

C03-TESTING-P02

→ Trabajo realizado por:

- David Rivera Concepción
- Roberto Ortuño Blanco

ET.02.03 Segundo Problema:

En su nuevo plan de transportes públicos de la JCCM, se nos ha pedido implementar una aplicación que determine el precio del billete del medio de transporte, en función de las condiciones de salud y etarias de una persona, y en función del estado de la pandemia. En este sentido, se vigilará la IA de Castilla-La Mancha (podemos suponerla “constante” durante la ejecución del proyecto, lo que requerirá una inicialización del entorno), con una reducción de plazas como sigue: si es menor que 100 no habrá restricciones de espacio (nivel 0), si está entre 100 y 200 (nivel 1) el aforo del medio de transporte se reduce al 80%, si está entre 201 y 300 (nivel 2), se reduce al 60%, si está entre 301 y 500 (nivel 3) el aforo se reduce al 40%, y si es superior a 501 (nivel 4) al 30%.

A fin de evitar movimientos innecesarios, se establecerá un incremento del precio del billete. Para ello se tendrá en cuenta las siguientes reglas para calcular el precio del billete:

- Independientemente del estado de la IA, una persona enferma, con contacto reciente en los últimos 10 días con infectados o con síntomas sospechosos de COVID no podrá viajar. Una persona con pasaporte COVID y no enferma podrá viajar si hay plaza, independientemente de su tipo de profesión.
- Si no hay restricciones de espacio (nivel 0), cualquier persona, independientemente de su edad podrá viajar, y tendrá un descuento del 60% si es menor que 23 años, y del 80% si es mayor que 65. No se establecen prioridades de transporte.
- En Nivel 1, los menores de 23 años tendrán un descuento de 30%, y los mayores de 65 tendrán un descuento de 50%. No se establecen prioridades de transporte, pero se reducen las plazas, con lo que solo se dará precio (plantéese lanzar algún tipo de excepción) si hay plazas disponibles.
- En Nivel 2, los menores de 23 años no tendrán descuentos, y los mayores de 65 años tendrá un incremento del 20%. En este nivel, de la capacidad disminuida posible, se reserva un 60% de las plazas a profesionales con profesiones imprescindibles.
- En Nivel 3, los menores de 23 años tendrán una recarga del 20%, y los mayores de 65 años tendrá un incremento del 50%. En este nivel de capacidad, se reserva un espacio disponible del 80% de la capacidad reducida a profesionales imprescindibles.
- En Nivel 4, los menores de 23 años tendrán una recarga del 50%, y los mayores de 65 años no podrán viajar. Se reserva un espacio del 90% para personas con profesiones imprescindibles.

1) Identificar las variables que se deben tener en cuenta para probar el método de interés.

→ puedeViajar:

- Persona:
 - `private int edad`
 - `private boolean enfermo`
 - `private boolean contactoInfectado10dias`
 - `private boolean sintomas`
 - `private boolean pasaporteCovid`
 - `private boolean profesionImprescindible`
- Pandemia:
 - `private int nivel`

→ calcularPrecio:

- Persona:
 - `private int edad`
- Pandemia:
 - `private int nivel`
- `public static boolean puedeViajar(Persona persona, Pandemia pandemia)`

→ calcularNivelPandemia:

- Pandemia:
 - `private int IACastillaLaMancha`

→ `static int IACastillaLaMancha`

2) Identificar los valores de pruebas para cada una de las variables anteriores usando las tres técnicas vistas en teoría, especificando para cada una cual es la que ha sido usada.

PARÁMETRO	CLASE DE EQUIVALENCIA	CONJUNTO DE VALORES	VALORES LÍMITE	CONJETURA DE ERRORES	Nº Valores
			(Variante pesada)		
edad	$(-\infty, 0)$ [0,23) [23,65) [65, $+\infty$)	-5 22 50 68	0 23 65 -1,1 ; 22 ,24 ; 64,66	200	13
enfermo	True False	True False	-	null	3
contactoInfectado10dias	True False	True False	-	null	3
sintomas	True False	True False	-	null	3
pasaporteCovid	True False	True False	-	null	3
profesionImprescindible	True False	True False	-	null	3
nivel	$(-\infty, 0)$ [0,4] (4, $+\infty$)	-2 3 5	0 4 -1,1 ; 3 ,4	5	9
IACastillaLaMancha	$(-\infty, 0)$ [0,100] [100,200] [201,300] [301,500] (500, $+\infty$)	-10 50 150 250 350 550	0 100 200 300 500 -1,1 ; 99,101 ; 199,201 ; 299,301; 499,501	5000000000000000000	22
numPlazas	$(-\infty, 0)$ [0,100] (100, $+\infty$)	-8 95 200	0 100 -1,1 ; 99,101	300	10

3) Calcular el número máximo posible de casos de pruebas que se podrían generar a partir de los valores de pruebas (combinatoria).

→Numero de valores por parámetro:

- edad: 13
- enfermo: 3
- contactoInfectado10dias: 3
- síntomas: 3
- pasaporteCovid: 3
- profesionImprescindible: 3
- nivel: 9
- IACastillaLaMancha: 22
- numPlazas: 10

→Número máximo de casos de prueba = $13 \times 3 \times 3 \times 3 \times 3 \times 3 \times 9 \times 22 \times 10 = 6254820$ casos de prueba

4) Defina un conjunto de casos de pruebas para cumplir con each use (cada valor una vez)

- **puedeViajar():**

enfermo	contactoInfectado10Dias	sintomas	pasaporteCovid	enfermo	nivel	profesionImprescindible	numPlazas
true	true	true	true	true	-2	true	-8
false	false	false	false	false	3	false	95
null	null	null	null	null	5	null	200

- **calcularPrecio():**

nivel	edad
-2	-5
3	22
5	68

- **calcularNivelPandemia():**

IACastillaLaMancha
-10
50
150
250
350
550

5) Defina conjuntos de pruebas para alcanzar cobertura pairwise usando el algoritmo explicado en clase. Se pueden comprobar los resultados con el programa PICT1:

- puedeViajar():

enfermo	contactoInfectado10dias
false	true
false	null
false	false
null	true
null	null
null	false
true	false
true	null
true	true

enfermo	sintomas
false	true
false	null
false	false
null	true
null	null
null	false
true	false
true	null
true	true

contactoInfectado10dias	sintomas
false	true
false	null
false	false
null	true
null	null
null	false
true	false
true	null
true	true

pasaporteCovid	enfermo
false	true
false	null
false	false
null	true
null	null
null	false
true	false
true	null
true	true

nivel	profesionImprescindible
-2	false
-2	null
-2	true
3	true
3	null
3	false
5	true
5	null
5	false

nivel	numPlazas
-2	95
-2	200
-2	-8
3	-8
3	200
3	95
5	-8
5	200
5	95

enfermo	numPlazas
false	-8
false	200
false	95
null	-8
null	200
null	95
true	95
true	200
true	-8

pasaporteCovid	numPlazas
false	-8
false	200
false	95
null	-8
null	200
null	95
true	95
true	200
true	-8

profesionImprescindible	numPlazas
false	-8
false	200
false	95
null	-8
null	200
null	95
true	95
true	200
true	-8

Test suit:

enfermo	contactoInfectado10dias	sintomas	pasaporteCovid	profesionImprescindible	nivel	numPlazas
true	null	true	false	true	5	200
null	true	null	true	null	5	-8
false	false	false	null	false	-2	-8
true	false	false	false	null	3	95
null	true	null	false	true	-2	95
null	false	true	true	false	3	200
false	null	false	true	false	5	95
false	false	null	null	true	5	200
true	true	true	true	true	-2	-8
null	null	true	null	null	3	95
null	true	false	null	null	-2	200
true	null	null	false	false	3	-8
true	true	false	null	true	3	-8
false	true	true	false	null	3	95
null	true	false	null	false	-2	-8
null	null	false	false	true	-2	95

- calcularPrecio():

nivel	edad
-2	68
-2	50
-2	22
-2	-5
3	50
3	-5
3	22
3	68
5	68
5	50
5	-5
5	22

- **calcularNivelPandemia():** como solo tenemos el parámetro de IACastillaLaMancha no podemos hacer pairwise.

6) Para los trozos de código que incluyan decisiones, proponga conjunto de casos de prueba para alcanzar cobertura de decisiones:

- **calcularNivelPandemia:**

- Nivel 0: `if(pandemia.getIACastillaLaMancha() < 100)`

A
T
F

- Nivel 1: `if(pandemia.getIACastillaLaMancha() >=100 && pandemia.getIACastillaLaMancha() <=200)`

A
T
F

- Nivel 2: `if(pandemia.getIACastillaLaMancha() >=201 && pandemia.getIACastillaLaMancha() <=300)`

A
T
F

- Nivel 3: `if(pandemia.getIACastillaLaMancha() >=301 && pandemia.getIACastillaLaMancha() <=500)`

A
T
F

- Nivel 4: `if(pandemia.getIACastillaLaMancha() >=501)`

A
T
F

- **calcularPrecio():**

- `if(pandemia.getNivel() == 0 && persona.getEdad() < 23)`

A	B	A AND B	CONDICION DOMINANTE
T	T	T	A, B
T	F	F	B
F	T	F	A
F	F	F	A, B

- `if(pandemia.getNivel() == 0 && persona.getEdad() > 65)`

A	B	A AND B	CONDICION DOMINANTE
T	T	T	A, B
T	F	F	B
F	T	F	A
F	F	F	A, B

- `if(pandemia.getNivel() == 1 && persona.getEdad() < 23)`

A	B	A AND B	CONDICION DOMINANTE
T	T	T	A, B
T	F	F	B
F	T	F	A
F	F	F	A, B

- `if(pandemia.getNivel() == 1 && persona.getEdad() > 65)`

A	B	A AND B	CONDICION DOMINANTE
T	T	T	A, B
T	F	F	B
F	T	F	A
F	F	F	A, B

- `if(pandemia.getNivel() == 2 && persona.getEdad() < 23)`

A	B	A AND B	CONDICION DOMINANTE
T	T	T	A, B
T	F	F	B
F	T	F	A
F	F	F	A, B

- `if(pandemia.getNivel() == 2 && persona.getEdad() > 65)`

A	B	A AND B	CONDICION DOMINANTE
T	T	T	A, B
T	F	F	B
F	T	F	A
F	F	F	A, B

- `if(pandemia.getNivel() == 3 && persona.getEdad() < 23)`

A	B	A AND B	CONDICION DOMINANTE
T	T	T	A, B
T	F	F	B
F	T	F	A
F	F	F	A, B

- `if(pandemia.getNivel() == 3 && persona.getEdad() > 65)`

A	B	A AND B	CONDICION DOMINANTE
T	T	T	A, B
T	F	F	B
F	T	F	A
F	F	F	A, B

- `if(pandemia.getNivel() == 4 && persona.getEdad() < 23)`

A	B	A AND B	CONDICION DOMINANTE
T	T	T	A, B
T	F	F	B
F	T	F	A
F	F	F	A, B

- **puedeViajar():**

- `if(persona.isEnfermo() || persona.isContactoInfectado10dias() || persona.isSintomas())`

A	B	C	A OR B OR C	CONDICION DOMINANTE
T	T	T	T	A,B,C
T	T	F	T	A,B
T	F	T	T	A,C
T	F	F	T	A
F	T	T	T	B,C
F	T	F	T	B
F	F	T	T	C
F	F	F	F	A,B,C

- else if(persona.isPasaporteCovid() && !persona.isEnfermo() && numPlazas>=1)

A	B	C	A AND B AND C	CONDICION DOMINANTE
T	T	T	T	A,B,C
T	T	F	F	C
T	F	T	F	B
T	F	F	F	B,C
F	T	T	F	A
F	T	F	F	A,C
F	F	T	F	A,B
F	F	F	F	A,B,C

- if(pandemia.getNivel() == 0 && numPlazas>=1)

A	B	A AND B	CONDICION DOMINANTE
T	T	T	A, B
T	F	F	B
F	T	F	A
F	F	F	A, B

- if(pandemia.getNivel() == 2 && persona.isProfesionImprescindible() && numPlazas>=1)

A	B	C	A AND B AND C	CONDICION DOMINANTE
T	T	T	T	A,B,C
T	T	F	F	C
T	F	T	F	B
T	F	F	F	B,C
F	T	T	F	A
F	T	F	F	A,C
F	F	T	F	A,B
F	F	F	F	A,B,C

- else if(pandemia.getNivel() == 2 && !persona.isProfesionImprescindible() && numPlazas*0.40 >= 1)

A	B	C	A AND B AND C	CONDICION DOMINANTE
T	T	T	T	A,B,C
T	T	F	F	C
T	F	T	F	B
T	F	F	F	B,C
F	T	T	F	A
F	T	F	F	A,C
F	F	T	F	A,B
F	F	F	F	A,B,C

- if(pandemia.getNivel() == 3 && persona.isProfesionImprescindible() && numPlazas>=1)

A	B	C	A AND B AND C	CONDICION DOMINANTE
T	T	T	T	A,B,C
T	T	F	F	C
T	F	T	F	B
T	F	F	F	B,C
F	T	T	F	A
F	T	F	F	A,C
F	F	T	F	A,B
F	F	F	F	A,B,C

- else if(pandemia.getNivel() == 3 && !persona.isProfesionImprescindible() && numPlazas*0.20 >= 1)

A	B	C	A AND B AND C	CONDICION DOMINANTE
T	T	T	T	A,B,C
T	T	F	F	C
T	F	T	F	B
T	F	F	F	B,C
F	T	T	F	A
F	T	F	F	A,C
F	F	T	F	A,B
F	F	F	F	A,B,C

- if(pandemia.getNivel() == 4 && persona.isProfesionImprescindible() && numPlazas>=1)

A	B	C	A AND B AND C	CONDICION DOMINANTE
T	T	T	T	A,B,C
T	T	F	F	C
T	F	T	F	B
T	F	F	F	B,C
F	T	T	F	A
F	T	F	F	A,C
F	F	T	F	A,B
F	F	F	F	A,B,C

- else if(pandemia.getNivel() == 4 && !persona.isProfesionImprescindible() && numPlazas*0.10 >= 1)

A	B	C	A AND B AND C	CONDICION DOMINANTE
T	T	T	T	A,B,C
T	T	F	F	C
T	F	T	F	B
T	F	F	F	B,C
F	T	T	F	A
F	T	F	F	A,C
F	F	T	F	A,B
F	F	F	F	A,B,C

- `if(pandemia.getNivel() == 4 && persona.getEdad() > 65)`

A	B	A AND B	CONDICION DOMINANTE
T	T	T	A, B
T	F	F	B
F	T	F	A
F	F	F	A, B

7) Para los trozos de código que incluyan decisiones, proponga conjunto de casos de prueba para alcanzar cobertura MC/DC.

- **calcularNivelPandemia:**

- Nivel 0: `if(pandemia.getIACastillaLaMancha() < 100)`

A
T
F

- Nivel 1: `if(pandemia.getIACastillaLaMancha() >=100 && pandemia.getIACastillaLaMancha() <=200)`

A
T
F

- Nivel 2: `if(pandemia.getIACastillaLaMancha() >=201 && pandemia.getIACastillaLaMancha() <=300)`

A
T
F

- Nivel 3: `if(pandemia.getIACastillaLaMancha() >=301 && pandemia.getIACastillaLaMancha() <=500)`

A
T
F

- Nivel 4: `if(pandemia.getIACastillaLaMancha() >=501)`

A
T
F

- **calcularPrecio():**

- if(pandemia.getNivel() == 0 && persona.getEdad() < 23)

A	B	A AND B	CONDICION DOMINANTE
T	T	T	A, B
T	F	F	B
F	T	F	A
F	F	F	A, B

- if(pandemia.getNivel() == 0 && persona.getEdad() > 65)

A	B	A AND B	CONDICION DOMINANTE
T	T	T	A, B
T	F	F	B
F	T	F	A
F	F	F	A, B

- if(pandemia.getNivel() == 1 && persona.getEdad() < 23)

A	B	A AND B	CONDICION DOMINANTE
T	T	T	A, B
T	F	F	B
F	T	F	A
F	F	F	A, B

- if(pandemia.getNivel() == 1 && persona.getEdad() > 65)

A	B	A AND B	CONDICION DOMINANTE
T	T	T	A, B
T	F	F	B
F	T	F	A
F	F	F	A, B

- if(pandemia.getNivel() == 2 && persona.getEdad() < 23)

A	B	A AND B	CONDICION DOMINANTE
T	T	T	A, B
T	F	F	B
F	T	F	A
F	F	F	A, B

- `if(pandemia.getNivel() == 2 && persona.getEdad() > 65)`

A	B	A AND B	CONDICION DOMINANTE
T	T	T	A, B
T	F	F	B
F	T	F	A
F	F	F	A, B

- `if(pandemia.getNivel() == 3 && persona.getEdad() < 23)`

A	B	A AND B	CONDICION DOMINANTE
T	T	T	A, B
T	F	F	B
F	T	F	A
F	F	F	A, B

- `if(pandemia.getNivel() == 3 && persona.getEdad() > 65)`

A	B	A AND B	CONDICION DOMINANTE
T	T	T	A, B
T	F	F	B
F	T	F	A
F	F	F	A, B

- `if(pandemia.getNivel() == 4 && persona.getEdad() < 23)`

A	B	A AND B	CONDICION DOMINANTE
T	T	T	A, B
T	F	F	B
F	T	F	A
F	F	F	A, B

- puedeViajar():

- if(persona.isEnfermo() || persona.isContactoInfectado10dias() || persona.isSintomas())

A	B	C	A OR B OR C	CONDICION DOMINANTE
T	T	T	T	A,B,C
T	T	F	T	A,B
T	F	T	T	A,C
T	F	F	T	A
F	T	T	T	B,C
F	T	F	T	B
F	F	T	T	C
F	F	F	F	A,B,C

- if(persona.isPasaporteCovid() && !persona.isEnfermo() && numPlazas>=1)

A	B	C	A AND B AND C	CONDICION DOMINANTE
T	T	T	T	A,B,C
T	T	F	F	C
T	F	T	F	B
T	F	F	F	B,C
F	T	T	F	A
F	T	F	F	A,C
F	F	T	F	A,B
F	F	F	F	A,B,C

- if(pandemia.getNivel() == 0 && numPlazas>=1)

A	B	A AND B	CONDICION DOMINANTE
T	T	T	A, B
T	F	F	B
F	T	F	A
F	F	F	A, B

- if(pandemia.getNivel() == 2 && persona.isProfesionImprescindible() && numPlazas>=1)

A	B	C	A AND B AND C	CONDICION DOMINANTE
T	T	T	T	A,B,C
T	T	F	F	C
T	F	T	F	B
T	F	F	F	B,C
F	T	T	F	A
F	T	F	F	A,C
F	F	T	F	A,B
F	F	F	F	A,B,C

- else if(pandemia.getNivel() == 2 && !persona.isProfesionImprescindible() && numPlazas*0.40 >= 1)

A	B	C	A AND B AND C	CONDICION DOMINANTE
T	T	T	T	A,B,C
T	T	F	F	C
T	F	T	F	B
T	F	F	F	B,C
F	T	T	F	A
F	T	F	F	A,C
F	F	T	F	A,B
F	F	F	F	A,B,C

- if(pandemia.getNivel() == 3 && persona.isProfesionImprescindible() && numPlazas>=1)

A	B	C	A AND B AND C	CONDICION DOMINANTE
T	T	T	T	A,B,C
T	T	F	F	C
T	F	T	F	B
T	F	F	F	B,C
F	T	T	F	A
F	T	F	F	A,C
F	F	T	F	A,B
F	F	F	F	A,B,C

- else if(pandemia.getNivel() == 3 && !persona.isProfesionImprescindible() && numPlazas*0.20 >= 1)

A	B	C	A AND B AND C	CONDICION DOMINANTE
T	T	T	T	A,B,C
T	T	F	F	C
T	F	T	F	B
T	F	F	F	B,C
F	T	T	F	A
F	T	F	F	A,C
F	F	T	F	A,B
F	F	F	F	A,B,C

- if(pandemia.getNivel() == 4 && persona.isProfesionImprescindible() && numPlazas>=1)

A	B	C	A AND B AND C	CONDICION DOMINANTE
T	T	T	T	A,B,C
T	T	F	F	C
T	F	T	F	B
T	F	F	F	B,C
F	T	T	F	A
F	T	F	F	A,C
F	F	T	F	A,B
F	F	F	F	A,B,C


- else if(pandemia.getNivel() == 4 && !persona.isProfesionImprescindible() && numPlazas*0.10 >= 1)

A	B	C	A AND B AND C	CONDICION DOMINANTE
T	T	T	T	A,B,C
T	T	F	F	C
T	F	T	F	B
T	F	F	F	B,C
F	T	T	F	A
F	T	F	F	A,C
F	F	T	F	A,B
F	F	F	F	A,B,C

8) Comente los resultados del número de los casos de pruebas conseguidos en los apartados 4, 5 y 6 ¿qué podría decirse algo de la cobertura alcanzada?

Tras realizar el testing con junit en Eclipse, hemos obtenido una cobertura de 96.0% e instrucciones cubiertas 831. Se nos ha generado el siguiente informe con surefire:

C03-Testing-P02

Project Documentation
Project Information
Project Reports
Surefire Report


Surefire Report

Summary

[Summary] [Package List] [Test Cases]

Tests	Errors	Failures	Skipped	Success Rate	Time
3	0	0	0	100%	0.044

Note: failures are anticipated and checked for with assertions while errors are unanticipated.

Package List

[Summary] [Package List] [Test Cases]

Package	Tests	Errors	Failures	Skipped	Success Rate	Time
C03.IS02_2021_Testing_P2	3	0	0	0	100%	0.044

Note: package statistics are not computed recursively, they only sum up all of its testsuites numbers.

C03.IS02_2021_Testing_P2

Class	Tests	Errors	Failures	Skipped	Success Rate	Time
mainTest	3	0	0	0	100%	0.044

Test Cases

[Summary] [Package List] [Test Cases]

mainTest

Class	Test	Time
mainTest	testCalculaIrraPandemia	0.003
mainTest	testCalculaPrecio	0
mainTest	testPuedeViajar	0