

PRÁCTICA LABORATORIO

TESTING

GRUPO B01
NÉSTOR LÓPEZ TORRES
YIRMEL SÁNCHEZ BALDERA

- VALORES DE PRUEBAS PARA LAS VARIABLES.

Variable	Clases de equivalencia	Valores límites	Valores límite variante pesada	Conjetura de errores	Total valores
Contagiados ($-\infty, 0$), [0, 100), [100, 200], (200, 300], (300, 500], (500, $+\infty$)	-5, 10, 150, 243, 400, 505	0, 100, 200, 300, 500	-1, 1, 99, 101, 199, 201, 299, 301, 499, 501	-50	22
Nivel_pandemia ($-\infty, 0$), [0, 1), [1, 2), [2, 3), [3, 4), [4, 5), [5, $+\infty$)	-1, 0, 1, 2, 3, 4, 23	0, 1, 2, 3, 4	-1, 1, 2, 3, 4, 5	-2, 12	10
Plazas_disponibles ($-\infty, +\infty$)	-	-	-	-5, 0	2
Edad ($-\infty, 0$), [0, 23], (23, 65], (65, $+\infty$)	-5, 14, 45, 89	0, 23, 65	-1, 1, 22, 24, 64, 66	-5, 200	14
Num_disponibles_profesionales ($-\infty, +\infty$)	-	-	-	-5, 0	2
Num_disponibles_noProfesionales ($-\infty, +\infty$)	-	-	-	-5, 0	2
Plazas_ocupadas_profesionales ($-\infty, +\infty$)	-	-	-	-5	1
Plazas_ocupadas_noProfesionales ($-\infty, +\infty$)	-	-	-	-5	1
Sana	sí, no				2
Síntomas	sí, no				2
Contacto	sí, no				2
Pasaporte_COVID	sí, no				2
Es_profesional	sí, no				2

Para obtener los valores de prueba de cada uno de los valores anteriores, hemos utilizado:

- Criterio de clases de equivalencia.

Obtenemos un valor cualquiera de cada uno de los intervalos que posee cada una de las variables. *Por ejemplo, en la variable contagiados, existe el intervalo ($-\infty, 0$), del cual obtenemos el valor -5.*

-

- Criterio de los valores límites.

Obtenemos los valores que limitan los intervalos de cada una de las variables. *Por ejemplo, en la variable contagiados, los intervalos ($-\infty, 0$) y [0, 100) están limitados por el valor 0.*

-

- Criterio de los valores límites de variante pesada.

Obtenemos el valor inmediatamente superior e inferior de cada uno de los valores límites. *Por ejemplo, el valor límite 0 obtenido anteriormente, posee el -1 y el 1 como valores límite de variante pesada.*

- **Conjetura de errores.** Obtenemos aquellos valores que consideramos que son propensos a causar errores, como pueden ser números negativos, el 0 o números demasiado grandes.

Por ejemplo, en el caso de la variable contagiados, consideramos conjetura de errores valores como el -50.

Las variables booleanas solo pueden poseer dos valores (si o no), por lo que no podremos aplicar los criterios descritos anteriormente.

- COBERTURA DE DECISIONES.

Se escriben los casos de prueba suficientes para que cada decisión en el programa se ejecute una vez con resultado verdadero y otra con resultado falso. Entendiendo camino como una secuencia de sentencias encadenadas desde la entrada del programa hasta su salida.

■ Casos de prueba utilizados:

	Contagiados	Num plazas	Plazas ocuProf	Plazas ocuNoProf	Persona						Resultado
					Edad	Sana	Contacto	Síntomas	Pasaporte COVID	Profesional	
N0_NV	50	100	10	20	18	false	false	false	true	false	NoPosibilidadDeViajarException
N0_NA	50	100	50	50	18	true	false	false	true	false	NoHayPlazasDisponiblesException
N0_J	50	100	10	20	18	true	false	false	true	false	PrecioBillete = 20 * (1 - 0.6)
N0_M	50	100	10	20	70	true	false	false	true	false	PrecioBillete = 20 * (1 - 0.8)
N1_J	150	100	10	20	18	true	false	false	true	false	PrecioBillete = 20 * (1 - 0.3)
N1_M	150	100	10	20	70	true	false	false	true	false	PrecioBillete = 20 * (1 - 0.5)
N2_M	250	100	10	20	70	true	false	false	true	false	PrecioBillete = 20 * 1.2
N3_NAP	350	100	32	6	18	true	false	false	true	true	NoHayPlazasDisponiblesException
N3_NANP	350	100	30	8	18	true	false	false	true	false	NoHayPlazasDisponiblesException
N3_J	350	100	20	6	18	true	false	false	true	false	PrecioBillete = 20 * 1.2
N3_M	350	100	20	6	70	true	false	false	true	false	PrecioBillete = 20 * 1.5
N4_JP	550	100	20	2	18	true	false	false	true	true	PrecioBillete = 20 * 1.5
N4_J	550	100	20	2	18	true	false	false	true	false	PrecioBillete = 20 * 1.5
N4_M	550	100	20	2	70	true	false	false	true	false	NoPosibilidadDeViajarException

Leyenda	
N0 – N4	Nivel Pandemia
NV	No Viajar
J	Persona Joven
M	Persona Mayor
NAP	No Asientos Profesionales
NANP	No Asientos No Profesionales
JP	Persona Joven Profesional

- CORRECCIONES DE CÓDIGO.

Existen un conjunto de variables que no son muy influyentes en el resultado final del programa independientemente de su valor o su valor es calculado en la misma clase Java antes de inicializar el programa:

Variables No Influyentes	
NivelPandemia	
PlazasDisponibles	100
nDisponiblesProf	
nDisponiblesNoProf	

Por simplicidad del programa hemos dejados la variable *PlazasDisponibles* constante con valor 100 para facilitar la comprobación de los cálculos.

- Al principio contemplamos el programa como una IA a la cual una persona accedía e introducía sus datos, que posteriormente serían utilizados por el programa, de la siguiente manera:

```
//Crear persona que quiere comprar un billete
private static Persona crearPersona() {
    Persona p1 = new Persona();

    //Se supone que todos los datos son correctos
    System.out.println("RELLENE EL SIGUIENTE FORMULARIO");
    System.out.println();

    System.out.print("Escriba su nombre: ");
    p1.setNombre(entradaEscaner.nextLine());
    System.out.print("Escriba su edad: ");
    p1.setEdad(entradaEscaner.nextInt());
    System.out.println();
    System.out.println("----- Rellene con true o false -----");
    System.out.print("¿Está usted sano?: ");
    p1.setSano(entradaEscaner.nextBoolean());
    System.out.print("¿Ha tenido usted contacto con algún contagiado recientemente?: ");
    p1.setContacto(entradaEscaner.nextBoolean());
    System.out.print("¿Tiene usted síntomas?: ");
    p1.setSintomas(entradaEscaner.nextBoolean());
    System.out.print("¿Tiene el pasaporte COVID en regla?: ");
    p1.setPasaporte_COVID(entradaEscaner.nextBoolean());
    System.out.print("¿Es un profesional con profesión imprescindible?: ");
    p1.setEs_Profesional(entradaEscaner.nextBoolean());
    System.out.println();

    return p1;
}
```

- Posteriormente al realizar los test con JUnit 4 vimos que no testeaba correctamente tecleando los datos de la persona por teclado, y cambiamos el código haciendo que el programa reciba automáticamente los datos de la persona. También eliminamos los Setters de la clase Persona, ya que no se utilizaban.

```
Persona pasajero = new Persona("Juan", 18, true, false, false, true, false);
principal(100, 500, 50, 80, pasajero);
```