**2.1. Pruebas unitarias de la clase Seguro**

Método: *Seguro(potecia, cliente, cobertura)*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Clases Válidas | Valores | Clases No Válidas | Valores |
| Potencia | (0,90)  [90,110]  >110 | 1  43  89  90  100  110  150 | <=0 | 0  -30 |
| Cobertura | TODORIESGO  TERCEROSLUNAS  TERCEROS | 1000  600  400 | ¡{TODORIESGO, TERCEROSLUNAS, TERCEROS}  null | CUARTOS  null |
| Cliente | !null | cliente | null | null |

Método: *precio()*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Clases Válidas | Valores | Clases No Válidas | Valores |
| Cobertura | TODORIESGO  TERCEROSLUNAS  TERCEROS | 1000  600  400 | ¡{TODORIESGO, TERCEROSLUNAS, TERCEROS}  null | CUARTOS  null |
| Potencia | (0,90)  [90,110]  >110 | 1  43  89  90  100  110  150 | <= 0 | 0  -30 |
| Siniestralidad | [hoy-1año,hoy]  [hoy-3años,hoy-1año)  < hoy-3años  nunca | hoy-1año  hoy-0.5año  hoy  hoy-3años  hoy-2años  hoy-1año y 1 dia  hoy-3años y 1 dia  null | > hoy | hoy+1dia |
| Minusvalia | true  false | true  false |  |  |

Válidas:

1.1 (1,cliente,TERCEROS): 400

1.2 (43,cliente,TERCEROSLUNAS): 600 (comprobamos dos valores del tirón)

1.3 (89,cliente,TODORIESGO): 1000 (todos las coberturas probadas)

2.1 (90,cliente,TERCEROS): 400\*1.05 = 420

2.2 (100,cliente,TERCEROS): 420

2.3 (110,cliente,TERCEROS): 420

3.1 (150,cliente,TERCEROS): 400\*1.20 = 480 (todas las potencias comprobadas)

No Válidas:

1. (1,cliente,CUARTOS): DatoIncorrectoException

2. (1,cliente,null): DatoIncorrectoException (coberturas comprobadas)

3. (0,cliente,TERCEROS): DatoIncorrectoException

4. (-30,cliente,TERCEROS): DatoIncorrectoException (potencias comprobadas)

5. (1,null,TERCEROS): DatoIncorrectoException (cliente comprobado)

Método: *setFechaUltimoSiniestro(fechaUltimoSiniestro)*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Clases Válidas | Valores | Clases No Válidas | Valores |
| Siniestralidad | < hoy | hoy-1dia | > hoy | hoy+1dia |

Válidas:

{hoy - 1 dia}

{hoy + 1 dia}: DatoIncorrectoException

**2.2 GUI**

Pruebas de integración:

* Escribir fecha siniestro (3 casos)
* Elegir cobertura (3 casos)
* Escribir potencia (7 casos)
* Marcar minusvalia (2 casos)
* Pulsar CALCULAR
* Comprobar que el precio es el correcto

Implemetados:

{hace 100 dias, TERCEROS, 60, minusvalia}: 450.0

{hace 2 años, TODO\_RIESGO, 100, no minusvalia}: 1100.0

{hace 10 años, TERCEROS\_LUNAS, 200, no minusvalia}: 720.0

**2.3 ListaOrdenada**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Clases Válidas | Valores | Clases No Válidas | Valores |
| add(e) | e != null  lista vacía  lista con un elemento  lista con elementos | 2 en [] ([2])  4 en [2] ([2,4])  3 en [2,4]: ([2,3,4]) | e == null | null en []: NullPExc  null en [1,2,4]: NullPExc |
| get(i): e | 0 <= i < tamaño lista  lista con un elemento  lista con elementos | i=0 en [2]: (2, [2])  i=1 en [2,4]: (4, [2,4]) | i < 0  i > tamaño  lista vacía | i=-1 en [2,3]: IOOBExc  i=5 en [2,3]: IOOBExc  i=0 en []: IOOBExc |
| remove(i): e | 0 < i < tamaño lista  lsita con un elemento  lista con elementos | i=0 en [4]: (4, [])  i=2 en [2,4,8]: (8, [2,4]) | i < 0  i > tamaño  lista vacía | i=-1 en [2,3]: IOOBExc  i=5 en [2,3]: IOOBExc |
| size(): i | lista vacía  lista con un elemento  lista con elementos | []: 0  [2]: 1  [2,5,8,9]: 4 |  |  |
| clear() | lista vacía  lista con un elemento  lista con elementos | []: []  [6]: []  [2,4,5,7,8]: [] |  |  |

Método: Cliente(nombre, dni, minusvalia) (ME PARECE QUE NO HACE FALTA)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Clases Válidas | Valores | Clases No Válidas | Valores |
| Nombre | string con caracteres | “Manuel” | null  vacío | null  “” |
| DNI | string 8 digitos y 1 caracter | “73193719M” | null  vacío  sin formato | null  “”  “AS342”  “123456789” |
| Minusvalia | true  false | true  false |  |  |