

Pruebas setDescuento(int descuento) de la clase Cliente:

Dado que descuento es un porcentaje lo hemos definido como el rango [0-100] por lo tanto se seleccionarán dos valores inválidos(uno por debajo del rango y otro por arriba).

Por lo tanto se han definido las clases de equivalencia:

Condición de entrada	Clases de equivalencia válidas	Clases de equivalencia inválidas
descuento	V1. 44	N1. -1 N2. 101

Por lo tanto los casos de pruebas serán:

Casos de prueba	Combinaciones de clases de equivalencia	Salida
CP1	V1	Se introduce el valor a la variable descuento
CP2	N1	No se introduce el valor a la variable descuento
CP3	N2	No se introduce el valor a la variable descuento

```
@Test
public void setDescuentoCP1() {
    assertEquals(Cliente.validarDescuento(44), actual: true);
}

@Test
public void setDescuentoCP2() {
    assertEquals(Cliente.validarDescuento(-1), actual: false);
}

@Test
public void setDescuentoCP3() {
    assertEquals(Cliente.validarDescuento(101), actual: false);
}
```

Por último hemos realizado una prueba para comprobar que realice su funcionalidad correctamente:

```
@Test
public void setDescuentoFuncional() {
    cliente.setDescuento(50);
    assertEquals(cliente.getDescuento(), actual: 50);
}
```

Comprobando así que el descuento ha cambiado a 50 estando inicializado por defecto a 0.

-Pruebas addInscripcionCurso(InscripcionCurso inscripcionCurso):

El parámetro inscripcionCurso es una referencia de la clase InscripcionCurso, no te deja ni compilar si intentas meter cualquier otra cosa, exceptuando null el cual se introducirá en la lista sin problemas pudiendo causar problemas a futuro, por lo que esto se deberá comprobar.

Por lo tanto se han definido las clases de equivalencia:

Condición de entrada	Clases de equivalencia válidas	Clases de equivalencia inválidas
InscripcionCurso	V1. InscripcionCurso inscripcionCurso	N1. null

Por lo tanto los casos de pruebas serán:

Casos de prueba	Combinaciones de clases de equivalencia	Salida
CP1	V1	La inscripción del curso ha sido añadida con éxito a la lista de inscripciones del cliente.
CP2	N1	No se introduce el null, por lo que la lista se mantiene inalterada.

```
// CP1= V1(InscripcionCurso inscripcionCurso)
@Test
public void addInscripcionCursoCP1() {
    this.cliente.addInscripcionCurso(inscripcionCurso);
    assertEquals(this.cliente.getInscripcionesCurso().contains(inscripcionCurso),true);
}

// CP2= N1 (null)
@Test
public void addInscripcionCursoCP2() {
    this.cliente.addInscripcionCurso(null);
    assertEquals(this.cliente.getInscripcionesCurso().contains(null),false);
}
```