

# **PRÁCTICA DE LA ASIGNATURA** **DESARROLLO DE PROGRAMAS**

***Curso 2017/18***

Grado en Ingeniería en Informática en:

Ingeniería del Software

Ingeniería de Computadores

(Idea original de [Roberto Rodríguez Echeverría](#))



## **Entrega 1 (Viernes, 27 de octubre)**

En este documento se especifican las acciones que deben llevarse a cabo para implementar la primera entrega del proyecto. El objetivo es construir un prototipo con algunas funcionalidades del sistema, que incluya principalmente las siguientes funcionalidades:

1. Creación del mapa o escenario del juego según unas dimensiones dadas. En una entrega posterior se definirá un mecanismo para la generación del mapa siguiendo un algoritmo concreto.
2. Reparto de armas por el mapa. Al inicio de la simulación, se dejará una serie de armas por ciertas casillas del mapa. En esta entrega, estas casillas serán siempre las mismas, mientras que en entregas posteriores serán calculadas.
3. Creación del hombre puerta y configuración de sus armas.
4. Primeras pruebas con algunos personajes (acciones de recoger armas e interactuar con el hombre puerta).

A continuación se detalla el contenido de cada una de las funcionalidades que debe incluir el sistema.

### **1.- El mapa o tablero**

El mapa o tablero será la estructura más general del sistema y contendrá un número de salas que vendrá dado por las dimensiones del mismo: ancho x alto. De este modo, un mapa de 6x6 (como aparece en la figura 1) tendrá 36 salas numeradas de 0 a 35. Las dimensiones alto y ancho

pueden ser diferentes, de modo que el mapa podrá ser rectangular (no sólo cuadrado).

0	1	2	3	4	5
6	7	8	9	10	11
12	13	14	15	16	17
18	19	20	21	22	23
24	25	26	27	28	29
30	31	32	33	34	35

*Figura 1: Ejemplo de la distribución de las salas del mapa en esta entrega.*

## 2.- Las salas

En esta segunda entrega, el mapa estará completamente abierto, de modo que cada sala tendrá conexión con cualquiera de las 4 posibles salas vecinas (no existen paredes, excepto las que limitan el mapa con el exterior). En cada sala podrá existir un conjunto de armas que los diferentes personajes podrán ir recogiendo según pasen por la misma. La forma de recoger las armas será diferente para los diferentes tipos de personajes y será especificada en más detalle posteriormente. Las salas mantendrán las llaves **ordenadas según su poder (de más alto a más bajo)**.

Dado que los personajes del juego se irán moviendo a través de las salas, éstas deberán disponer de algún espacio de almacenamiento en el que hospedarlos. En este sentido, se debe tener en cuenta que en cada sala podrán coexistir varios personajes de igual o diferente tipo al mismo tiempo. Los personajes se almacenarán en las salas en **orden histórico**, de modo que aquellos que llegaron primero, serán los primeros en salir de la misma.

## 3.- Las armas

Como se ha comentado en el apartado anterior, algunas salas del mapa contendrán armas que podrán ser recogidas por los diferentes personajes. El reparto de las armas por esta salas será llevado a cabo al inicio de la simulación. En concreto, el conjunto de armas que será repartido a través de las diferentes salas y su poder asociado es el siguiente (representado según (nombre de arma, poder)):

[{Mjolnir,29}, {Anillo,1}, {Garra,27}, {Armadura,3}, {Red,25}, {Escudo,5}, {Lucille,23}, {Lawgiver,7}, {GuanteInfinito,21}, {LazoVerdad,9}, {CadenaFuego,19}, {Capa,11}, {Flecha,17}, {Tridente,13}, {Antorcha,15}, {Baston,28}, {Latigo,2}, {MazaOro,26}, {CampoMagnetico,4}, {Tentaculo,24}, {CampoEnergia,6}, {Cetro,22}, {RayoEnergia,8}, {Laser,20}, {Bola,10}, {Espada,18}, {Sable,12}, {Acido,16}, {Gema,14}, {Nullifier,23}, {Mjolnir,1}, {Anillo,29}, {Garra,3}, {Armadura,27}, {Red,5}, {Escudo,25}, {Lucille,7}, {Lawgiver,23}, {GuanteInfinito,9}, {LazoVerdad,21}, {CadenaFuego,11}, {Capa,19}, {Flecha,13}, {Tridente,17}, {Antorcha,28}, {Baston,15}, {Latigo,26}, {MazaOro,2}, {CampoMagnetico,24}, {Tentaculo,4}, {CampoEnergia,22}, {Cetro,6}, {RayoEnergia,20}, {Laser,8}, {Bola,18}, {Espada,10}, {Sable,16}, {Acido,12}, {Gema,1}, {Nullifier,3}]

Las armas generadas serán repartidas de 5 en 5 (según el orden en el que aparecen en el listado anterior) por diferentes salas del mapa. Por tanto, dado que el número total de armas generadas es 60, éstas serán repartidas en 12 salas. En esta primera entrega, las salas en las que se

depositarán armas serán las siguientes: 1, 2, 8, 14, 15, 21, 27, 35, 28, 29, 33, 34 (especificadas también en el código del anexo de este documento). En esta entrega vamos a suponer que **el mapa será de dimensiones fijas de 6x6, por lo que se crearán 35 salas**, por lo que siempre se repartirán las 60 armas. En entregas posteriores, estas salas serán calculadas mediante un algoritmo para conocer las salas con mayor frecuencia de paso (por las que pasen más caminos del mapa).

Cada vez que un personaje pase por una sala en la que hay un arma, éste recogerá el arma con mayor poder. Sin embargo, la forma de recogerla será diferente en función del tipo de personaje:

- a) **Superhéroes**. Si no tenía previamente este arma, será almacenada en su contenedor de armas (**que debe permitir búsquedas de la forma más eficiente posible**), mientras que si ya la tenía, se le sumará al poder del arma que tenía el poder del arma que ha recogido.
- b) **Villanos**. También recogerán de la sala el arma con mayor poder, pero dado que éstos sólo podrán llevar consigo **un arma** (que será configurada para cada personaje al principio de la simulación), sólo se quedarán con el arma de mayor poder entre las dos (la recogida y la que llevaba previamente), dejando en la sala la de menor poder.

## 4.- El hombre puerta

Este superhéroe posee la habilidad de abrir "portales" para cruzar objetos sólidos, colocándose cerca de ellos. Esto permite que los demás personajes puedan atravesar su cuerpo pasando por dicho portal para poder obtener el *Teseracto*.

El portal del hombre puerta podrá presentar dos estados diferentes: "cerrado" y "abierto". Este personaje dispondrá de un contenedor de armas (que permita también búsquedas eficientes) que debe ser configurado inicialmente (al principio de la simulación) antes de poder usarlas. Una vez configurado, el portal pasará a estar en el estado de cerrado. Sólo es posible abrir el portal cuando los demás superhéroes utilicen una secuencia de armas adecuada que sea capaz de cambiar el estado del portal a abierto. El cambio de estado dependerá de que se cumpla o no una condición de apertura (que se explicará a continuación). La apertura del portal implicará que el personaje que lo haya conseguido podrá acceder al *Teseracto* y dominará la ciudad.

Cuando cualquier personaje llegue a la sala del hombre puerta, interactuará con él para intentar abrir el portal. Esta interacción, dependerá del tipo de personaje que interactúe con el hombre puerta:

- a) **Superhéroes**. Los Superhéroes interactuarán con el hombre puerta de la siguiente forma. El superhéroe buscará en su contenedor de armas el arma con mayor peso. A continuación, el hombre puerta buscará el mismo arma en su contenedor. Si el hombre puerta encuentra este arma y el arma del superhéroe tiene un peso mayor que la del hombre puerta, el hombre puerta eliminará de su contenedor el arma. En caso contrario, no se eliminará. Si el hombre puerta no tiene el arma buscada, no se realizará ninguna interacción. En todos los casos, el superhéroe eliminará de su contenedor de armas el arma utilizada.
- b) **Villanos**. Los Villanos interactuarán con el hombre puerta de la siguiente forma. Se comparará el poder del arma del Villano con el del arma con mayor poder del hombre puerta. En caso de tener más poder el arma del Villano, el hombre puerta eliminará su arma de su contenedor de armas. Si el hombre puerta no tiene el arma buscada, no se

realizará ninguna acción. Dado que el Villano sólo dispone de un arma, ésta no será eliminada en ningún caso.

Una vez realizada la interacción entre el personaje (del tipo que sea) y el hombre puerta, se llevará a cabo una operación para comprobar si la condición de apertura del portal se cumple. En concreto, el portal sólo se abrirá cuando se cumplan las siguientes dos condiciones:

- La profundidad del contenedor de armas del hombre puerta (**estructura binaria de búsquedas**) debe ser menor que un valor predeterminado (constante).

El contenedor de armas del hombre puerta será cargado inicialmente con las siguientes armas:

[(CampoEnergia, 5), (Armadura,13), (Anillo,11), (Acido,1), (Antorcha,5), (Bola,3), (Baston,22), (CadenaFuego,11), (Espada,11), (Cetro,20), (Capa,10), (CampoMagnetico,5), (Escudo,3), (Garra,22), (Flecha,12), (Gema,4)]

## 5.- Objetivo de esta entrega.

En esta primera entrega, las pruebas que se realizarán consistirán en situar a varios personajes en la sala en la que se encuentra el Daily Planet y ejecutar durante 5 turnos las acciones de recogida de arma de la sala e interacción con el hombre puerta (en ese orden).

La simulación terminará o bien por abrirse el portal a través del hombre puerta o por terminarse los 5 turnos. En caso de abrir el portal del hombre puerta, se debe terminar el turno

## Consideraciones

- Para **comprobar** el correcto funcionamiento de esta primera entrega del proyecto, en el anexo de este documento se entrega un método que inicializa una configuración de armas en el *hombre puerta*, así como la creación de varios superhéroes y villanos (su uso es orientativo, no obligatorio).
- Cada grupo de alumnos **no debe limitarse** a probar el proyecto únicamente con esta configuración inicial (la generada por el código del anexo) sino que deberá probar la corrección de su proyecto utilizando diferentes configuraciones que permitan detectar cualquier posible error en el mismo. En este sentido, los grupos podrán discutir en el **aula virtual** los **resultados** obtenidos para así poder compararlos con los de otros compañeros.
- La **salida por pantalla** en esta primera entrega no tiene un formato predefinido. Cada grupo puede hacerla como quiera. Sin embargo, en las siguientes entregas es recomendable que el proyecto muestre al final de la ejecución los datos en los formatos que se vayan especificando. Un ejemplo de salida para esta EC1 podría ser la siguiente:

```
<para el mapa>
(mapa:<id sala salida>)
(hombrepuerta:<estado portal>:<altura del contenedor de armas>:<armas de hombre
puerta>)
```

- El programa entregado deberá ejecutar de principio a fin **sin pedir ningún dato por teclado**.

- Deben implementarse las clases **necesarias para poder tener un mapa inicial, armas en las salas especificadas en este enunciado y personajes situados en la sala Daily Planet que interactúen con el hombre puerta.**
- El proyecto debe incluir la implementación de la estructura de datos **Árbol Binario de Búsqueda (ABB) y Estructuras de Datos Lineales** (las necesarias para la implementación de esta entrega). Además, las estructuras de datos utilizadas deben ser genéricas, de modo que puedan ser utilizadas con diferentes tipos de objetos sin necesidad de cambiar su implementación.
- Generación de la **documentación externa** del sistema en ficheros HTML siguiendo el formato **JavaDoc** (en el proyecto entregado no será necesario incluir los ficheros HTML generados), sólo haber comentado el código siguiendo el formato **JavaDoc**. Será estudiado en la siguiente sesión de laboratorio.
- Entrega de la primera versión del diagrama de clases del sistema (diagrama conceptual o diagrama de clases).

## **PRIMERA ENTREGA DE LA EVALUACIÓN CONTINUA (EC1)**

Cada grupo de alumnos deberá realizar la **entrega de esta primera fase del proyecto en la fecha indicada en la planificación de la asignatura y en este documento (ver abajo)**. El código de la entrega contendrá **exclusivamente el código necesario para satisfacer las especificaciones** de la misma y nada más. Para realizar esta entrega se habilitará una **tarea en el aula virtual**.

### **Requisitos sobre el código de la EC1:**

- El código debe estar perfectamente documentado siguiendo el formato JavaDoc.
- El código debe estar escrito siguiendo un único estilo de programación.
- En la documentación interna del código debe indicarse:

**Nombre del grupo:**

**Nombre y Apellidos de cada alumno:**

**Curso:**

Para realizar la entrega, la carpeta creada para el proyecto será comprimida en un fichero en **formato ZIP**. El nombre del fichero comprimido será “ec1\_nombregrupo.zip” o, si se hace de manera individual, “ec1\_idcorreoalumnosuex.zip”. El fichero debe ser subido **por los dos miembros del grupo** (para que quede constancia en el registro de ambos usuarios en el aula virtual).

La entrega de esta fase será: **Viernes, 27 de octubre.**

*ANEXO. Ejemplo de código main que inicializa los diferentes objetos y pone en marcha la simulación del sistema (este código es meramente orientativo).*

```
/**
 * EC1 - proyecto17_18
 * Implementacion de la clase Mapa que representa la EC1 dentro del proyecto
 * @version 3.0
 * @author
 * <b> Profesores DP </b><br>
 * Asignatura Desarrollo de Programas<br>
 * Curso 17/18
 */
public class Mapa {
    /**
     * Programa principal - EC1.
     * @param args Argumentos que recibe el programa principal
     * @return Retorna la salida del programa
     */
    public static void main (String args[]) {

        // Creación del mapa
        // Parámetros: sala del Daily Planet, nº columnas, nº filas y profundidad de apertura del portal
        int dimX = 6;
        int dimY = 6;
        int salaDailyPlanet = (dimX * dimY) - 1;
        int alturaApertura = 4;
        Mapa mapa = new Mapa(salaDailyPlanet, dimX, dimY, alturaApertura);

        // Creación de las armas para el hombre puerta
        int numArmasPuerta = 15;
        int [] armasPuerta = {new Arma("CampoEnergia", 5), new Arma("Armadura",13), new
            Arma("Anillo",11), new Arma("Acido",1), new Arma("Antorcha",5), new
            Arma("Bola",3), new Arma("Baston",22), new Arma("CadenaFuego",11), new
            Arma("Espada", 11), new Arma("Cetro",20), new Arma("Capa",10), new
            Arma("CampoMagnetico",5), new Arma("Escudo",3), new Arma("Garra",22), new
            Arma("Flecha",12), new Arma("Gema",4)};

        // Creación del hombre puerta y configuración
        HombrePuerta doorMan = new HombrePuerta();
        // Configurar el hombre puerta introduciendo la combinación de armas
        doorMan.configurar(armasPuerta);
        // Cerrar el portal, por si inicialmente está abierto
        doorMan.cerrar();
        // Añadir el hombre puerta al mapa
        mapa.insertarHombrePuerta(doorMan);
    }
}
```

```

// Creación de las armas para repartir en salas
int numArmasSalas = 60;
int [] armasSalas = {new Arma("Mjolnir",29), new Arma("Anillo",1), new Arma("Garra",27),
    new Arma("Armadura",3), new Arma("Red",25), new Arma("Escudo",5),
    new Arma("Lucille",23), new Arma("Lawgiver",7), new Arma("GuanteInfinito",21),
    new Arma("LazoVerdad",9), new Arma("CadenaFuego",19), new Arma("Capa",11),
    new Arma("Flecha",17), new Arma("Tridente",13), new Arma("Antorcha",15),
    new Arma("Baston",28), new Arma("Latigo",2), new Arma("MazaOro",26),
    new Arma("CampoMagnetico",4), new Arma("Tentaculo",24),
    new Arma("CampoEnergia",6), new Arma("Cetro",22), new Arma("RayoEnergia",8),
    new Arma("Laser",20), new Arma("Bola",10), new Arma("Espada",18),
    new Arma("Sable",12), new Arma("Acido",16), new Arma("Gema",14),
    new Arma("Nullifier",23), new Arma("Mjolnir",1), new Arma("Anillo",29),
    new Arma("Garra",3), new Arma("Armadura",27), new Arma("Red",5),
    new Arma("Escudo",25), new Arma("Lucille",7), new Arma("Lawgiver",23),
    new Arma("GuanteInfinito",9), new Arma("LazoVerdad",21),
    new Arma("CadenaFuego",11), new Arma("Capa",19), new Arma("Flecha",13),
    new Arma("Tridente",17), new Arma("Antorcha",28), new Arma("Baston",15),
    new Arma("Latigo",26), new Arma("MazaOro",2), new Arma("CampoMagnetico",24),
    new Arma("Tentaculo",4), new Arma("CampoEnergia",22), new Arma("Cetro",6),
    new Arma("RayoEnergia",20), new Arma("Laser",8), new Arma("Bola",18),
    new Arma("Espada",10), new Arma("Sable",16), new Arma("Acido",12),
    new Arma("Gema",1), new Arma("Nullifier",3)};

int [] idSalasConArmas = {1, 2, 8, 14, 15, 21, 27, 35, 28, 29, 33, 34};
mapa.distribuirArmas(idSalasConArmas, armasSalas);
// La distribución de armas quedará de la siguiente forma:
// (sala:1: {Mjolnir,29}, {Anillo,1}, {Garra,27}, {Armadura,3}, {Red,25},)
// (sala:2: {Escudo,5}, {Lucille,23}, {Lawgiver,7}, {GuanteInfinito,21}, {LazoVerdad,9},)
// (sala:8: {CadenaFuego,19}, {Capa,11}, {Flecha,17}, {Tridente,13}, {Antorcha,15},)
// (sala:14: {Baston,28}, {Latigo,2}, {MazaOro,26}, {CampoMagnetico,4}, {Tentaculo,24},)
// (sala:15: {CampoEnergia,6}, {Cetro,22}, {RayoEnergia,8}, {Laser,20}, {Bola,10},)
// (sala:21: {Espada,18}, {Sable,12}, {Acido,16}, {Gema,14}, {Nullifier,23},)
// (sala:27: {Mjolnir,1}, {Anillo,29}, {Garra,3}, {Armadura,27}, {Red,5},)
// (sala:35: {Escudo,25}, {Lucille,7}, {Lawgiver,23}, {GuanteInfinito,9}, {LazoVerdad,21},)
// (sala:28: {CadenaFuego,11}, {Capa,19}, {Flecha,13}, {Tridente,17}, {Antorcha,28},)
// (sala:29: {Baston,15}, {Latigo,26}, {MazaOro,2}, {CampoMagnetico,24}, {Tentaculo,4},)
// (sala:33: {CampoEnergia,22}, {Cetro,6}, {RayoEnergia,20}, {Laser,8}, {Bola,18},)
// (sala:34: {Espada,10}, {Sable,16}, {Acido,12}, {Gema,1}, {Nullifier,3})

// Creación de varios personajes
SuperHeroe thor = new SuperHeroe("Thor", 'T');
mapa.insertarPersonaje(thor,salaDailyPlanet);

SuperHeroe ironMan = new SuperHeroe("IronMan", 'I');
mapa.insertarPersonaje(ironMan,salaDailyPlanet);

SuperHeroe storm = new SuperHeroe("Storm", 'S');
mapa.insertarPersonaje(storm,salaDailyPlanet);

SuperHeroe captainAmerica = new SuperHeroe("Capitan América", 'C');
mapa.insertarPersonaje(captainAmerica,salaDailyPlanet);

```

```
Villano deadPool = new Villano("Dead Pool", 'D', new Arma ("Sable",17));
mapa.insertarPersonaje(deadPool,salaDailyPlanet);

Villano kurtConnors = new Villano("Kurt Connors", 'K', new Arma ("CampoEnergia",15));
mapa.insertarPersonaje(kurtConnors,salaDailyPlanet);

Villano nebula = new Villano("Nebula", 'N', new Arma ("RayoEnergia",15));
mapa.insertarPersonaje(nebula,salaDailyPlanet);

mapa.simulacion();
mapa.pintar(); //se mostrará en este caso únicamente la información del mapa
... // Realizar más pruebas
}
```