

Nome: Filipe Meirim
Número de aluno: 45642

Exercício 1:

a)

|value| = 2, pois:

value <= 0 (**Invalid Partition**) (Classe A1)

value > 0 (**Valid Partition**) (Classe A2)

|weight| = 6 pois:

Weight <= 0 (**Invalid Partition**) (Classe B1)

weight > 0 && weight < 500 (**Valid partition**) (Classe B2)

weight >= 500 && weight < 1000 (**Valid partition**) (Classe B3)

weight >= 1000 && weight < 1500 (**Valid partition**) (Classe B4)

weight >= 1500 && weight < 2000 (**Valid partition**) (Classe B5)

weight >= 2000 (**Valid partition**) (Classe B6)

|distance| = 4 pois:

Distance <= 0 (**Invalid Partition**) (Classe C1)

distance > 0 && distance <= 20 (**Valid partition**) (Classe C2)

distance > 20 && distance <= 50 (**Valid partition**) (Classe C3)

distance > 50 (**Valid partition**) (Classe C4)

Número de testes neste caso é igual a $\max(|value|, |weight|, |distance|) = 6$

b)

Test Case Id	Test Data			Expected Output	Covered Partition
	Value	Weight	Distance		
1	0	0	0	Invalid	A1,B1,C1
2	10	400	10	9.5	A2,B2,C2
3	0	600	30	Invalid	A1,B3,C3
4	10	1100	60	9.8	A2,B4,C4
5	0	1600	0	Invalid	A1,B5,C1
6	10	2100	10	9.1	A2,B6,C2

c)

|value| = 2, pois:

value <= 0 (**Invalid Partition**) (Classe A1)

value > 0 (**Valid Partition**) (Classe A2)

|weight| = 6 pois:

Weight <= 0 (**Invalid Partition**) (Classe B1)

weight > 0 && weight < 500 (**Valid partition**) (Classe B2)

weight >= 500 && weight < 1000 (**Valid partition**) (Classe B3)

weight >= 1000 && weight < 1500 (**Valid partition**) (Classe B4)

weight >= 1500 && weight < 2000 (**Valid partition**) (Classe B5)

weight >= 2000 (**Valid partition**) (Classe B6)

|distance| = 4 pois:

Distance <= 0 (**Invalid Partition**) (Classe C1)

distance > 0 && distance <= 20 (**Valid partition**) (Classe C2)

distance > 20 && distance <= 50 (**Valid partition**) (Classe C3)

distance > 50 (**Valid partition**) (Classe C4)

Número de testes neste caso é igual a $|value| * |weight| * |distance| = 48$

d)

Test Case Id	Test Data			Expected Output	Covered Partitions
	Value	Weight	Distance		
1	0	0	0	Invalid	A1,B1,C1
2	0	0	10	Invalid	A1,B1,C2
3	0	0	30	Invalid	A1,B1,C3
4	0	0	60	Invalid	A1,B1,C4
5	0	400	0	Invalid	A1,B2,C1
6	0	400	10	Invalid	A1,B2,C2
7	0	400	30	Invalid	A1,B2,C3
8	0	400	60	Invalid	A1,B2,C4
9	0	600	0	Invalid	A1,B3,C1
10	0	600	10	Invalid	A1,B3,C2
11	0	600	30	Invalid	A1,B3,C3
12	0	600	60	Invalid	A1,B3,C4
13	0	1100	0	Invalid	A1,B4,C1
14	0	1100	10	Invalid	A1,B4,C2
15	0	1100	30	Invalid	A1,B4,C3
16	0	1100	60	Invalid	A1,B4,C4
17	0	1600	0	Invalid	A1,B5,C1
18	0	1600	10	Invalid	A1,B5,C2
19	0	1600	30	Invalid	A1,B5,C3
20	0	1600	60	Invalid	A1,B5,C4
21	0	2100	0	Invalid	A1,B6,C1
22	0	2100	10	Invalid	A1,B6,C2
23	0	2100	30	Invalid	A1,B6,C3
24	0	2100	60	Invalid	A1,B6,C4
...	...	...	...	...	...

A tabela prossegue e vai ter 48 casos de teste sendo que a variável value vai tomar o valor 10 e as variáveis weight e distance vão tomar os seguintes valores, respetivamente {0,400,600,1100,1600,2100} e {0,10,30,60} seguindo a mesma estratégia que nesta primeira metade da tabela.

e) Ver ficheiro Exercise1.java

f) Ver ficheiro Exercise1.java

Exercício 2:

Dominio(Monitor Brands) = {Acer, Alienware, AOC, Asus, Benq, Dell, HP, Lenovo, LG, Samsung}

Dominio(Resolutions) = {HD, FullHD, QHD, UHD, 5k}

Dominio(Refresh Rates) = { 60hz, 75hz, 144hz, 180hz, 240hz }

Dominio(Panel Types) = { TN, VA, IPS,OLED }

Dominio(Monitors) = {Flat, Curved}

Monitor Brands	Resolutions	Refresh Rates	Panel Types	Monitors
Acer	HD	60hz	TN	Flat
Acer	FullHD	75hz	VA	Curved
Acer	QHD	144hz	IPS	Flat
Acer	UHD	180hz	OLED	Curved
Acer	5k	240hz	TN	Curved
Alienware	HD	240hz	VA	Flat
Alienware	FullHD	60hz	IPS	Curved
Alienware	QHD	75hz	OLED	Flat
Alienware	UHD	144hz	TN	Flat
Alienware	5k	180hz	VA	Curved
AOC	HD	180hz	IPS	Flat
AOC	FullHD	240hz	OLED	Curved
AOC	QHD	60hz	TN	Curved
AOC	UHD	75hz	VA	Flat
AOC	5k	144hz	IPS	Curved
Asus	HD	144hz	OLED	Flat
Asus	FullHD	180hz	TN	Flat
Asus	QHD	240hz	VA	Curved
Asus	UHD	60hz	IPS	Flat
Asus	5k	75hz	OLED	Curved
Benq	HD	75hz	TN	Curved
Benq	FullHD	144hz	VA	Flat
Benq	QHD	180hz	IPS	Curved
Benq	UHD	240hz	OLED	Flat
Benq	5k	60hz	TN	Flat
Dell	HD	60hz	VA	Curved
Dell	FullHD	75hz	IPS	Flat
Dell	QHD	144hz	OLED	Curved
Dell	UHD	180hz	TN	Curved
Dell	5k	240hz	VA	Flat
HP	HD	240hz	IPS	Curved
HP	FullHD	60hz	OLED	Flat
HP	QHD	75hz	TN	Flat
HP	UHD	144hz	VA	Curved
HP	5k	180hz	IPS	Flat
Lenovo	HD	180hz	OLED	Curved
Lenovo	FullHD	240hz	TN	Curved
Lenovo	QHD	60hz	VA	Flat
Lenovo	UHD	75hz	IPS	Curved
Lenovo	5k	144hz	OLED	Flat
LG	HD	144hz	TN	Flat

Product price is not 500 euros	0	0	0	0	0	0	0	0	indifferent	1	1	1	0
Effects													
10 % discount	1	0	0	0	0	0	0	0	0	0	0	0	0
Transport fee is 10 euros	1	1	1	0	1	1	0	0	1	1	1	0	0
Free transportation	0	0	0	1	0	0	1	0	0	0	0	1	0
10 euros discount	0	0	0	0	1	1	1	0	0	0	0	0	0
No 10 euros discount	0	0	0	0	0	0	0	1	1	0	0	0	0
No 10% discount	0	0	0	0	0	0	0	0	0	1	1	1	1

c)

Test Case Id	Input (Causes)				Expected Outcome (Effects)
	Price	Weight	Birthday month	Informatics product	
1	= 500	<5kg	False	False	10% discount && Transportation fee of 10 euros
2	<1000	<5kg	False	False	Transportation fee of 10 euros
3	>=1000	<5kg	False	False	Transportation fee of 10 euros
4	>=1000	>5kg	False	False	Free Transportation
5	>=1000	<5kg	True	False	10 euros discount && Transportation fee of 10 euros
6	<1000 && >200	<5kg	True	False	10 euros discount && Transportation fee of 10 euros
7	>=1000	>5kg	True	False	10 euros discount && Free Transportation
8	>=1000	>5kg	False	False	No 10 euros discount
9	<200	<5kg	False	False	No 10 euros discount && 10 euros transportation fee
10	<1000 && !=500	<5kg	False	False	No 10% discount && 10 euros transportation fee

11	>=1000	<5kg	False	False	No 10% discount && 10 euros transportation fee
12	>=1000	>5kg	False	False	No 10% discount && Free transportation
13	= 500	<5kg	False	True	No 10% discount && 10 euros transportation fee

d) Ver ficheiro Exercise3.java

Para este exercício assumiu-se que o construtor da classe Purchase é o seguinte:

```
public Purchase(int value, int weight, boolean informatics, boolean isBirthday){...}
```

Em que value é o valor do produto em centimos, weight é o seu peso em gramas, informatics é uma variável que diz se o produto é informático e isBirthday diz se é o mês de aniversário ou não.

Exercício 4:

a)

|width| = 3, pois:

width <= 0 (**Invalid Partition**) (Classe A1)

width > 0 && width <= 45 (**Valid partition**) (Classe A2)

width > 45 (**Invalid partition**) (Classe A3)

|height| = 3, pois:

height <= 0 (**Invalid Partition**) (Classe B1)

height > 0 && height <= 56 (**Valid partition**) (Classe B2)

height > 56 (**Invalid partition**) (Classe B3)

|depth| = 3, pois:

depth <= 0 (**Invalid Partition**) (Classe C1)

depth > 0 && depth <= 25 (**Valid partition**) (Classe C2)

depth > 25 (**Invalid partition**) (Classe C3)

b)

Logo o número de testes neste caso é igual a $\max(|width|, |height|, |depth|) = 3$

c)

Logo o número de testes neste caso é igual a $|width| * |height| * |depth| = 27$

d)

Test Case Id	Test Data			Expected Outcome	Covered Classes
	Width	Height	Depth		
1	0	0	0	False	A1,B1,C1
2	10	20	15	True	A2,B2,C2
3	50	70	30	False	A3,B3,C3

e)

Test Case Id	Test Data			Expected Outcome	Covered Classes
	Width	Height	Depth		
1	0	0	0	False	A1,B1,C1
2	0	0	15	False	A1,B1,C2
3	0	0	26	False	A1,B1,C3
4	0	30	0	False	A1,B2,C1
5	0	30	15	False	A1,B2,C2
6	0	30	26	False	A1,B2,C3
7	0	60	0	False	A1,B3,C1
8	0	60	15	False	A1,B3,C2
9	0	60	26	False	A1,B3,C3
...	...	...	...	...	...

A tabela prossegue e vamos ter 27 testes como é dito na alínea c) e a coluna width toma os valores 30 e 50, a coluna height toma os valores 0, 30 e 60 e a coluna depth toma os valores 0, 15 e 26. E continuamos a tabela da mesma forma como fizemos nesta primeira parte da tabela.

f)

Test Case Id	Test Data			Expected Output	Covered Partitions
	Width	Height	Depth		
1	1	28	13	True	A2,B2,C2
2	2	28	13	True	A2,B2,C2
3	23	28	13	True	A2,B2,C2
4	44	28	13	True	A2,B2,C2
5	45	28	13	True	A2,B2,C2
6	23	1	13	True	A2,B2,C2
7	23	2	13	True	A2,B2,C2
8	23	28	13	True	A2,B2,C2
9	23	55	13	True	A2,B2,C2
10	23	56	13	True	A2,B2,C2
11	23	28	1	True	A2,B2,C2
12	23	28	2	True	A2,B2,C2
13	23	28	13	True	A2,B2,C2
14	23	28	24	True	A2,B2,C2
15	23	28	25	True	A2,B2,C2

$D(\text{width}) = \{1, 2, 23, 44, 45\}$

$D(\text{height}) = \{1, 2, 28, 55, 56\}$

$D(\text{depth}) = \{1, 2, 13, 24, 25\}$

g) Junta-se aos testes da alínea f) os seguintes testes:

Test Case Id	Test Data			Expected Output	Covered Partitions
	Width	Height	Depth		
1	0	28	13	False	A1,B2,C2
2	46	28	13	False	A3,B2,C2
3	23	0	13	False	A2,B1,C2
4	23	57	13	False	A2,B3,C2
5	23	28	0	False	A2,B2,C1
6	23	28	26	False	A2,B2,C3

h) Junta-se aos testes da alínea f) os seguintes testes:

Test Case Id	Test Data			Expected Output	Covered Partitions
	Width	Height	Depth		
1	1	1	13	True	A2,B2,C2
2	1	2	13	True	A2,B2,C2
3	1	55	13	True	A2,B2,C2
4	1	56	13	True	A2,B2,C2
5	1	28	1	True	A2,B2,C2
6	1	28	2	True	A2,B2,C2
7	1	28	24	True	A2,B2,C2
8	1	28	25	True	A2,B2,C2
9	2	1	13	True	A2,B2,C2
10	2	2	13	True	A2,B2,C2
11	2	55	13	True	A2,B2,C2
12	2	56	13	True	A2,B2,C2
13	2	28	1	True	A2,B2,C2
14	2	28	2	True	A2,B2,C2
15	2	28	24	True	A2,B2,C2
16	2	28	25	True	A2,B2,C2
17	44	1	13	True	A2,B2,C2
18	44	2	13	True	A2,B2,C2
19	44	55	13	True	A2,B2,C2
20	44	56	13	True	A2,B2,C2
21	44	28	1	True	A2,B2,C2
22	44	28	2	True	A2,B2,C2
23	44	28	24	True	A2,B2,C2
24	44	28	25	True	A2,B2,C2
25	45	1	13	True	A2,B2,C2
26	45	2	13	True	A2,B2,C2
27	45	55	13	True	A2,B2,C2
28	45	56	13	True	A2,B2,C2
29	45	28	1	True	A2,B2,C2
30	45	28	2	True	A2,B2,C2
31	45	28	24	True	A2,B2,C2
32	45	28	25	True	A2,B2,C2
...	...	...	...	...	...

D(width) = {1,2,23,44,45}

D(height) = {1,2,28,55,56}

D(depth) = {1,2,13,24,25}

No resto da tabela o objetivo é variar as variáveis height e depth como foi feito com a variável width nesta primeira parte da tabela sendo que height irá tomar os valores 1,2,55 e 56 e depth irá tomar os valores 1,2,24 e 25 quando estivermos a variar essas variáveis. O objetivo é testar as extremidades da região válida seguindo o mesmo raciocínio que foi feito nesta primeira parte da tabela. Por final faltam as combinações entre as 3 variáveis em que fazemos variar as 3 variáveis por exemplo (width = 1, height = 1, depth = 1).

i) Junta-se aos testes da alínea h) e aos testes da alínea g) os seguintes testes:

Test Case Id	Test Data			Expected Output	Covered Partitions
	Width	Height	Depth		
1	0	0	13	False	A1,B1,C2
2	0	1	13	False	A1,B2,C2
3	0	2	13	False	A1,B2,C2
4	0	55	13	False	A1,B2,C2
5	0	56	13	False	A1,B2,C2
6	0	57	13	False	A1,B3,C2
7	0	28	0	False	A1,B2,C1
8	0	28	1	False	A1,B2,C2
9	0	28	2	False	A1,B2,C2
10	0	28	24	False	A1,B2,C2
11	0	28	25	False	A1,B2,C2
12	0	28	26	False	A1,B2,C3
13	46	0	13	False	A3,B2,C2
14	46	1	13	False	A3,B2,C2
15	46	2	13	False	A3,B2,C2
16	46	55	13	False	A3,B2,C2
17	46	56	13	False	A3,B2,C2
18	46	57	13	False	A3,B3,C2
19	46	28	0	False	A3,B2,C1
20	46	28	1	False	A3,B2,C2
21	46	28	2	False	A3,B2,C2
22	46	28	24	False	A3,B2,C2
23	46	28	25	False	A3,B2,C2
24	46	28	26	False	A3,B2,C3
...	...	...	...	...	...

D(width) = {0,1,2,23,44,45,46}

D(height) = {0,1,2,28,55,56,57}

D(depth) = {0,1,2,13,24,25,26}

j) Ver ficheiros Bag.java e Exercise4.java

Mr.X's team qualifies to the final	Condition C1
Mr.X's friend can get the invitation	Condition C2
Mr.X can pay for the ticket	Condition C3
Mr.X cleared all his justice problems	Condition C4
Mr.X is waiting for trial but without travel restriction	Condition C5

Exercício 6:

Clothes name length > 20 (Invalid partition) (Classe C3)

Sizes list members ≥ 32 && Sizes list members ≤ 54 && Sizes list members mod 2 != 0 (Invalid partition) (Classe B4)

Number of sizes > 8 (**Invalid partition**) (Classe A3)

a) Simple Boundary Test Analysis

Test Case Id	Test Data			Expected Output	Covered Partitions
	Number of Sizes	Sizes list	Clothes Name		
1	4	"32,34,52,54"	"ab"	Valid	A2,B2,C2
2	4	"32,34,52,54"	"abc"	Valid	A2,B2,C2
3	4	"32,34,52,54"	"abcdefghijk"	Valid	A2,B2,C2
4	4	"32,34,52,54"	"abcdefghijklmnopqrs"	Valid	A2,B2,C2
5	4	"32,34,52,54"	"abcdefghijklmnopqrst"	Valid	A2,B2,C2
6	1	"32"	"abcdefghijk"	Valid	A2,B2,C2
7	2	"32,54"	"abcdefghijk"	Valid	A2,B2,C2
8	4	"32,34,52,54"	"abcdefghijk"	Valid	A2,B2,C2
9	7	"32,34,38,40,42,52,54"	"abcdefghijk"	Valid	A2,B2,C2
10	8	"32,34,38,40,42,44,52,54"	"abcdefghijk"	Valid	A2,B2,C2

b) Simple Robust Test Analysis

Aos testes da alínea a) juntam-se os seguintes testes:

Test Case Id	Test Data			Expected Output	Covered Partitions
	Number of Sizes	Sizes list	Clothes Name		
1	4	"32,34,52,54"	"a"	Invalid	A2,B2,C1
2	4	"32,34,52,54"	"aaaaaaaaaaaaaaaaaaaaa"	Invalid	A2,B2,C3
3	0	""	"aaaaaaaaaaaa"	Invalid	A1,B1,C2
4	4	"32,34,36,56"	"aaaaaaaaaaaa"	Invalid	A2,B3,C2
5	4	"30,32,34,36"	"aaaaaaaaaaaa"	Invalid	A2,B1,C2
6	4	"32,33,34,36"	"aaaaaaaaaaaa"	Invalid	A2,B4,C2
7	9	"32,34,36,38,40,42,44,46,48"	"aaaaaaaaaaaa"	Invalid	A3,B2,C2

c) Worst Case Test Analysis

Aos testes da alínea a) juntam-se os seguintes testes:

Test Case Id	Test Data			Expected Output	Covered Partitions
	Number of Sizes	Sizes list	Clothes Name		
1	1	"32"	"aaaaaaaaaaaa"	Valid	A2,B2,C2
2	1	"34"	"aaaaaaaaaaaa"	Valid	A2,B2,C2
3	1	"52"	"aaaaaaaaaaaa"	Valid	A2,B2,C2
4	1	"54"	"aaaaaaaaaaaa"	Valid	A2,B2,C2
5	2	"32,34"	"aaaaaaaaaaaa"	Valid	A2,B2,C2
6	2	"32,52"	"aaaaaaaaaaaa"	Valid	A2,B2,C2
7	2	"32,54"	"aaaaaaaaaaaa"	Valid	A2,B2,C2
8	2	"34,52"	"aaaaaaaaaaaa"	Valid	A2,B2,C2
9	2	"34,54"	"aaaaaaaaaaaa"	Valid	A2,B2,C2
10	2	"52,54"	"aaaaaaaaaaaa"	Valid	A2,B2,C2
...	...	...	...	...	...

A tabela prossegue, a coluna “Number of sizes” vai tomar os valores 7 e 8 e o sizes list vai poder ser uma lista de ordem crescente de tamanhos, como pede o enunciado, sendo que os tamanhos só podem estar no domínio de 32 a 54 apenas valores pares e a coluna “Clothes Name” irá tomar valores com número de caracteres igual a 2,3,19 e 20 juntando a isso valores das extremidades das outras duas variáveis.

d) Robust Worst Case Test Analysis

Aos testes da alínea b) e c) juntam-se os seguintes testes:

Test Case Id	Test Data			Expected Output	Covered Partitions
	Number of Sizes	Sizes list	Clothes Name		
1	0	""	"a"	Invalid	A1,B1,C1
2	0	""	"aa"	Invalid	A1,B1,C2
3	0	""	"aaa"	Invalid	A1,B1,C2
4	0	""	"aaaaaaaaaaaaaaaaaaaaa"	Invalid	A1,B1,C2
5	0	""	"aaaaaaaaaaaaaaaaaaaaa"	Invalid	A1,B1,C2
6	0	""	"aaaaaaaaaaaaaaaaaaaaa"	Invalid	A1,B1,C3
...	...	...		...	

A tabela prossegue e a coluna “Number of sizes” vai tomar os valores 9 e a coluna Sizes list vai poder ser uma lista de ordem crescente de tamanhos, como pede o enunciado, e os tamanhos podem ir de 30 a 56, incluindo números ímpares e a coluna “Clothes Name” irá variar quando fixamos os valores da coluna “Number of sizes” e “Sizes list” assim como foi feito nesta primeira parte da tabela. Nos casos em que temos números ímpares o resultado é inválido, nos casos em que temos valores abaixo de 32 ou acima de 54 é inválido e as restantes permutações possíveis para quando temos Number of sizes igual a 9 também são inválidos. Nesta tabela não podem existir testes que tenham mais que uma causa para falhar em relação á mesma variável como por exemplo termos um valor abaixo de 32 que seja ímpar na Sizes list pois assim não é possível saber de onde vem o erro especificamente.

Exercício 7: Ver ficheiro Door.java

States:

- Open (open = true)
- Closed (open = false)
- Locked (locked = true)

Transitions:

- closeDoor
- openDoor
- lockDoor
- unlockDoor

Estado/Transição	closeDoor	openDoor	lockDoor	unlockDoor
Open	Closed	-	-	-
Closed	-	Open	Locked	-
Locked	-	-	-	Closed

Test Battery (Supondo que começamos no estado Closed):

openDoor; closeDoor; lockDoor; unlockDoor; closeDoor; unlockDoor; openDoor;
 lockDoor; unlockDoor; openDoor; closeDoor; lockDoor; openDoor; lockDoor;
 closeDoor

Exercício 8:

a) Ver ficheiro control_flow.jpg

b)

Statement Coverage:

Inputs	Outputs			
Scanner	Average Grade	Average Passing Grade	Passing Students	Total Students
"-1.0"	0.0	0.0	0	0
"10"	10.0	10.0	1	1

Condition Coverage:

Inputs	Outputs			
Scanner	Average Grade	Average Passing Grade	Passing Students	Total Students
"-1.0"	0.0	0.0	0	0
"10.0, 9.4, 21.0"	9.7	10.0	1	2

Decision Coverage:

Inputs	Outputs			
Scanner	Average Grade	Average Passing Grade	Passing Students	Total Students
"-1.0"	0.0	0.0	0	0
"10.0, 9.4"	9.7	10.0	1	2

Modified condition/decision coverage:

Inputs	Outputs			
Scanner	Average Grade	Average Passing Grade	Passing Students	Total Students
"-1.0"	0.0	0.0	0	0
"10.0, 9.4, 21.0"	9.7	10.0	1	2

Independent path coverage:

Number of independent paths: $18 - 15 + 2 = 5$

Inputs	Outputs			
Scanner	Average Grade	Average Passing Grade	Passing Students	Total Students
"-1.0"	0.0	0.0	0	0
"9.4"	9.4	0.0	0	1
"10.0"	10.0	10.0	1	1

Neste caso é impossível fazer dois dos "independent paths" pois se o totalStudents é igual a 0 então o passingStudents obrigatoriamente será também igual a 0.

c) Ver ficheiro data_flow.jpg

d)

All definitions coverage:

Inputs	Outputs			
Scanner	Average Grade	Average Passing Grade	Passing Students	Total Students
"-1.0"	0.0	0.0	0	0
"10.0"	10.0	10.0	1	1
"9.4"	9.4	0.0	0	1

All uses:

Inputs	Outputs			
Scanner	Average Grade	Average Passing Grade	Passing Students	Total Students
"-1.0"	0.0	0.0	0	0
"10.0"	10.0	10.0	1	1
"9.4"	9.4	0.0	0	1

All def-uses:

Inputs	Outputs			
Scanner	Average Grade	Average Passing Grade	Passing Students	Total Students
"-1.0"	0.0	0.0	0	0

Não é possível verificar todos os def-clear paths de todas as variáveis pois para isso teria de se passar por um loop o que faria com que tivesse de passar por nós mais que uma vez.