

Trabajo teórico 2



Componentes:

Sandra Rodríguez Sánchez-Gil.

Steven Tipantuña Aquieta.

Álvaro Mozos Olmo.

Raúl Mata Muela.

Alonso Antonio Zamora Zamora.

Manuel Villanueva Alises.

Fecha: 20-12-2024

Índice

ET.02.02 Primer Problema	4
1. Escribir, al menos el pseudocódigo correspondiente al método o a los métodos identificados.	4
2) Identificar las variables que se deben tener en cuenta para probar el método de interés.	5
3) 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.	5
4) Calcular el número máximo posible de casos de pruebas que se podrían generar a partir de los valores de pruebas (combinatoria).	5
5) Defina un conjunto de casos de pruebas para cumplir con each use (cada valor una vez)	6
6) Defina conjuntos de pruebas para alcanzar cobertura pairwise usando el algoritmo explicado en clase. Se pueden comprobar los resultados con el programa PICT2	6
7) Para los trozos de código que incluyan decisiones, proponga conjunto de casos de prueba para alcanzar cobertura de decisiones	8
8) Para los trozos de código que incluyan decisiones, proponga conjunto de casos de prueba para alcanzar cobertura MC/DC.	8
9) Comente los resultados del número de los casos de pruebas conseguidos en los apartados 4, 5 y 6 ¿qué podría decirse de la cobertura alcanzada?	9
ET.02.02 Segundo Problema	11
1. Escribir, al menos el pseudocódigo correspondiente al método o a los métodos identificados.	11
2) Identificar las variables que se deben tener en cuenta para probar el método de interés.	12
3) 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.	13
4) Calcular el número máximo posible de casos de pruebas que se podrían generar a partir de los valores de pruebas (combinatoria).	13
5) Defina un conjunto de casos de pruebas para cumplir con each use (cada valor una vez).	14
6) Defina conjuntos de pruebas para alcanzar cobertura pairwise usando el algoritmo explicado en clase. Se pueden comprobar los resultados con el programa PICT	14
7) Para los trozos de código que incluyan decisiones, proponga conjunto de casos de prueba para alcanzar cobertura de decisiones.	15
8) Para los trozos de código que incluyan decisiones, proponga conjunto de casos de prueba para alcanzar cobertura MC/DC.	17
9) Comente los resultados del número de los casos de pruebas conseguidos en los apartados 4, 5 y 6 ¿qué podría decirse de la cobertura alcanzada?	18
ET.02.02 Tercer Problema	20
1. Escribir, al menos el pseudocódigo correspondiente al método o a los métodos identificados.	20
2. Identificar las variables que se deben tener en cuenta para probar el método de interés.	20
3. 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.	21
4) Calcular el número máximo posible de casos de pruebas que se podrían generar a partir de los valores de pruebas (combinatoria).	21
5) Defina un conjunto de casos de pruebas para cumplir con each use (cada valor una vez)	22
6) Defina conjuntos de pruebas para alcanzar cobertura pairwise usando el algoritmo explicado en clase. Se pueden comprobar los resultados con el programa PICT	23
7) Para los trozos de código que incluyan decisiones, proponga conjunto de casos de prueba para alcanzar cobertura de decisiones.	24
8) Para los trozos de código que incluyan decisiones, proponga conjunto de casos de prueba para alcanzar cobertura MC/DC.	32
9) Comente los resultados del número de los casos de pruebas conseguidos en los apartados 4, 5 y 6 ¿qué podría decirse de la cobertura alcanzada?	33

Enunciado problema 1:

Se trata de escribir y probar una clase Persona que inicialice objetos con estados válidos. Los datos que se deben tomar de la Persona deben ser nombre, apellidos, fecha de nacimiento, nacionalidad, titulación, certificación de inglés, número de teléfono, y correo electrónico. Además, teniendo en cuenta la semántica del problema nos interesa incluir métodos que nos ayuden a determinar si la persona es mayor de edad, si es europea o no y si puede matricularse en un programa de doctorado porque tenga una titulación igual o superior a un máster. Valore en la medida de lo posible, la gestión y uso de excepciones, así como la posibilidad de crear un programa donde se use la clase persona. Tenga en cuenta, además de las consiguientes consideraciones semánticas del problema, los siguientes requisitos para el programa:

- La interfaz de usuario será en línea de comando. Se tienen que desarrollar clases con métodos específicos para leer de teclado y escribir en pantalla cadenas y números.
- Utilice excepciones en aquellos puntos del programa donde sea necesario controlar el flujo del programa.
- No acople las clases de dominio con interfaz de usuario (p.ej. no genere salidas por pantalla en las clases de la lógica de dominio)

ET.02.02 Primer Problema

1. Escribir, al menos el pseudocódigo correspondiente al método o a los métodos identificados.

```
package codigo_probi;
import java.time.LocalDate;

public class prpb1 {

    public class Persona {
        private String nombre;
        private String apellidos;
        private LocalDate fechaNacimiento;
        private String nacionalidad;
        private String titulacion;
        private boolean certingles;
        private String telefono;
        private String email;

        // Constructor para inicializar una Persona con validaciones
        public Persona(String nombre, String apellidos, LocalDate fechaNacimiento,

        // Método para verificar si es mayor de edad
        public boolean esMayorDeEdad() {
            return calcularEdad() >= 18;
        }

        // Método para verificar si es europea
        public boolean esEuropea() {
            String[] paisesEuropeos = {"España", "Francia", "Alemania", "Italia", "Portugal"};
            return Arrays.asList(paisesEuropeos).contains(this.nacionalidad);
        }

        // Método para verificar si puede matricularse en un doctorado
        public boolean puedeDoctorado() {
            return this.titulacion.equalsIgnoreCase("Máster") || this.titulacion.equalsIgnoreCase("Doctorado");
        }

        // Método auxiliar para calcular la edad
        private int calcularEdad() {
            return Period.between(this.fechaNacimiento, LocalDate.now()).getYears();
        }

        // Métodos de validación (simulados como placeholders)
        private boolean validarNombre(String nombre) {
            return nombre != null && !nombre.trim().isEmpty();
        }

        private boolean validarApellidos(String apellidos) {
            return apellidos != null && !apellidos.trim().isEmpty();
        }

        private boolean validarFechaNacimiento(LocalDate fechaNacimiento) {
            return fechaNacimiento != null && fechaNacimiento.isBefore(LocalDate.now());
        }

        private boolean validarNacionalidad(String nacionalidad) {
            return nacionalidad != null && !nacionalidad.trim().isEmpty();
        }

        private boolean validarTitulacion(String titulacion) {
            return titulacion != null && (titulacion.equalsIgnoreCase("Licenciatura") ||
                titulacion.equalsIgnoreCase("Máster") ||
                titulacion.equalsIgnoreCase("Doctorado"));
        }

        private boolean validarEmail(String email) {
            return email != null && email.contains("@");
        }

        private boolean validarTelefono(String telefono) {
            return telefono != null && telefono.matches("\\d{9}");
        }
    }
}
```

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

- Edad: int
- Nacionalidad: PaisesEuropa [String]
- NacionalidaEuropeo: europeo/ no europeo
- Titulación: Dos aspectos
 - Máster: True/False
 - Superior_Master: True/False
- Certificado_ingles: True/False

3) 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.

	<i>Clases de equivalencia</i>	<i>Conjunto de valores</i>	<i>Conjetura de errores</i>	<i>Valores Totales</i>
Nombre	{Alonso}	Manuel	" "	3
Apellidos	{Zamora}	Sanchez	" "	3
Telefono	{123456789}	987654321	Telefono.length() != 9	3
Email	EsUnEjemplo@gmail.com	OtroEjemplo@gmail.com	!email.contains("@")	3
Edad	(-oo,0) u (0,18) u [18, +oo)	-1,15, 50, 0, 1, 18, 17,19	500	9
Nacionalidad	{España, Francia, Italia, Brasil, Panamá, Japón}	España, Francia, Italia, Brasil, Panamá, Japón	" "	7
NacionalidadEuropeo	EUROPEA U NO EUROPEA	TRUE / FALSE	NULL	3
Certificado_ingles	TRUE U FALSE	TRUE / FALSE	NULL	3
Titulacion	MÁSTER / DOCTORADO	M D M, D	[]	4

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

- Para una cobertura para cada uso tomaremos el valor el máximo del número de valores de una variable, para así combinar todos los valores una vez:

EACH-USE =9 casos de pruebas.

La variable con más el máximo número de valores es **Edad**

- Para una combinación por pares de valores o pair-wise debemos tomar los dos máximos números de valores de dos variables y multiplicarlos:

PAIR-WISE = 9 * 7 = 63 casos de pruebas.

- Para la combinación de n-veces, es la que mayor número de casos de pruebas nos dará debido a que debemos multiplicar los números totales de valores de cada variable.

N-WISE = 3*3*3*3*9*7*3*3*4 = 183.707 casos de prueba.

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

Tomamos el valor de número de valores máximo:

Caso	Nombre	Apellidos	Teléfono	Email	Edad	Nacionalidad	Nacionalidad Europeo	Certificado Inglés	Titulación
1	Alonso	Zamora	123456789	EsUnEjemplo@gmail.com	-1	España	TRUE	TRUE	M
2	Manuel	Sánchez	987654321	OtroEjemplo@gmail.com	15	Francia	TRUE	FALSE	D
3	""	""	Inválido	Inválido (sin @)	50	Italia	NULL	NULL	NULL
4	Alonso	Sánchez	123456789	EsUnEjemplo@gmail.com	0	Brasil	FALSE	TRUE	M, D
5	Manuel	Zamora	987654321	OtroEjemplo@gmail.com	1	Panamá	FALSE	FALSE	NULL
6	""	Sánchez	Inválido	Inválido (sin @)	18	Japón	FALSE	TRUE	M
7	Alonso	""	123456789	EsUnEjemplo@gmail.com	17	España	TRUE	FALSE	D
8	Manuel	""	987654321	OtroEjemplo@gmail.com	19	Francia	TRUE	NULL	M, D
9	""	Zamora	Inválido	Inválido (sin @)	500	Italia	NULL	TRUE	D

6) Defina conjuntos de pruebas para alcanzar cobertura pairwise usando el algoritmo explicado en clase. Se pueden comprobar los resultados con el programa PICT2

Tenemos que comprobar los resultados de 63 casos de prueba, para ello tal como se nos recomienda utilizaremos el programa PICT2. El programa no se encuentra disponible, por lo que hemos usado la siguiente página web:

<https://pairwise.yuuniworks.com/>

A continuación, mostraremos la tabla con los resultados:

Nombre	Apellidos	Telefono	Email	Edad	Nacionalidad	NacionalidadEuropeo	Certificado_ingles	Titulacion
VACIO	Zamora	1,23E+08	ejemplo@gmail.com	19	Italia	SinValor	True	Master
Manuel	VACIO	VACIO	otro@gmail.com	1	Espana	Europea	SinValor	Master-Doctorado
VACIO	Sanchez	9,88E+08	VACIO	17	VACIO	NoEuropea	False	Doctorado
Alonso	VACIO	1,23E+08	VACIO	50	Brasil	Europea	True	SinValor
Alonso	Zamora	9,88E+08	otro@gmail.com	1	Panama	SinValor	True	Doctorado
Alonso	Sanchez	VACIO	ejemplo@gmail.com	15	Francia	SinValor	False	Master-Doctorado
Manuel	VACIO	9,88E+08	ejemplo@gmail.com	500	Italia	NoEuropea	SinValor	SinValor
Manuel	Zamora	9,88E+08	VACIO	18	Espana	Europea	False	Master
Manuel	Sanchez	1,23E+08	otro@gmail.com	17	Italia	NoEuropea	True	Master-Doctorado
VACIO	Zamora	VACIO	VACIO	0	Italia	Europea	SinValor	Doctorado
VACIO	VACIO	1,23E+08	otro@gmail.com	0	Espana	SinValor	False	SinValor
Alonso	Sanchez	VACIO	otro@gmail.com	1	Brasil	NoEuropea	SinValor	Master
VACIO	Sanchez	VACIO	VACIO	1	Japon	SinValor	False	SinValor
Manuel	VACIO	1,23E+08	ejemplo@gmail.com	17	Espana	SinValor	SinValor	Doctorado
Manuel	Zamora	VACIO	otro@gmail.com	19	VACIO	Europea	SinValor	Master-Doctorado
Manuel	VACIO	VACIO	ejemplo@gmail.com	17	Panama	Europea	False	Master
VACIO	Zamora	1,23E+08	otro@gmail.com	18	Francia	NoEuropea	True	SinValor
VACIO	Sanchez	1,23E+08	VACIO	500	Panama	Europea	True	Master-Doctorado
Manuel	Zamora	9,88E+08	ejemplo@gmail.com	50	Panama	NoEuropea	SinValor	Master-Doctorado
Manuel	Zamora	1,23E+08	otro@gmail.com	15	Panama	Europea	True	SinValor
Alonso	Zamora	VACIO	otro@gmail.com	17	Japon	Europea	True	Master
Alonso	VACIO	9,88E+08	otro@gmail.com	500	Francia	SinValor	SinValor	Master
Manuel	Sanchez	9,88E+08	ejemplo@gmail.com	0	Japon	NoEuropea	True	Master-Doctorado
Alonso	Sanchez	VACIO	otro@gmail.com	500	Espana	NoEuropea	True	Doctorado
VACIO	Sanchez	VACIO	otro@gmail.com	50	Espana	SinValor	False	Master
Alonso	VACIO	1,23E+08	VACIO	19	Japon	NoEuropea	SinValor	Doctorado
Manuel	Sanchez	1,23E+08	VACIO	50	Francia	Europea	SinValor	Doctorado
Alonso	VACIO	VACIO	ejemplo@gmail.com	18	Italia	SinValor	False	Doctorado
VACIO	Zamora	9,88E+08	ejemplo@gmail.com	19	Brasil	SinValor	False	SinValor
Alonso	Zamora	9,88E+08	VACIO	0	Panama	Europea	True	Master
VACIO	VACIO	9,88E+08	VACIO	15	Espana	NoEuropea	SinValor	Master
Alonso	VACIO	1,23E+08	ejemplo@gmail.com	15	VACIO	SinValor	True	Doctorado
Alonso	Sanchez	1,23E+08	VACIO	19	Francia	NoEuropea	True	SinValor
Alonso	Sanchez	9,88E+08	VACIO	15	Japon	SinValor	False	SinValor
Manuel	Zamora	1,23E+08	ejemplo@gmail.com	-1	Brasil	SinValor	SinValor	Doctorado
VACIO	Sanchez	9,88E+08	ejemplo@gmail.com	0	Francia	NoEuropea	SinValor	SinValor
Manuel	Sanchez	VACIO	VACIO	18	VACIO	SinValor	SinValor	Master-Doctorado
Manuel	VACIO	1,23E+08	ejemplo@gmail.com	18	Brasil	NoEuropea	False	Master-Doctorado
Manuel	VACIO	1,23E+08	ejemplo@gmail.com	15	Brasil	SinValor	SinValor	SinValor
VACIO	Sanchez	VACIO	ejemplo@gmail.com	50	VACIO	Europea	True	SinValor
Alonso	Zamora	1,23E+08	ejemplo@gmail.com	1	Italia	NoEuropea	True	SinValor
VACIO	Zamora	1,23E+08	VACIO	17	Brasil	NoEuropea	True	SinValor
VACIO	VACIO	VACIO	otro@gmail.com	-1	Italia	NoEuropea	True	SinValor
Alonso	Sanchez	9,88E+08	VACIO	-1	Panama	Europea	False	Master-Doctorado
Manuel	Zamora	1,23E+08	ejemplo@gmail.com	500	Japon	NoEuropea	False	Master
Manuel	Zamora	1,23E+08	VACIO	-1	Espana	NoEuropea	SinValor	Master
VACIO	Sanchez	1,23E+08	ejemplo@gmail.com	500	VACIO	Europea	True	Master
Manuel	VACIO	VACIO	VACIO	15	Italia	Europea	False	Master-Doctorado
Manuel	VACIO	9,88E+08	ejemplo@gmail.com	0	VACIO	NoEuropea	SinValor	Master-Doctorado
VACIO	Sanchez	9,88E+08	VACIO	-1	Francia	Europea	True	Master
Alonso	VACIO	1,23E+08	VACIO	17	Francia	Europea	False	Master-Doctorado
Alonso	Sanchez	1,23E+08	otro@gmail.com	500	Brasil	SinValor	True	Master-Doctorado
VACIO	Zamora	9,88E+08	ejemplo@gmail.com	1	Francia	Europea	False	Master-Doctorado
Manuel	Zamora	VACIO	VACIO	18	Panama	Europea	False	Master-Doctorado
VACIO	VACIO	VACIO	VACIO	-1	Japon	Europea	True	Master-Doctorado
Alonso	Sanchez	VACIO	otro@gmail.com	50	Japon	Europea	SinValor	SinValor
Alonso	VACIO	1,23E+08	ejemplo@gmail.com	19	Panama	Europea	False	Doctorado
Manuel	VACIO	VACIO	otro@gmail.com	18	Japon	SinValor	True	Master-Doctorado
Alonso	Zamora	9,88E+08	ejemplo@gmail.com	50	Italia	NoEuropea	False	Master
VACIO	VACIO	9,88E+08	ejemplo@gmail.com	-1	VACIO	SinValor	SinValor	Master-Doctorado
Manuel	Zamora	1,23E+08	VACIO	0	Brasil	Europea	True	SinValor
VACIO	Zamora	VACIO	otro@gmail.com	19	Espana	SinValor	False	Master-Doctorado
Alonso	VACIO	1,23E+08	VACIO	1	VACIO	Europea	True	Master-Doctorado

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

1. Primera decisión: *if* (*edad* >= 18)

A	Valor A
0	17
1	24

2. Segunda decisión: *if* (*nacionalidad* == “España” || *nacionalidad* == “Francia” || *nacionalidad* == “Italia”)

A	B	C	A or B or C	Valor A	Valor B	Valor C
0	0	0	0	“Japón”		
0	0	1	1			“Italia”
0	1	0	1		“Francia”	
0	1	1	1			
1	0	0	1	“España”		
1	0	1	1			
1	1	0	1			
1	1	1	1			

En esta tabla, solo una variable puede ser TRUE, porque la nacionalidad no puede ser a la vez “España”, “Francia” y “Italia”.

3. - Tercera decisión: *if* (*titulación* == “Máster” || *titulación* == “Doctorado”):

A	B	A or B	Valor A	Valor B
0	0	0	“Grado”	
0	1	1		“Doctorado”
1	0	1	“Máster”	
1	1	1		

En esta tabla pasa lo mismo que en la anterior, dado que la titulación no puede ser a la vez “Máster” y “Doctorado”

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

- Primera decisión: *if* (*edad* >= 18)

A	DOMINANTE	DECISIÓN
0	A	Edad = “17”
1	A	Edad = “24”

- Segunda decisión: *if* (*nacionalidad* == “España” || *nacionalidad* == “Francia” || *nacionalidad* == “Italia”)

A	B	C	A or B or C	DOMINANTE	DECISIÓN
0	0	0	0	A, B, C	Nacionalidad = “Japón”
0	0	1	1	C	Nacionalidad = “Italia”
0	1	0	1	B	
0	1	1	1	B, C	
1	0	0	1	A	
1	0	1	1	A, C	
1	1	0	1	A, B	
1	1	1	1	A, B, C	

- Tercera decisión: *if* (*titulación* == “Máster” || *titulación* == “Doctorado”):

A	B	A or B	DOMINANTE	DECISIÓN
0	0	0	A, B	Titulación = “Grado”
0	1	1	B	
1	0	1	A	Titulación = “Máster”
1	1	1	A, B	

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

1. Apartado 4: Combinatoria total (N-wise)

Esta combinación cubre todas las interacciones posibles entre los valores de todas las variables. Es ideal para encontrar defectos en sistemas críticos, pero es impráctico debido al número elevado de pruebas.

2. Apartado 5: Each-use

Cada valor de cada variable se usa al menos una vez. Es una estrategia básica que garantiza cobertura mínima, pero no asegura que todas las interacciones sean probadas. Es útil como un enfoque inicial, pero insuficiente para detectar defectos que dependen de interacciones.

3. Apartado 6: Pairwise

Es el enfoque más eficiente y efectivo. Garantiza que todas las combinaciones de pares sean probadas, lo que cubre la mayoría de los defectos encontrados en aplicaciones reales. La cobertura es significativamente mejor que la de "each-use" con un costo mucho menor que "n-wise".

Enunciado problema 2:

Escribir y probar un método que permita determinar qué tipo de cuenta bancaria se le puede ofrecer a una persona que se considera cliente joven potencial (el método aceptará un objeto de tipo cliente con tantos atributos como sean necesarios – use los métodos set y get-) en función de unas determinadas condiciones que nuestro departamento de marketing usando las más poderosas técnicas de ML/IA ha establecido:

- 1) Si es menor de **edad**, está estudiando, y **vive habitualmente en casa de sus padres**, el tipo de producto que se le ofrecerá será la “Cuenta Confort”.
- 2) Si tiene menos de 25 años y **está estudiando**, pero desplazado del domicilio familiar, se le ofrecerá la “Cuenta Vamos que tú puedes”.
- 3) Si es mayor de edad, y si **ha empezado a trabajar**, pero vive con sus padres, se le ofrecerá la “Cuenta Ahorra ahora que puedes”; por el contrario, si ya no vive con sus padres, se le ofrecerá la “Cuenta Saltando del Nido”
- 4) Si es mayor de 25 años, y está trabajando, pero vive con sus padres, se le ofrecerá el producto “Cuenta Independízate que va siendo hora”
- 5) Si es mayor de 25 años, está trabajando y ya no vive en el domicilio familiar, se le ofrecerá la “Cuenta Bienvenido a la Vida Adulta”. Se supone que no puede haber ambigüedades en el tipo de cuenta que se le puede ofrecer a un determinado cliente. Si las encontrara, escriba las suposiciones y excepciones que considere más adecuadas.

ET.02.02 Segundo Problema

1. Escribir, al menos el pseudocódigo correspondiente al método o a los métodos identificados.

Solo se ha identificado un método en este problema:

determinarTipoDeCuenta(cliente: Cliente): String

// Primero verificamos las condiciones en orden de prioridad según las reglas del marketing

Caso 1: Si es menor de edad, está estudiando y vive con sus padres

si cliente.edad < 18 y cliente.estaEstudiando == verdadero y cliente.viveConPadres == verdadero entonces

devolver "Cuenta Confort"

Caso 2: Si es menor de 25 años, está estudiando y vive desplazado (no con sus padres)

sí cliente.edad < 25 y cliente.estaEstudiando == verdadero y cliente.viveConPadres == falso entonces

devolver "Cuenta Vamos que tú puedes"

Caso 3: Si es mayor de edad, trabaja y vive con sus padres

sí cliente.edad >= 18 y cliente.estaTrabajando == verdadero y cliente.viveConPadres == verdadero entonces

devolver "Cuenta Ahorra ahora que puedes"

Caso 4: Si es mayor de edad, trabaja y no vive con sus padres

sí cliente.edad >= 18 y cliente.estaTrabajando == verdadero y cliente.viveConPadres == falso entonces

devolver "Cuenta Saltando del Nido"

Caso 5: Si es mayor de 25 años, trabaja y vive con sus padres

sí cliente.edad > 25 y cliente.estaTrabajando == verdadero y cliente.viveConPadres == verdadero entonces

devolver "Cuenta Independízate que va siendo hora"

Caso 6: Si es mayor de 25 años, trabaja y no vive con sus padres

sí cliente.edad > 25 y cliente.estaTrabajando == verdadero y cliente.viveConPadres == falso entonces

devolver "Cuenta Bienvenido a la Vida Adulta"

Si no se cumple ninguna de las condiciones anteriores, lanzar una excepción o mensaje de error

lanzar "No se puede determinar el tipo de cuenta"

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

Para probar el método que determina el tipo de cuenta que se le ofrece al cliente, es necesario considerar las siguientes variables clave:

1. Cliente Edad (entero):

Este atributo es crucial para identificar si el cliente es **menor de 18 años**, **menor de 25 años** o **mayor de 25 años**, lo que influye directamente en el tipo de cuenta a ofrecer.

2. Está estudiando(boolean):

Dependiendo de si el valor es true o false, el cliente podrá optar a diferentes productos. Por ejemplo, si el cliente está estudiando, puede acceder a productos específicos para estudiantes, como la "Cuenta Confort" o la "Cuenta Vamos que tú puedes".

3. Vive con sus padres(boolean):

Esta variable ayuda a determinar el tipo de cuenta a ofrecer según la situación de vivienda del cliente. Si el cliente vive con sus padres, es posible que se le ofrezcan productos orientados a la seguridad financiera, como la "Cuenta Ahorra ahora que puedes", mientras que, si no vive con sus padres, se le pueden ofrecer cuentas como la "Cuenta Saltando del Nido".

4. Está trabajando(boolean):

El estado de empleo del cliente es importante para ofrecer productos específicos para personas con estabilidad económica, como la "Cuenta Ahorra ahora que puedes" o la "Cuenta Bienvenido a la Vida Adulta". Si el cliente está trabajando, se le puede ofrecer una gama de productos más amplia y adaptada a su situación laboral.

Estas variables son esenciales para probar las condiciones de negocio del método, y su combinación determinará el tipo de cuenta que se le ofrece a cada cliente según su perfil personal y situación

3) 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.

En este apartado hemos identificado las **clases de equivalencia** de cada variable basándonos en los casos de cada una.

Para el **conjunto de valores** hemos seleccionado primero un valor para cada una de las clases de equivalencia y luego hemos añadido los **valores límite** que corresponden a los valores límite de cada uno de los casos.

Para la **conjetura de valores** hemos tomado valores que suponemos que darán error en la ejecución y prueba de los casos.

Para el **número de valores** hemos contado todos los valores totales incluyendo el de conjetura de errores.

VARIABLE	CLASES DE EQUIVALENCIA	CONJUNTO DE VALORES	CONJETURA DE ERRORES	NÚMERO DE VALORES
Cliente Edad	$(-\infty, 0) \cup (0, 18) \cup [18, 25) \cup [25, +\infty)$	-1, 15, 18, 25, 0, 1, 17, 19, 24, 26	400	11
Está estudiando	FALSO, VERDADERO	FALSO, VERDADERO	""	3
Vive con sus padres	FALSO, VERDADERO	FALSO, VERDADERO	""	3
Está trabajando	FALSO, VERDADERO	FALSO, VERDADERO	""	3

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

- Para una cobertura para cada uso tomaremos el valor el máximo del número de valores de una variable, para así combinar todos los valores una vez:
EACH-USE = 11 casos de pruebas
- Para una combinación por pares de valores o pair-wise debemos tomar los dos máximos números de valores de dos variables y multiplicarlos:
PAIR-WISE = $11 * 3 = 33$ casos de pruebas
- Para la combinación de n-veces, es la que mayor número de casos de pruebas nos dará debido a que debemos multiplicar los números totales de valores de cada variable.
N-WISE = $11 * 3 * 3 * 3 = 297$ casos de prueba

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

Para garantizar que cada valor identificado en el conjunto de pruebas sea usado al menos una vez, diseñamos los siguientes casos de prueba, cubriendo las clases de equivalencia y valores límites.

Caso	Edad	Está Estudiando	Vive con Padres	Está Trabajando	Resultado esperado
1	-1	FALSO	FALSO	FALSO	Error: "No se puede determinar el tipo de cuenta"
2	15	VERDADERO	VERDADERO	FALSO	"Cuenta Confort"
3	18	VERDADERO	FALSO	FALSO	"Cuenta Vamos que tú puedes"
4	25	FALSO	VERDADERO	VERDADERO	"Cuenta Ahorra ahora que puedes"
5	26	FALSO	FALSO	VERDADERO	"Cuenta Bienvenido a la Vida Adulta"
6	17	VERDADERO	FALSO	VERDADERO	"Cuenta Vamos que tú puedes"
7	19	FALSO	VERDADERO	FALSO	"Cuenta Ahorra ahora que puedes"
8	24	FALSO	VERDADERO	VERDADERO	"Cuenta Independízate que va siendo hora"
9	18	VERDADERO	VERDADERO	VERDADERO	"Cuenta Ahorra ahora que puedes"
10	25	FALSO	FALSO	FALSO	Error: "No se puede determinar el tipo de cuenta"

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

Para optimizar la cobertura combinatoria, utilizamos el método **pairwise testing**, que garantiza que todas las combinaciones posibles de dos variables sean probadas al menos una vez.

Caso	Edad	Está Estudiando	Vive con Padres	Está Trabajando	Resultado esperado
1	15	VERDADERO	VERDADERO	FALSO	"Cuenta Confort"
2	15	VERDADERO	FALSO	VERDADERO	"Cuenta Vamos que tú puedes"
3	15	FALSO	VERDADERO	VERDADERO	"Cuenta Ahorra ahora que puedes"
4	18	FALSO	FALSO	VERDADERO	"Cuenta Saltando del Nido"
5	18	VERDADERO	VERDADERO	FALSO	"Cuenta Confort"
6	18	VERDADERO	FALSO	FALSO	"Cuenta Vamos que tú puedes"
7	18	FALSO	VERDADERO	FALSO	"Cuenta Ahorra ahora que puedes"
8	24	FALSO	VERDADERO	VERDADERO	"Cuenta Ahorra ahora que puedes"
9	24	VERDADERO	FALSO	VERDADERO	"Cuenta Saltando del Nido"
10	24	FALSO	FALSO	VERDADERO	"Cuenta Ahorra ahora que puedes"
11	25	VERDADERO	VERDADERO	FALSO	"Cuenta Ahorra ahora que puedes"
12	25	FALSO	VERDADERO	VERDADERO	"Cuenta Independízate que va siendo hora"
13	25	FALSO	FALSO	FALSO	Error: "No se puede determinar el tipo de"
14	26	VERDADERO	VERDADERO	FALSO	"Cuenta Confort"
15	26	FALSO	FALSO	VERDADERO	"Cuenta Bienvenido a la Vida Adulta"
16	26	FALSO	VERDADERO	FALSO	"Cuenta Independízate que va siendo hora"
17	15	FALSO	FALSO	VERDADERO	"Cuenta Bienvenido a la Vida Adulta"
18	18	VERDADERO	FALSO	VERDADERO	"Cuenta Vamos que tú puedes"
19	26	VERDADERO	FALSO	VERDADERO	"Cuenta Bienvenido a la Vida Adulta"
20	15	VERDADERO	VERDADERO	VERDADERO	"Cuenta Ahorra ahora que puedes"
21	18	FALSO	VERDADERO	FALSO	"Cuenta Ahorra ahora que puedes"
22	24	VERDADERO	VERDADERO	FALSO	"Cuenta Ahorra ahora que puedes"
23	25	VERDADERO	VERDADERO	VERDADERO	"Cuenta Ahorra ahora que puedes"
24	18	VERDADERO	VERDADERO	FALSO	"Cuenta Confort"
25	24	FALSO	FALSO	VERDADERO	"Cuenta Saltando del Nido"
26	26	FALSO	VERDADERO	FALSO	"Cuenta Independízate que va siendo hora"
27	25	VERDADERO	FALSO	FALSO	"Cuenta Saltando del Nido"
28	18	FALSO	VERDADERO	VERDADERO	"Cuenta Ahorra ahora que puedes"
29	25	FALSO	FALSO	VERDADERO	"Cuenta Saltando del Nido"
30	24	VERDADERO	FALSO	VERDADERO	"Cuenta Saltando del Nido"
31	15	FALSO	VERDADERO	VERDADERO	"Cuenta Ahorra ahora que puedes"
32	18	FALSO	VERDADERO	VERDADERO	"Cuenta Ahorra ahora que puedes"
33	25	VERDADERO	VERDADERO	FALSO	"Cuenta Ahorra ahora que puedes"

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

Caso 1

```
if (cliente.getEdad() < 18 && cliente.isEstaEstudiando() && cliente.isViveConPadres()) {
    return "Cuenta Confort";
}
```

$A = \text{Edad}$, $B = \text{EstaEstudiando}$ y $C = \text{ViveConPadres}$

A	B	C	A and B and C	Valor A	Valor B	Valor C
0	0	0	0	34	FALSE	FALSE
0	0	1	0			
0	1	0	0			
0	1	1	0			
1	0	0	0			
1	0	1	0			
1	1	0	0			
1	1	1	1	12	TRUE	TRUE

Caso 2

```
if (cliente.getEdad() < 25 && cliente.isEstaEstudiando() && !cliente.isViveConPadres()) {
    return "Cuenta Vamos que tú puedes";
}
```

$A = \text{Edad}$, $B = \text{EstaEstudiando}$ y $C = \text{ViveConPadres}$

A	B	C	A and B and $\neg C$	Valor A	Valor B	Valor C
0	0	0	0			
0	0	1	0	36	FALSE	TRUE
0	1	0	0			
0	1	1	0			
1	0	0	0			
1	0	1	0			
1	1	0	1	23	TRUE	FALSE
1	1	1	0			

Caso 3

```

if (cliente.getEdad() >= 18 && cliente.isEstaTrabajando() && cliente.isViveConPadres())
{
    return "Cuenta Ahorra ahora que puedes";}

```

$A = \text{Edad} \geq 18$, $B = \text{EstaTrabajando}$ y $C = \text{ViveConPadres}$

A	B	C	A and B and C	Valor A	Valor B	Valor C
0	0	0	0	15	FALSE	FALSE
0	0	1	0			
0	1	0	0			
0	1	1	0			
1	0	0	0			
1	0	1	0			
1	1	0	0			
1	1	1	1	22	TRUE	TRUE

Caso 4

```

if (cliente.getEdad() >= 18 && cliente.isEstaTrabajando() && !cliente.isViveConPadres())
{
    return "Cuenta Saltando del Nido";}

```

$A = \text{Edad}$, $B = \text{EstaTrabajando}$ y $C = \text{ViveConPadres}$

A	B	C	A and B and $\neg C$	Valor A	Valor B	Valor C
0	0	0	0			
0	0	1	0	8	FALSE	TRUE
0	1	0	0			
0	1	1	0			
1	0	0	0			
1	0	1	0			
1	1	0	1	23	TRUE	FALSE
1	1	1	0			

Caso 5

```

if (cliente.getEdad() > 25 && cliente.isEstaTrabajando() && cliente.isViveConPadres()) {
    return "Cuenta Independízate que va siendo hora";}

```

$A = \text{Edad}$, $B = \text{EstaTrabajando}$ y $C = \text{ViveConPadres}$

A	B	C	A and B and C	Valor A	Valor B	Valor C
0	0	0	0	20	FALSE	FALSE
0	0	1	0			
0	1	0	0			
0	1	1	0			
1	0	0	0			
1	0	1	0			
1	1	0	0			
1	1	1	1	30	TRUE	TRUE

Caso 6

```

if (cliente.getEdad() > 25 && cliente.isEstaTrabajando() && !cliente.isViveConPadres())
{
    return "Cuenta Bienvenido a la Vida Adulta";
}

```

$A = \text{Edad}$, $B = \text{EstaTrabajando}$ y $C = \text{ViveConPadres}$

A	B	C	A and B and $\neg C$	Valor A	Valor B	Valor C
0	0	0	0			
0	0	1	0	20	FALSE	TRUE
0	1	0	0			
0	1	1	0			
1	0	0	0			
1	0	1	0			
1	1	0	1	32	TRUE	FALSE
1	1	1	0			

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

```

if (cliente.getEdad() < 18 && cliente.isEstaEstudiando() && cliente.isViveConPadres()) {
    return "Cuenta Confort";
}

```

A	B	C	A and B and C	Dominante	Valor A	Valor B	Valor C
0	0	0	0	A, B, C	34	FALSE	FALSE
0	0	1	0	A, B			
0	1	0	0	A, C			
0	1	1	0	A			
1	0	0	0	B, C			
1	0	1	0	B			
1	1	0	0	C			
1	1	1	1	A, B, C	12	TRUE	TRUE

```

if (cliente.getEdad() > 25 && cliente.isEstaTrabajando() && cliente.isViveConPadres()) {
    return "Cuenta Independízate que va siendo hora";
}

```

$A = \text{Edad}$, $B = \text{EstaTrabajando}$ y $C = \text{ViveConPadres}$

A	B	C	A and B and C	Dominante	Valor A	Valor B	Valor C
0	0	0	0	A, B, C	14	FALSE	FALSE
0	0	1	0	A, B			
0	1	0	0	A, C			
0	1	1	0	A			
1	0	0	0	B, C			
1	0	1	0	B			
1	1	0	0	C			
1	1	1	1	A, B, C	48	TRUE	TRUE

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

Cobertura alcanzada en cada apartado

1. Apartado 4: Número máximo posible de casos (n-wise = 297 casos)
 - Este método genera todas las combinaciones posibles de los valores de las variables. Esta cobertura es completa ya que asegura que todas las combinaciones son probadas y detecta cualquier error relacionado con interacciones entre variables. Sin embargo, el número de casos puede ser muy alto, lo que incrementa el tiempo y costo de las pruebas.
2. Apartado 5: Each-use = 11 casos
 - Este enfoque selecciona un conjunto reducido de pruebas en el que cada valor de cada variable se utiliza al menos una vez. Esto garantiza que todas las clases de equivalencia y límites se prueben. La cobertura es parcial, ya que no considera combinaciones entre valores de diferentes variables, pero es suficiente para validar individualmente cada valor y reducir la cantidad de pruebas.
3. Apartado 6: Pair-wise = 33 casos
 - Este método combina pares de valores de las variables, asegurando que cada par de valores sea probado al menos una vez. Se enfoca en detectar errores en interacciones entre dos variables. La cobertura es moderada, ya que asegura que los errores que dependen de interacciones entre pares de variables sean detectados, pero no cubre interacciones complejas entre tres o más variables.

Enunciado problema 3:

Se pretende desarrollar una aplicación que genere una recomendación sobre actividades lúdicas en función de las condiciones meteorológicas, del estado de salud de los usuarios y de las características de los espacios donde se puede disfrutar de las actividades de ocio. Para ello se tendrá en cuenta los siguientes aspectos:

- Si la persona está sana y sin síntomas, no ha estado en contacto en las dos últimas semanas con nadie infectado, ha pasado el COVID y tiene cartilla de vacunación (pasaporte COVID en regla) puede realizar cualquiera de las actividades propuestas. En otro caso no podrá realizar ninguna actividad.
- Si la temperatura meteorológica está por debajo de 0 grados, la humedad relativa es menor que 15%, y hay precipitaciones de nieve o de agua, entonces lo mejor es quedarse en casa.
- Si la temperatura meteorológica está por debajo de 0 grados, la humedad relativa es menor que 15%, y no hay precipitaciones de nieve o de agua, se puede ir a esquiar, si no se supera el aforo permitido por la legislación pertinente.
- Si la temperatura meteorológica está entre 0 y 15 grados, y no hay precipitaciones de agua, entonces es posible ir a hacer senderismo, si no se supera aforo del espacio previsto.
- Si la temperatura meteorológica está entre 15 y 25 grados, no llueve, y no está nublado y no hay una humedad relativa superior al 60%, entonces se puede ir a hacer turismo al aire libre, si la ciudad no tiene restricciones de confinamiento.
- Si la temperatura meteorológica está entre 25 y 35 grados, y no llueve, la recomendación es irse de cañas, si el establecimiento no tiene problemas de aforo.
- Si la temperatura meteorológica es mayor que 30 grados, y no llueve, la recomendación es irse a la playa o a la piscina. La piscina no puede superar el aforo permitido.

ET.02.02 Tercer Problema

1. Escribir, al menos el pseudocódigo correspondiente al método o a los métodos identificados.

Método recomendarActividad:

Si estadoSalud == falso:

Devolver “No puede realizar ninguna actividad”

Si temperatura < 0 && humedad < 15% && precipitaciones == verdadero:

Devolver “Quédate en casa”

Si temperatura < 0 && humedad < 15% && precipitaciones == falso && aforoPermitido == verdadero:

Devolver “Ir a esquiar”

Si 0 <= temperatura < 15 && precipitaciones == falso && aforoPermitido == verdadero

Devolver “Hacer senderismo”

Si 15 <= temperatura < 25 && precipitaciones == falso && nublado == falso && humedad <= 60% && restriccionesLocales == falso:

Devolver “Hacer turismo al aire libre.”

Si 25 <= temperatura <= 35 && precipitaciones == falso && aforoPermitido == verdadero

Devolver “Ir de cañas”

Si temperatura > 30 && precipitaciones == falso && aforoPermitido == verdadero:

Devolver “Ir a la playa o piscina.”

Si no se cumple ninguna de las condiciones anteriores, devuelve el mensaje “Quédate en casa”

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

- estadoSalud (boolean): Si el usuario cumple con los requisitos de salud para participar en actividades
- temperatura (float): Temperatura en grados Celsius
- humedad (int): Humedad relativa en porcentaje.
- precipitaciones (boolean): indica si hay lluvia o nieve.
- nublado (boolean): indica si el clima está nublado
- aforoPermitido (boolean): Si el lugar tiene espacio suficiente para las actividades
- restriccionesLocales (boolean): Si hay restricciones locales como confinamiento.

3. 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.

	CLASES DE EQUIVALENCIA	CONJUNTO DE VALORES	CONJETURA DE ERRORES	VALORES TOTALES
ESTADOSALUD	TRUE U FALSE	TRUE, FALSE	Null	3
TEMPERATURA	$(-\infty, 0) \cup [0, 15) \cup [15, 25) \cup [25, 35] \cup (35, +\infty)$	-10, 14, 20, 27, 40, 0, 1, 15, 16, 24, 25, 35	200	12
HUMEDAD	$[0, 15) \cup [15, 60] \cup (60, +\infty)$	13, 40, 65, 0, 14, 15, 59, 60	150	8
PRECIPITACIONES	TRUE U FALSE	TRUE, FALSE	Null	3
NUBLADO	TRUE U FALSE	TRUE, FALSE	Null	3
AFOROPERMITIDO	TRUE U FALSE	TRUE, FALSE	Null	3
RESTRICCIONESLOCALES	TRUE U FALSE	TRUE, FALSE	Null	3

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

Para garantizar la cobertura de cada uso, tomaremos el máximo número de valores que puede asumir una variable, de modo que combinemos todos los valores al menos una vez.

EACH-USE = 12 casos de prueba

En este caso, la variable con el mayor número de valores es **Temperatura**.

Para una combinación por pares de valores o pair-wise debemos tomar los dos máximos números de valores de dos variables y multiplicarlos:

PAIR-WISE = $12 * 8 = 96$ casos de pruebas

Para la combinación de n-veces, es la que mayor número de casos de pruebas nos dará debido a que debemos multiplicar los números totales de valores de cada variable.

N-WISE = $3 * 12 * 8 * 3 * 3 * 3 * 3 = 23.328$ x1casos de prueba

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

La tabla presentada muestra un conjunto de **12 casos de prueba** diseñados para cumplir con la estrategia **Each-Use**, que garantiza que cada valor de las variables especificadas sea utilizado al menos una vez.

Caso	ESTADOSALUD	TEMPERATURA	HUMEDAD	PRECIPITACIONES	NUBLADO	AFOROPERMITIDO	RESTRICCIONESLOCALES
1	TRUE	-10	13	TRUE	TRUE	TRUE	TRUE
2	FALSE	14	40	FALSE	FALSE	FALSE	FALSE
3	TRUE	20	65	TRUE	FALSE	TRUE	FALSE
4	FALSE	27	0	FALSE	TRUE	FALSE	TRUE
5	TRUE	40	14	TRUE	TRUE	TRUE	FALSE
6	FALSE	0	15	FALSE	FALSE	TRUE	TRUE
7	TRUE	1	59	TRUE	FALSE	FALSE	TRUE
8	FALSE	15	60	FALSE	TRUE	TRUE	FALSE
9	TRUE	16	13	TRUE	FALSE	FALSE	TRUE
10	FALSE	24	40	FALSE	TRUE	TRUE	FALSE
11	TRUE	25	65	TRUE	TRUE	FALSE	TRUE
12	FALSE	35	14	FALSE	FALSE	TRUE	FALSE

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

ESTADO	SALUD	TEMPERATURA	HUMEDAD	PRECIPITACIONES	NUBLADO	AFORO	PERMITIDO	RESTRICCIONES	LOCALES
FALSE		15	65	TRUE	TRUE		FALSE		FALSE
FALSE		25	0	FALSE	FALSE		TRUE		TRUE
TRUE		27	60	TRUE	FALSE		TRUE		FALSE
TRUE		40	59	FALSE	TRUE		FALSE		TRUE
TRUE		0	14	TRUE	FALSE		FALSE		TRUE
FALSE		20	40	FALSE	TRUE		TRUE		FALSE
FALSE		0	60	FALSE	TRUE		TRUE		FALSE
TRUE		24	13	FALSE	TRUE		FALSE		TRUE
FALSE		35	60	TRUE	FALSE		FALSE		TRUE
FALSE		14	13	TRUE	FALSE		TRUE		FALSE
FALSE		0	15	FALSE	TRUE		FALSE		TRUE
TRUE		-10	65	FALSE	FALSE		TRUE		TRUE
FALSE		15	59	TRUE	FALSE		TRUE		TRUE
FALSE		27	14	FALSE	TRUE		FALSE		FALSE
TRUE		15	40	TRUE	FALSE		FALSE		TRUE
TRUE		14	40	FALSE	TRUE		FALSE		TRUE
TRUE		40	0	TRUE	FALSE		TRUE		FALSE
FALSE		-10	0	TRUE	TRUE		FALSE		FALSE
TRUE		20	14	TRUE	FALSE		TRUE		TRUE
FALSE		24	65	TRUE	FALSE		TRUE		FALSE
TRUE		20	15	TRUE	FALSE		FALSE		FALSE
TRUE		1	65	TRUE	FALSE		TRUE		FALSE
TRUE		35	15	FALSE	TRUE		TRUE		FALSE
TRUE		16	60	TRUE	FALSE		TRUE		FALSE
TRUE		15	15	FALSE	FALSE		FALSE		TRUE
TRUE		25	60	TRUE	TRUE		FALSE		FALSE
TRUE		35	40	TRUE	TRUE		FALSE		TRUE
TRUE		15	60	FALSE	TRUE		FALSE		TRUE
FALSE		16	14	FALSE	TRUE		FALSE		TRUE
FALSE		40	60	TRUE	FALSE		FALSE		FALSE
FALSE		25	59	FALSE	FALSE		FALSE		FALSE
TRUE		20	0	FALSE	TRUE		FALSE		FALSE
TRUE		35	13	TRUE	TRUE		FALSE		TRUE
FALSE		-10	40	FALSE	FALSE		FALSE		TRUE
TRUE		27	40	FALSE	TRUE		TRUE		TRUE
TRUE		15	13	TRUE	TRUE		TRUE		TRUE
FALSE		14	59	FALSE	FALSE		FALSE		FALSE
FALSE		27	0	FALSE	FALSE		FALSE		FALSE
FALSE		14	15	FALSE	TRUE		TRUE		FALSE
FALSE		24	60	FALSE	FALSE		TRUE		FALSE
TRUE		40	40	TRUE	FALSE		FALSE		FALSE
TRUE		27	13	TRUE	FALSE		FALSE		FALSE
FALSE		27	15	FALSE	TRUE		TRUE		FALSE
FALSE		24	59	TRUE	TRUE		FALSE		FALSE
TRUE		40	65	FALSE	TRUE		TRUE		TRUE
TRUE		0	65	FALSE	TRUE		TRUE		FALSE
TRUE		27	59	TRUE	FALSE		TRUE		FALSE
FALSE		24	0	TRUE	FALSE		FALSE		FALSE
FALSE		1	13	FALSE	TRUE		FALSE		TRUE
FALSE		25	15	TRUE	FALSE		TRUE		TRUE
FALSE		16	59	FALSE	FALSE		TRUE		TRUE
TRUE		14	0	FALSE	TRUE		FALSE		TRUE
TRUE		-10	15	FALSE	FALSE		TRUE		TRUE
FALSE		16	0	TRUE	FALSE		TRUE		TRUE
FALSE		-10	59	TRUE	TRUE		TRUE		TRUE
TRUE		16	65	FALSE	FALSE		TRUE		FALSE
TRUE		25	13	FALSE	TRUE		TRUE		FALSE
FALSE		1	60	TRUE	FALSE		TRUE		TRUE
TRUE		24	15	TRUE	FALSE		FALSE		FALSE
TRUE		14	14	TRUE	FALSE		TRUE		TRUE
FALSE		40	14	TRUE	FALSE		FALSE		TRUE
FALSE		14	65	TRUE	FALSE		TRUE		TRUE
FALSE		-10	13	TRUE	FALSE		TRUE		TRUE
FALSE		16	40	FALSE	FALSE		TRUE		FALSE
TRUE		15	14	FALSE	TRUE		FALSE		FALSE
TRUE		35	59	FALSE	FALSE		FALSE		TRUE
FALSE		1	59	TRUE	TRUE		TRUE		FALSE
FALSE		35	0	TRUE	TRUE		TRUE		FALSE
TRUE		0	13	TRUE	TRUE		FALSE		TRUE
TRUE		0	40	TRUE	TRUE		FALSE		TRUE
FALSE		35	14	TRUE	FALSE		TRUE		TRUE
TRUE		25	65	TRUE	FALSE		TRUE		TRUE
FALSE		24	40	TRUE	FALSE		FALSE		TRUE
TRUE		20	65	FALSE	TRUE		FALSE		FALSE
FALSE		14	60	TRUE	FALSE		FALSE		TRUE
FALSE		20	13	FALSE	TRUE		FALSE		FALSE
TRUE		0	0	TRUE	FALSE		TRUE		TRUE
TRUE		35	65	TRUE	TRUE		FALSE		TRUE
FALSE		-10	60	TRUE	TRUE		TRUE		FALSE
TRUE		15	0	FALSE	FALSE		TRUE		TRUE
TRUE		16	15	FALSE	TRUE		FALSE		FALSE
TRUE		1	40	FALSE	TRUE		TRUE		FALSE
FALSE		1	15	FALSE	FALSE		FALSE		FALSE
TRUE		25	14	TRUE	TRUE		TRUE		FALSE
FALSE		20	60	FALSE	TRUE		FALSE		TRUE
TRUE		-10	14	FALSE	FALSE		FALSE		FALSE
TRUE		24	14	TRUE	FALSE		TRUE		TRUE
FALSE		25	40	TRUE	FALSE		TRUE		FALSE
FALSE		40	15	TRUE	FALSE		TRUE		FALSE
TRUE		27	65	FALSE	FALSE		FALSE		FALSE
FALSE		16	13	TRUE	FALSE		FALSE		TRUE
FALSE		20	59	FALSE	FALSE		FALSE		TRUE
TRUE		0	59	FALSE	FALSE		TRUE		TRUE
FALSE		1	14	FALSE	TRUE		TRUE		TRUE
TRUE		1	0	TRUE	TRUE		FALSE		TRUE
TRUE		40	13	TRUE	FALSE		FALSE		FALSE

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

Tenemos 8 if, por lo tanto, tenemos que hacer 8 tablas:

if (! estadoSalud)

A	¬A	VA
0	1	FALSE
1	0	TRUE

A: ! estadoSalud

if (temperatura < 0 && humedad < 15 && precipitaciones)

A	B	C	A AND B AND C	VA	VB	VC
0	0	0	0	10	20	FALSE
0	0	1	0	12	23	TRUE
0	1	0	0	5	8	FALSE
0	1	1	0	3	10	TRUE
1	0	0	0	-6	21	FALSE
1	0	1	0	-8	22	TRUE
1	1	0	0	-12	12	FALSE
1	1	1	1	-10	14	TRUE

A: temperatura < 0 / B: humedad < 15 / C: precipitaciones

if (temperatura < 0 && humedad < 15 &&! precipitaciones && aforoPermitido)

A	B	C	D	A AND B AND ¬C AND D	VA	VB	VC	VD
0	0	0	0	0	1	16	TRUE	FALSE
0	0	0	1	0	2	18	TRUE	TRUE
0	0	1	0	0	4	19	FALSE	FALSE
0	0	1	1	0	7	20	FALSE	TRUE
0	1	0	0	0	9	13	TRUE	FALSE
0	1	0	1	0	2	12	TRUE	TRUE
0	1	1	0	0	12	10	FALSE	FALSE
0	1	1	1	0	2	7	FALSE	TRUE
1	0	0	0	0	-2	18	TRUE	FALSE
1	0	0	1	0	-9	21	TRUE	TRUE
1	0	1	0	0	-4	19	FALSE	FALSE
1	0	1	1	0	-5	22	FALSE	TRUE
1	1	0	0	0	-7	13	TRUE	FALSE
1	1	0	1	0	-2	7	TRUE	TRUE
1	1	1	0	0	-6	5	FALSE	FALSE
1	1	1	1	1	-7	3	FALSE	TRUE

A: temperatura < 0 / B: humedad < 15 / C!: precipitaciones / D: aforoPermitido

if (temperatura >= 0 && temperatura < 15 &&! precipitaciones && aforoPermitido)

A	B	C	D	A AND B AND ¬C AND D	VA	VB	VC	VD
0	0	0	0	0	-3	16	TRUE	FALSE
0	0	0	1	0	-5	18	TRUE	TRUE
0	0	1	0	0	-8	19	FALSE	FALSE
0	0	1	1	0	-2	20	FALSE	TRUE
0	1	0	0	0	0	13	TRUE	FALSE
0	1	0	1	0	2	12	TRUE	TRUE
0	1	1	0	0	4	10	FALSE	FALSE
0	1	1	1	0	5	7	FALSE	TRUE
1	0	0	0	0	-3	18	TRUE	FALSE
1	0	0	1	0	-4	21	TRUE	TRUE
1	0	1	0	0	-10	19	FALSE	FALSE
1	0	1	1	0	-12	22	FALSE	TRUE
1	1	0	0	0	4	13	TRUE	FALSE
1	1	0	1	0	5	7	TRUE	TRUE
1	1	1	0	0	25	5	FALSE	FALSE
1	1	1	1	1	5	3	FALSE	TRUE

A: temperatura ≥ 0 / B: humedad < 15 / C!: precipitaciones / D: aforoPermitido

if (temperatura ≥ 15 && temperatura < 25 &&! precipitaciones &&! nublado && humedad ≤ 60 &&! restriccionesLocales)

$X = A \text{ AND } \neg B \text{ AND } \neg C \text{ AND } D \text{ AND } \neg E$

A	B	C	D	E	X	VA	VB	VC	VD	VE
0	0	0	0	0	0	26	T	T	62	T
0	0	0	0	1	0	27	T	T	65	F
0	0	0	1	0	0	27	T	T	40	T
0	0	0	1	1	0	28	T	T	56	F
0	0	1	0	0	0	30	T	F	70	T
0	0	1	0	1	0	31	T	F	76	F
0	0	1	1	0	0	29	T	F	35	T
0	0	1	1	1	0	28	T	F	43	F
0	1	0	0	0	0	11	F	T	76	T
0	1	0	0	1	0	11	F	T	89	F
0	1	0	1	0	0	10	F	T	20	T
0	1	0	1	1	0	9	F	T	12	F
0	1	1	0	0	0	3	F	F	98	T
0	1	1	0	1	0	2	F	F	88	F
0	1	1	1	0	0	5	F	F	21	T
0	1	1	1	1	0	8	F	F	2	F
1	0	0	0	0	0	16	T	T	78	T
1	0	0	0	1	0	16	T	T	76	F

1	0	0	1	0	0	17	T	T	21	T
1	0	0	1	1	0	19	T	T	23	F
1	0	1	0	0	0	23	T	F	87	T
1	0	1	0	1	0	21	T	F	78	F
1	0	1	1	0	0	22	T	F	34	T
1	0	1	1	1	0	22	T	F	39	F
1	1	0	0	0	0	19	F	T	98	T
1	1	0	0	1	0	21	F	T	99	F
1	1	0	1	0	0	24	F	T	33	T
1	1	0	1	1	0	23	F	T	21	F
1	1	1	0	0	0	19	F	F	78	T
1	1	1	0	1	0	19	F	F	76	F
1	1	1	1	0	0	18	F	F	1	T
1	1	1	1	1	0	20	F	F	56	F

A: temperatura ≥ 15 $\&\&$ temperatura < 25

B!: precipitaciones

C!: nublado

D: humedad ≤ 60

E!: restriccionesLocales

if (temperatura ≥ 25 $\&\&$ temperatura ≤ 35 $\&\&$! precipitaciones $\&\&$ aforoPermitido)

A	B	C	A AND \neg B AND C	VA	VB	VC
0	0	0	0	10	TRUE	FALSE
0	0	1	0	12	FALSE	TRUE
0	1	0	0	35	TRUE	FALSE
0	1	1	0	33	FALSE	TRUE
1	0	0	0	6	TRUE	FALSE
1	0	1	0	8	FALSE	TRUE
1	1	0	0	28	TRUE	FALSE
1	1	1	1	30	FALSE	TRUE

A: temperatura >= 25 && temperatura <= 35
B!: precipitaciones
C: aforoPermitido

if (temperatura > 30 &&! precipitaciones && aforoPermitido)

A	B	C	A AND ¬B AND C	VA	VB	VC
0	0	0	0	10	TRUE	FALSE
0	0	1	0	12	FALSE	TRUE
0	1	0	0	35	TRUE	FALSE
0	1	1	0	33	FALSE	TRUE
1	0	0	0	6	TRUE	FALSE
1	0	1	0	8	FALSE	TRUE
1	1	0	0	42	TRUE	FALSE
1	1	1	1	40	FALSE	TRUE

A: temperatura > 30
B: !precipitaciones
C: aforoPermitido
if (true)

A: true

A	VA
0	FALSE
1	TRUE

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

Primera condición:

```
if (temperatura < 0 && humedad < 15 && precipitaciones) {
    return "Quédate en casa";
}
```

Caso	A: temperatura < 0	B: humedad < 15	C: precipitaciones	Resultado (Quédate en casa)	Dominante	Valores (temperatura, humedad, precipitaciones)
1	0 (False)	0 (False)	0 (False)	0 (False)	A, B, C	10°C, 20%, false
2	0 (False)	0 (False)	1 (True)	0 (False)	C	10°C, 20%, true
3	1 (True)	0 (False)	0 (False)	0 (False)	A	-5°C, 20%, false
4	1 (True)	1 (True)	0 (False)	0 (False)	B	-5°C, 10%, false
5	1 (True)	1 (True)	1 (True)	1 (True)	A, B, C	-5°C, 10%, true

Segunda condición:

```
if (temperatura < 0 && humedad < 15 &&! precipitaciones && aforoPermitido) {
    return "Ir a esquiar";
}
```

Caso	A: temperatura < 0	B: humedad < 15	C!: precipitaciones	D: aforoPermitido	Resultado (Ir a esquiar)	Dominante
1	0 (False)	0 (False)	0 (False)	0 (False)	0 (False)	A, B, C, D
2	1 (True)	0 (False)	0 (False)	0 (False)	0 (False)	A
3	1 (True)	1 (True)	0 (False)	0 (False)	0 (False)	B
4	1 (True)	1 (True)	1 (True)	0 (False)	0 (False)	C
5	1 (True)	1 (True)	1 (True)	1 (True)	1 (True)	A, B, C, D

Tercera condición:

```
if (temperatura >= 15 && temperatura < 25 && ! precipitaciones && !nublado &&
    humedad <= 60 && !restriccionesLocales) {
```

```
    return "Hacer turismo al aire libre.";
```

```
}
```

Caso	A: 15 <= temperatura < 25	B: ! precipitaciones	C: ! nublado	D: humedad <= 60	E: !restriccionesLocales	Resultado
1	0 (False)	0 (False)	0 (False)	0 (False)	0 (False)	0 (False)
2	1 (True)	0 (False)	0 (False)	0 (False)	0 (False)	0 (False)
3	1 (True)	1 (True)	0 (False)	0 (False)	0 (False)	0 (False)
4	1 (True)	1 (True)	1 (True)	0 (False)	0 (False)	0 (False)
5	1 (True)	1 (True)	1 (True)	1 (True)	0 (False)	0 (False)

Resumen de los casos generados:

- Primera decisión: 5 casos de prueba.
- Segunda decisión: 5 casos de prueba.
- Tercera decisión: 6 casos de prueba.

Total: 16 casos de prueba para cobertura MC/DC.

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

Cobertura alcanzada en cada apartado

Apartado 4: Número máximo posible de casos de prueba (n-wise = 23328 casos)

Este método genera todas las combinaciones posibles de los valores de las variables. Garantiza una cobertura completa, ya que todas las combinaciones son evaluadas, detectando cualquier error relacionado con interacciones entre múltiples variables. Sin embargo, el número total de casos es muy alto (23328), lo que aumenta significativamente el tiempo y costo de las pruebas.

Apartado 5: Each-use = 12 casos

Este enfoque selecciona un conjunto reducido de pruebas donde cada valor de cada variable es utilizado al menos una vez. En este caso, se seleccionaron 12 casos porque la variable con el mayor número de valores es *Temperatura*. La cobertura es parcial, ya que no considera

combinaciones entre valores de diferentes variables, pero asegura que todas las clases de equivalencia y límites individuales sean probadas, optimizando la cantidad de pruebas necesarias.

Apartado 6: Pair-wise = 96 casos

Este método genera combinaciones de pares de valores entre las variables, asegurando que cada par de valores sea probado al menos una vez. Se basa en multiplicar los máximos valores de dos variables con más combinaciones (*Temperatura* y otra variable con 8 valores). La cobertura es moderada, permitiendo detectar errores en interacciones entre pares de variables, aunque no cubre interacciones más complejas entre tres o más variables.