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	O	4
1.1. Identifi	icador do documento	. 4
1.2. Âmbito	o	. 4
1.3. Refere	nces	. 4
1.4 Glossá	rio	. 4
2. Detalhes		4
 Verifical 	r se estabelece o número de employees necessários	4
 Verifical 	r se escalona employee para trabalhar	. 5
 Verifical 	r se retorna os employees que foram escalonados	. 5
-	r se retorna o tempo mais próximo de employees necessários ainda não fora	
2.1. Casos	de teste	.6
 Verifical 	r se estabelece o número de employees necessários	6
 Verifical 	r se escalona employee para trabalhar	. 7
 Verifical 	r se retorna os employees que foram escalonados	. 7
	r se retorna o tempo mais próximo de employees necessários ainda não fora	_
escalonados		. 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 (Equivalence Class Partitioning)	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 (Boundary Value Analysis)	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

	• Vermedi se estabelece o namero de employees necessarios						
	ECP (Equivalence Class Partition) Classe: WorkSchedule Método: public void setRequiredNumber(int nemployee, int starttime, int endtime)						
Pressuposto	Verificar se estabe endtime.	elece o número de employees necess	ários para	todas as horas entre o intervalo star	ttime a		
Pré- Condições	Ter uma instância	de WorkSchedule no sistema.					
Casos de Uso	Critérios	Classe Válida	[ECP]	Classe Inválida	[ECP]		
	Nº argumentos de entrada	3		i= 3			
Varificar on	Tipo argumento de entrada	int nemployee, int starttime, int endtime		!= (int nemployee, int starttime, int endtime)			
Verificar se estabelece o número de employees necessários.	Critérios de validade	starttime >= 0 && endtime < size && starttime <= endtime && nemployee >= 0	[ECP1]	starttime < 0 endtime >= size starttime > endtime nemployee < 0	[ECP2]		
	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	Inputs min max					
nemployee	nemployee 0 MAX_INT					

starttime	0	Size - 1
endtime	0	size - 1

• Verificar se escalona employee para trabalhar

	vermed be esserted employee para crasamar						
	ECP (Equivalence Class Partition) Classe: WorkSchedule Método: public boolean addWorkingPeriod(String employee, int starttime, int endtime)						
Pressuposto		ona employee para trabalhar durante a					
Pré-		de WorkSchedule no sistema.					
Condições	Definir o método s	setRequiredNumber no sistema.					
Casos de Uso	Critérios	Classe Válida	[ECP]	Classe Inválida	[ECP]		
	Nº argumentos de entrada	3		!= 3			
Verificar se	Tipo argumento de entrada	Stringf employee, int starttime, int endtime		!= (String employee, int starttime, int endtime)			
escalona employee para trabalhar.	Critérios de validade	employee != null && employee != "" && starttime >= 0 && starttime <= endtime && endtime < size	[ECP3]	employee = null employee = "" starttime < 0 starttime > endtime endtime > size	[ECP4]		
	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)			

Métod	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

	Vermear se retorna os emproyees que roram escaronados					
	ECP (Equivalence Class Partition) Classe: WorkSchedule					
		odo: public String[] workingEmployees				
Pressuposto	Verificar se retorn naquela hora.	a os nomes dos employees que, até a	no momen	to, foram escalonados para trabalhar	-	
Pré- Condições	Definir o método s	i de WorkSchedule no sistema. setRequiredNumber no sistema. addWorkingPeriod no sistema.				
	T		1		ı	
Casos de Uso	Critérios	Classe Válida	[ECP]	Classe Inválida	[ECP]	
	Nº argumentos de entrada	2		!= 2		
Verificar se	Tipo argumento de entrada	int starttime, int endtime		!= (int starttime, int endtime)		
escalona employee para trabalhar.	Critérios de validade	starttime >= 0 && starttime <= endtime && endtime < size	[ECP5]	starttime < 0 starttime > endtime endtime >= size	[ECP6]	
	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	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		a o tempo mais próximo a contar o te a não foi escalonada.	empo/hora	a atual para o qual a totalidade de em	ployees	
Pré- Condições	Ter uma instância	de WorkSchedule(size) no sistema.				
Casos de Uso	Critérios	Classe Válida	[ECP]	Classe Inválida	[ECP]	
Verificar se retorna o	Nº argumentos de entrada	1		!= 1		
tempo mais próximo de	Tipo argumento de entrada	int currenttime	!= (int currenttime)			
employees necessários	Critérios de validade	currenttime >= 0 && currenttime < size	[ECP7]	currenttime < 0 currenttime > size	[ECP8]	
ainda não foram	Resultado esperado	3		IndexOutOfBoundsException		
escalonados	Exemplo	nextIncomplete(3)		nextIncomplete(-2)		

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

2.1. Casos de teste

Procedimento / Passos do user para esta função

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 W	orkSchedule no sistema.						
Procedimento / Pass	os do <i>user</i> para esta função						
1. Executar o méto	odo construtor.						
Executar o méto	odo setRequiredNumber.						
Requisito/Use	Inputs	Valores do Inputs	Resultados	Pass/Fail/Untested			
Case/			Esperados				
Funcionalidade							
[ECP1]			Atualização bem-				
Verificar se			sucedida da tabela				
estabelece o	int nemployee, int starttime,	5, 3, 6	para as horas	Pass			
número de	int endtime	5, 5, 0	especificadas com	1 055			
employees			requiredNumber				
necessários			correto				

Test Case ID	testSetRequiredNumber_ReturnIndexOutOfBoundE xception_Valid	Dependên cias		
Data de	Hora de Execução	Tester	Pass/Fail	Resumo da

21/11/2023 21:30	Maria Dias	Pass	
Pré-condições / estado inicial			
Ter uma instância WorkSchedule no sistema.			

	Executar o método construtor. Executar o método setRequiredNumber.					
Requisito/U se Case/ Funcionalid ade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Unte sted		
[ECP2] Verificar se estabelece o número de employees necessários	int nemployee, int starttime, int endtime	1, -2, 4	IndexOutOfBoundsExc eption	Pass		

• Verificar se escalona employee para trabalhar

Test Case ID	testAddWorkingPeriod_ReturnTrue_Valid	Dependências	

Data de Execução F	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
1 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

Executar o método construtor.

- 2. Executar o método addWorkingPeriod.
- Executar o método setRequiredNumber.

Requisito/Use Case/	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Funcionalidade		mpato	Loperadoo	
[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.
- Executar o método addWorkingPeriod.
 Executar o método setRequiredNumber.

	2. Execution of metodo serveguireary arriber.				
Requisito/Use	Inputs	Valores do	Resultados	Pass/Fail/Untested	
Case/		Inputs	Esperados		
Funcionalidade					
[ECP4]					
Verificar se	String employee, int starttime, int				
escalona	endtime	"David", 2, 8	False	Pass	
employee para	endunte				
trabalhar					

• Verificar se retorna os employees que foram escalonados

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

21/11/2023	21:32	Maria Dias	Pass				
Pré-condições / est	Pré-condições / estado inicial						
Ter uma instância V	WorkSchedule no sistema.						
Procedimento / Pas	ssos do <i>user</i> para esta função						
1. Executar o mé	todo construtor.						
2. Executar o mé	todo setRequiredNumber.						
3. Executar o mé	todo addWorkingPeriod.						
4. Executar o mé	todo workingEmployees.						
Requisito/Use	Inputs	Valores do	Resultados	Pass/Fail/Untested			
Case/		Inputs	Esperados				
Funcionalidade							
[ECP5]							
Verificar se							
retorna os	int starttime, int endtime	3, 6	["John Doe",	Pass			
employees que	int startume, intendume	5, 0	"Alice Dias"]	1 055			
foram							

escalonados

escalonados

Test Case ID	testWorkingEmployees_ReturnEmptyList_Valid	Dependências			
Data de	Hora de Execução	Tester	Pass/Fail	Resumo da Falha	
Execução					
21/11/2023	21:33	Maria Dias	Pass		
Pré-condições / es	stado inicial				
Ter uma instância	WorkSchedule no sistema.				
Procedimento / Pa	assos do <i>user</i> para esta função				
 Executar o m 	Executar o método construtor.				
Executar o m	2. Executar o método setRequiredNumber.				
Executar o m	3. Executar o método addWorkingPeriod.				
Executar o m	nétodo workingEmployees.				
Requisito/Use	Inputs	Valores do	Resultados	Pass/Fail/Untested	
Case/		Inputs	Esperados		
Funcionalidade					
[ECP6]					
Verificar se					
retorna os	int starttime, int endtime	2, 8	П	Pass	
employees que	int starttime, intenutime	۷, ٥	П	F d 5 5	
foram					

Verificar se retorna o tempo mais próximo de employees necessários ainda não foram escalonados 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 / est	ado inicial			
Ter uma instância V	VorkSchedule no sistema.			
Procedimento / Pas	sos do <i>user</i> para esta função			
1. Executar o mé	todo construtor.			
2. Executar o método setRequiredNumber.				
3. Executar o método nextIncomplete.				
Requisito/Use	Inputs	Valores do Inputs	Resultados	Pass/Fail/Untested
Case/			Esperados	
Funcionalidade				
[ECP7]				
Verificar se				
retorna o tempo	int currenttime	3	3	Pass
mais próximo de				

necessários ainda não foram		
escalonados		

Test Case	testNextIncomplete_ReturnIndexOutOfBoundsExc	Dependênc
ID	eption_Valid	ias

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/U se Case/ Funcionalid ade		Valores do Inputs	Resultados Esperados	Pass/Fail/Unte sted
[ECP7] Verificar se retorna o tempo mais próximo de employees necessários ainda não foram escalonado	int currenttime	-2	IndexOutOfBoundsExc eption	Pass