

6. Conversor de Idades a Categorías

Implementa `AgeClassifier.classify(int age)`, que devuelve:

- "Infantil" para idades entre 0 e 12 anos.
- "Adolescente" entre 13 e 17 anos.
- "Adulto" entre 18 e 64 anos.
- "Senior" a partir de 65 anos.

```
public class AgeClassifier {  
    public static String classify(int age) {  
        if (age < 0 || age > 120) { // NP1  
            throw new IllegalArgumentException("Idade non  
válida");  
        }  
        if (age <= 12) { // NP2  
            return "Infantil";  
        } else if (age <= 17) { // NP3  
            return "Adolescente";  
        } else if (age < 65) { // NP4  
            return "Adulto";  
        }  
        return "Infantil";  
    }  
}
```

COMPLEJIDAD CICLOMÁTICA:

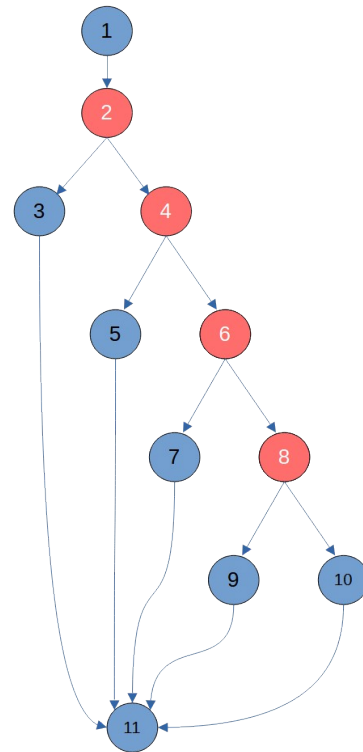
NODOS PREDICADOS = 4

$V(G) = NP + 1 = 4 + 1 = 5$

Errores corregidos: En el documento original, la última condición devolvía "Infantil" en lugar de "Senior", lo cual era incorrecto.

Nodos

1. Nodo de inicio.
2. Nodo de condición ($\text{age} < 0 \parallel \text{age} > 120$).
3. Nodo de retorno de excepción.
4. Nodo de condición ($\text{age} \leq 12$).
5. Nodo de retorno "Infantil".
6. Nodo de condición ($\text{age} \leq 17$).
7. Nodo de retorno "Adolescente".
8. Nodo de condición ($\text{age} \leq 64$).
9. Nodo de retorno "Adulto".
10. Nodo de retorno "Senior".



CAMINOS POSIBLES:

- 1, 2, 3, 11
- 1, 2, 4, 5, 11
- 1, 2, 4, 6, 7, 11
- 1, 2, 4, 6, 8, 9, 11
- 1, 2, 4, 6, 8, 10, 11

Tabla de Clases de Equivalencia

Condición de Entrada	Clases Válidas	Clases No Válidas
Un número entre 0 y 120, inclusive	$0 \leq \text{age} \leq 120$ (1)	$\text{age} < 0$ (2) $\text{age} > 120$ (3)

Categoría de Edad	Clases Válidas	Clases Inválidas
Infantil (0-12 años)	Edad entre 0 y 12 (4)	No aplica
Adolescente (13-17 años)	Edad entre 13 y 17 (5)	No aplica
Adulto (18-64 años)	Edad entre 18 y 64 (6)	No aplica
Senior (65 años o más)	Edad ≥ 65 (7)	No aplica

Casos de prueba con clases de equivalencia válidas

Entrada: age	Clases incluidas
5	(1), (4)
15	(1), (5)
30	(1), (6)
70	(1), (7)

Casos de prueba con clases de equivalencia no válidas

Entrada: password	Clases incluidas
-5	(2)
130	(3)

Código corregido

```
package contornos.ud3;

public class AgeClassifier {
    public static String classify(int age) {
        if (age < 0 || age > 120) {
            throw new IllegalArgumentException("Idade non válida");
        }
        if (age <= 12) {
            return "Infantil";
        } else if (age <= 17) {
            return "Adolescente";
        } else if (age < 65) {
            return "Adulto";
        }
        return "Senior";
    }
}
```

Pruebas Unitarias

- **Edad Infantil (0-12):**
 - Prueba con edad 5, el resultado esperado es "Infantil".
- **Edad Adolescente (13-17):**
 - Prueba con edad 15, el resultado esperado es "Adolescente".
- **Edad Adulto (18-64):**
 - Prueba con edad 30, el resultado esperado es "Adulto".
- **Edad Senior (65 y más):**
 - Prueba con edad 70, el resultado esperado es "Senior".
- **Edad Inválida:**
 - Prueba con edad -5, que debería lanzar una excepción `IllegalArgumentException` indicando "Idade non válida".
 - Prueba con edad 130, que también debería lanzar una excepción.

```
package contornos.ud3;

import org.junit.jupiter.api.Test;
import static org.junit.jupiter.api.Assertions.*;

class AgeClassifierTest {

    @Test
    void testInfantil() {
        assertEquals("Infantil", AgeClassifier.classify(5));
    }

    @Test
    void testAdolescente() {
        assertEquals("Adolescente", AgeClassifier.classify(15));
    }

    @Test
    void testAdulto() {
        assertEquals("Adulto", AgeClassifier.classify(30));
    }

    @Test
    void testSenior() {
        assertEquals("Senior", AgeClassifier.classify(70));
    }

    @Test
    void testEdadInvalidaNegativa() {
        assertThrows(IllegalArgumentException.class, () -> {
            AgeClassifier.classify(-5);
        });
    }

    @Test
    void testEdadInvalidaMayorA120() {
        assertThrows(IllegalArgumentException.class, () -> {
            AgeClassifier.classify(130);
        });
    }
}
```

