

# Enunciado de Ejercicio

Ismael Sallami Moreno

**Titulación:** Ingeniería Informática + ADE

**Grupo:** A1

**Profesor:** Francisco Velasco

## Enunciado del Ejercicio

### 1. Super Player (Java y Ruby)

Implementar una clase llamada `SuperPlayer` que extienda a la clase `Player`. Esta clase debe cumplir con los siguientes requisitos:

- **Enumerado:** Crear un enumerado con los valores `Fuzzy` y `Super`.
- **Clase Dice:** Añadir una función en la clase `Dice` que devuelva un valor del enumerado `Fuzzy` o `Super`, en función de las siguientes probabilidades:
  - 20 % de probabilidad de devolver `Fuzzy`.
  - 80 % de probabilidad de devolver `Super`.
- **Clase SuperPlayer:** Crear la clase `SuperPlayer`, que cumpla con las siguientes características:
  - Heredar de la clase `Player`.
  - Sobrescribir el método de ataque para que ataque como su clase padre, pero multiplicando el valor del ataque por la inteligencia del personaje.
  - Sobrescribir el método `toString` para que devuelva la cadena: "Soy super".
  - Modificar el comportamiento en caso de muerte: si el jugador muere, debe resucitar como `Fuzzy` o como `Super`, de acuerdo con las probabilidades definidas en la clase `Dice`.

- **Integración:** Realizar las modificaciones necesarias en las clases relacionadas (**Labyrinth**, **Game**, y otras que correspondan) para garantizar la correcta funcionalidad del comportamiento descrito.

**Nota:** Asegúrate de implementar este ejercicio tanto en **Ruby** como en **Java**.

## 2. Interfaz Cursors (Solo en Java)

En la interfaz **Cursors** de Java, implementar un botón adicional que, al pulsarlo, mueva al jugador en una dirección aleatoria. Las posiciones posibles son:

- Arriba.
- Abajo.
- Izquierda.
- Derecha.

## Entrega

Incluye el código fuente de las implementaciones en ambos lenguajes, con comentarios explicativos y pruebas funcionales que demuestren el correcto funcionamiento de las modificaciones.

## Solución

Para ver la solución pinche aquí.