



Secretaría  
de Educación  
de Guanajuato



**SEP**  
SECRETARÍA DE  
EDUCACIÓN PÚBLICA

**UTP**  
DIRECCIÓN GENERAL DE UNIVERSIDADES  
TECNOLOGICAS Y POLITÉCNICAS

**UFG**  
Universidad Tecnológica  
del Norte de Guanajuato  
Organismo Público Descentralizado del Gobierno del Estado  
"Educación y progreso para la vida"



# UNIVERSIDAD TECNOLÓGICA DEL NORTE DE GUANAJUATO



Universidad Tecnológica  
del Norte de Guanajuato  
Organismo Público Descentralizado del Gobierno del Estado  
"Educación y progreso para la vida"

## Ingeniería en Desarrollo y Gestión de Software

### Unidad II.

## Programación orientada a objetos para videojuegos

**Asignatura:**

**Creación de Videojuegos**

**PRESENTA:**

**Herrera Ramiro**

**Grupo:**

**GIDS5101-e**

**A 03 de octubre del 2025**

**Dolores Hidalgo Cuna de la Independencia Nacional, Guanajuato.**

**UNIVERSIDAD TECNOLÓGICA DEL NORTE DE GUANAJUATO**

Av. Educación Tecnológica No. 34 Fracc. Universidad | C.P. 37800 | Tel. (418) 182 5500  
Dolores Hidalgo Cuna de la Independencia Nacional, Guanajuato  
utng.edu.mx | [f](#) [t](#) [g](#) @UTNGDOLORESHIDALGO

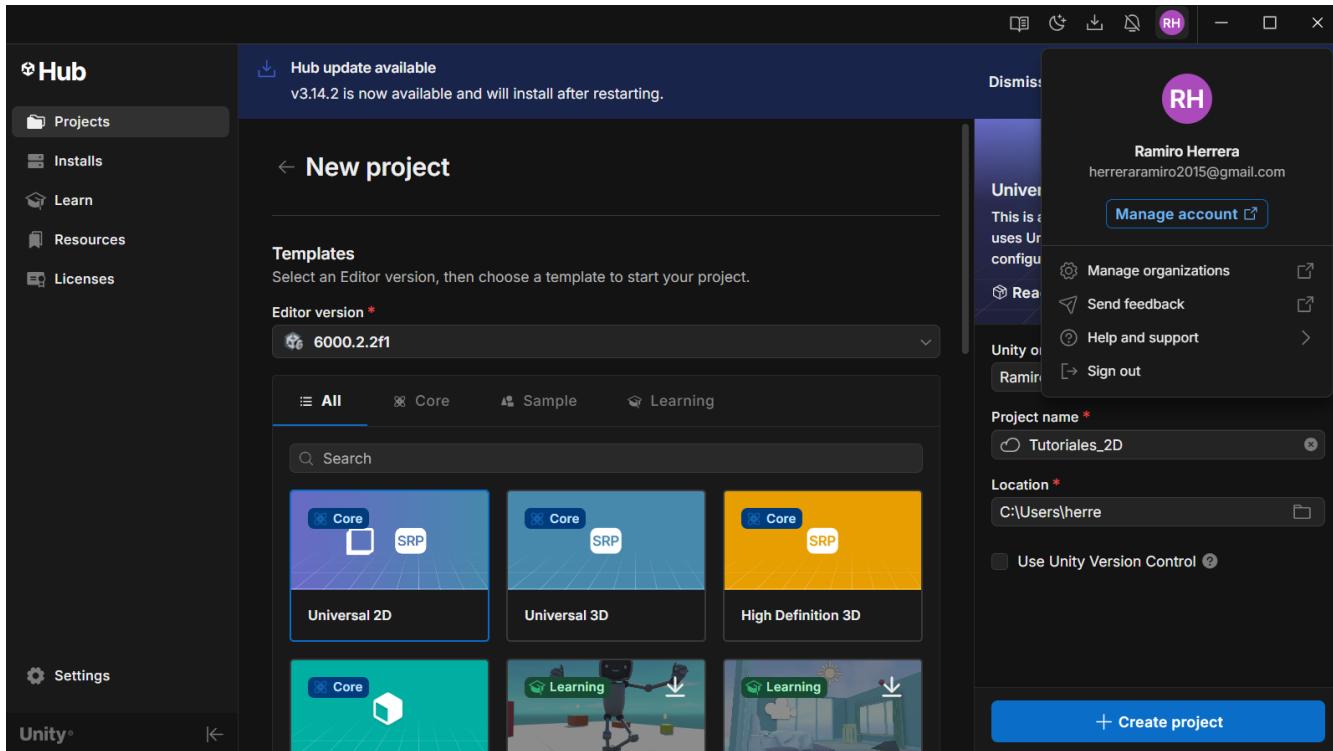
## TABLA DE CONTENIDO

Tutorial 01 - ✎ PROCEDIMIENTO - PASO A PASO ✎ .....	1
CONFIGURACIÓN INICIAL .....	1
CREACIÓN DE OBJETOS .....	3
TEORÍA: DISEÑO ENTIDAD-COMPONENTE .....	4
CONFIGURACIÓN DE SPRITES .....	4
EDITOR DE SPRITES .....	7
CREACIÓN DE ANIMACIONES .....	11
ANIMATOR CONTROLLER .....	14
PRUEBA Y AJUSTES .....	16
🎯 DESAFÍO FINAL .....	18
RESULTADO FINAL .....	21

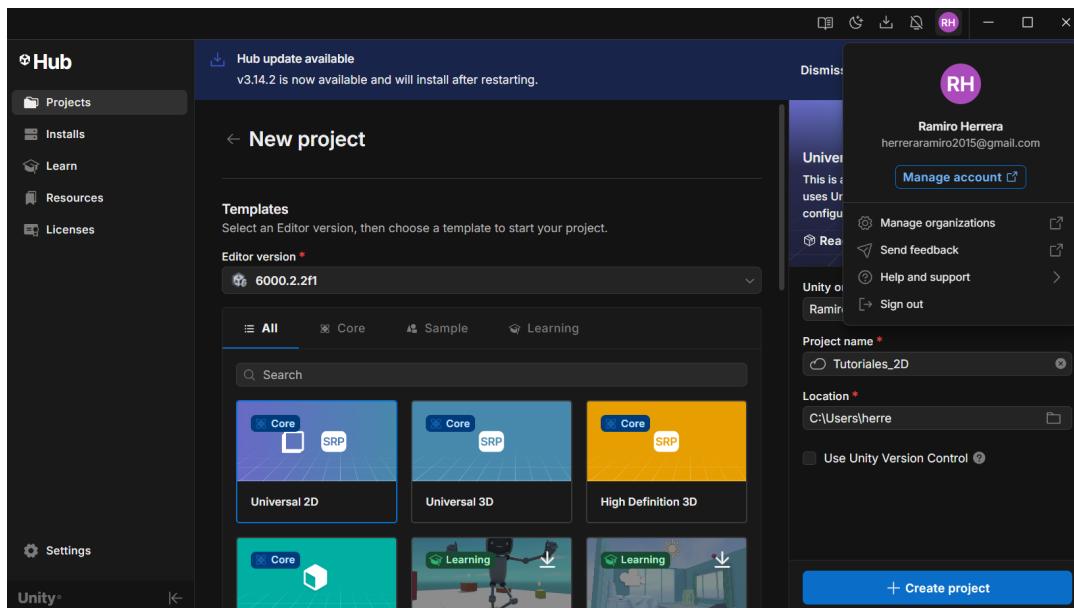
## Tutorial 01 - ✎ PROCEDIMIENTO - PASO A PASO ✎

### CONFIGURACIÓN INICIAL

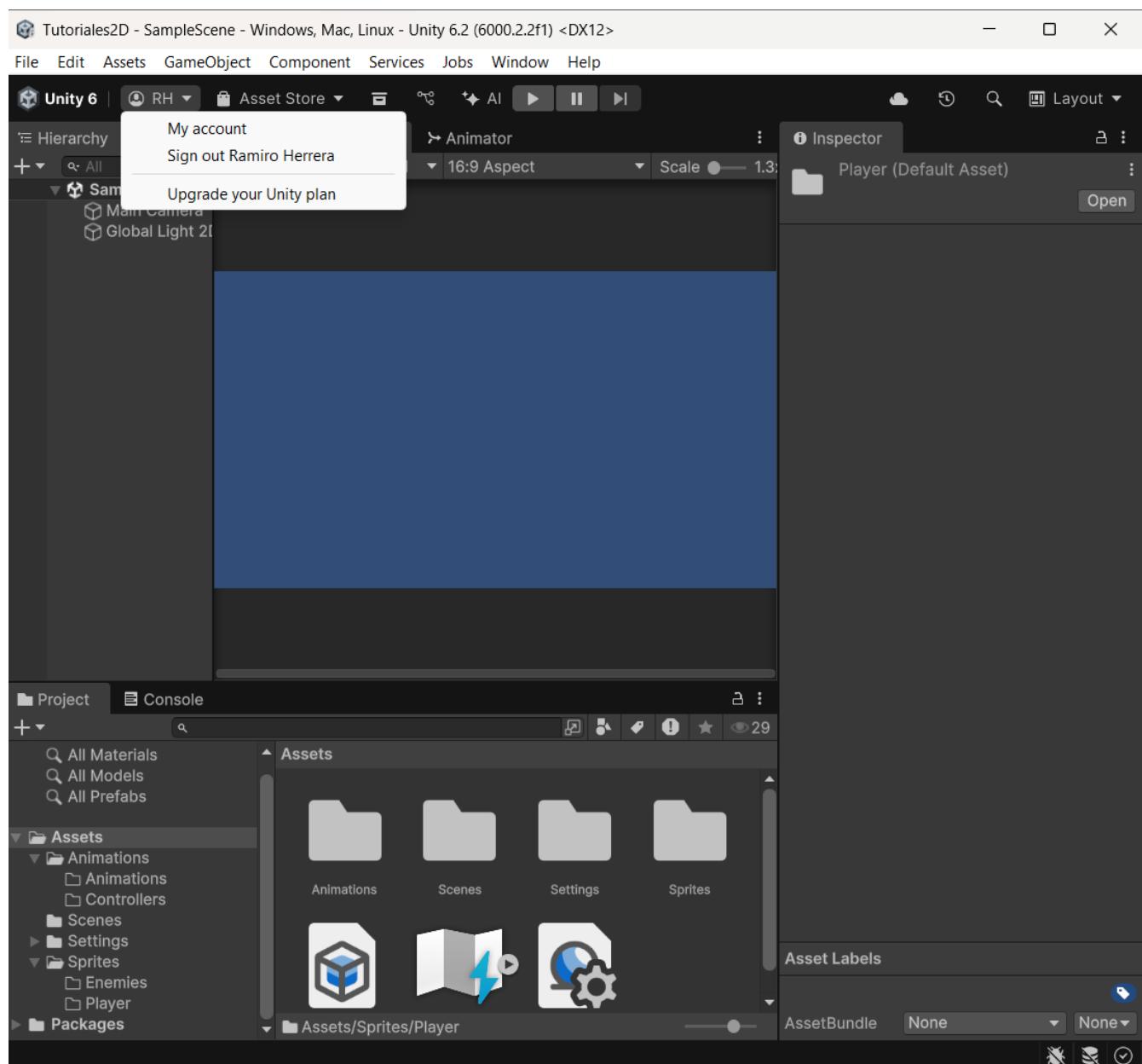
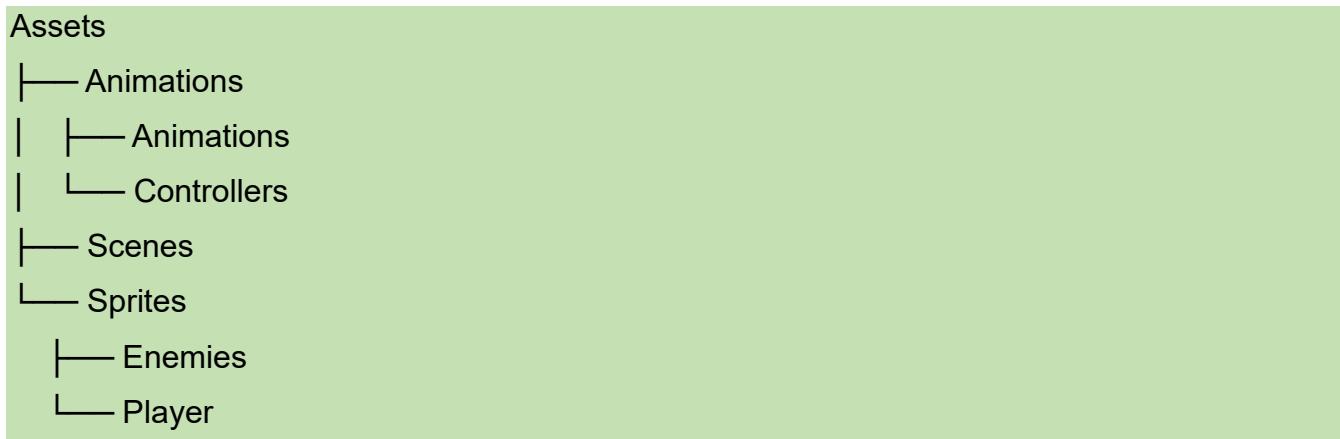
**Paso 1:** Crear un nuevo proyecto a través de Unity Hub. Seleccionar el número de versión más actual de acuerdo con las versiones instaladas en tu Unity Hub.



**Paso 2:** Una vez que aparezca el cuadro de diálogo, seleccionar la opción **2D RPG**, escribir un nombre **RPG** para el proyecto y la ruta para almacenarlo. Presionar el botón **CREATE** y esperar a que Unity cargue e inicialice el proyecto.



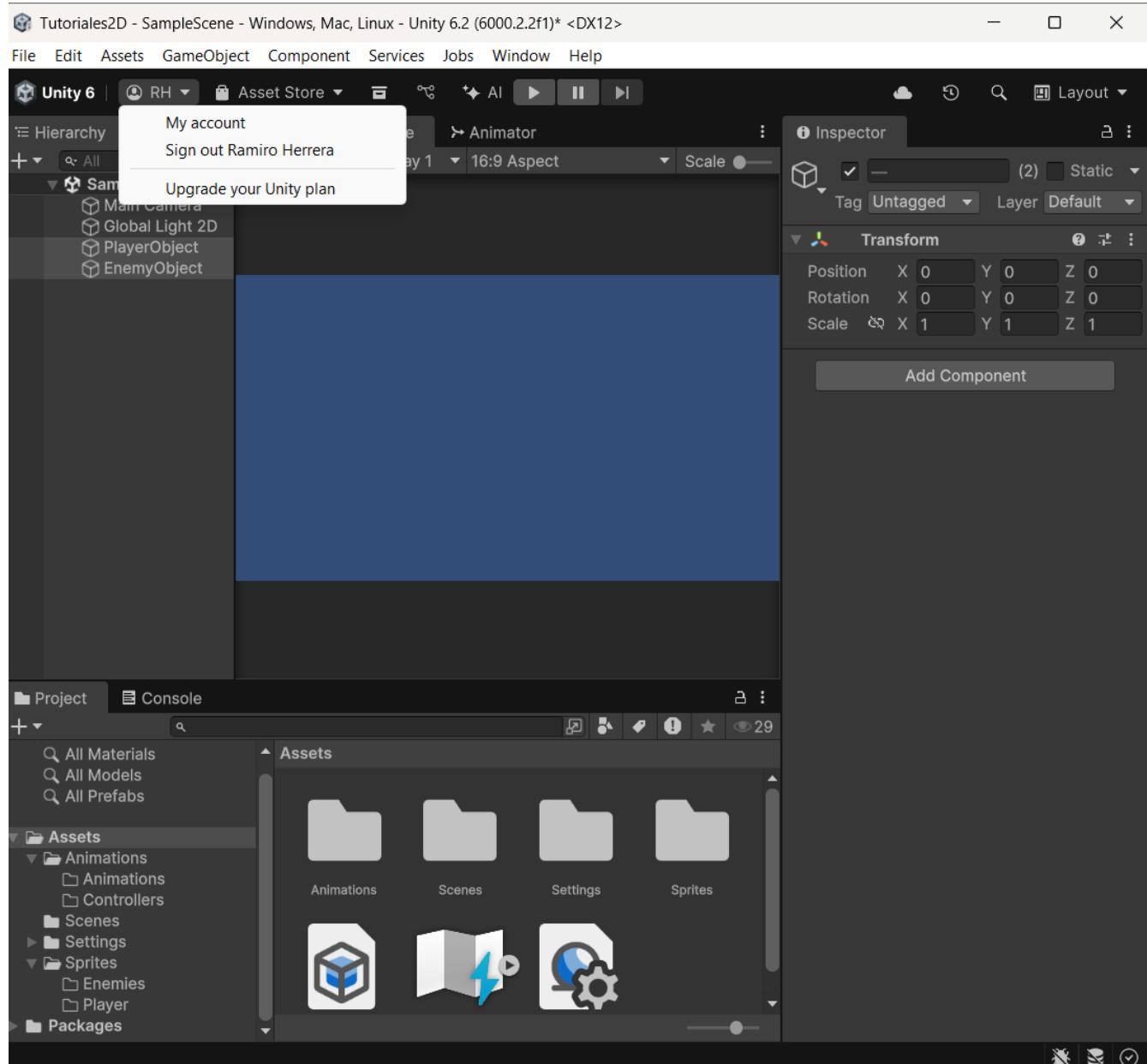
**Paso 3:** Crear la estructura de directorios tal y como se muestra:



## CREACIÓN DE OBJETOS

**Paso 4:** Abrir el IDE de Unity, crear dos objetos vacíos mediante la vista **Hierarchy** opción crear que se encuentra en la esquina superior izquierda. Renombrar los objetos como:

- **PlayerObject**
- **EnemyObject**

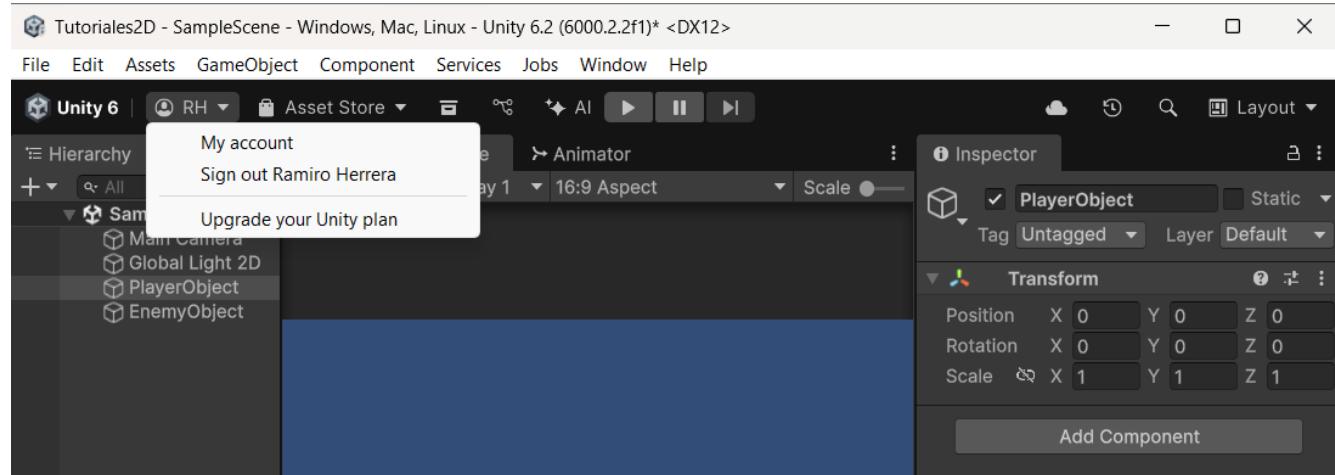


## TEORÍA: DISEÑO ENTIDAD-COMPONENTE

**Paso 5:** Entender el concepto de "separación de intereses" y el patrón Entity-Component de Unity:

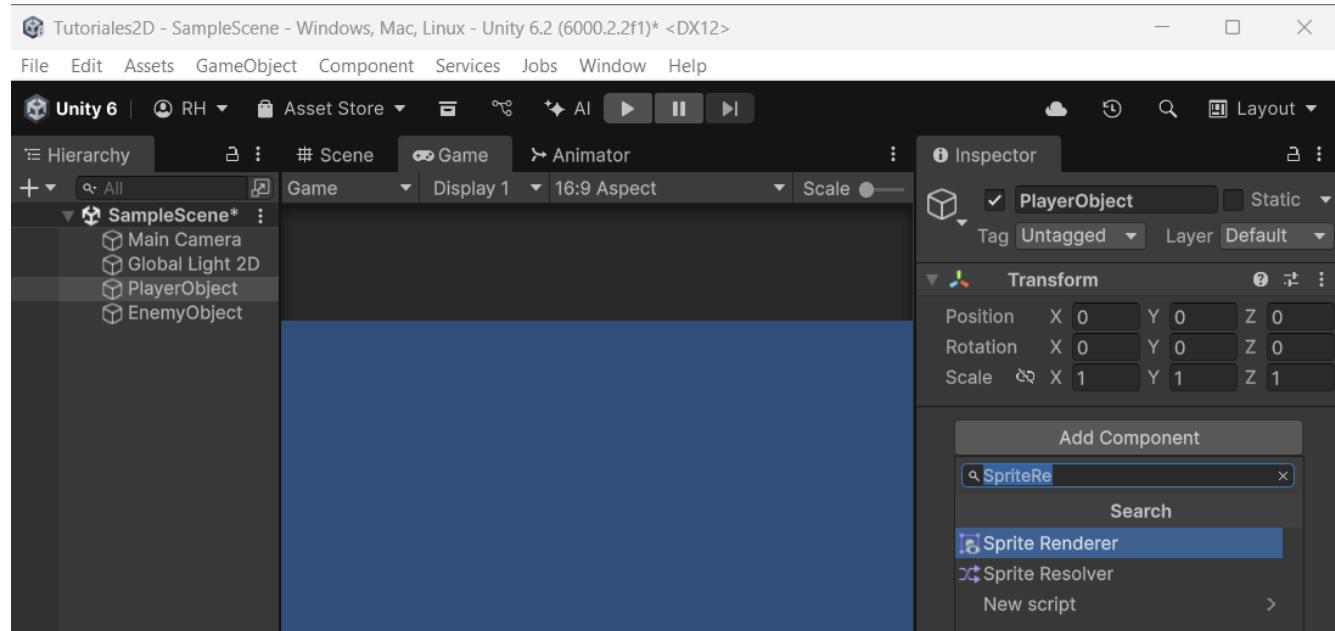
- **GameObject** = Entidad
- **Componentes** = Funcionalidades específicas
- Todo en una escena de Unity se considera un GameObject

**Paso 6:** Seleccionar el objeto PlayerObject dentro de la vista Hierarchy y observar los valores dentro de la vista Inspector.

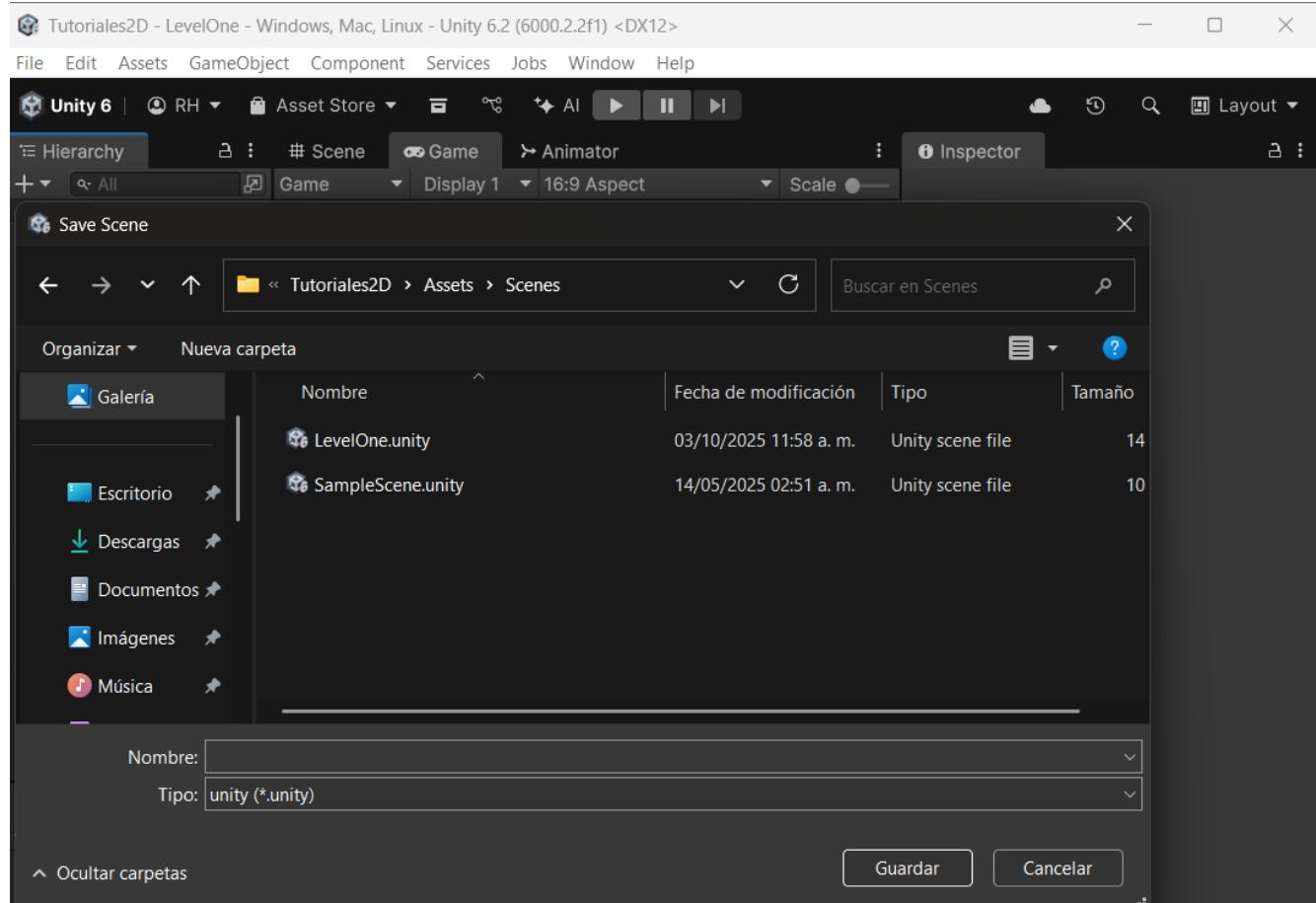


## CONFIGURACIÓN DE SPRITES

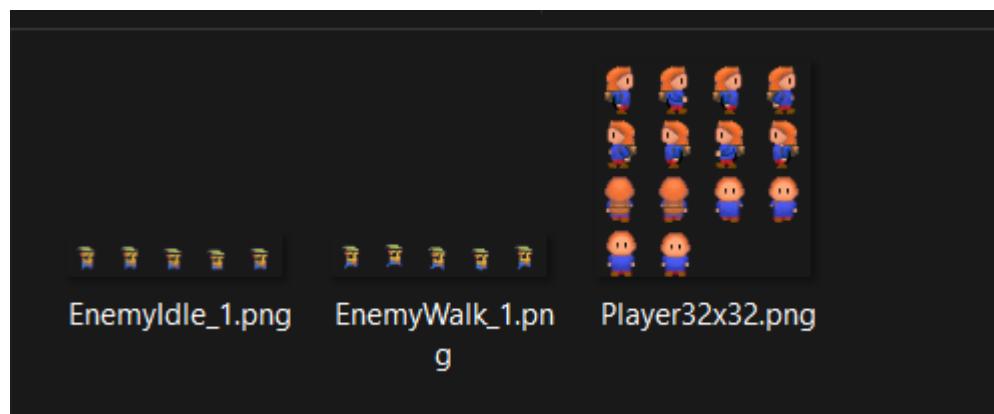
**Paso 7:** Para el objeto PlayerObject seleccionar la opción **Add Component** → **Sprite Renderer**. Utilizar el mismo proceso para el objeto EnemyObject.



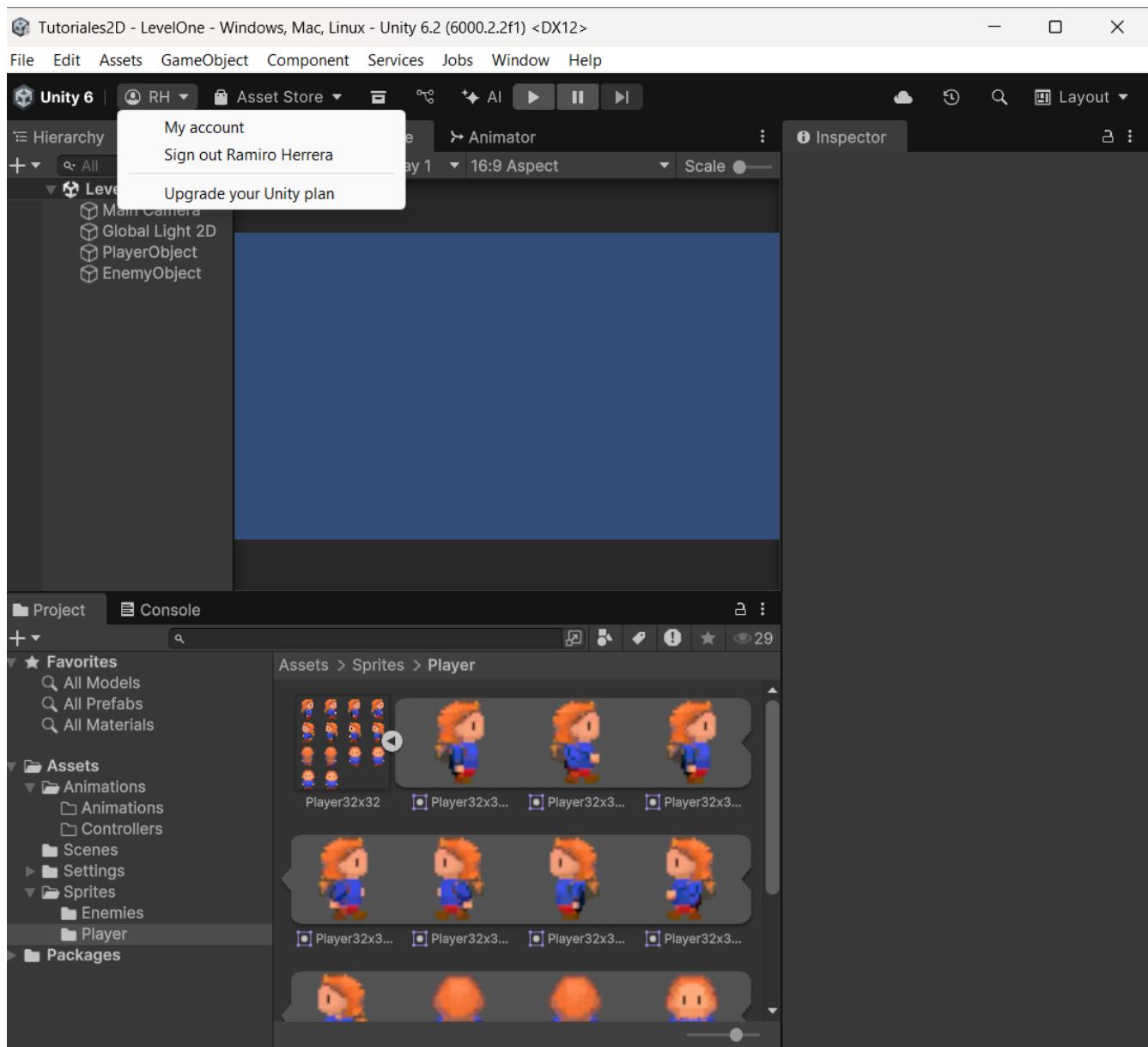
**Paso 8:** Guardar la escena como **LevelOne**, primeramente, crear un folder llamado **Scenes** y dentro de ella guardar.



**Paso 9:** Descargar el archivo de imágenes que viene junto al tutorial, seleccionar el archivo **Player.png**, **EnemyIdle\_1.png**, **EnemyWalk\_1.png** y arrastrarlos al folder correspondiente dentro de la carpeta sprites.



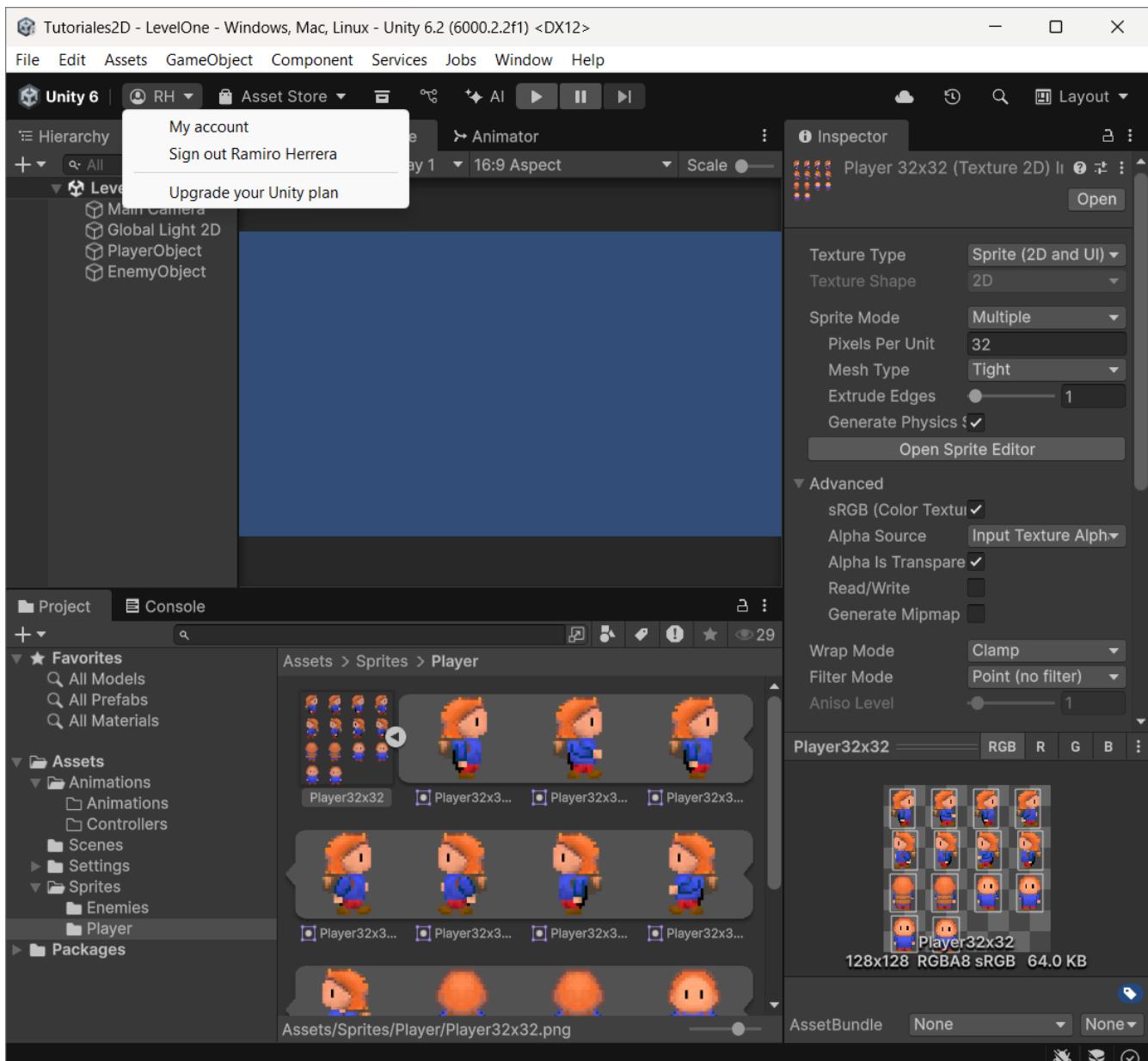
## Unidad I. Programación orientada a objetos para videojuegos



**Paso 10:** Cambiar las propiedades del objeto Player.png dentro de la vista Inspector:

- **Texture Type:** Sprite (2D and UI)
- **Sprite Mode:** Multiple
- **Pixels Per Unit:** 32
- **Filter Mode:** Point (no filter)
- **Compression:** None
- Hacer clic en **Apply**

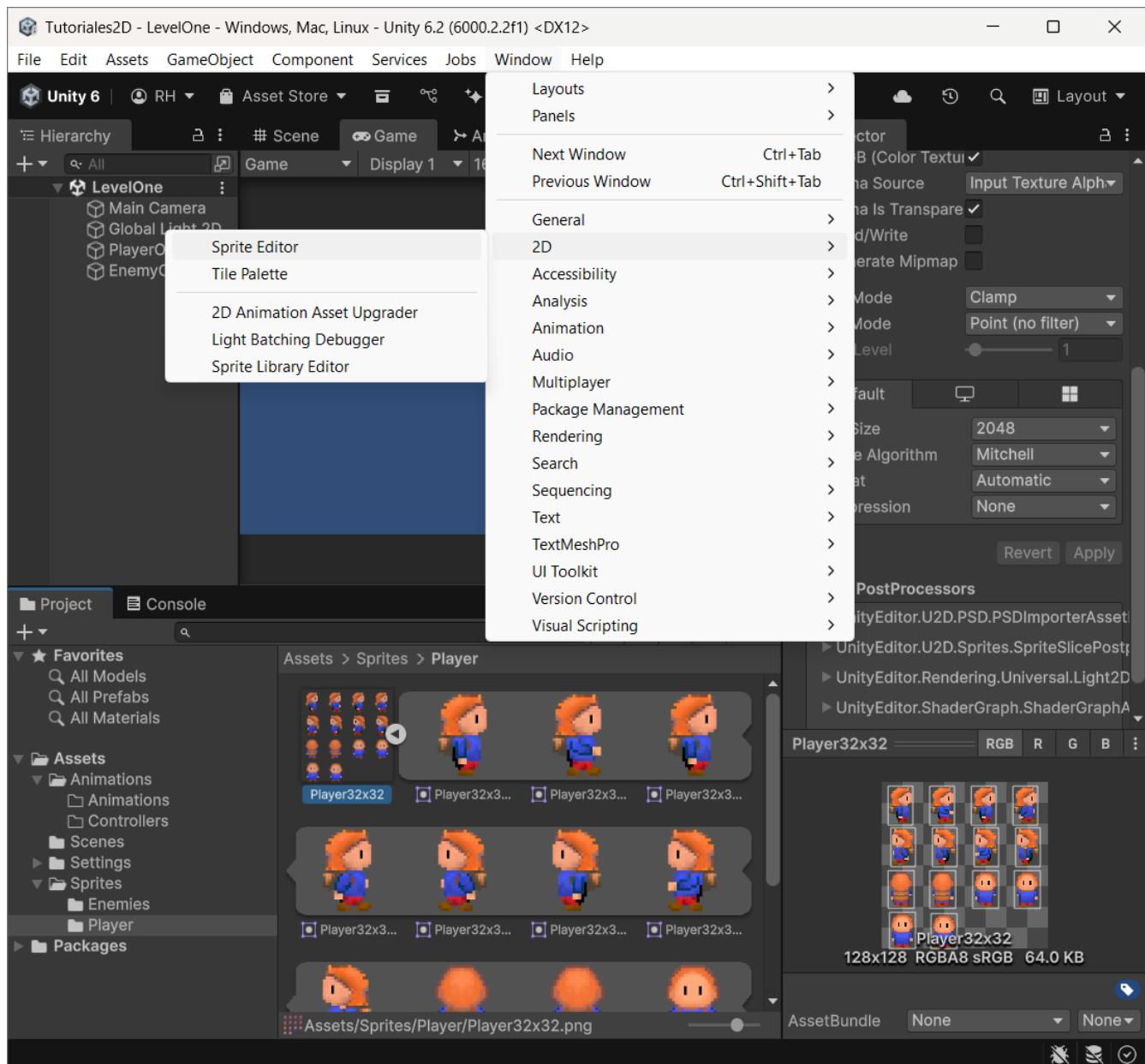
## Unidad I. Programación orientada a objetos para videojuegos



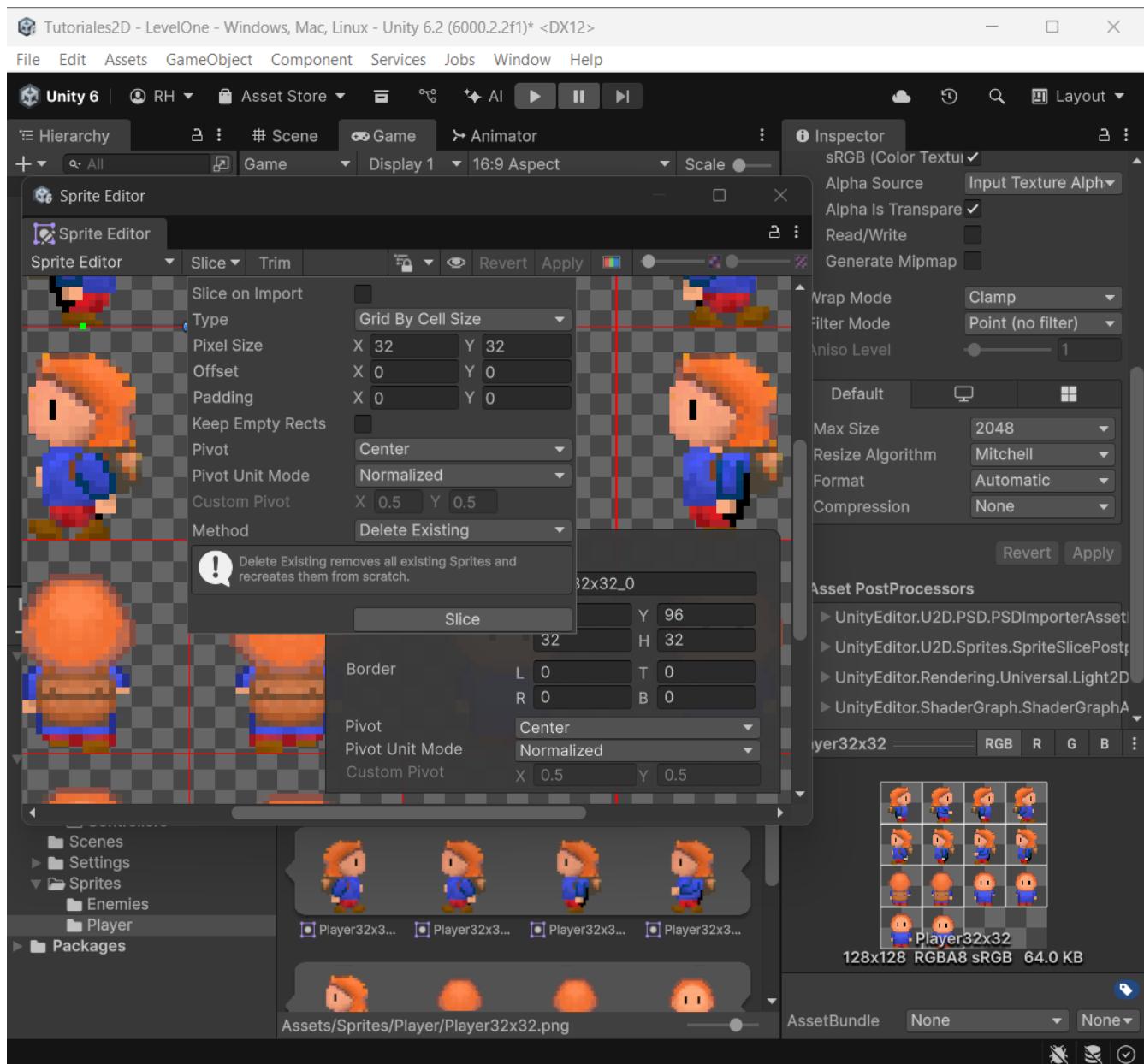
### EDITOR DE SPRITES

**Paso 11:** Ubicar la opción **Window → 2D → Sprite Editor** de la vista Inspector para dividir la hoja de sprites en individuales. Presionar el botón **Slice** y configurar las propiedades. Presionar **Apply** para aplicar el corte.

## Unidad I. Programación orientada a objetos para videojuegos

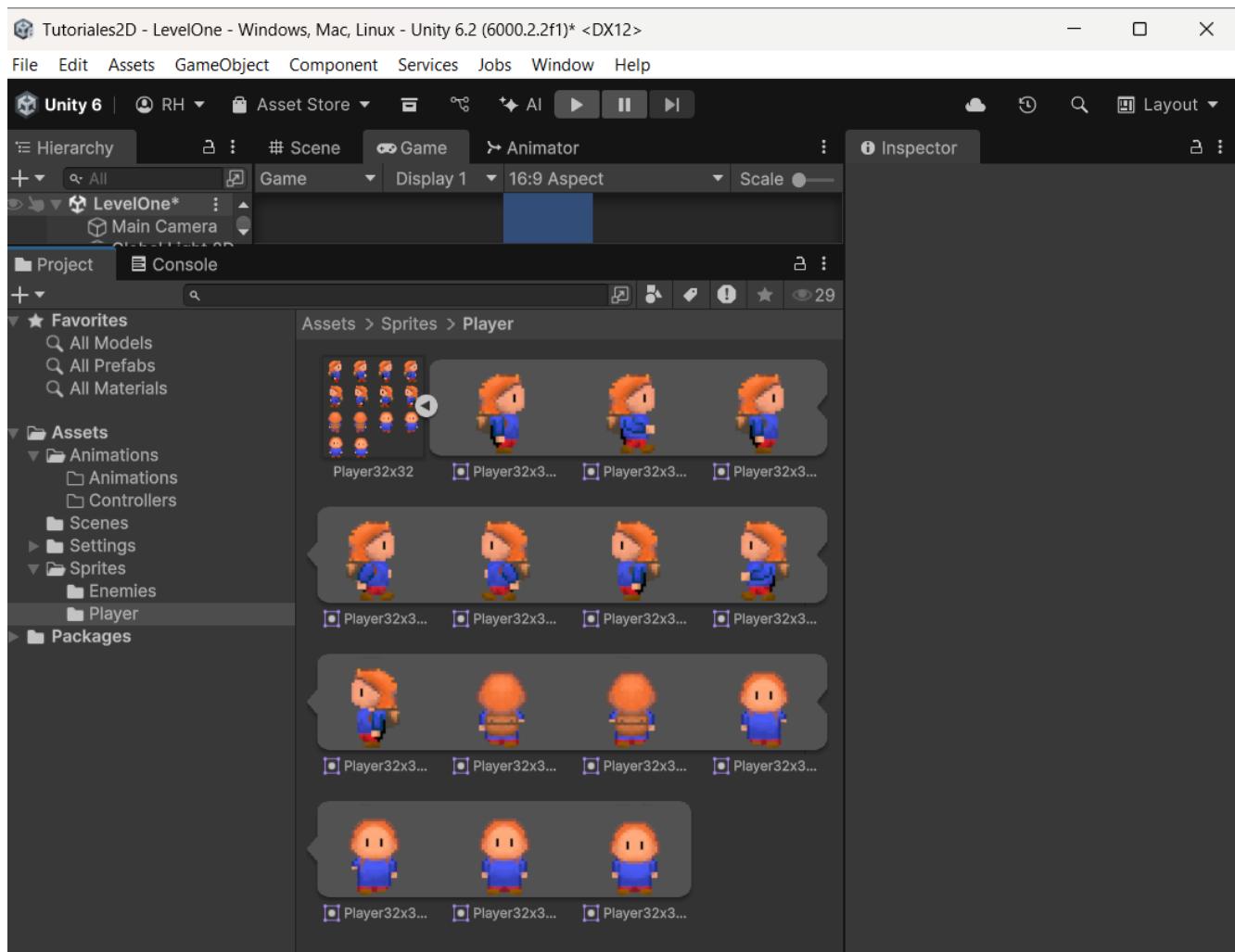


## Unidad I. Programación orientada a objetos para videojuegos

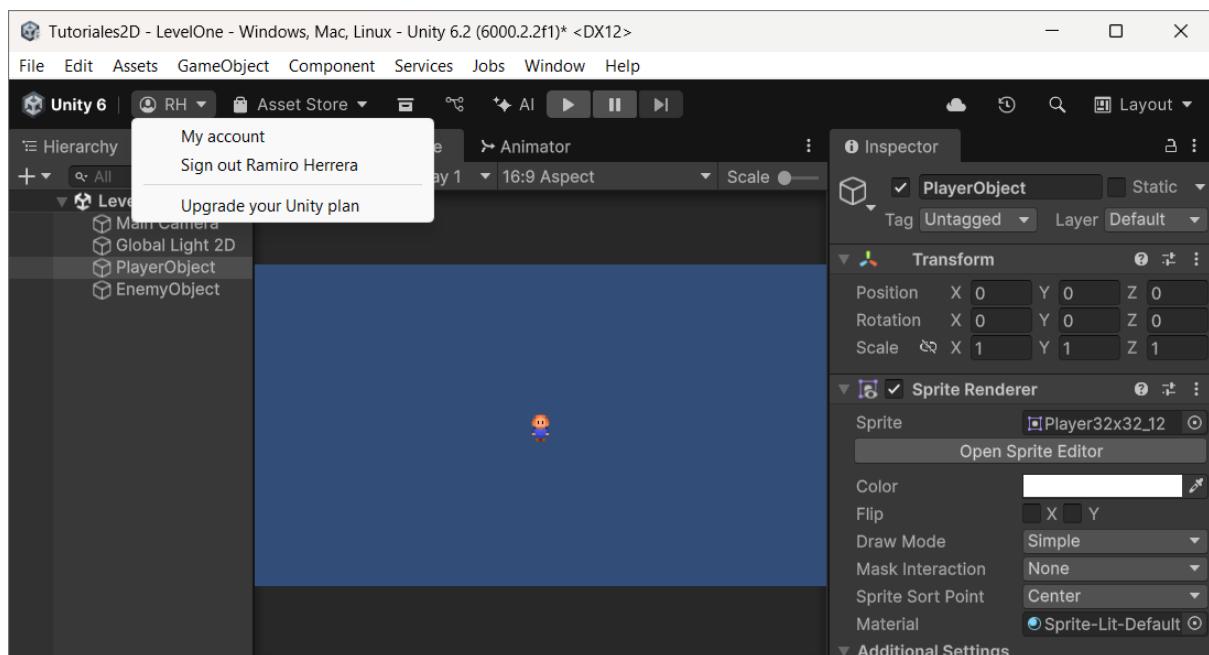


**Paso 12:** Hacer clic en el pequeño triángulo junto a la hoja de sprites del jugador para ver todos los sprites individuales extraídos.

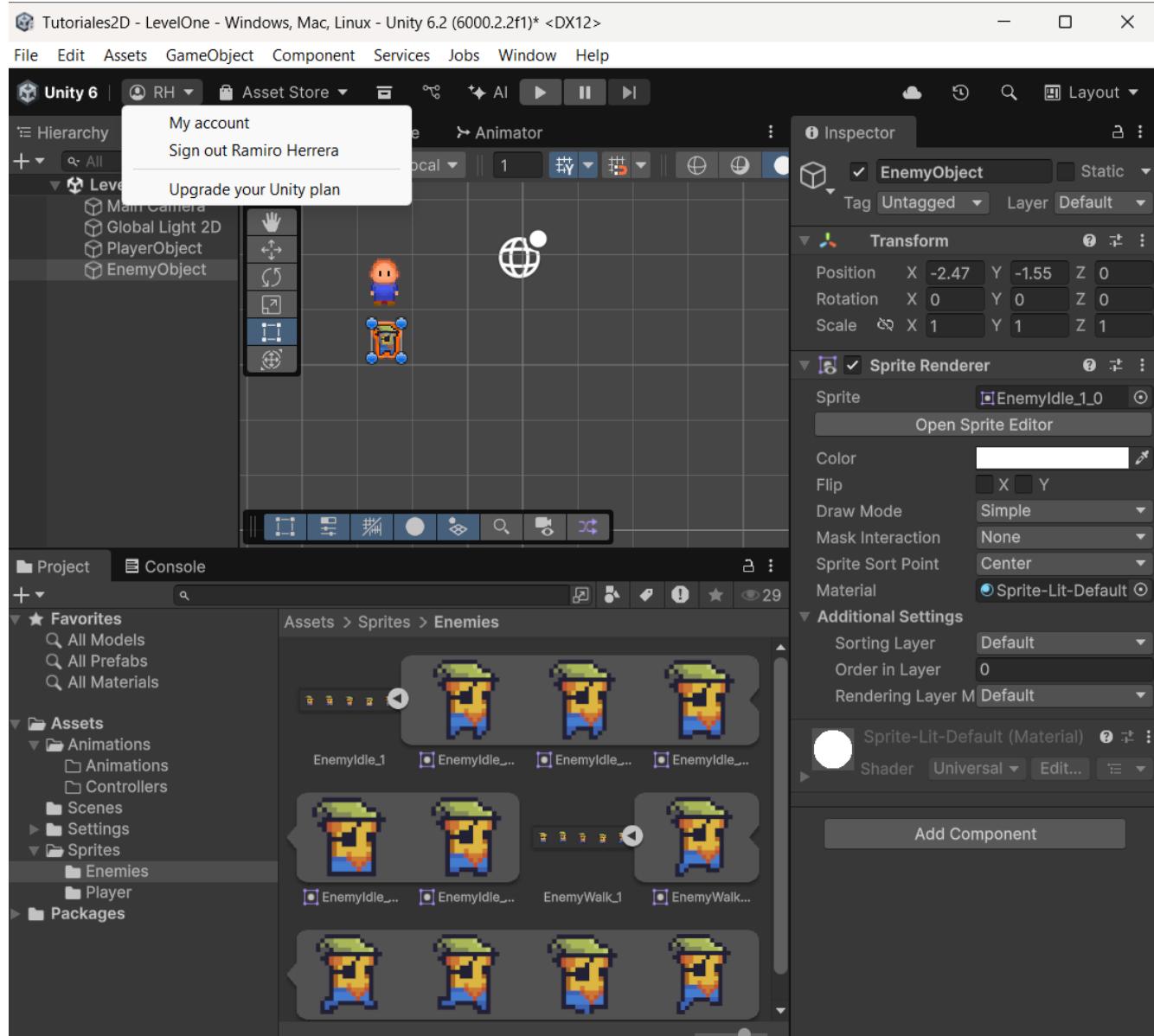
## Unidad I. Programación orientada a objetos para videojuegos



**Paso 13:** Seleccionar el PlayerObject. En la vista Inspector, a la derecha de la propiedad Sprite hacer clic en el pequeño círculo y seleccionar un sprite para la vista Game.



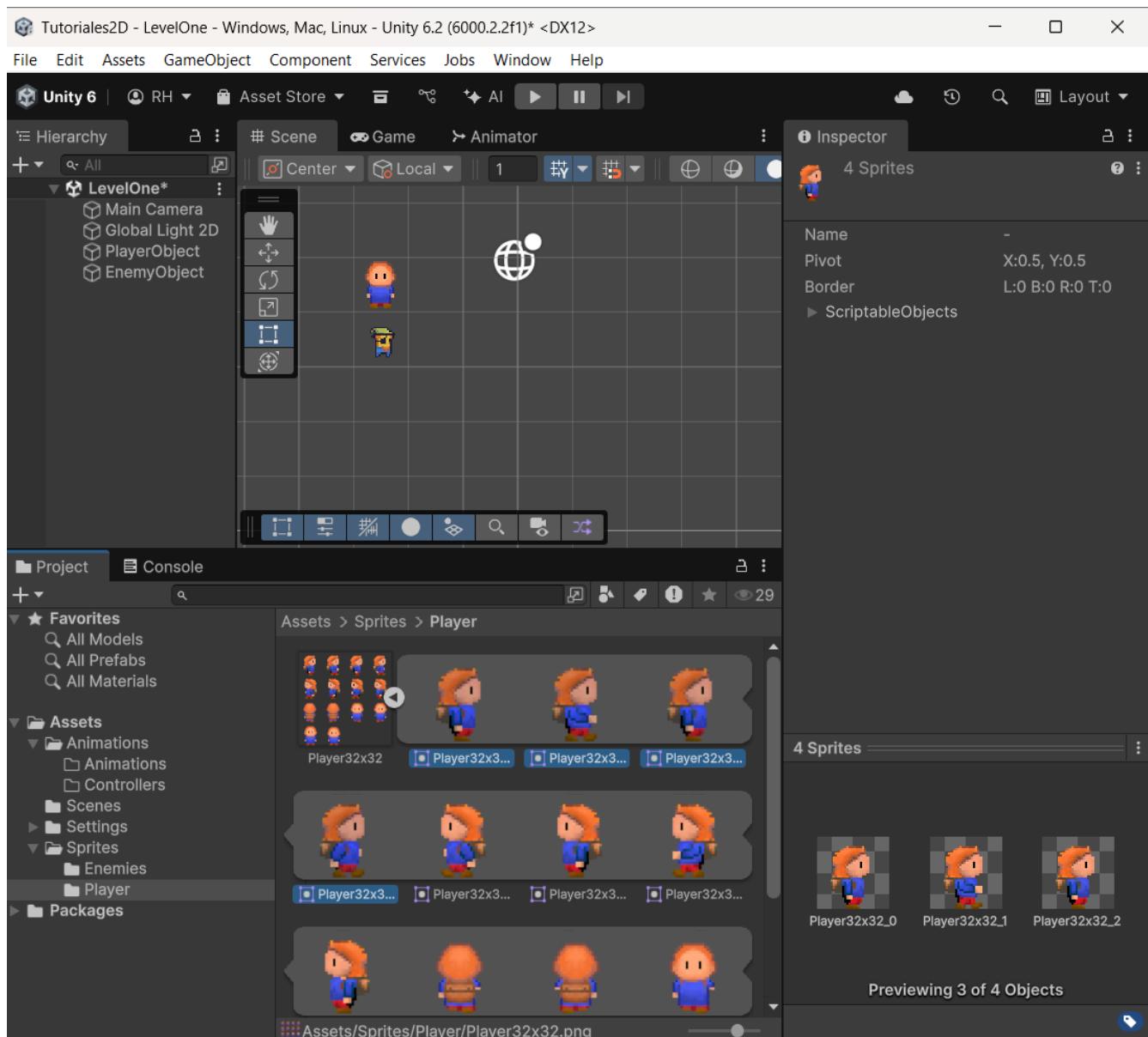
**Paso 14:** Repetir el mismo proceso con EnemyIdle\_1 y EnemyWalk\_1 (configurar como Multiple y aplicar).



## CREACIÓN DE ANIMACIONES

**Paso 15:** Expandir los sprites del Player haciendo clic en la pequeña flecha junto a él en la Vista de proyecto. Seleccionar los cuatro primeros sprites con **Ctrl + clic izquierdo** del ratón.

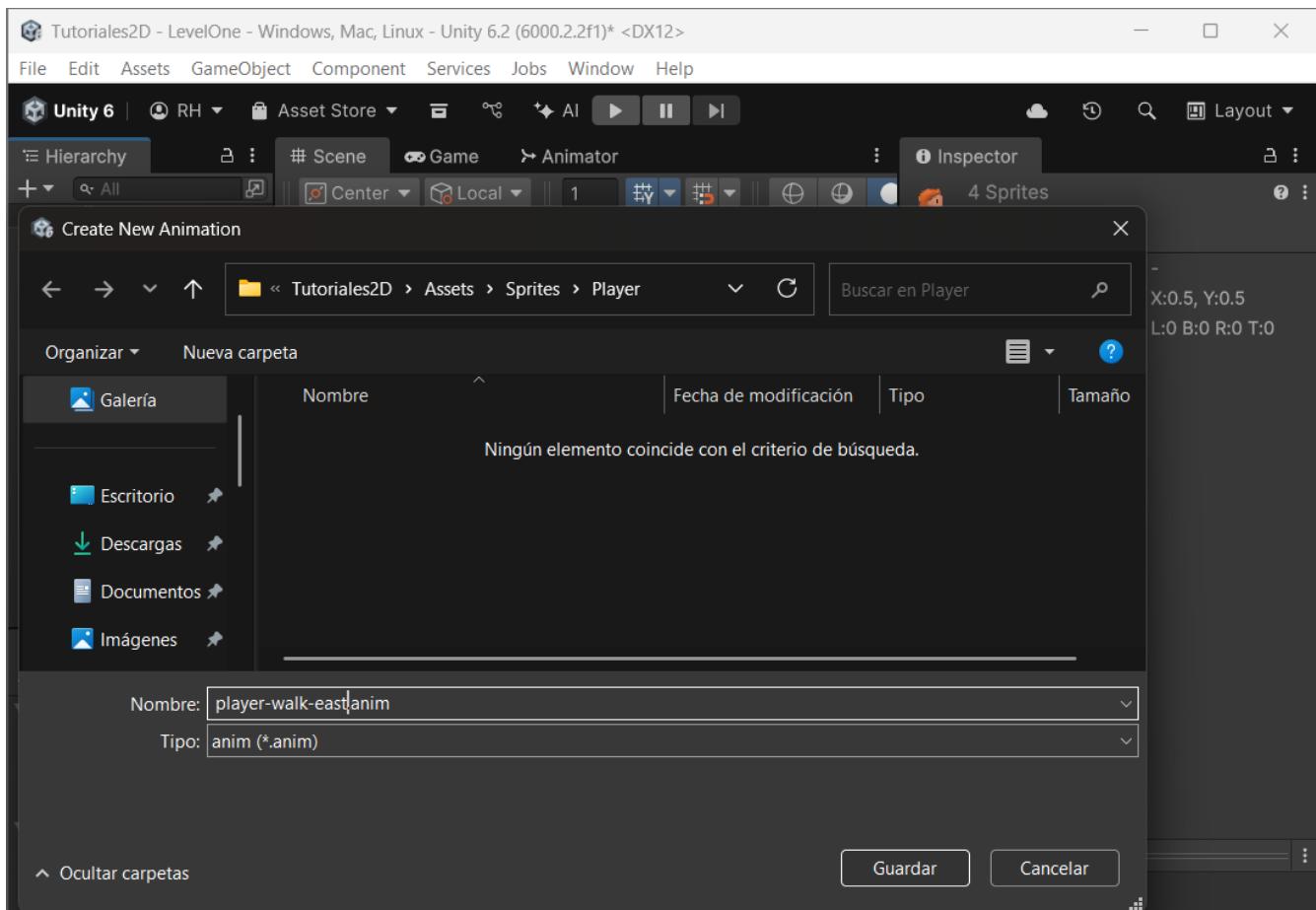
## Unidad I. Programación orientada a objetos para videojuegos



**Paso 16:** Arrastrar los sprites seleccionados sobre el objeto **PlayerObject**.

**Paso 17:** Guardar la animación en la carpeta **Animaciones** y nombrar como **player-walk-east**.

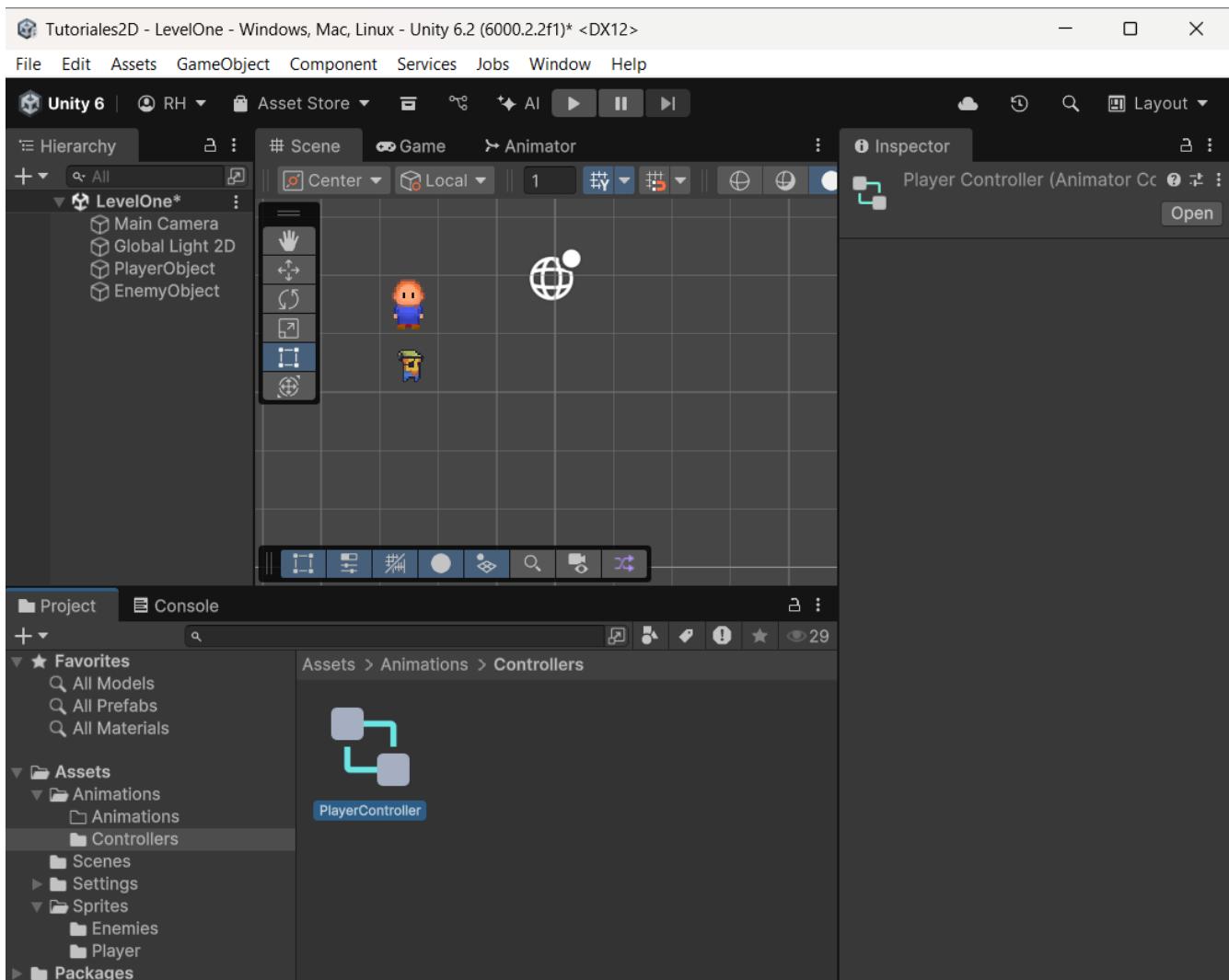
## Unidad I. Programación orientada a objetos para videojuegos



**Paso 18:** Seleccionar PlayerObject y observar la vista del Inspector. Verificar que tenemos dos componentes nuevos: **Sprite Renderer** y **Animator**.

**Paso 19:** Renombrar el controller como **PlayerController** y arrastrarlo a la carpeta **Animations/Controllers**.

## Unidad I. Programación orientada a objetos para videojuegos

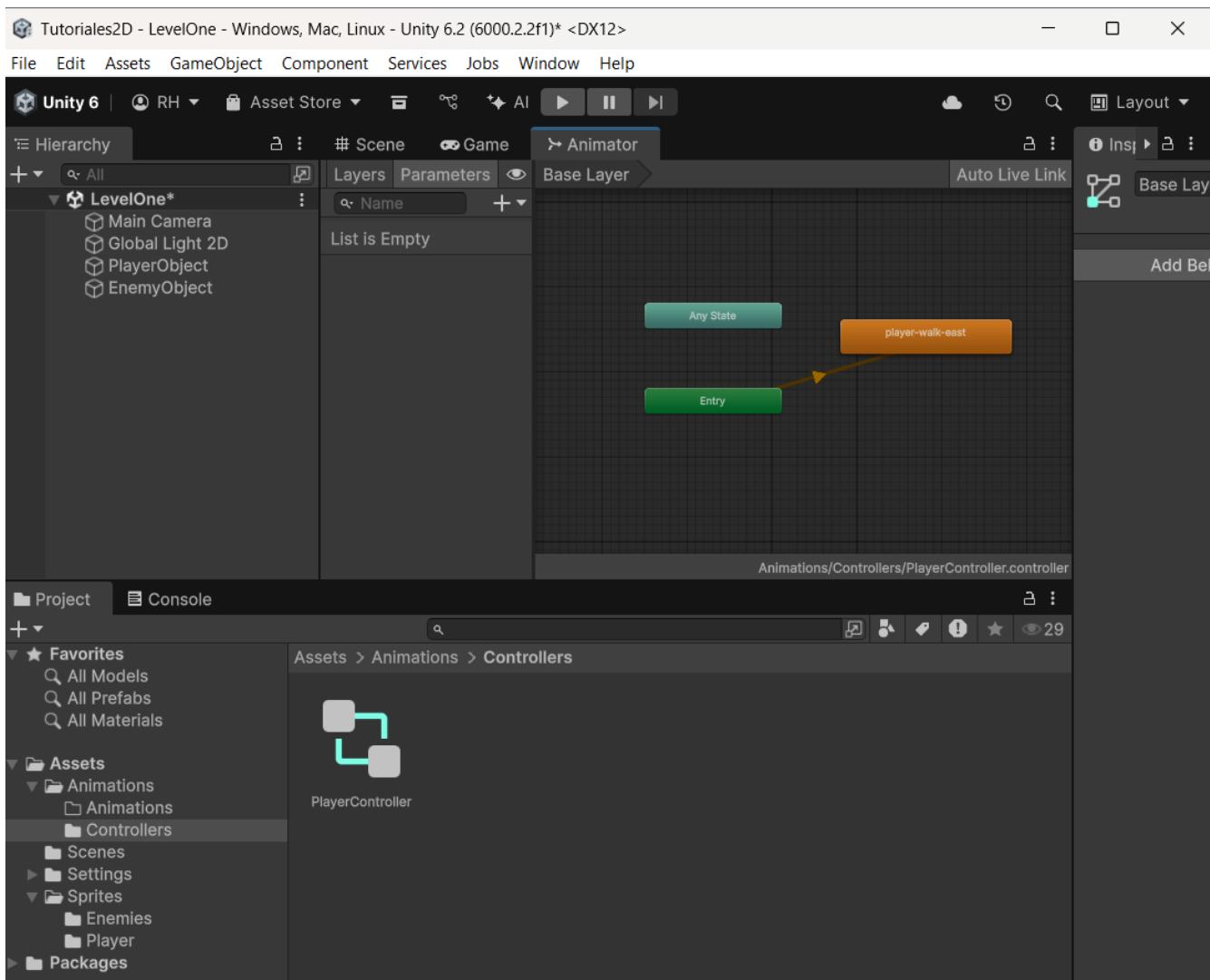


## ANIMATOR CONTROLLER

**Paso 20:** Hacer doble clic sobre **PlayerController** para que abra la ventana Animator.

**Concepto:** El controlador de animación mantiene un conjunto de reglas, llamada **máquina de estado**, que determina qué clip de animación se reproduce según el estado del jugador (caminar, atacar, inactivo, etc.).

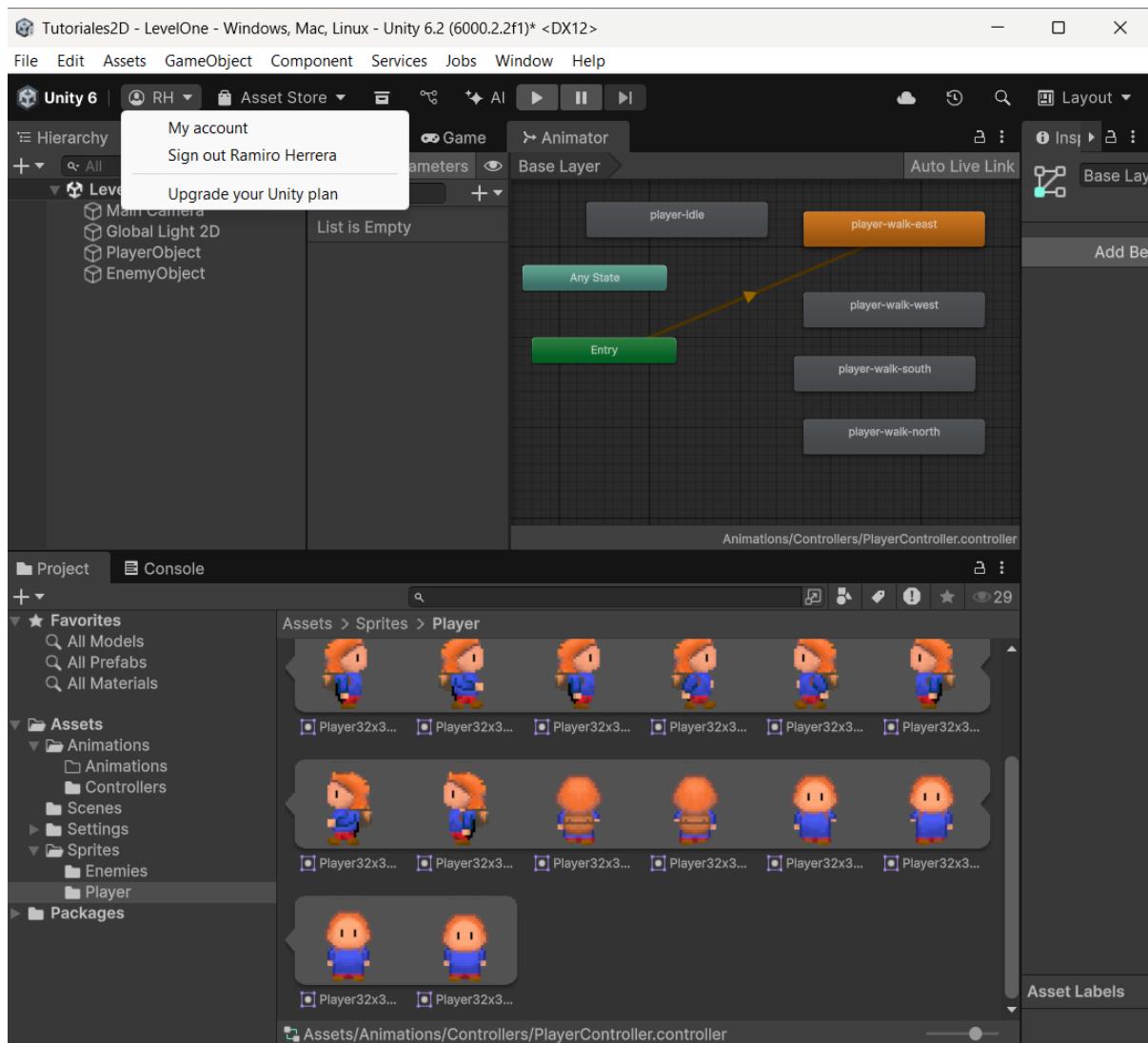
## Unidad I. Programación orientada a objetos para videojuegos



**Paso 21:** Agregar el resto de las animaciones. Volver a la carpeta Sprites y seleccionar las animaciones y volverlas a arrastrar sobre el objeto PlayerObject:

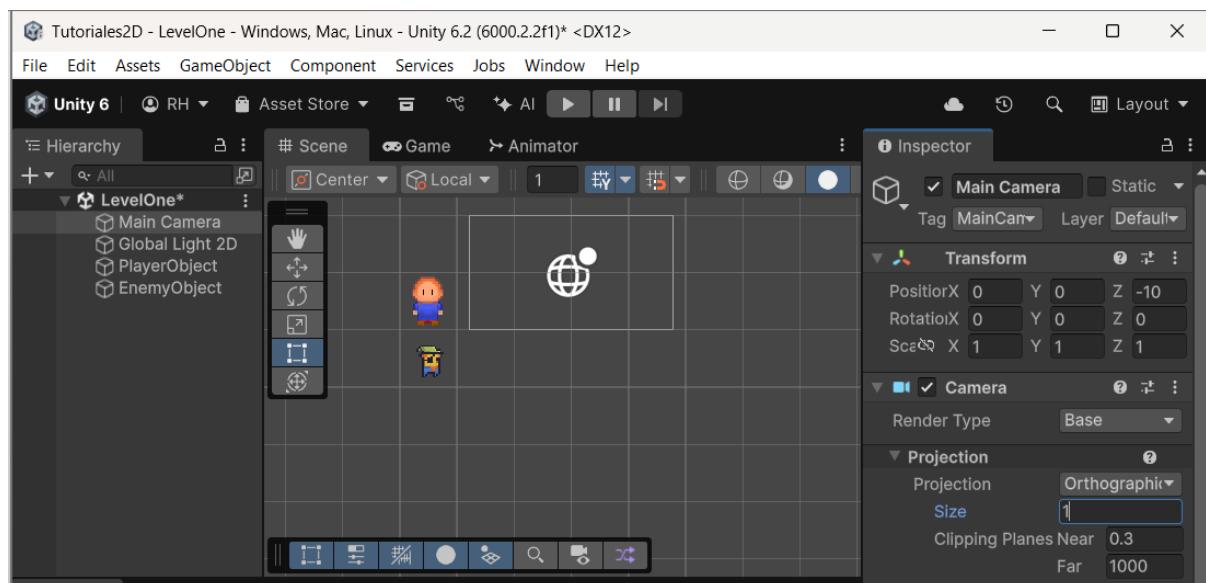
- **player-walk-west** (siguientes cuatro animaciones)
- **player-walk-south** (siguientes dos animaciones)
- **player-walk-north** (siguientes dos animaciones)
- **player-idle** (siguientes dos animaciones)

## Unidad I. Programación orientada a objetos para videojuegos

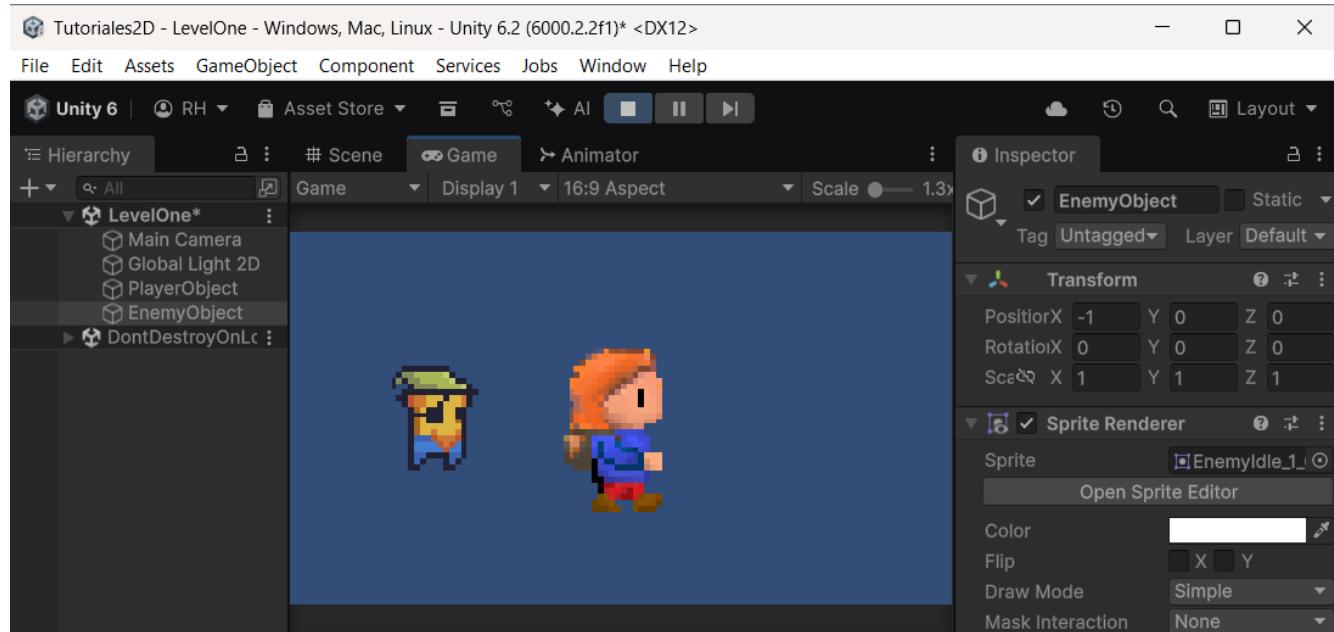


## PRUEBA Y AJUSTES

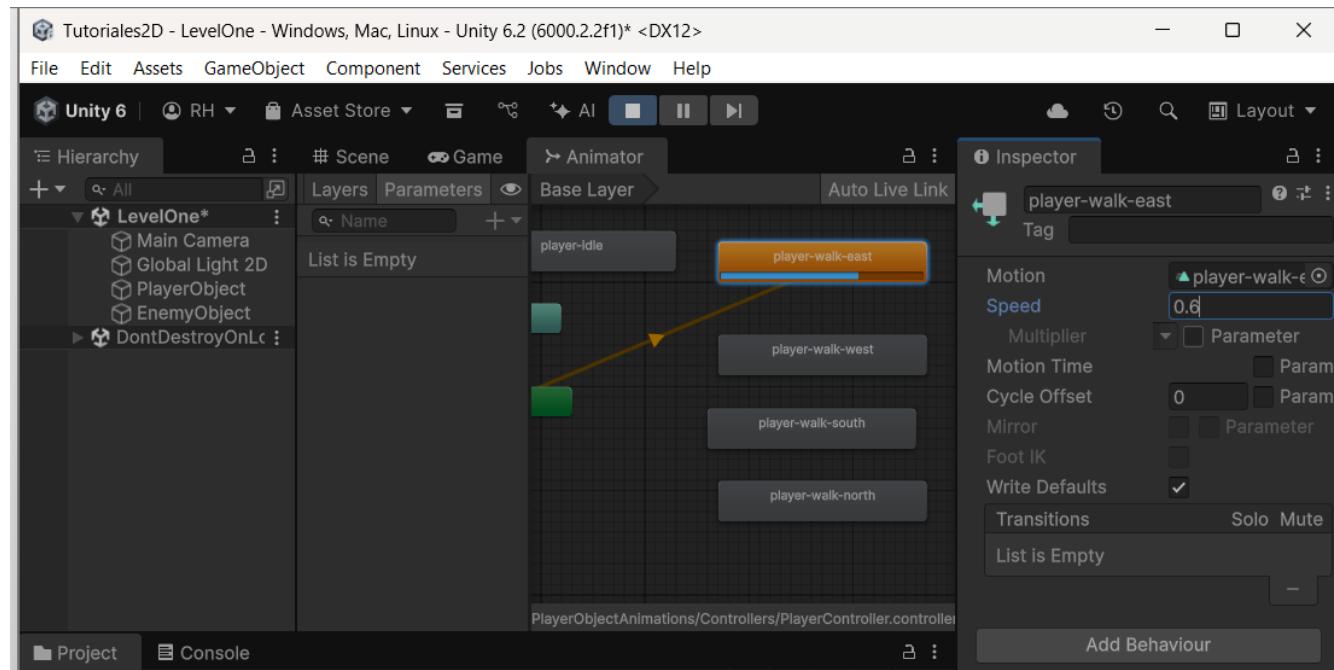
**Paso 22:** Seleccionar el objeto **Main Camera** y establecer la propiedad **size** a 1.



**Paso 23:** Presionar el botón Play para probar las animaciones.



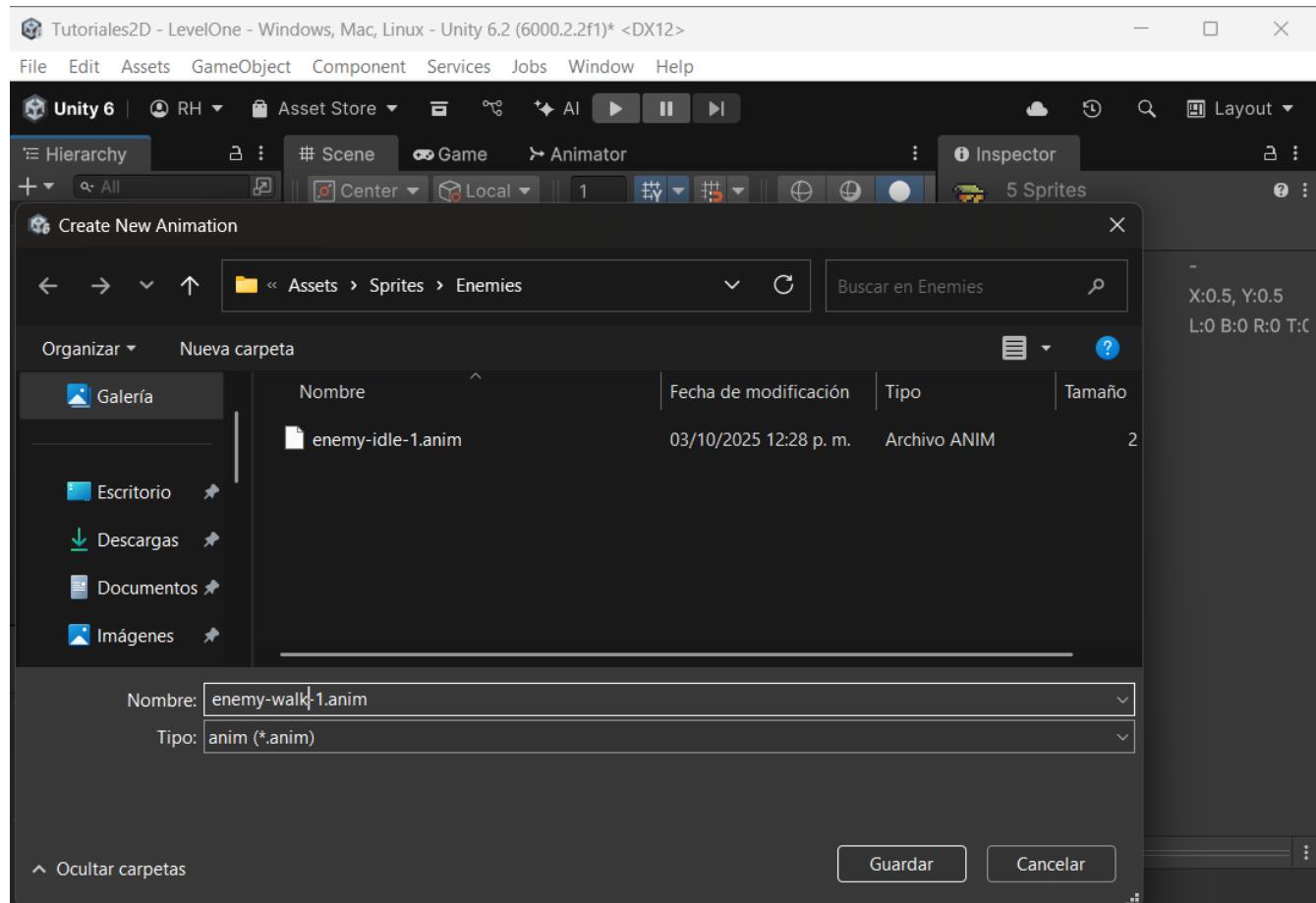
**Paso 24:** Reducir la velocidad del jugador. Abrir la ventana Animator haciendo doble clic en PlayerObject Animator. Seleccionar la animación "player-walk-east" y cambiar el valor para **Velocidad** a **0.6**.



## 🎯 DESAFÍO FINAL

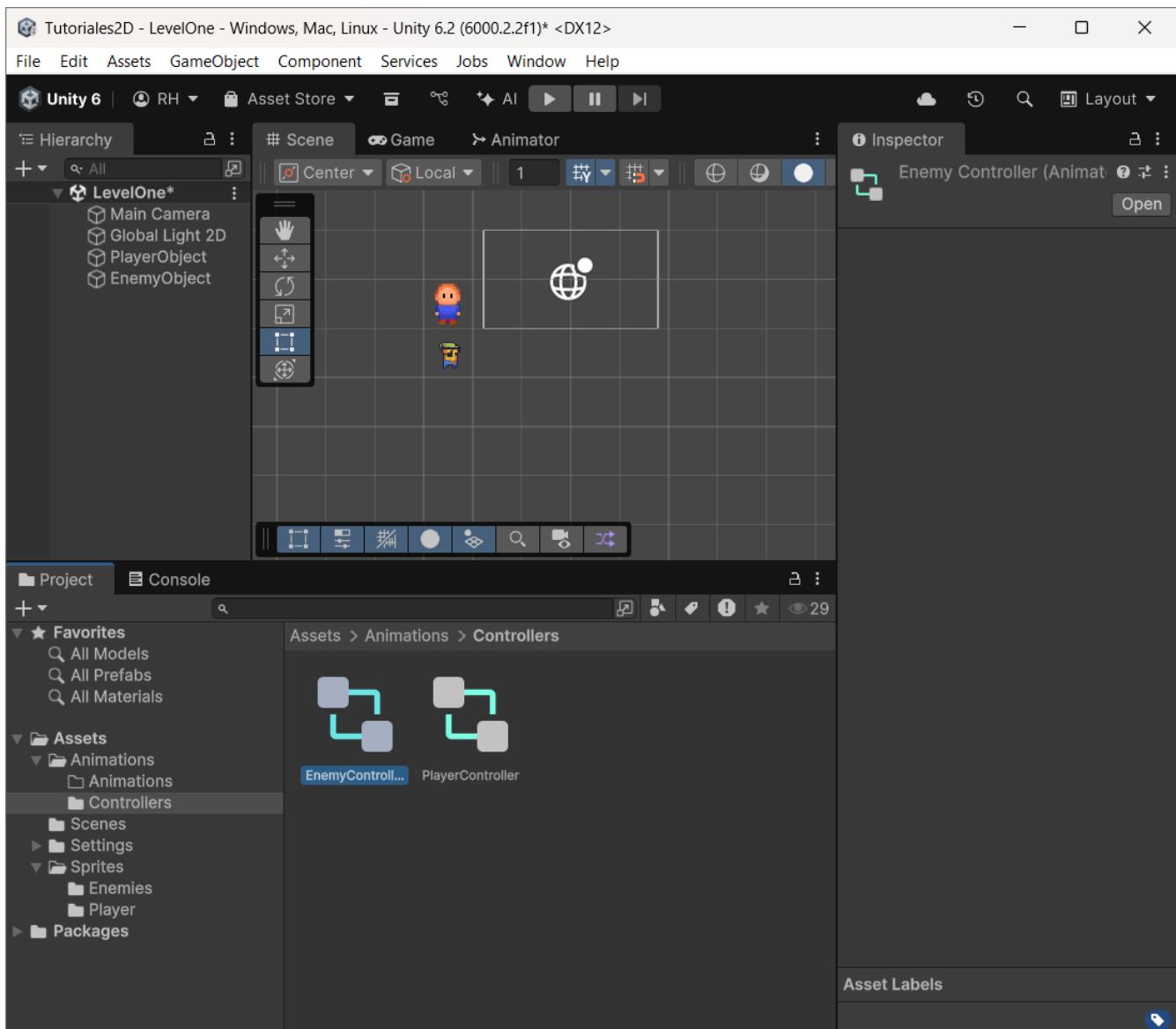
**Paso 25:** Crear y guardar las animaciones para nuestro EnemyWalk\_1 y EnemyIdle\_1. Cada una contiene cinco sprites cada una. Nombrar las animaciones:

- **enemy-walk-1**
- **enemy-idle-1**



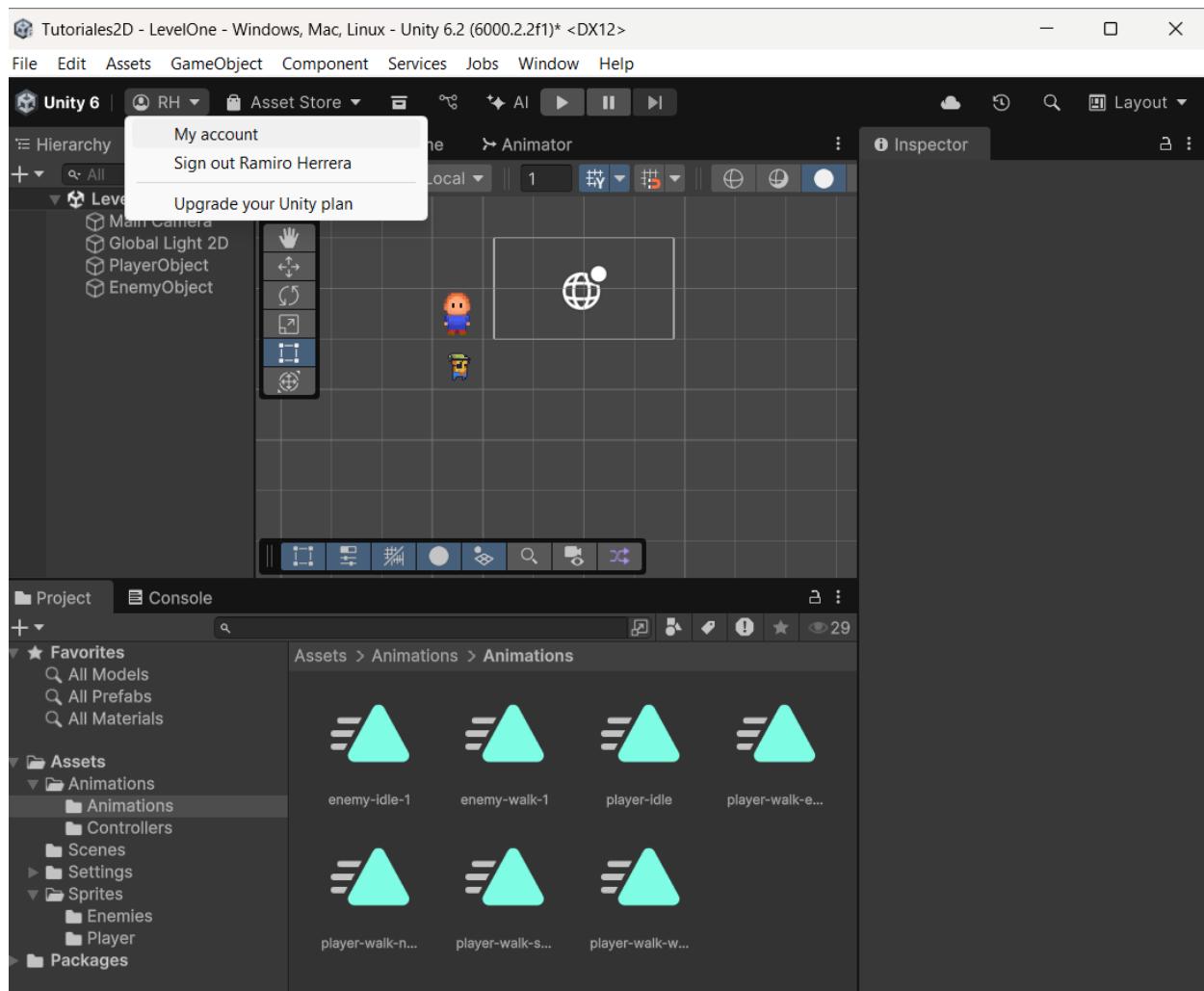
**Paso 26:** Renombrar EnemyObject Animation Controller a **EnemyController** y moverlo a la subcarpeta **Animaciones/Controladores**.

## Unidad I. Programación orientada a objetos para videojuegos

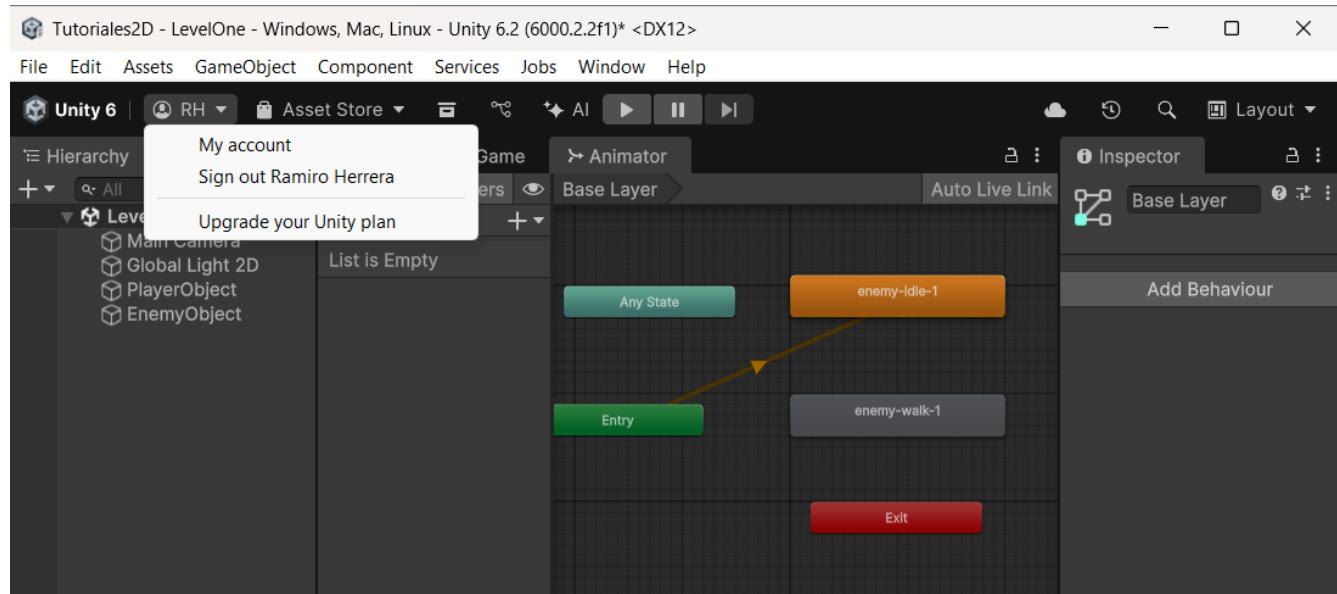


**Paso 27:** Mover las animaciones enemigas a la subcarpeta **Animaciones/Animaciones**.

## Unidad I. Programación orientada a objetos para videojuegos



**Paso 28:** Crear, copiar o importar un personaje que tenga la función de enemigo y agregar animación en ventana Animator.



## RESULTADO FINAL

