

```

package practica2junit;

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

import org.junit.jupiter.api.Test;

import practica2junit.JUnitPractica2_1.Cuenta;

public class JUnitPractica2_1Test {

    @Test
    public void testDepositar1() {
        int cod = -1506;
        int saldoinicial = -5000;
        int saldoesperado = -5250;
        Cuenta pruebadecuenta = new Cuenta(cod,saldoinicial);
        pruebadecuenta.depositar (-250);
        int saldoactual = pruebadecuenta.obtenersaldo();
        assertEquals (saldoesperado, saldoactual);
        //Comprobamos si el programa admite n meros negativos como
        //c digo de cuenta y saldo inicial.
        //Los admite, y el deposito es correcto. El problema es que
        //una vez que empezamos con un saldo negativo, es posible seguir
        //sacando dinero.
    }

    @Test
    public void testDepositar2() {
        int cod = 999999999;
        int saldoinicial = 999999999;
        int saldoesperado = 999999999;
        Cuenta pruebadecuenta = new Cuenta(cod,saldoinicial);
        pruebadecuenta.depositar (999999999);
        int saldoactual = pruebadecuenta.obtenersaldo();
        assertEquals (saldoesperado, saldoactual);
        //Comprobamos los l mites del programa.
        //No se pueden hacer dep sitos de m s de 5000  y
        //no es posible tener un saldo superior a 999 millones.
    }

    @Test
    public void testDepositar3() {
        int cod = 5656644;
        int saldoinicial = 250;
        int saldoesperado = -250;
    }

```

```

    Cuenta pruebadecuenta = new Cuenta(cod,saldoinicial);
    pruebadecuenta.depositar (-500);
    int saldoactual = pruebadecuenta.obtenersaldo();
    assertEquals (saldoesperado, saldoactual);
    //Comprobamos si el programa admite depositos negativos.
    //Se puede realizar depositos negativos, lo cual es incorrecto.
    //Ya disponemos de la función "retirar" para retirar dinero.
}

@Test
public void testDepositar5() {
    int cod = 4984984;
    int saldoinicial = 250;
    int saldoesperado = 0;
    Cuenta pruebadecuenta = new Cuenta(cod,saldoinicial);
    pruebadecuenta.retirar (250);
    int saldoactual = pruebadecuenta.obtenersaldo();
    assertEquals (saldoesperado, saldoactual);
    //El 0 cuenta como saldo en números rojos. Lo cual es erróneo,
debemos
    //de poder vaciar la cuenta.

}

@Test
public void testDepositar6() {
    int cod = 4645213;
    int saldoinicial = 500;
    int saldoesperado = -500;
    Cuenta pruebadecuenta = new Cuenta(cod,saldoinicial);
    pruebadecuenta.retirar (1000);
    int saldoactual = pruebadecuenta.obtenersaldo();
    assertEquals (saldoesperado, saldoactual);
    //Nos avisa de que no podemos dejar la cuenta negativo, es
correcto.
}

@Test
public void testDepositar7() {
    int cod = 4645213;
    int saldoinicial = 1500;
    int saldoesperado = 500;
    Cuenta pruebadecuenta = new Cuenta(cod,saldoinicial);
    pruebadecuenta.retirar (-1000);
    int saldoactual = pruebadecuenta.obtenersaldo();

```

```
    assertEquals (saldoesperado, saldoactual);  
  }  
  //Podemos realizar retiros negativos, lo que equivale a un ingreso.  
  //Para ello ya tenemos el método "deposito()".  
}
```