Descripción breve

Documento de explicación de las pruebas realizadas al software desarrollado.

PRUEBAS

Videojuego de Combate

Contenido

[2. Introducción 3](#_Toc137511323)

[3. NombreClase1 4](#_Toc137511324)

[3.1. Método nombreMétodo1 4](#_Toc137511325)

[3.1.1. Pruebas no exitosas 4](#_Toc137511326)

[3.2. Método nombreMétodo2 4](#_Toc137511327)

[3.2.1. Pruebas no exitosas 5](#_Toc137511328)

[4. NombreClase2 6](#_Toc137511329)

[4.1. Método nombreMétodo1 6](#_Toc137511330)

[4.1.1. Pruebas no exitosas 6](#_Toc137511331)

[4.2. Método nombreMétodo2 6](#_Toc137511332)

[4.2.1. Pruebas no exitosas 7](#_Toc137511333)

Tabla de ilustraciones

Ilustración 1 Diagrama de clases NombreClase1 4

Ilustración 2 Diagrama de clases NombreClase2 6

# Introducción

En este documento realizaremos una revisión de los casos de prueba realizados en la práctica de la creación de un videojuego para la empresa *MetProg Urjc S.L.*

Para el análisis, lo realizaremos clase por clase, centrándonos en los Test que hemos realizado en cada clase.

De cada Test, explicaremos en qué consiste en si mismo, así como los resultados obtenidos.

# App

Tabla

Descripción generada automáticamente

Ilustración Diagrama de la clase App

Para la clase “App” hemos diseñado diferentes test, cada uno de ellos focalizado en probar/chequear el funcionamiento de cada método.

Como todos ellos necesitan la instantación de app, lo primero que haremos será crear un objeto de la clase app, mediante el comando @Before.

## Método “loadRanking”

Comprobamos que el archivo “ranking.ser” existe y se puede acceder a él.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Entrada | Condiciones de ejecución | Respuesta Esperada | Respuesta Obtenida | Comentarios |
| 1 | - | La clase App está instanciada | No devuelve excepción (el ranking existe o se ha creado correctamente) | No devuelve excepción | - |

## Método “saveRanking”

Verificamos que se guarda correctamente la estructura ranking en el archivo.

Para ello creamos un ranking, lo guardamos en el fichero y lo leemos para verificar que coinciden.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Entrada | Condiciones de ejecución | Respuesta Esperada | Respuesta Obtenida | Comentarios |
| 1 | Ranking | La clase App está instanciada y el archivo “ranking.ser” existe | NotNull | NotNull | - |

## Método “loadEquipos”

Comprobamos que el archivo “equipos.ser” existe y se puede acceder a él.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Entrada | Condiciones de ejecución | Respuesta Esperada | Respuesta Obtenida | Comentarios |
| 1 | - | La clase App está instanciada | NotNull | NotNull | - |

## Método “saveEquipos”

Comprobamos que se puede acceder con permiso de escritura al archivo “equipos.ser”.

Para ello, cargamos en una variable los equipos almacenados en el fichero, añadimos un nuevo equipo a la lista y guardamos los cambios en el fichero.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Entrada | Condiciones de ejecución | Respuesta Esperada | Respuesta Obtenida | Comentarios |
| 1 | Equipo | La clase App está instanciada y el archivo “equipos.ser” existe | No se produce excepción | No se produce excepción | En caso de que no se guardase correctamente, saltaría una excepción |

## Método “loadGenerador”

Comprobamos que el archivo “generador.ser” existe y se puede acceder a él.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Entrada | Condiciones de ejecución | Respuesta Esperada | Respuesta Obtenida | Comentarios |
| 1 | - | La clase App está instanciada | NotNull | NotNull | - |

## Método “saveGenerador”

Comprobamos que se puede acceder con permiso de escritura al archivo “generador.ser”.

Para ello, cargamos en una variable los generadores almacenados en el fichero (lectura), añadimos nuevos generadores a la lista y guardamos los cambios en el archivo (escritura).

A continuación leemos de nuevo el fichero (actualizado con los nuevos generadores) y los guardamos en otra variable. Verificamos que los cambios se han realizado correctamente, comprobando la extensión de los generadores actualizos en la variable (sin guardarse en el archivo) y los cargados del fichero. También comprobamos que los generadores añadimos son los mismos.

Para concluir, eliminamos los generadores de prueba añadidos, restaurando el generador.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Entrada | Condiciones de ejecución | Respuesta Esperada | Respuesta Obtenida | Comentarios |
| 1 | Generador | La clase App está instanciada y el archivo “generador.ser” existe | Equals | Equals | Si el generador actualizado en memoria y el generador creado fueran distintos, su longitud no coincidiría |

## Método “loadHabilidades”

Comprobamos que el archivo “habilidades.ser” existe y se puede acceder a él.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Entrada | Condiciones de ejecución | Respuesta Esperada | Respuesta Obtenida | Comentarios |
| 1 | - | La clase App está instanciada | NotNull | NotNull | Si el archivo “habilidades.ser” no se encontrase o no se pudiese crear, devolvería Null |

## Método “saveHabilidades”

Comprobamos que se puede acceder con permiso de escritura al archivo “habilidades.ser”.

Para ello, cargamos en una variable las habilidades almacenadas en el fichero (lectura), añadimos nuevas habilidades a la lista y guardamos los cambios en el archivo (escritura).

A continuación leemos de nuevo el fichero (actualizado con las nuevas habilidades) y los guardamos en otra variable. Verificamos que los cambios se han realizado correctamente, comprobando la extensión de la lista de habilidades actualiza en la variable (sin guardarse en el archivo) y las cargadas del fichero. También comprobamos que las habilidades que añadimos son las mismas.

Para concluir, eliminamos las habilidades de prueba añadidas, restaurando la lista original.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Entrada | Condiciones de ejecución | Respuesta Esperada | Respuesta Obtenida | Comentarios |
| 1 | Habilidades | La clase App está instanciada y el archivo “habilidades.ser” existe | True | True | - |
| 2 | Talento | La clase App está instanciada y el archivo “habilidades.ser” existe | True | True | - |
| 3 | Disciplina | La clase App está instanciada y el archivo “habilidades.ser” existe | True | True | - |
| 4 | Don | La clase App está instanciada y el archivo “habilidades.ser” existe | True | True | - |

## Método “run”

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Entrada | Condiciones de ejecución | Respuesta Esperada | Respuesta Obtenida | Comentarios |
| 1 | -1 | Lorem ipsum | 5 | 5 | Lorem ipsum |
| 2 | 0 | Lorem ipsum | “hola” | “hola” | Lorem ipsum |
| 3 | 1 | Lorem ipsum | true | false | Lorem ipsum |
| 4 | null | Lorem ipsum | NullPointerException | NullPointerException | Lorem ipsum |

@Test

public void testRun() throws Exception {

System.out.println("run");

app.run();

// TODO review the generated test code and remove the default call to fail.

fail("The test case is a prototype.");

}

### Pruebas no exitosas

# Personaje

Diagrama

Descripción generada automáticamente

Ilustración Diagrama de la clase Personaje

## Método AnadirItem

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Entrada | Condiciones de ejecución | Respuesta Esperada | Respuesta Obtenida | Comentarios |
| 1 | -1 | Lorem ipsum | 5 | 5 | Lorem ipsum |
| 2 | 0 | Lorem ipsum | “hola” | “hola” | Lorem ipsum |
| 3 | 1 | Lorem ipsum | true | false | Lorem ipsum |
| 4 | null | Lorem ipsum | NullPointerException | NullPointerException | Lorem ipsum |

public void testAnadirItem() {

System.out.println("anadirItem");

Equipo item = new Arma("arma", 1, 0, 0, 2);

Personaje instance = new Licantropo("lican", new Don("don", 1,1,1));

instance.anadirItem(item);

// TODO review the generated test code and remove the default call to fail.

fail("The test case is a prototype.");

}

### Pruebas no exitosas

## Método ModificarOro

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Entrada | Condiciones de ejecución | Respuesta Esperada | Respuesta Obtenida | Comentarios |
| 1 | -1 | Lorem ipsum | 5 | 5 | Lorem ipsum |
| 2 | 0 | Lorem ipsum | “hola” | “hola” | Lorem ipsum |
| 3 | 1 | Lorem ipsum | true | false | Lorem ipsum |
| 4 | null | Lorem ipsum | NullPointerException | NullPointerException | Lorem ipsum |

public void testModificarOro() {

System.out.println("modificarOro");

int cant = 0;

Personaje instance = new Licantropo("lican", new Don("don", 1,1,1));

instance.modificarOro(cant);

// TODO review the generated test code and remove the default call to fail.

assertTrue(instance.getOro()==cant);

fail("The test case is a prototype.");

}

## Método PonerArmaActiva

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Entrada | Condiciones de ejecución | Respuesta Esperada | Respuesta Obtenida | Comentarios |
| 1 | -1 | Lorem ipsum | 5 | 5 | Lorem ipsum |
| 2 | 0 | Lorem ipsum | “hola” | “hola” | Lorem ipsum |
| 3 | 1 | Lorem ipsum | true | false | Lorem ipsum |
| 4 | null | Lorem ipsum | NullPointerException | NullPointerException | Lorem ipsum |

public void testPonerArmaActiva() {

System.out.println("ponerArmaActiva");

Arma arma = new Arma("arma",1,0,1,2);

Personaje instance = new Licantropo("lican", new Don("don", 1,1,1));

instance.ponerArmaActiva(arma);

// TODO review the generated test code and remove the default call to fail.

fail("The test case is a prototype.");

assertTrue(instance.getArmas\_activas().get(0).getNombre().equals(arma.getNombre())); //si el nombre del primer arma activa coincide con el del arma que acabamos de poner como activa

}

## Método OroSuficiente

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Entrada | Condiciones de ejecución | Respuesta Esperada | Respuesta Obtenida | Comentarios |
| 1 | -1 | Lorem ipsum | 5 | 5 | Lorem ipsum |
| 2 | 0 | Lorem ipsum | “hola” | “hola” | Lorem ipsum |
| 3 | 1 | Lorem ipsum | true | false | Lorem ipsum |
| 4 | null | Lorem ipsum | NullPointerException | NullPointerException | Lorem ipsum |

public void testOroSuficiente() {

System.out.println("oroSuficiente");

int oro = 5;

Personaje instance = new Licantropo("lican", new Don("don", 1,1,1));

boolean expResult = true;

boolean result = instance.oroSuficiente(oro); //por defecto tiene 50 de oro

assertEquals(expResult, result);

// TODO review the generated test code and remove the default call to fail.

fail("The test case is a prototype.");

}

## Método AñadirModificador

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Entrada | Condiciones de ejecución | Respuesta Esperada | Respuesta Obtenida | Comentarios |
| 1 | -1 | Lorem ipsum | 5 | 5 | Lorem ipsum |
| 2 | 0 | Lorem ipsum | “hola” | “hola” | Lorem ipsum |
| 3 | 1 | Lorem ipsum | true | false | Lorem ipsum |
| 4 | null | Lorem ipsum | NullPointerException | NullPointerException | Lorem ipsum |

public void testAñadirModificador() {

System.out.println("a\u00f1adirModificador");

int ataque =2;

Modificador mod = new Modificador("mod", ataque, true);

Personaje instance = new Licantropo("lican", new Don("don", 1,1,1));

instance.añadirModificador(mod);

assertTrue(instance.calcularModificadoresAtaque()== ataque);

// TODO review the generated test code and remove the default call to fail.

fail("The test case is a prototype.");

}

## Método AñadirEsbirro

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Entrada | Condiciones de ejecución | Respuesta Esperada | Respuesta Obtenida | Comentarios |
| 1 | -1 | Lorem ipsum | 5 | 5 | Lorem ipsum |
| 2 | 0 | Lorem ipsum | “hola” | “hola” | Lorem ipsum |
| 3 | 1 | Lorem ipsum | true | false | Lorem ipsum |
| 4 | null | Lorem ipsum | NullPointerException | NullPointerException | Lorem ipsum |

public void testAñadirEsbirro() {

System.out.println("a\u00f1adirEsbirro");

Esbirro es = new Humano("humano", 1, Lealtad.ALTA);

Personaje instance = new Licantropo("lican", new Don("don", 1,1,1));

instance.añadirEsbirro(es);

assertTrue(instance.getEsbirros().size() >0);

// TODO review the generated test code and remove the default call to fail.

fail("The test case is a prototype.");

}

## Método CalcularSaludEsbirros

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Entrada | Condiciones de ejecución | Respuesta Esperada | Respuesta Obtenida | Comentarios |
| 1 | -1 | Lorem ipsum | 5 | 5 | Lorem ipsum |
| 2 | 0 | Lorem ipsum | “hola” | “hola” | Lorem ipsum |
| 3 | 1 | Lorem ipsum | true | false | Lorem ipsum |
| 4 | null | Lorem ipsum | NullPointerException | NullPointerException | Lorem ipsum |

public void testCalcularSaludEsbirros() {

System.out.println("calcularSaludEsbirros");

int salud=1;

Esbirro es = new Humano("humano", salud, Lealtad.ALTA);

Personaje instance = new Licantropo("lican", new Don("don", 1,1,1));

instance.añadirEsbirro(es);

instance.calcularSaludEsbirros();

assertTrue(instance.getSalud\_esbirros()==salud);

// TODO review the generated test code and remove the default call to fail.

fail("The test case is a prototype.");

}

## Método CalcularAtaqueEquipo

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Entrada | Condiciones de ejecución | Respuesta Esperada | Respuesta Obtenida | Comentarios |
| 1 | -1 | Lorem ipsum | 5 | 5 | Lorem ipsum |
| 2 | 0 | Lorem ipsum | “hola” | “hola” | Lorem ipsum |
| 3 | 1 | Lorem ipsum | true | false | Lorem ipsum |
| 4 | null | Lorem ipsum | NullPointerException | NullPointerException | Lorem ipsum |

public void testCalcularAtaqueEquipo() {

System.out.println("calcularAtaqueEquipo");

Personaje instance = new Licantropo("lican", new Don("don", 1,1,1));

int ataque=2;

Arma arma = new Arma("arma", ataque, 0, 1, 2);

instance.ponerArmaActiva(arma);

int expResult = ataque;

int result = instance.calcularAtaqueEquipo();

assertEquals(expResult, result);

// TODO review the generated test code and remove the default call to fail.

fail("The test case is a prototype.");

}

## Método CalcularModificadoresAtaque

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Entrada | Condiciones de ejecución | Respuesta Esperada | Respuesta Obtenida | Comentarios |
| 1 | -1 | Lorem ipsum | 5 | 5 | Lorem ipsum |
| 2 | 0 | Lorem ipsum | “hola” | “hola” | Lorem ipsum |
| 3 | 1 | Lorem ipsum | true | false | Lorem ipsum |
| 4 | null | Lorem ipsum | NullPointerException | NullPointerException | Lorem ipsum |

public void testCalcularModificadoresAtaque() {

System.out.println("calcularModificadoresAtaque");

Personaje instance = new Licantropo("lican", new Don("don", 1,1,1));

int ataque=2;

instance.añadirModificador(new Modificador("mod", ataque, true));

int expResult = ataque;

int result = instance.calcularModificadoresAtaque();

assertEquals(expResult, result);

// TODO review the generated test code and remove the default call to fail.

fail("The test case is a prototype.");

}

## Método CalcularModificadoresDefensa

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Entrada | Condiciones de ejecución | Respuesta Esperada | Respuesta Obtenida | Comentarios |
| 1 | -1 | Lorem ipsum | 5 | 5 | Lorem ipsum |
| 2 | 0 | Lorem ipsum | “hola” | “hola” | Lorem ipsum |
| 3 | 1 | Lorem ipsum | true | false | Lorem ipsum |
| 4 | null | Lorem ipsum | NullPointerException | NullPointerException | Lorem ipsum |

public void testCalcularModificadoresDefensa() {

System.out.println("calcularModificadoresDefensa");

Personaje instance = new Licantropo("lican", new Don("don", 1,1,1));

int defensa=2;

instance.añadirModificador(new Modificador("mod", defensa, false));

int expResult = defensa;

int result = instance.calcularModificadoresDefensa();

assertEquals(expResult, result);

// TODO review the generated test code and remove the default call to fail.

fail("The test case is a prototype.");

}

### Pruebas no exitosas

# NombreClase2

Una señal de alto

Descripción generada automáticamente con confianza media

Ilustración Diagrama de clases NombreClase2

## Método nombreMétodo1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Entrada | Condiciones de ejecución | Respuesta Esperada | Respuesta Obtenida | Comentarios |
| 1 | -1 | Lorem ipsum | 5 | 5 | Lorem ipsum |
| 2 | 0 | Lorem ipsum | “hola” | “hola” | Lorem ipsum |
| 3 | 1 | Lorem ipsum | true | false | Lorem ipsum |
| 4 | null | Lorem ipsum | NullPointerException | NullPointerException | Lorem ipsum |

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed tempus fringilla laoreet. In hac habitasse platea dictumst. Praesent non nibh non tortor eleifend vulputate. Phasellus porttitor eget erat non auctor. Vestibulum mauris diam, feugiat eu nisl eget, iaculis lacinia ex. Praesent vitae tristique nunc. Sed mi ante, consequat vitae libero vel, consectetur consectetur neque. Maecenas sodales et est ut ultrices. Quisque eget nulla aliquam, vulputate purus luctus, venenatis arcu. Duis et lorem vel nulla lacinia volutpat. Donec viverra leo nec egestas rhoncus. Duis consequat eu ante sit amet euismod. Donec in erat vitae nibh consectetur auctor. Fusce vel libero laoreet, ornare ante in, tincidunt diam. Duis id semper elit.

### Pruebas no exitosas

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed tempus fringilla laoreet. In hac habitasse platea dictumst. Praesent non nibh non tortor eleifend vulputate. Phasellus porttitor eget erat non auctor. Vestibulum mauris diam, feugiat eu nisl eget, iaculis lacinia ex. Praesent vitae tristique nunc. Sed mi ante, consequat vitae libero vel, consectetur consectetur neque. Maecenas sodales et est ut ultrices. Quisque eget nulla aliquam, vulputate purus luctus, venenatis arcu. Duis et lorem vel nulla lacinia volutpat. Donec viverra leo nec egestas rhoncus. Duis consequat eu ante sit amet euismod. Donec in erat vitae nibh consectetur auctor. Fusce vel libero laoreet, ornare ante in, tincidunt diam. Duis id semper elit.

## Método nombreMétodo2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Entrada | Condiciones de ejecución | Respuesta Esperada | Respuesta Obtenida | Comentarios |
| 1 | -1 | Lorem ipsum | 5 | 5 | Lorem ipsum |
| 2 | 0 | Lorem ipsum | “hola” | “hola” | Lorem ipsum |
| 3 | 1 | Lorem ipsum | true | false | Lorem ipsum |
| 4 | null | Lorem ipsum | NullPointerException | NullPointerException | Lorem ipsum |

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed tempus fringilla laoreet. In hac habitasse platea dictumst. Praesent non nibh non tortor eleifend vulputate. Phasellus porttitor eget erat non auctor. Vestibulum mauris diam, feugiat eu nisl eget, iaculis lacinia ex. Praesent vitae tristique nunc. Sed mi ante, consequat vitae libero vel, consectetur consectetur neque. Maecenas sodales et est ut ultrices. Quisque eget nulla aliquam, vulputate purus luctus, venenatis arcu. Duis et lorem vel nulla lacinia volutpat. Donec viverra leo nec egestas rhoncus. Duis consequat eu ante sit amet euismod. Donec in erat vitae nibh consectetur auctor. Fusce vel libero laoreet, ornare ante in, tincidunt diam. Duis id semper elit.

### Pruebas no exitosas

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed tempus fringilla laoreet. In hac habitasse platea dictumst. Praesent non nibh non tortor eleifend vulputate. Phasellus porttitor eget erat non auctor. Vestibulum mauris diam, feugiat eu nisl eget, iaculis lacinia ex. Praesent vitae tristique nunc. Sed mi ante, consequat vitae libero vel, consectetur consectetur neque. Maecenas sodales et est ut ultrices. Quisque eget nulla aliquam, vulputate purus luctus, venenatis arcu. Duis et lorem vel nulla lacinia volutpat. Donec viverra leo nec egestas rhoncus. Duis consequat eu ante sit amet euismod. Donec in erat vitae nibh consectetur auctor. Fusce vel libero laoreet, ornare ante in, tincidunt diam. Duis id semper elit.