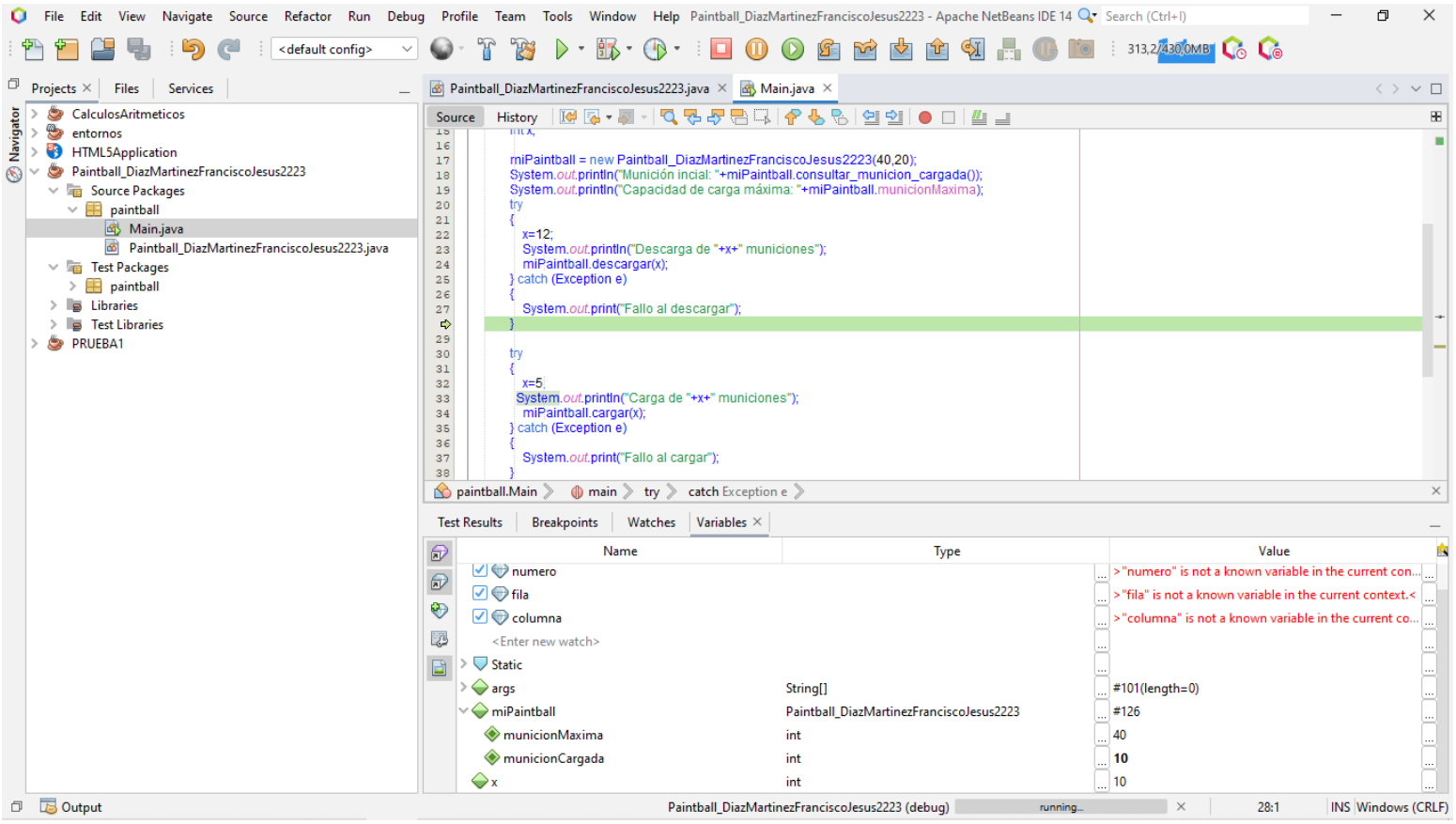
Justificación y capturas

Entornos de desarrollo, DAM

Francisco Jesús Díaz Martínez
IES AGUADULCE , ALMERÍA



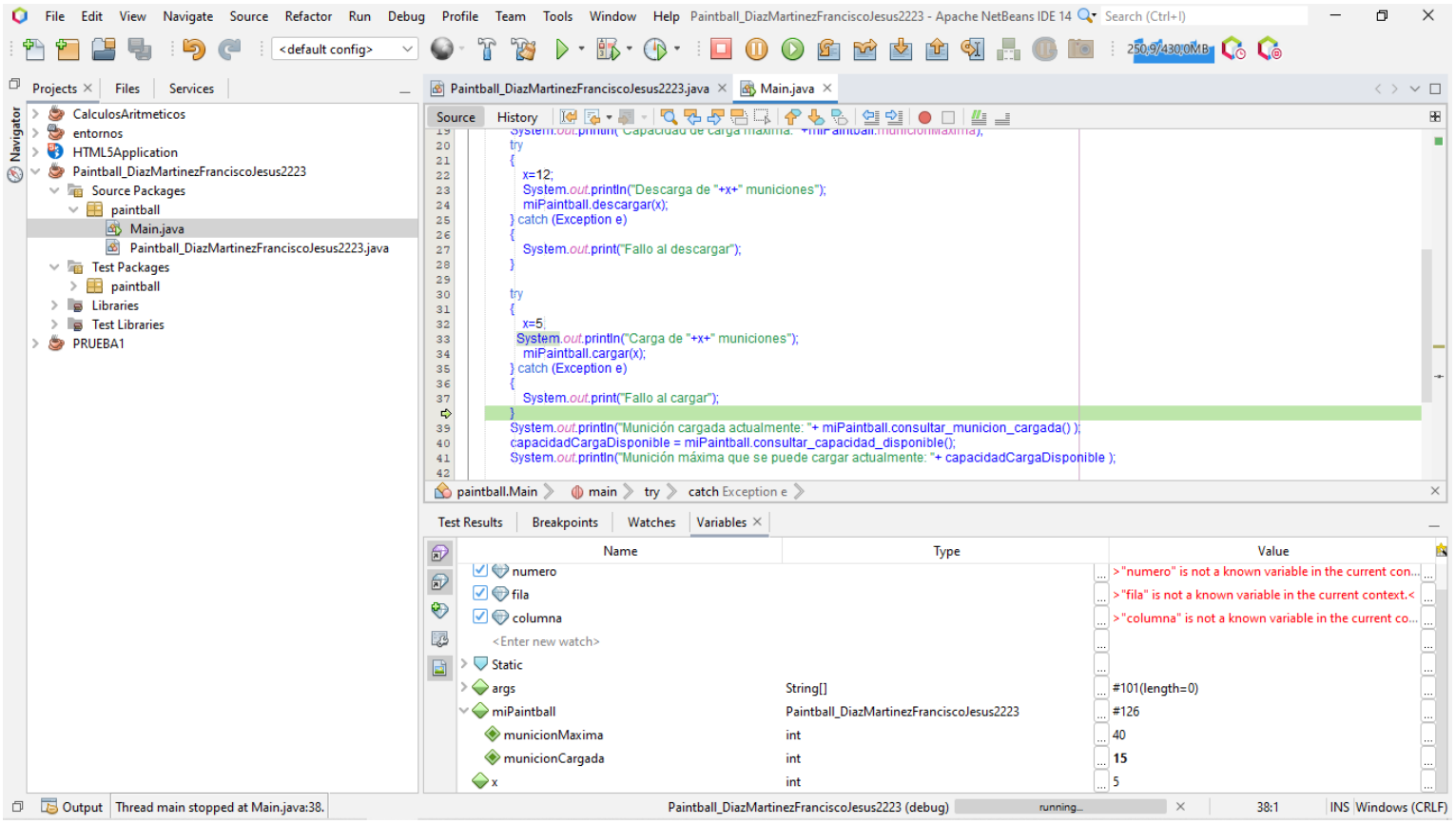
El apartado 1 consistente en cambiar de nombre el método y el proyecto está incluido en el propio archivo pero también se puede observar en la captura.

Capturas solicitadas, apartado 2.1

Breve explicación:

En este apartado, atendemos a **"miPaintball.descargar(10);"** lo que nos indica que cuando pase por el método "descargar" cambiemos el valor de la variable **X** a **10** mediante el inspector de variables**.** Para llegar al punto en el que pasa por le método, recurrimos a la ejecución paso por paso en el modo depuración.

Al indicar que **"x=10"** partiendo de (40 munMax, 20 munCarg) la variable de la **munición cargada** pasa a valer 10, porque se han retirado 10 balas.



Capturas solicitadas, apartado 2.2

Breve explicación:

En este apartado, atendemos a **"miPaintball.cargar(5);"** lo que nos indica que cuando pase por el método "cargar" cambiemos el valor de la variable **X** a **5** mediante el inspector de variables**.** Para llegar al punto en el que pasa por el método, recurrimos a la ejecución paso por paso en el modo depuración.

Al indicar que **"x=5"** partiendo de (40 munMax, 10 munCarg) la variable de la **munición cargada** pasa a valer 15, porque se han cargado 5 balas.

En este caso hay que recalcar, que el valor de la munición cargada que usamos, es el resultado de la ejecución del método anterior (miPaintball.descargar(10) ;).

3. Casos de prueba: Verificación del método “descargar” para valores límite:

Código para límite inferior y superior:

Entendemos valores límite aquellos que se encuentran inmediatamente antes y después de los valores contemplados en este supuesto; en esta ocasión es 20 que es el valor de carga máximo que viene por defecto.

Como se nos indica que lo verifiquemos, solo tomaremos los valores que sean **válidos** para el método (1 y 19) ya que el método define sus excepciones **en “x<=0;”** y “**x>munCarg;”** (no se pueden descargar 0 balas, y no se pueden descargar más balas de las que hay actualmente en el cargador.

A continuación, el código diseñado:

Límite inferior

```
                @Test
       public void testLimiteInferior() throws Exception {
        System.out.println("Test de prueba para comprobar el límite
inferior");
        int cantidad = 1;
        Paintball_DiazMartinezFranciscoJesus2223 instance = new
Paintball_DiazMartinezFranciscoJesus2223(40,20);
        try {
        instance.descargar(cantidad);
        assertTrue(instance.municionCargada==19);
        }catch (Exception e) {
            fail ("Se ha producido una excepción no esperada: " +e);
        }
```

Límite superior

```
         @Test
       public void testLimiteSuperior() throws Exception {
        System.out.println("Test de prueba para comprobar el límite
superior");
        int cantidad = 19;
        Paintball_DiazMartinezFranciscoJesus2223 instance = new
Paintball_DiazMartinezFranciscoJesus2223(40,20);
        try {
        instance.descargar(cantidad);
        assertTrue(instance.municionCargada==1);
        }catch (Exception e) {
            fail ("Se ha producido una excepción no esperada: " +e);
        }
```

También se adjuntan las pruebas de valores límite no validos en caso de ser necesarias (-1,21) no se contempla el 0 porque ya está definido en el código de serie.

Límite inferior

```
@Test
       public void testLimiteInferiorNo() throws Exception {
        System.out.println("Test de prueba para comprobar el límite
inferior, valor no válido");
        int cantidad = -1;
        Paintball_DiazMartinezFranciscoJesus2223 instance = new
Paintball_DiazMartinezFranciscoJesus2223(40,20);
        try {
        instance.descargar(cantidad);
            fail("Intento de descargar munición con valor negativo");
        }catch (Exception e) {
                        System.out.println(e);
           assertTrue(instance.municionCargada==20);
        }
     }
```

Límite superior

```
            @Test
       public void testLimiteSuperiorNo() throws Exception {
        System.out.println("Test de prueba para comprobar el límite
superior, valor no válido");
        int cantidad = 21;
        Paintball_DiazMartinezFranciscoJesus2223 instance = new
Paintball_DiazMartinezFranciscoJesus2223(40,20);
        try {
        instance.descargar(cantidad);
            fail("Intento de descargar más munición de la cargada
actualmente");
        }catch (Exception e) {
                        System.out.println(e);
           assertTrue(instance.municionCargada==20);
        }
     }
```

Todos los códigos que se han usado, al ser ejecutados pasan satisfactoriamente las instrucciones contenidas en el método.

Se han adjuntado en la plantilla test generada mediante J Unit, también los de valores no válidos en caso de que se necesiten.

4. Casos de prueba: Verificación de output correcto en el método “descargar” para valores no válidos:

Los valores que hemos contemplado como no válidos para el código original, son:

- Intentar descargar 0 balas
- Intentar descargar un número negativo de balas
- Intentar descargar un número superior de balas a las que hay actualmente en el cargador.

Test descargar 0 balas

```
@Test
     public void testDescargarMunicionCero () throws Exception {
        System.out.println("Test de prueba para descargar 0 balas");
        int cantidad = 0;
        Paintball_DiazMartinezFranciscoJesus2223 instance = new
Paintball_DiazMartinezFranciscoJesus2223(40,20);
       try {
        instance.descargar(cantidad);
        fail("Intento de descargar 0 balas");
       }
       catch (Exception e){
         System.out.println(e);
         assertTrue(instance.municionCargada==20);
       }
    }
```

Test descargar número negativo de balas

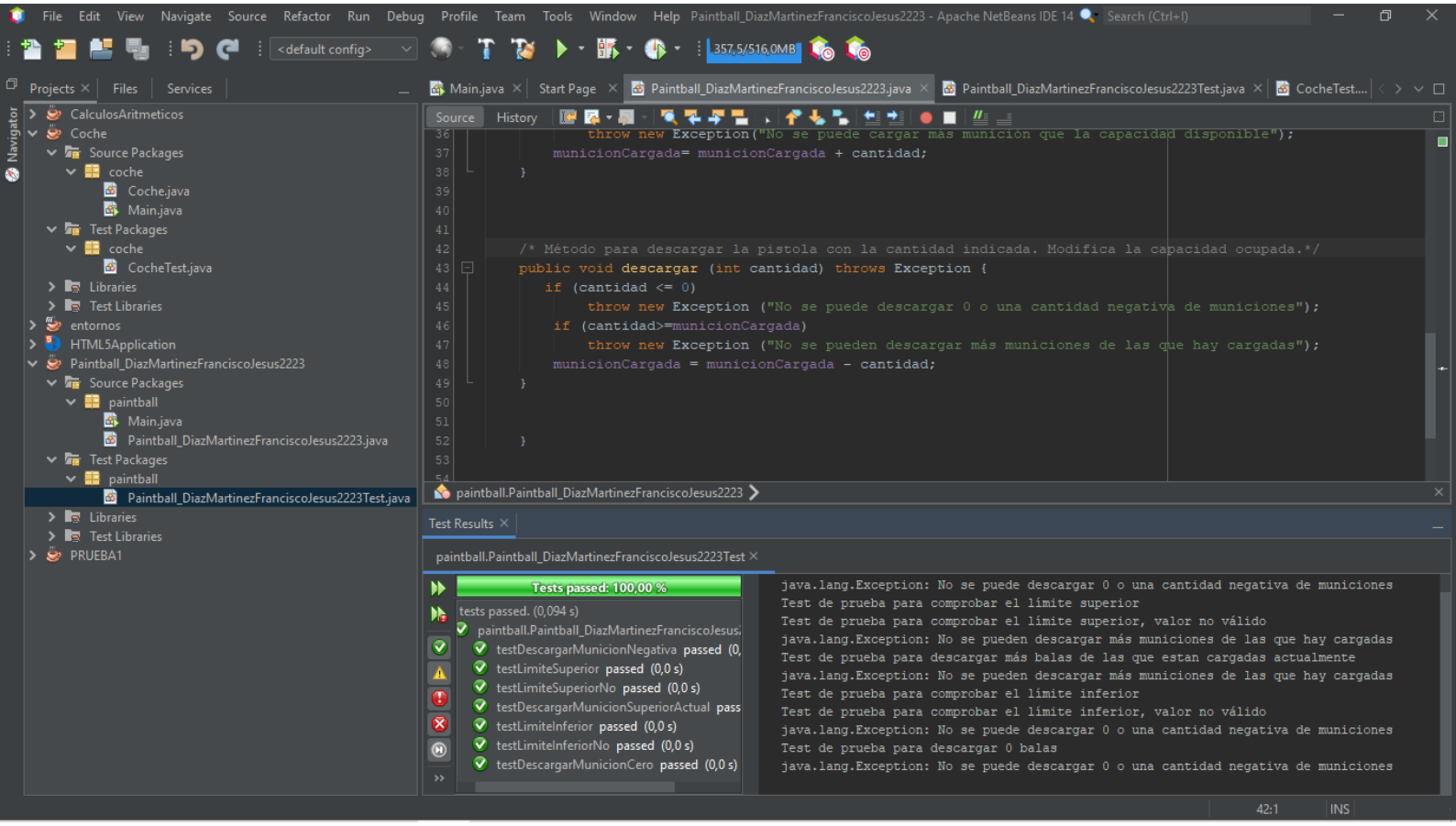
```
 @Test
     public void testDescargarMunicionNegativa() throws Exception {
        System.out.println("Test de prueba para descargar numero
negativo de balas");
        int cantidad = -4;
        Paintball_DiazMartinezFranciscoJesus2223 instance = new
Paintball_DiazMartinezFranciscoJesus2223(40,20);
       try {
        instance.descargar(cantidad);
        fail("Intento de descargar número negativo de balas");
       }
       catch (Exception e){
         System.out.println(e);
         assertTrue(instance.municionCargada==20);
       }
    }
```

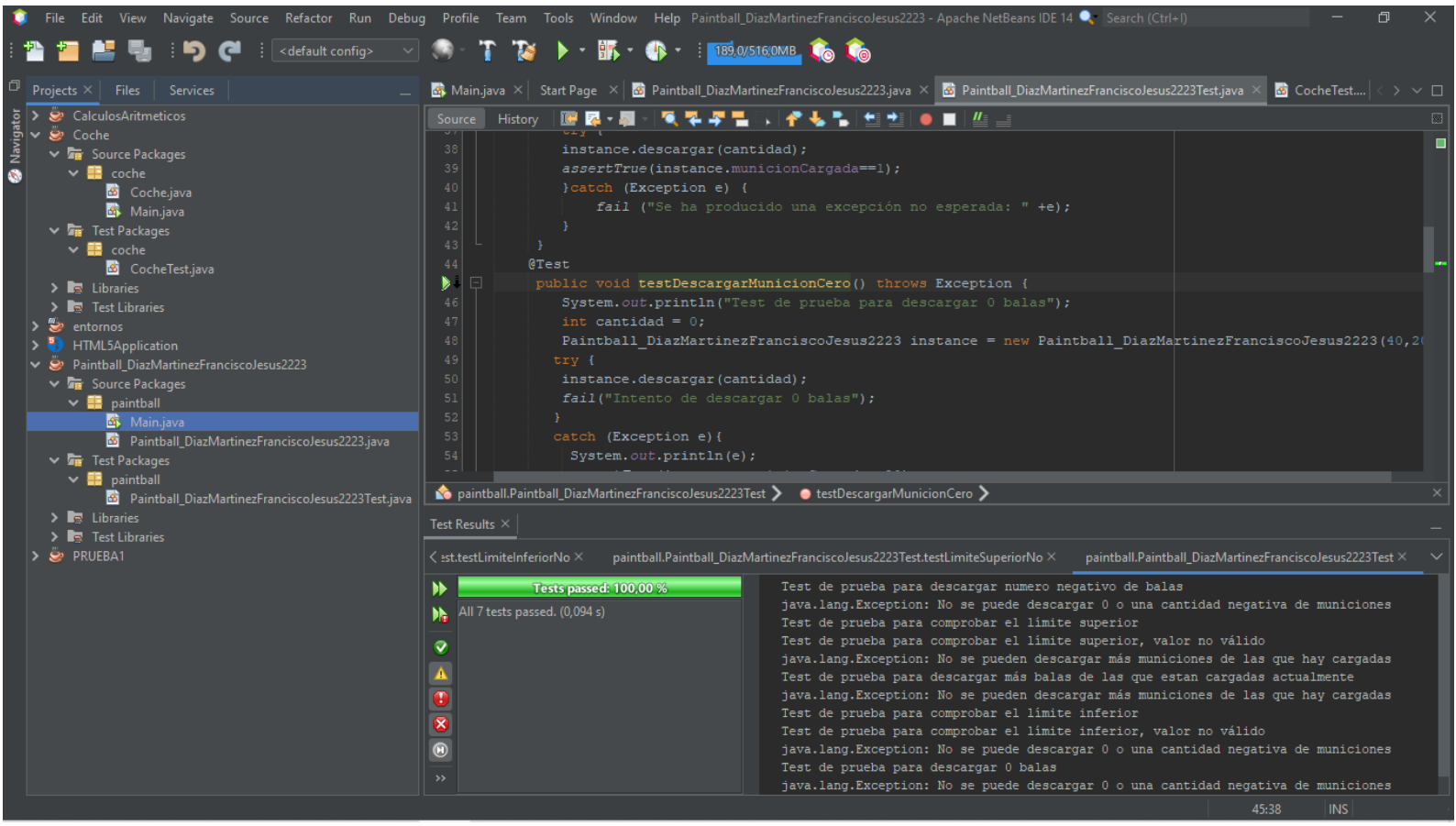
Test descargar cantidad superior de balas a las que hay en el cargador

```
 @Test
     public void testDescargarMunicionSuperiorActual () throws
Exception {
         System.out.println("Test de prueba para descargar más balas de
las que estan cargadas actualmente");
         int cantidad = 41;
         Paintball_DiazMartinezFranciscoJesus2223 instance = new
Paintball_DiazMartinezFranciscoJesus2223(40,20);
        try {
         instance.descargar(cantidad);
         fail("Intento de descargar más balas de las que hay en el
cargador");
        }
        catch (Exception e){
          System.out.println(e);
          assertTrue(instance.municionCargada==20);
        }
    }
```

5. Ejecución de test y comentarios sobre sus resultados:

Debajo se muestra la ejecución de todos los test y su resultado satisfactorio





Analisis de la salida de resultados al ejecutar el test file sobre el método "Paintball_DiazMartínezFranciscoJesus2223.java"

- **Test de prueba para descargar número negativo de balas:**

Podemos comprobar que este test es satisfactorio, pues al intentar descargar un número negativo de balas salta la excepción definida en el método original.

- **Test de prueba para comprobar el límite superior e inferior valores válidos:**

No salta ninguna excepcion, pues los valores que se introducen son completamente válidos y estan comprendidos entre los disponibles para el valor de X en cada caso.

- **Test de prueba comprobar el límite superior e inferior, valores no válidos:**

Al igual que en el primer apartado, observamos que al ejectura el test, saltan las excepciones difnidas en el método de municiones superiores a las cargadas y la imposibilidad de descargar balas de valor negativo.

- **Test de prueba para descargar 0 balas:**

Al igual que los anteriores, se puede ver como salta la excepción contenida en el método original al introducir un valor x=0.

- **Test de prueba para descargar más balas de las que hay en el cargador:**

En este último test, se puede ver como llama a la excepción contenida en el método original, al darle un valor “x>munCarg” al test.

La excepción de la que hablamos en estas justificaciones es “java.lang.Exception” que tiene como base la definición original del método “descargar” al definir los valores que acepta.

Los comentarios explicativos del código se encuentran contenidos en el archivo de test.