumber. 3. Executar o método nextIncomplete.				
Requisito/User Case/Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
[ECP7] Verificar se retorna o tempo mais próximo de employees necessários ainda não foram escalonados	int currenttime	-2	IndexOutOfBoundsException	Pass