# Juego Base

## Paso 1: Creación del proyecto

Desde el *Buscador de Proyectos* , seleccionamos la opción *Games*

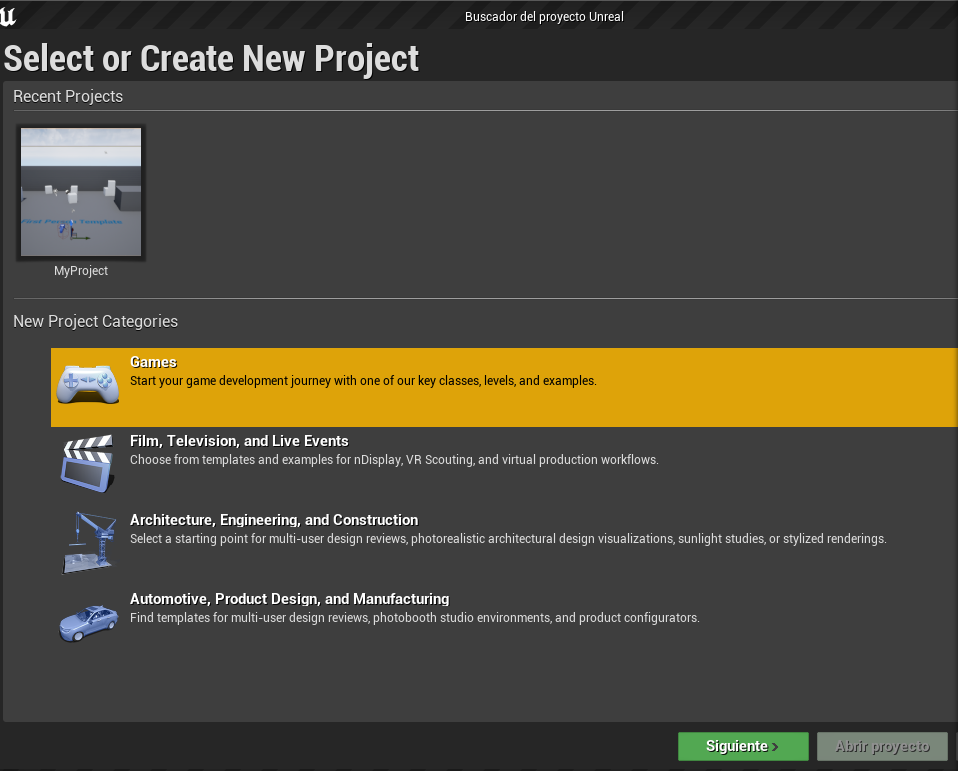


Ilustración 1: Buscador de Proyectos de Unreal 4.27

En la siguiente pantalla seleccionamos la opción Blank para comenzar con un proyecto vacío y pulsamos sobre siguiente:

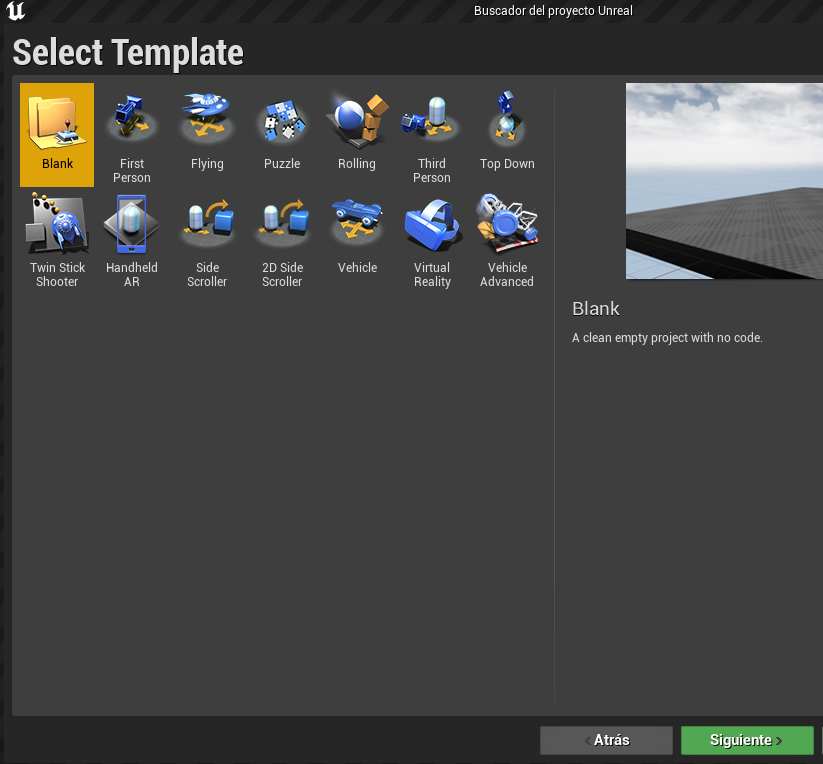


Ilustración 2: Selección de Template

En la pantalla de configuración dejamos las opciones por defecto tal como aparecen en la imagen siguiente y le damos un nombre a nuestro proyecto *GameVaquero*:



Ilustración 3: Parámetros de Proyecto

## Paso 2: Creación del nivel

Creamos un nuevo nivel seleccionando **File -> New Level del menu**

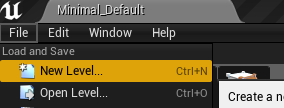


Ilustración 4: Menu File -> New level

En la nueva ventana seleccionamos el tipo **Default**

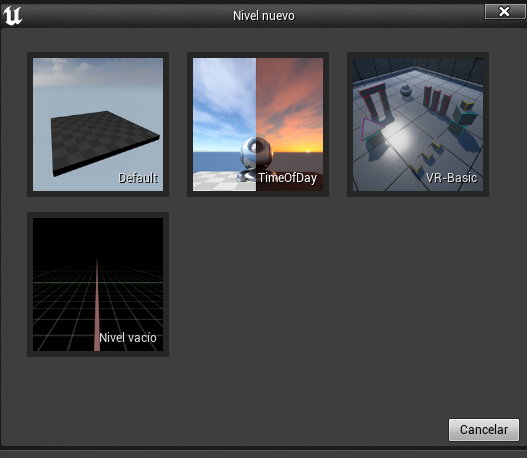


Ilustración 5: Tipos de Niveles

**Importante**: Salvamos el mapa **File -> Save Current**, lo llamaremos **NivelBase**

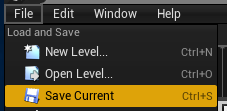


Ilustración 6: Guardar Nivel

Lo llamamos **nivel1**.

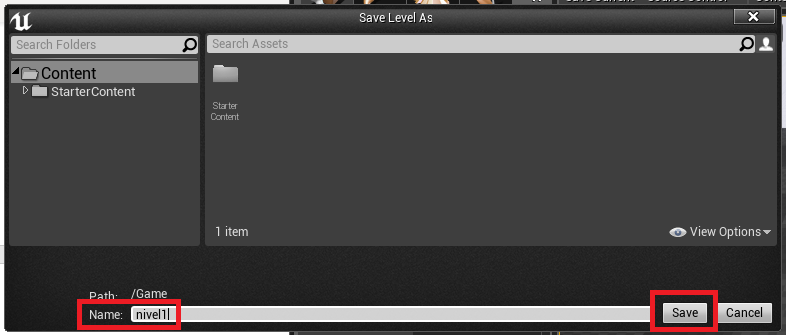


Ilustración 7: Nivel1

En el panel *World Outliner*, situado a la dcha, se muestran los elementos del nivel en forma de árbol. Vamos a eliminar todos los elementos **Floor**, **Light Source** , **Player Start** , **Sky Light , SphereReflectionCapture** , **Atmospheric Frog**  mapa (después los sustituiremos por otros).

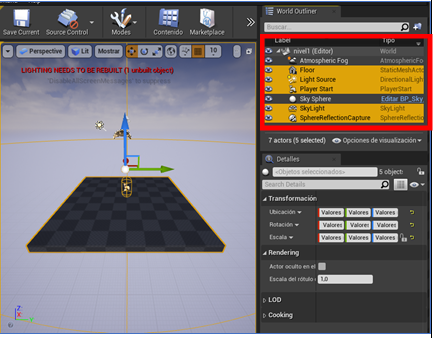


Ilustración 8: Eliminación elementos por defecto del nivel

Para crear el suelo de nuestro juego, vamos a usar un cubo que adaptaremos hasta alcanzar la forma necesaria. Para ello, en el panel *Place Actors* situado a la izda., seleccionaremos **Geometry -> box**

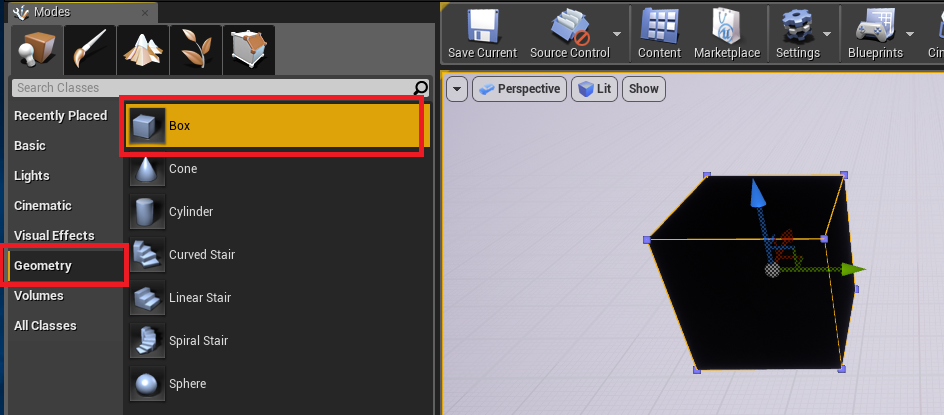


Ilustración 9: Menú Geometry - Box

Localizamos los **detalles** del elemento seleccionado (parte inferior derecha).

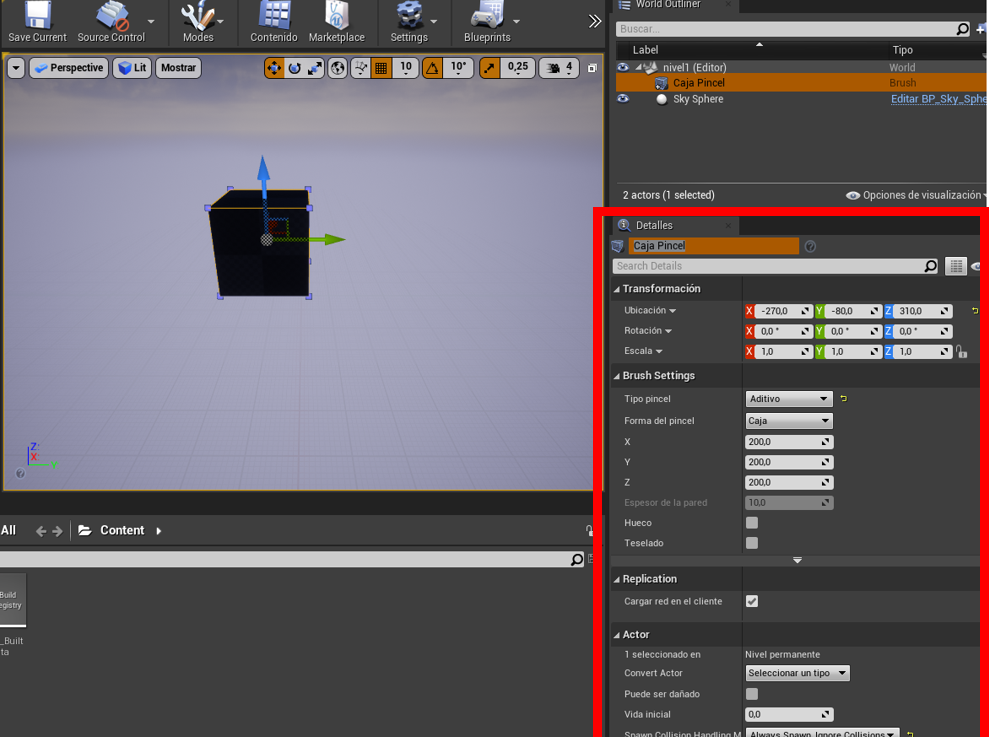


Ilustración 10: Panel detalle elemento seleccionado

Cambiamos el nombre a **Suelo**, y su tamaño a través de los valores X e Y, según se visualiza en la imagen

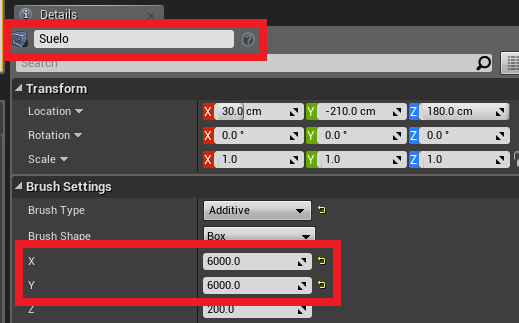


Ilustración 11: Modificación tamaño suelo

Ahora colocamos un componente **Basics -> PlayerStart** encima del **suelo**

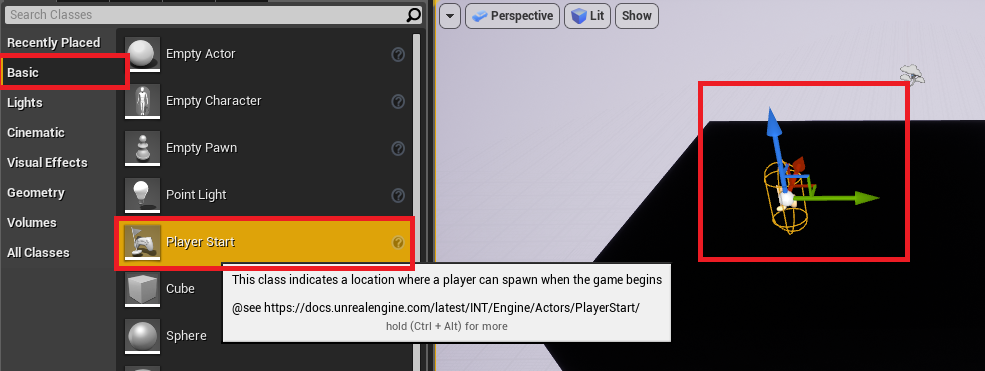


Ilustración 12: Añadir Player Start

Finalmente agregamos un componente **Lights -> SkyLight** (lo podemos soltar en cualquier zona del mapa).

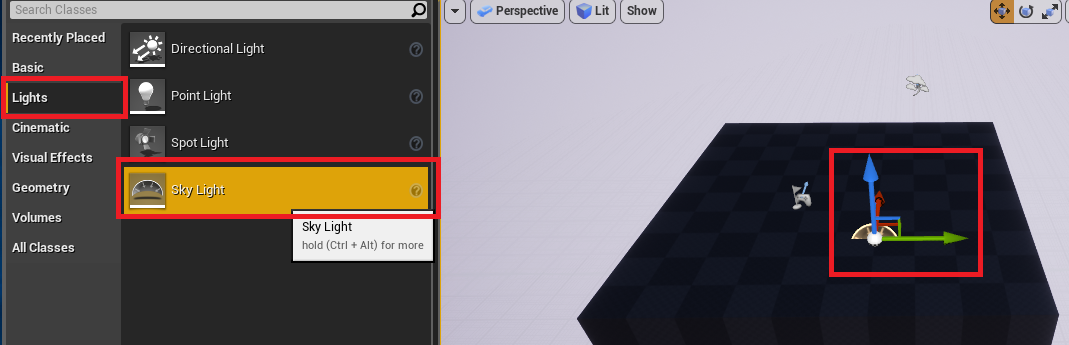


Ilustración 13: Añadir Luz del Cielo

Aumentamos la **intensidad** del **Sky Light** a 2, ya que por defecto es bastante baja. Modificamos la propiedad **Sky Distance** para minimizar las zonas sin luz.

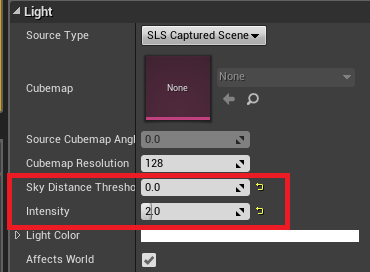


Ilustración 14: Propiedades Sky Light

## Paso 3: Importar recursos gráficos para el Jugador

En el panel *Buscador de contenidos*, situado en la parte inferior del editor, desplegamos la estructura de ficheros del proyecto con la opción *Show/Hide Sources Panel y* nos situamos sobre la carpeta **StarterContent.**

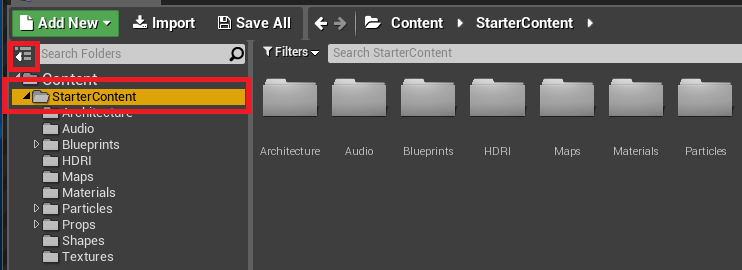


Ilustración 15: Panel Buscador de Contenidos

Creamos la carpeta **Jugador** (botón derecho, new Folder)

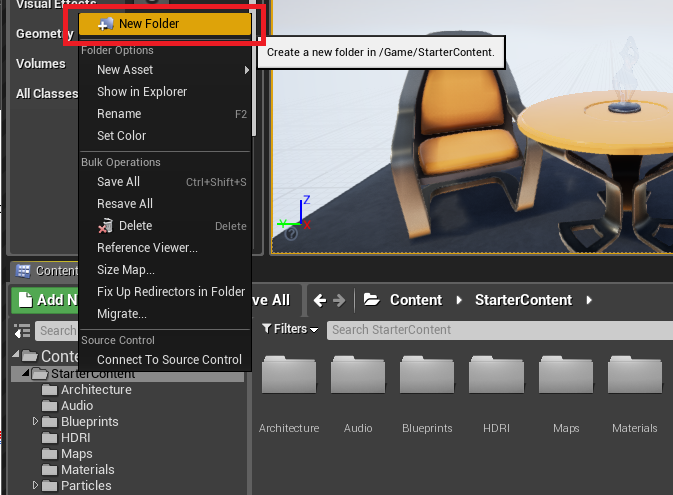


Ilustración 16: Menú New Folder

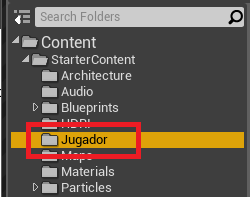


Ilustración 17: Carpeta Jugador

Descargamos y descomprimimos la carpeta *Nuevo res* del campus virtual con los recursos básicos y la descomprimimos en nuestro equipo.

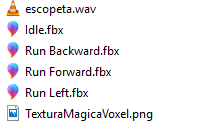
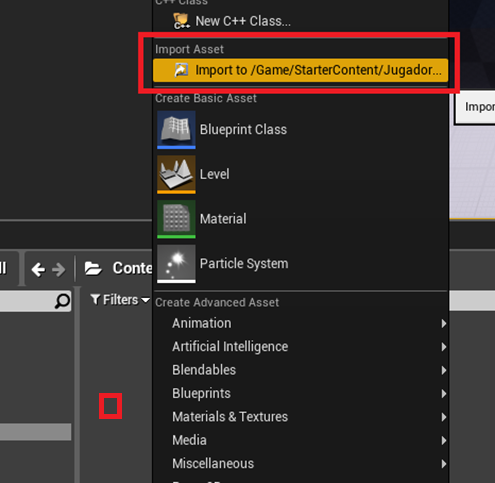


Ilustración 18: Contenido carpeta de recursos

### Importación de malla del jugador

Vamos a comenzar importando la **Malla(Con esqueleto, Skeletal Mesh)** y la **Animación de Idle** (ambas están contenidas en un mismo **FBX**).

Entramos en la carpeta **Jugador** , en la zona del contenido de la carpeta damos al botón derecho e **import to**



Esquema de los ficheros que van a componer el 3D de nuestro personaje

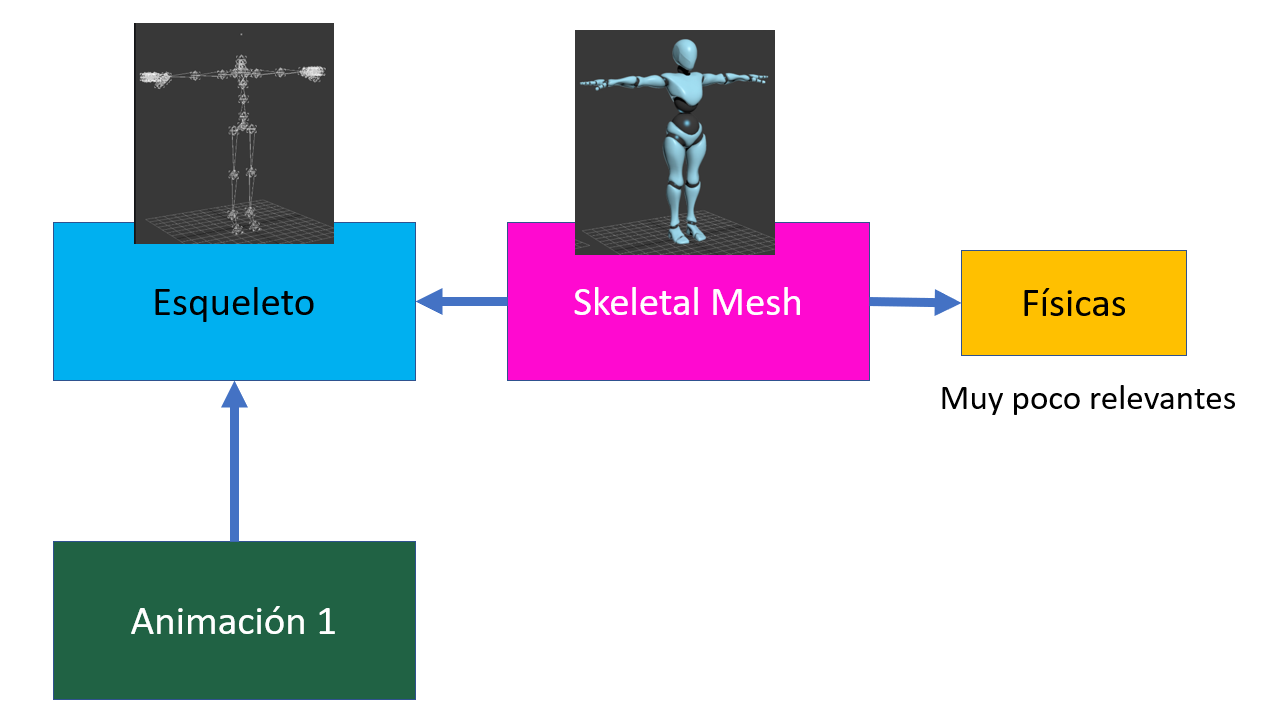


Ilustración 19: Esquema relación esqueleto - malla - animaciones

Seleccionamos el fichero **idle.fbx** Es la primera animación que incluimos, aun **NO tenemos** ni la **malla** ni el **esqueleto**

Se trata de una animación basada en una malla con esqueleto (hay que marcar la casilla **Skeletal mesh**)

Aun no tenemos ninguna malla para el jugador, para importarla marcamos la casilla **Import Mesh.** En la opción de **Material Import Method**seleccionamos *Create New Instanced* *Materials*, con Base *None*.

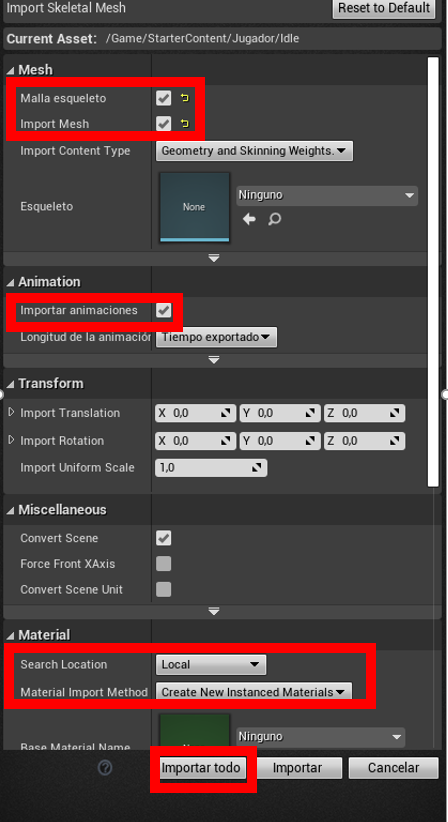


Ilustración 20: Panel Importacion fbx

Nota: si aparecen warnings los ignoramos, los importa de todos modos.

Ya tenemos (por orden: la malla 3d (que admite esqueleto), la animación de Idle, el esqueleto de físicas y el esqueleto (Los dos últimos son ficheros “complementarios” a la malla).



Ilustración 21: Carpeta Jugador

Vamos a comenzar renombrando el fichero de la malla, a **jugador\_malla** (click derecho -> rename).



Ilustración 22: Carpeta Jugador

También el esqueleto a **jugador\_esqueleto**.

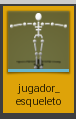


Ilustración 23: Esqueleto de jugador

Vemos que el jugador no tiene color, vamos a tener que crear un material, todos los modelos creados con la herramienta MagicaVoxel comparten el mismo material (esto es algo bastante poco común en el modelado 3D).

Importamos el fichero **TexturaMagicaVoxel.png** (de la misma forma que importamos el fichero anterior).

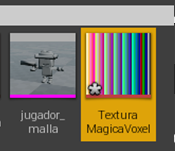
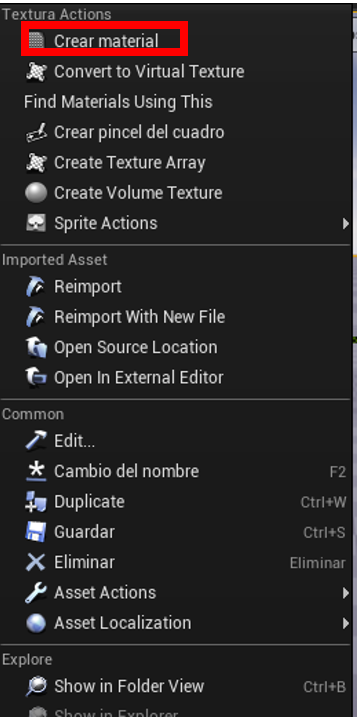
****

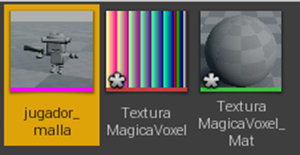
Ilustración 24: Textura Magical Voxel

Para crear el material hacemos click derecho sobre TexturaMagicaVoxel y **Create Material**.



El fichero que se acaba de crear es un material, puede ser asignado a cualquier malla 3D, un material es algo así como una “super textura”, no solo tiene una imagen, sino que puede definir multitud de propiedades más (refracción, emisión de luz, rugosidad, etc.) Aunque en nuestro caso es una mera textura.

Abrimos **jugador\_malla** (doble click), le asignamos el material que acabamos de crear.



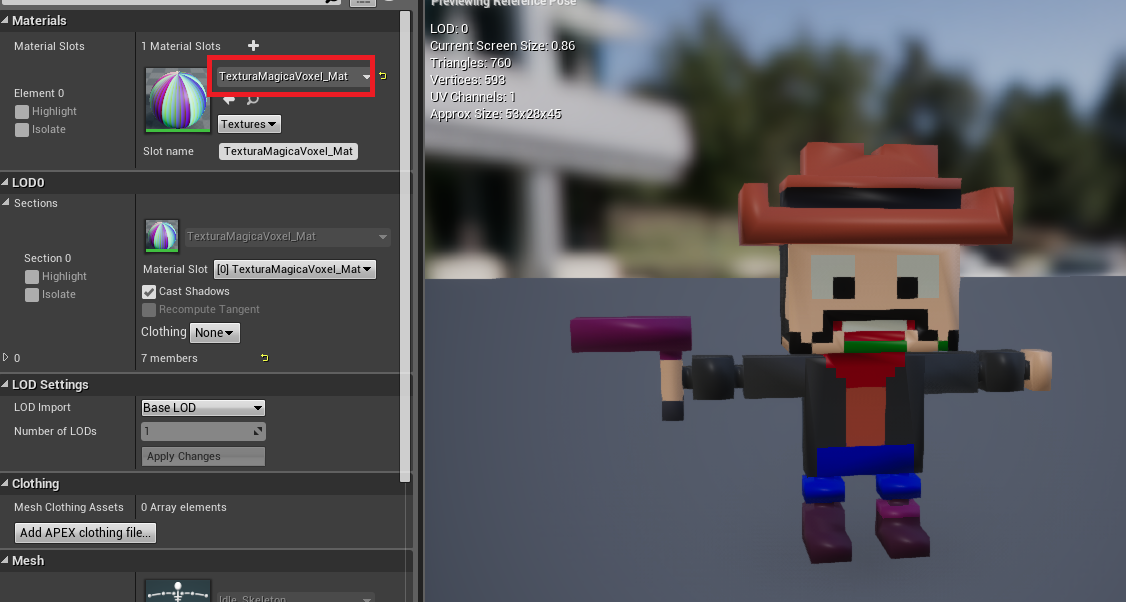
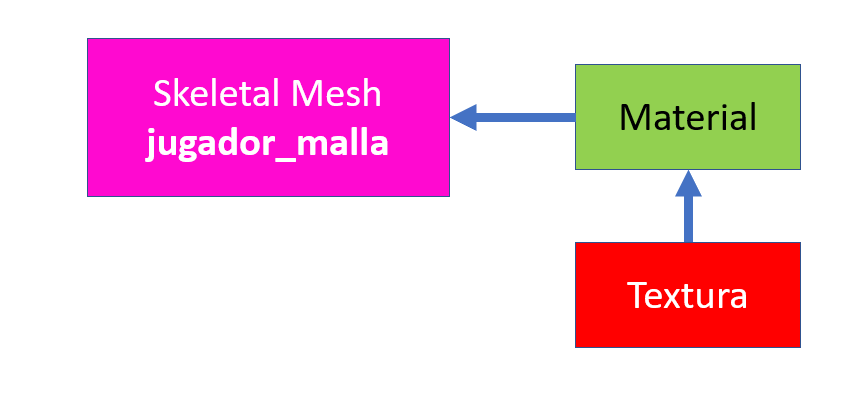
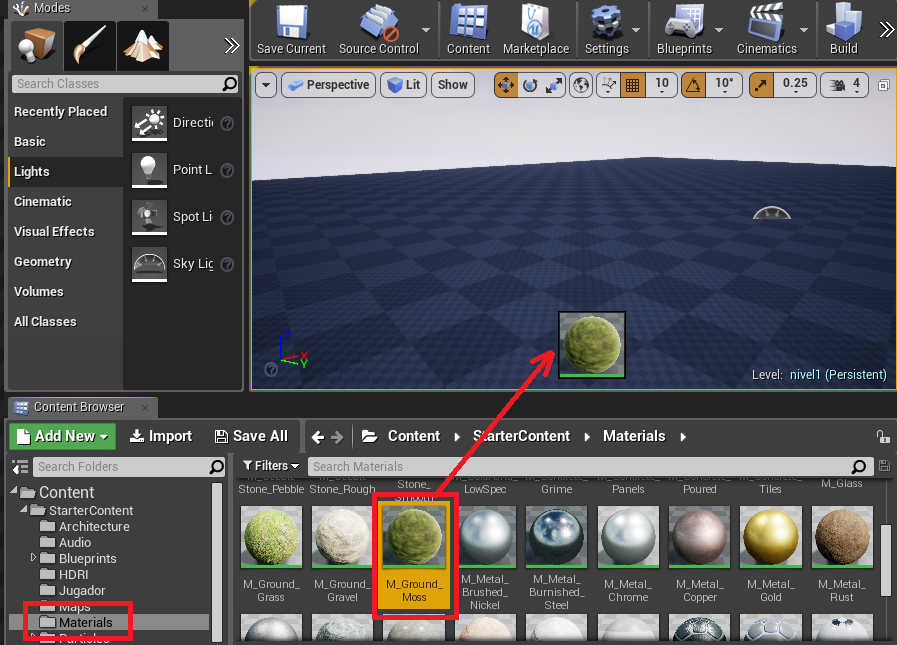


Ilustración 25: Asociación Textura y Material a la malla

  
Dentro de *Start Content* n, en la carpeta de *Materials* , nos encontramos con numerosos materiales



## Importar más animaciones

Para importar más animaciones de este jugador ya no necesitaríamos ni **la malla** ni **el esqueleto.**

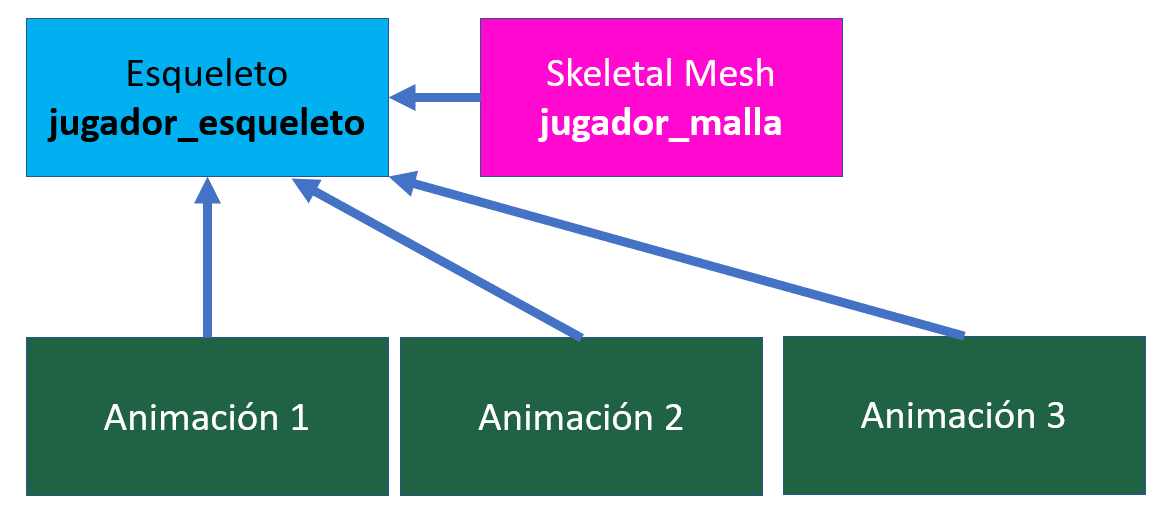


Ilustración 26: esquema animaciones - esqueleto - malla

Repetimos el proceso de **Import To,** seleccionando el fichero **Run Backward.fbx**

Debemos desmarcar la opción **Import Mesh** yseleccionar el **Skeleton** que va a usar como base la animación

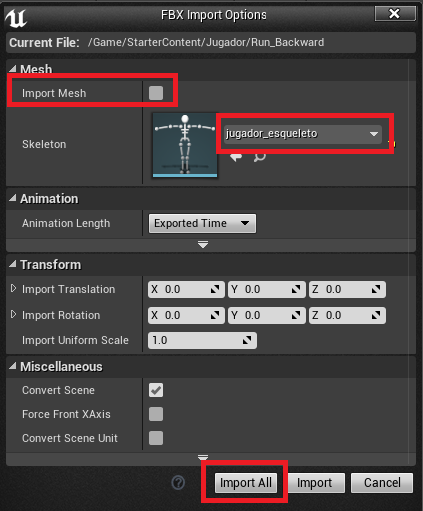


Ilustración 27: Importación animación sin malla

Con esto ya tenemos la segunda animación importada.

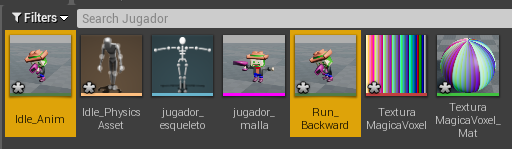
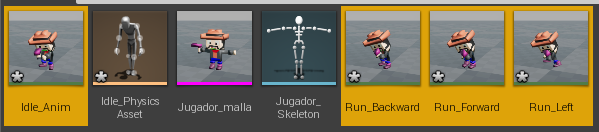
  
  
Para verlas en funcionamiento podemos hacer doble click sobre ellas.



Ilustración 28: Animación

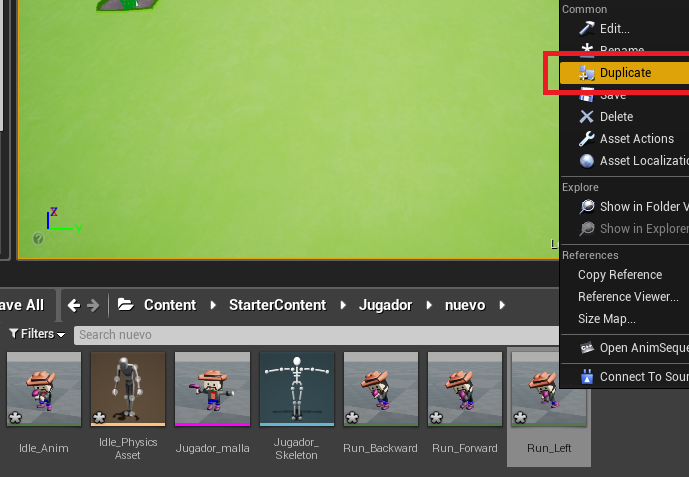
Repetimos el proceso de la misma forma para **Run** **Forward.fbx** , **Run Left.fbx**

Al finalizar deberíamos tener 4 animaciones, tal y como se muestra en la siguiente captura.

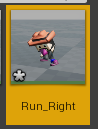


Nos falta la animación de **Run\_Right**, pero vamos a generarla copiando la de corriendo hacia la izquierda e invirtiéndola.

Botón derecho sobre la animación y **Duplicate**.

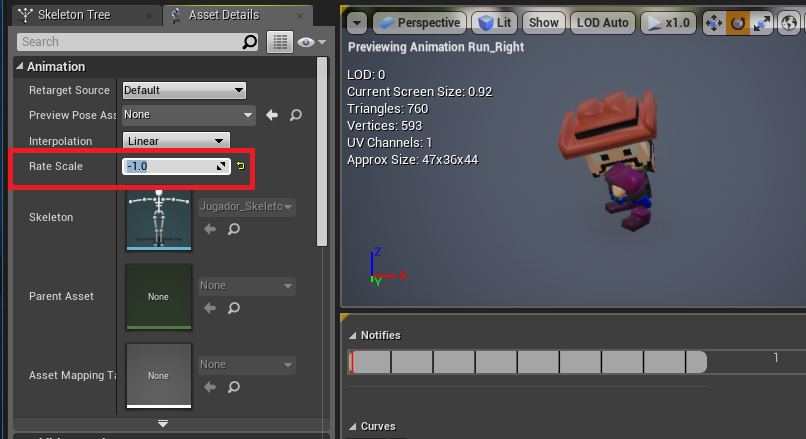


A la nueva la llamaremos **Run\_Right**.



Entramos dentro de la animación haciendo doble click.

Invertimos la animación invirtiendo el **Rate Scale.**



Salvamos y salimos.

