

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

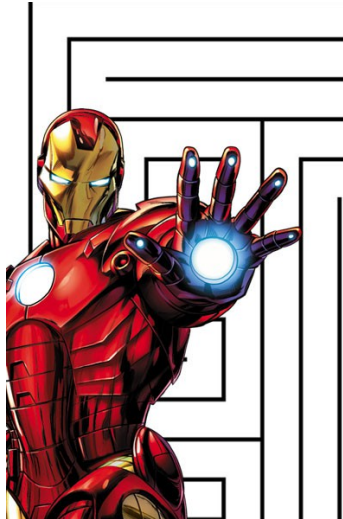
**Curso 2017/18**

Grado en Ingeniería en Informática en:

**Ingeniería del Software**

**Ingeniería de Computadores**

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



## **Entrega 3 (Miércoles, 20 de diciembre)**

En esta fase del proyecto el objetivo es construir el sistema final añadiendo al prototipo el cálculo de las rutas de los personajes y la distribución de armas en las salas más frecuentadas en los caminos entre la sala 0 y la sala Daily Planet. Opcionalmente, el proyecto puede incluir una Interfaz Gráfica de Usuario. Las funcionalidades que deben implementarse son:

1. Reparto de las armas en las salas según su frecuencia de aparición en los diferentes caminos.
2. Generación de las rutas de los personajes.
3. Registro de los resultados obtenidos a un fichero de texto con el formato especificado (no implementado en EC2).
4. Completar los Juegos de pruebas unitarias.
5. Documentación externa del proyecto.
6. Implementación de Interfaz Gráfica de Usuario (**opcional**)

### **1.- Reparto de armas en las salas más frecuentadas en el mapa**

Las armas que deben repartirse por el mapa se generarán de la misma manera que en la etapa anterior: se generarán 60 armas.

Sin embargo, a diferencia de la segunda etapa, en esta entrega las armas se distribuirán a través de las salas según su frecuencia de aparición en las rutas existentes entre la sala 0 y la sala Daily Planet. Para calcular estas salas, el alumno debe calcular todos los caminos existentes desde la sala 0 hasta la sala Daily Planet y todas las salas deben ser ordenadas (descendentemente) según su frecuencia de aparición en estos caminos. Como ejemplo, si una sala pertenece a 3 caminos diferentes, su frecuencia es 3, mientras que una sala que aparece en 1 caminos tiene

una frecuencia de 1 y una sala que no pertenece a ningún camino tiene una frecuencia de 0. Para dos salas con la misma frecuencia, se ordenarán según su identificador (ascendentemente). Luego, las armas se distribuirán en las salas que tienen una frecuencia más alta (las que aparecen en un mayor número de caminos). Como se especificó en la segunda entrega, cada sala recibirá 5 armas diferentes, por tanto, la distribución finaliza cuando se hayan distribuido todas las armas. Como el número de armas que se distribuirán es 60, solo las 12 salas con la frecuencia más alta generalmente recibirán armas (no consideraremos mapas con menos de 12 salas).

## 2.- Rutas de los personajes

Tanto el algoritmo de la simulación como el final de la misma serán los mismos que en los establecidos en las entregas anteriores. Sin embargo, habrá una diferencia importante con respecto al comportamiento de los personajes. Generarán sus rutas utilizando un algoritmo particular.

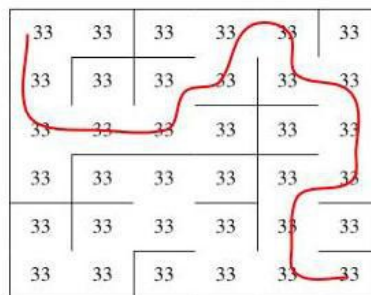
### ***Superhéroes con poderes físicos***

Antes del comienzo de la simulación, cada personaje de tipo SHPhysical debe calcular su ruta desde la sala inicial a la sala Daily Planet. Esta ruta se implementará siguiendo un recorrido en profundidad en el grafo que representa el laberinto generado en el mapa. Por lo tanto, la ruta será una de las rutas que existen en el Mapa desde la sala inicial a la sala Daily Planet. Basado en este algoritmo, el personaje SHPhysical generará una secuencia de direcciones (N, S, E, W) que representa su ruta. Esta secuencia de direcciones se seguirá para llegar a la sala Daily Planet (en la simulación). Por lo tanto, en esta entrega, la ruta del personaje no se introducirá manualmente (como en las etapas anteriores).

En el recorrido en profundidad que debe implementar el personaje, cuando el algoritmo se encuentre con **una sala que tiene dos vecinas posibles, debe siempre elegir las salas vecinas ordenadas ascendentemente por su orden de identificador** (es decir, desde una sala, se elegirá la sala vecina con menor identificador, la segunda la siguiente, y así sucesivamente...).

La ruta calculada por el personaje no contendrá posibles movimientos que le lleven a salas que no tienen salida (no forman parte del camino correcto). Por ejemplo, para el mapa mostrado en la Figura 1 la ruta de un personaje de tipo SHPhysical sería: :

**S, S, E, E, N, E, N, E, S, E, S, S, O, S, S, E**



*Figure 1: Ruta de un personaje de tipo SHPhysical*

Una vez haya comenzado la simulación, el personaje de tipo SHPhysical realizará las acciones especificadas en el documento EC2 (segunda entrega del proyecto) en cada turno.

## Superhéroes con poderes extrasensoriales

Antes del comienzo de la simulación, cada personaje de tipo SHExtraSensorial debe calcular su ruta desde la sala inicial a la sala Daily Planet. Para calcular esta ruta, un personaje de tipo SHExtraSensorial debe usar un algoritmo conocido de resolución de laberintos: *the Wall Follower* utilizando la regla de la mano derecha. Este algoritmo se basa en la idea de que el personaje lleve la mano derecha siempre pegada a la pared. De este modo, el personaje se mueve por las diferentes salas tocando siempre con su mano derecha la pared. Como ejemplo, teniendo en cuenta que la sala de inicio de los personajes SHExtraSensorial será la 0, la Figura 2 muestra un ejemplo de ruta que el personaje seguiría, considerando que la sala Daily Planet estaría situada en la última sala del mapa (sala 35 en este ejemplo). Inicialmente supondremos que el personaje tiene su mano derecha en la pared Oeste de la sala 0 (es decir, estaría mirando en dirección Sur).

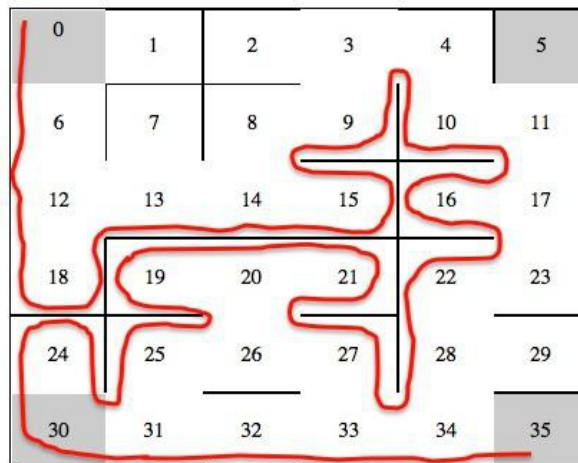


Figura 2: Movimiento de SHExtraSensorial según algoritmo de la mano derecha

Una vez haya comenzado la simulación, el personaje de tipo SHExtraSensorial realizará las acciones especificadas en el documento EC2 (segunda entrega del proyecto) en cada turno.

## Superhéroes con poderes de viaje

Antes de comenzar la simulación, cada personaje SHFlight debe calcular su ruta a través del mapa. La ruta para un personaje SHFlight se basa en moverse de su sala inicial a la sala Daily Planet siguiendo el camino más corto entre estas dos salas.

Una vez haya comenzado la simulación, el personaje de tipo SHFlight realizará las acciones especificadas en el documento EC2 (segunda entrega del proyecto) en cada turno.

## Villanos

Antes de comenzar la simulación, cada personaje de tipo Villain debe calcular su ruta a través del mapa. La ruta para un tipo Villain también se basa en el algoritmo *Wall Follower*. Sin embargo, en este caso, este tipo de personaje usará la regla de la mano izquierda, en lugar de la derecha (como el tipo SHExtraSensorial). Supondremos que este personaje toca inicialmente la pared Este de la sala inicial (es decir, estaría mirando hacia la dirección sur).

Una vez haya comenzado la simulación, el personaje de tipo Villain realizará las acciones especificadas en el documento EC2 (segunda entrega del proyecto) en cada turno.

### 3. Registro de resultados en fichero de texto

El registro de resultados en el fichero de texto se realizará de la misma forma que la especificada en la EC2.

### 4. Completar juego de pruebas unitarias

Las pruebas de unidad e integración para el sistema deben completarse, de acuerdo con las nuevas clases y funcionalidades agregadas en esta tercera etapa.

### 5. Documentación externa del sistema

En la entrega del proyecto en la **convocatoria oficial de enero** se debe presentar la documentación externa completa del sistema, consistente en el manual de usuario y programador acorde con la plantilla presentada en “*Tema02\_ADOO\_AnexoDocumentacion*”, explicado en la sesión teórica de la semana 03 del cuatrimestre. La plantilla para la documentación está disponible en la carpeta de teoría de la semana 03.

### 6. Interfaz gráfica de usuario (GUI)

De manera **opcional**, se podrá realizar una interfaz gráfica de usuario para el proyecto. Esta interfaz será libre, de modo que cada grupo podrá elegir el aspecto que desee y las opciones que considere oportunas. En cualquier caso, debe existir la opción de ejecutar el algoritmo de simulación del proyecto de manera completa sin solicitar intervención alguna por parte del usuario desde el inicio hasta el final. La creación de la interfaz de usuario no exime de la necesidad de registrar los resultados en el fichero “registro.log” por lo que éstos deben seguir apareciendo en el mismo.

Para la creación de la interfaz de usuario, en las sesiones de teoría y prácticas de la asignatura dedicadas a este tema, se entregará un ejemplo de interfaz (incompleta) que los estudiantes podrán extender y modificar para utilizarlo en sus proyectos.

## TERCERA ENTREGA (EC3)

### Consideraciones

- El fichero de registro debe denominarse **registro.log** y debe seguir **estrictamente el formato** definido en el enunciado EC2.
- El programa entregado debe ejecutarse de principio a fin **sin pedir ninguna tecla**.
- La **salida** por pantalla **no tiene un formato predefinido**, cada estudiante puede hacerla como desee.
- Los estudiantes deben hacer un uso correcto de los conceptos de **herencia y polimorfismo**.
- Es recomendable que el proyecto incluya la implementación y utilización de la estructura de datos **grafo** y los **patrones de diseño** estudiados en la asignatura.
- El nombre del fichero de inicio se especificará como un parámetro del ejecutable de nuestro proyecto (será explicado en una sesión de prácticas), según la sintaxis:  
<nombre\_del\_programa> <nombre\_fichero>

Ejemplo: `java Mapa inicio.txt`

- Se deben seguir **TODAS** las recomendaciones apuntadas en los **enunciados anteriores**.

La entrega de esta fase será: **miércoles, 20 de diciembre**.