

```
package model;
```

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

```
import java.time.LocalDateTime;
```

```
import java.util.ArrayList;
```

```
import org.junit.jupiter.api.Test;
```

```
import exceptions.NoPicoCedulaException;
```

```
import exceptions.NoWithTIEException;
```

```
class Mini_MarketTest {
```

```
    private int today = LocalDateTime.now().getDayOfMonth();
```

```
    private Mini_Market mini_market;
```

```
    public void setupScenary() {
```

```
        mini_market=new Mini_Market();
```

```
    }
```

```
    //(1) Puede ingresar al minimercado por cumplir todas las condiciones
```

```
    @Test
```

```
    public void testRegister1() {
```

```
        setupScenary1();
```

```
        if(today%2 == 0) {
```

```
            try {
```

```
                Type type = Type.CC;
```

```

        String numID= "1006584131";
        mini_market.registerUser(type, numID);
    } catch (NoWithTIEException nwt) {
        System.out.println("No se permite el ingreso a menores
de edad\n");

        fail();
    } catch (NoPicoCedulaException npc) {
        System.out.println("No cumple con el Pico y Cedula, por
lo que no puede entrar a la tienda\n");
        fail();
    }
} else {
    try {
        Type type = Type.CC;
        String numID= "1006584121";
        mini_market.registerUser(type, numID);
    } catch (NoWithTIEException nwt) {
        System.out.println("No se permite el ingreso a menores
de edad\n");

        fail();
    } catch (NoPicoCedulaException npc) {
        System.out.println("No cumple con el Pico y Cedula, por
lo que no puede entrar a la tienda\n");
        fail();
    }
}
}

```

```

ArrayList<User> userRegistered = mini_market.getUserRegistered();
assertEquals(1, userRegistered.size());// El "1" significa que se
registra en el arreglo

```

```
}
```

//(2) El usuario que sí pudo ingresar, queda registrado en el arreglo total y en el arreglo de personas ingresadas

```
@Test
```

```
public void testRegister2() {  
    setupScenary1();  
    if(today%2 == 0) {  
        try {  
            Type type = Type.CC;  
            String numID= "1006584131";  
            mini_market.registerUser(type, numID);  
  
        } catch (NoWithTIException nwt) {  
            System.out.println("No se permite el ingreso a menores  
de edad\n");  
            fail();  
        } catch (NoPicoCedulaException npc) {  
            System.out.println("No cumple con el Pico y Cedula, por  
lo que no puede entrar a la tienda\n");  
            fail();  
        }  
    }else {  
        try {  
            Type type = Type.CC;  
            String numID= "1006584121";  
            mini_market.registerUser(type, numID);  
        } catch (NoWithTIException nwt) {
```

```

        System.out.println("No se permite el ingreso a menores
de edad\n");

        fail();

    } catch (NoPicoCedulaException npc) {

        System.out.println("No cumple con el Pico y Cedula, por
lo que no puede entrar a la tienda\n");

        fail();

    }

}

```

```

        ArrayList<User> userRegistered = mini_market.getUserRegistered();
        assertEquals(1, userRegistered.size()); // El "1" significa que se
registra en el contador

        assertEquals(1, mini_market.getAttempts()); // El "1" significa que se
registra en el arreglo

    }

```

//(3) NO puede ingresar al minimercado por ser menor de edad

@Test

```

public void testRegister3() {

    setupScenary1();

    Type type = Type.TI;

    String numID= "1006071586";

    try {

        mini_market.registerUser(type, numID);

    } catch (NoWithTIException nwt) {

```

```

        System.out.println("No se permite el ingreso a menores de
edad\n");
    } catch (NoPicoCedulaException npc) {
        fail();
    }

    ArrayList<User> userRegistered = mini_market.getUserRegistered();
    assertEquals(0, userRegistered.size()); // El "0" significa que no se
pudo registrar al arreglo
}

```

//(4) NO se puede registrar porque no le corresponde el día de acuerdo con el número de su cédula y el día del mes

```

@Test
public void testRegister4() {
    setupScenary1();
    if(today%2 == 0) {
        try {
            Type type = Type.CC;
            String numID= "1006584121";
            mini_market.registerUser(type, numID);
        } catch (NoWithTIException nwt) {
            System.out.println("No se permite el ingreso a menores
de edad\n");
            fail();
        } catch (NoPicoCedulaException npc) {
            System.out.println("No cumple con el Pico y Cedula, por
lo que no puede entrar a la tienda\n");

```

```

        }
    }else {
        try {
            Type type = Type.CC;
            String numID= "1006584131";
            mini_market.registerUser(type, numID);
        } catch (NoWithTIException nwt) {
            System.out.println("No se permite el ingreso a menores
de edad\n");

            fail();
        } catch (NoPicoCedulaException npc) {
            System.out.println("No cumple con el Pico y Cedula, por
lo que no puede entrar a la tienda\n");

        }
    }

    ArrayList<User> userRegistered = mini_market.getUserRegistered();
    assertEquals(0, userRegistered.size()); // El "0" significa que no se
pudo registrar al arreglo
}

}

```