PRACTICA 3

Link del repositorio: **https://github.com/MaelGM/Pruebas_JUnit.git**

```java
public boolean compruebaIBAN(String anaves){
    boolean pais = false, digitos = false;
    if (anaves.length() != 24) return false;
    if (anaves.substring(0,2).equalsIgnoreCase( anotherString: "ES")) pais = true;
    try{
        BigInteger numeros = new BigInteger( val: anaves.substring(4)+"142800");
        BigInteger num = new BigInteger( val: "97");
        BigInteger resto = numeros.mod(num);
        if (98-resto.intValue() == Integer.parseInt(anaves.substring(2,4))) digitos = true;
    }catch (NumberFormatException | ArithmeticException e){
        e.printStackTrace();
        return false;
    }
    return pais && digitos;
}

public String generaIBAN(String entitat, String oficina, String dc, String compte){
    BigInteger numeros;
    String nums = entitat+oficina+dc+compte;
    if (entitat.length() != 4 || oficina.length() != 4 || dc.length() != 2 || compte.length() != 10) return null;
    try{
        numeros = new BigInteger( val: entitat+oficina+dc+compte+"142800");
    }catch (NumberFormatException | ArithmeticException e){
        return null;
    }
    BigInteger resto = numeros.mod(new BigInteger( val: "97"));
    String codigo = String.valueOf(98-resto.intValue());
    return "ES"+codigo+nums;
}
```

```java
import static org.junit.jupiter.api.Assertions.assertEquals;

public class CompteTest {
    @Test
    void compruebaIBAN_1(){
        Compte c = new Compte();
        assertEquals( expected: true, c.compruebaIBAN( anaves: "ES6621000418401234567891"));
    }
    @Test
    void compruebaIBAN_2(){
        Compte c = new Compte();
        assertEquals( expected: true, c.compruebaIBAN( anaves: "ES6000491500051234567892"));
    }
    @Test
    void compruebaIBAN_3(){
        Compte c = new Compte();
        assertEquals( expected: true, c.compruebaIBAN( anaves: "ES9420805801101234567891"));
    }
    @Test
    void compruebaIBAN_4(){
        Compte c = new Compte();
        assertEquals( expected: false, c.compruebaIBAN( anaves: "ES7600246912501234567891"));
    }
```

```java
    @Test
    void compruebaIBAN_5(){
        Compte c = new Compte();
        assertEquals( expected: false, c.compruebaIBAN( anaves: "ES4721000418401234567891"));
    }
    @Test
    void compruebaIBAN_6(){
        Compte c = new Compte();
        assertEquals( expected: false, c.compruebaIBAN( anaves: "ES8200491500051234567892"));
    }


    @Test
    void generaIBAN_1(){
        Compte c = new Compte();
        assertEquals( expected: "ES7100302053091234567895", c.generaIBAN( entitat: "0030", oficina: "2053", dc: "09", compte: "1234567895"));
    }
    @Test
    void generaIBAN_2(){
        Compte c = new Compte();
        assertEquals( expected: "ES1000492352082414205416", c.generaIBAN( entitat: "0049", oficina: "2352", dc: "08", compte: "2414205416"));
    }
    @Test
    void generaIBAN_3(){
        Compte c = new Compte();
        assertEquals( expected: "ES1720852066623456789011", c.generaIBAN( entitat: "2085", oficina: "2066", dc: "62", compte: "3456789011"));
    }
```

```java
@Test
void generaIBAN_4(){
    Compte c = new Compte();
    assertEquals( expected: null, c.generaIBAN( entitat: "2085", oficina: "2066", dc: "62", compte: "3456AE9011"));
}
@Test
void generaIBAN_5(){
    Compte c = new Compte();
    assertEquals( expected: null, c.generaIBAN( entitat: "208", oficina: "2066", dc: "62", compte: "3456789011"));
}
@Test
void generaIBAN_6(){
    Compte c = new Compte();
    assertEquals( expected: null, c.generaIBAN( entitat: "2080", oficina: "20A6", dc: "62", compte: "3456789011") );
}
@Test
void generaIBAN_7(){
    Compte c = new Compte();
    assertEquals( expected: null, c.generaIBAN( entitat: "2080", oficina: "2086", dc: "6", compte: "3456789011") );
}
@Test
void generaIBAN_8(){
    Compte c = new Compte();
    assertEquals( expected: null, c.generaIBAN( entitat: "2080", oficina: "2086", dc: "63", compte: "345678911"));
}
```