

# Introducción al desarrollo de videojuegos 2D con Godot Engine (Parte I)

*Webinar – 26 de Marzo de 2020*

---



# Temario

---

- Introducción al concepto de motor de videojuego.
- Presentación de Godot Engine.
- Enfoque de desarrollo en Godot Engine.
- Estudio de caso práctico: Arkanoid Godot:
  - ✓ *Nodos y Escenas.*
  - ✓ *Jerarquías y Grupos.*
  - ✓ *Señales.*
  - ✓ *Introducción al Scripting.*

# Repositorio de recursos

---

Disponible en GitHub:

- Presentación.
- Proyecto de caso práctico inicial.
- Proyecto de caso práctico final.
- Assets.



<https://github.com/gastoncaminiti/WebinarGodotIEEEGTC2020>

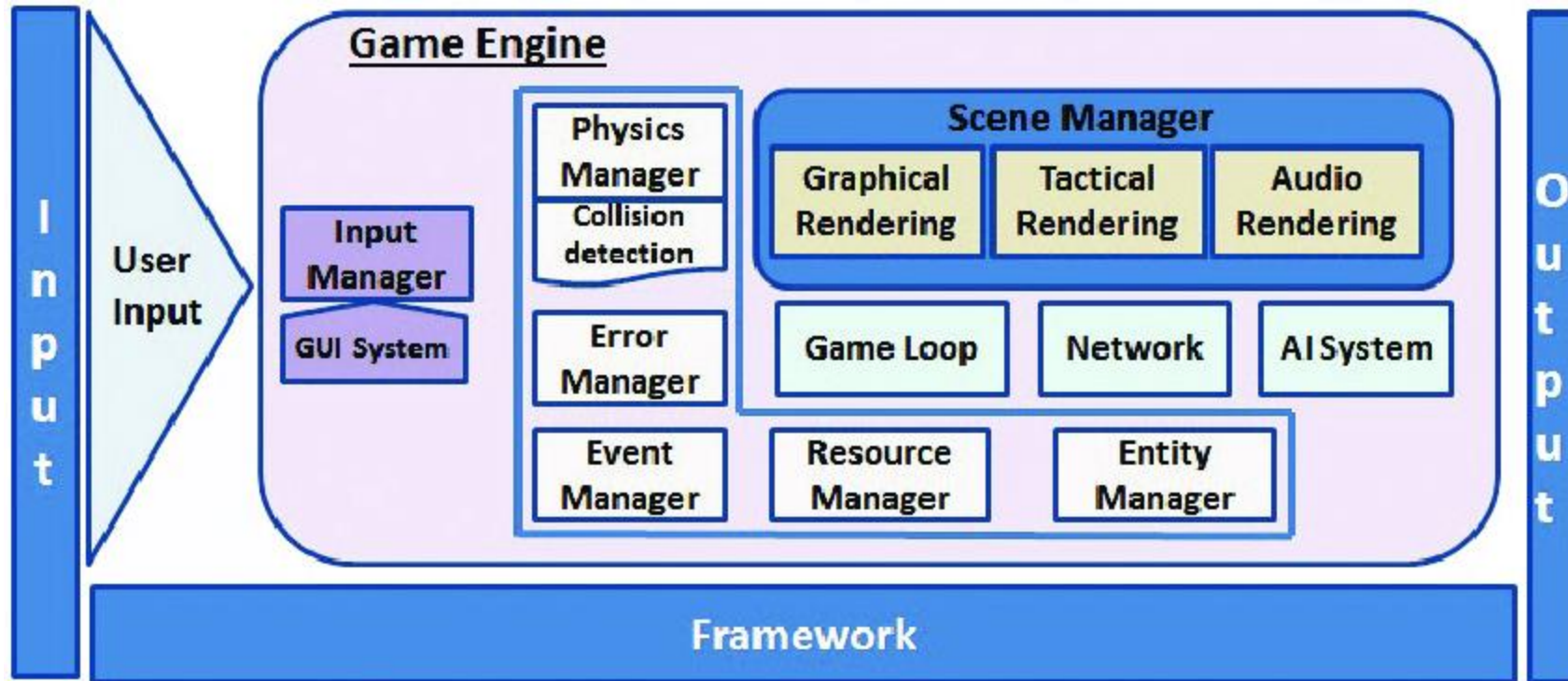
# Motor de Videojuegos

---

“Los motores de videojuegos son una nueva forma de desarrollar juegos de alta calidad de forma fácil y rápida sin necesidad de conocimientos intensivos de programación y recursos computacionales.”

Zarrad, A. (2018)

# Arquitectura del Game Engine



Zarrad, A. (2018) Game engine solutions

# La elección

---

*Analizar cómo el Game Engine favorece el desarrollo y publicación de nuestro videojuego.*



# Presentación de Godot Engine

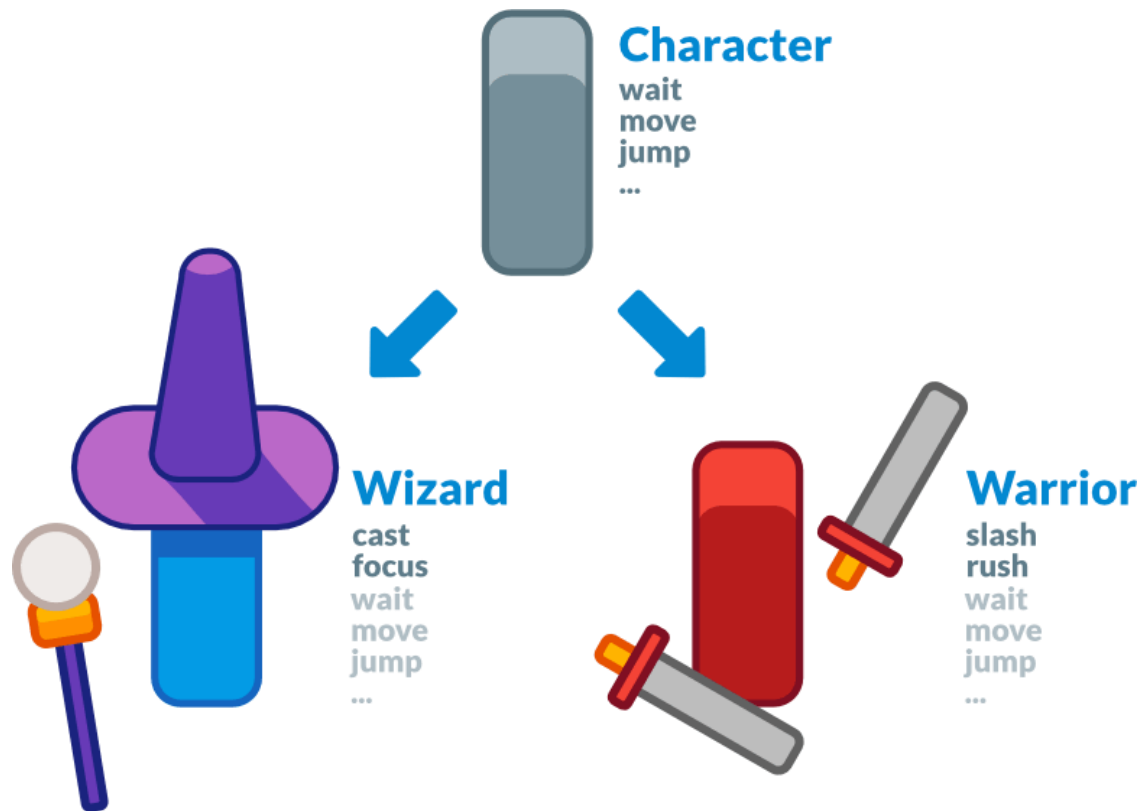
---

- Licencia MIT – Software Libre
- Multiplataforma: Editor y Publicación de Videojuegos.
- Soporte 2D y 3D.
- Lenguaje de Scripting: GDScript.
- Diseño orientado a objetos.
- Extendible por Plugins y GDNative.

Source: <https://docs.godotengine.org/es/latest/about/faq.html#doc-faq>



# Filosofía de Diseño y Enfoque de desarrollo



“Godot adopta un diseño orientado a objetos desde el núcleo con su sistema flexible de escenas y jerarquía de nodos.”

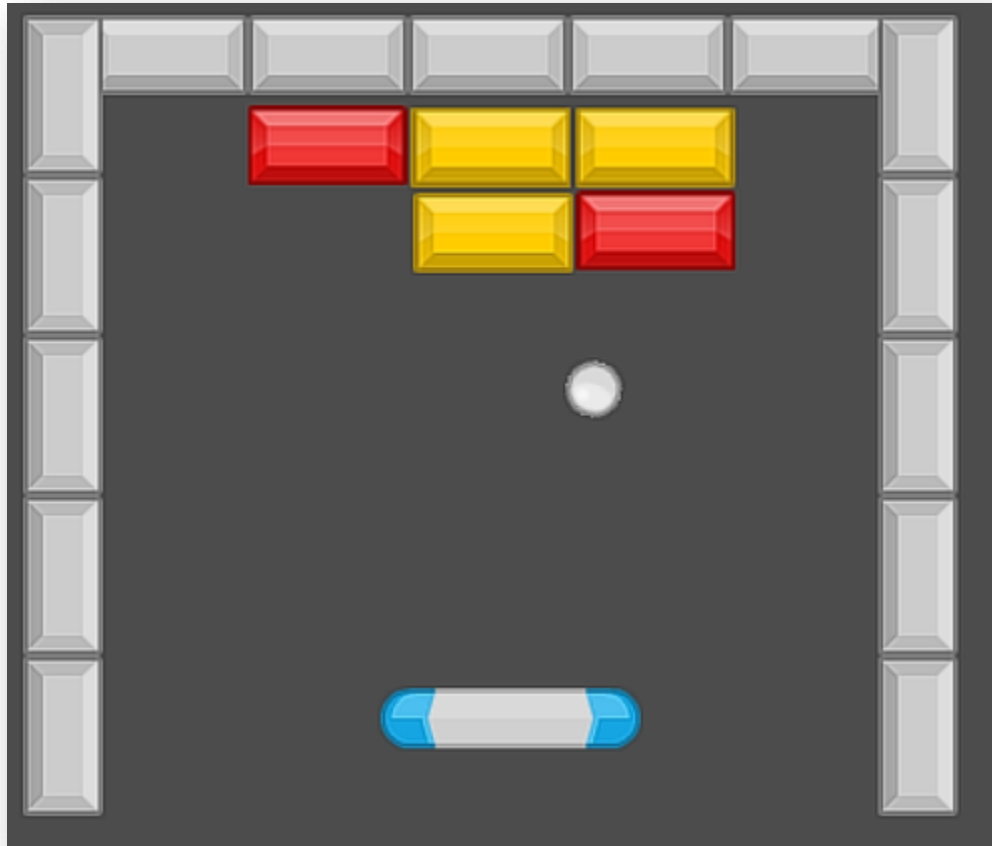


Source: [https://docs.godotengine.org/es/latest/getting\\_started/step\\_by\\_step/godot\\_design\\_philosophy.html](https://docs.godotengine.org/es/latest/getting_started/step_by_step/godot_design_philosophy.html)



# Estudio de caso práctico: Arkanoid Godot

---



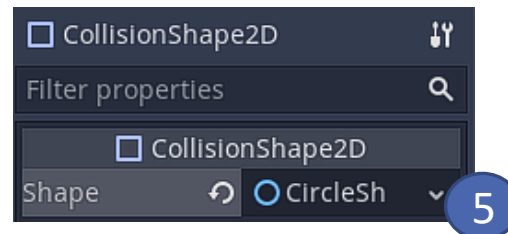
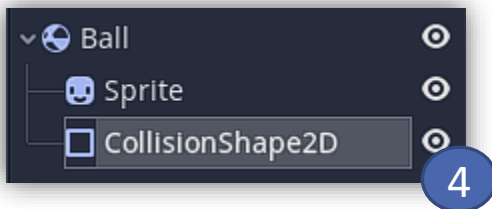
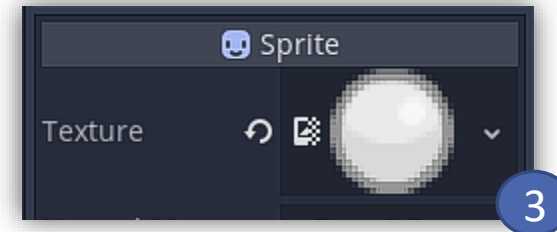
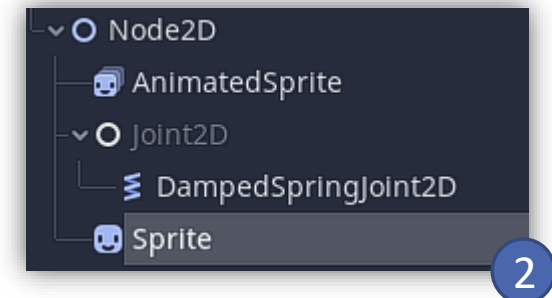
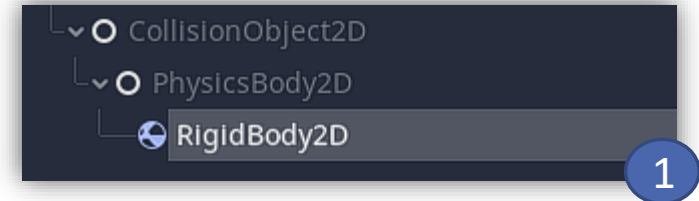
Realizaremos a continuación el desarrollo de elementos del videojuego Arkanoid partiendo del proyecto inicial en el repositorio.

`projects/Arkanoid Godot (Inicio)`



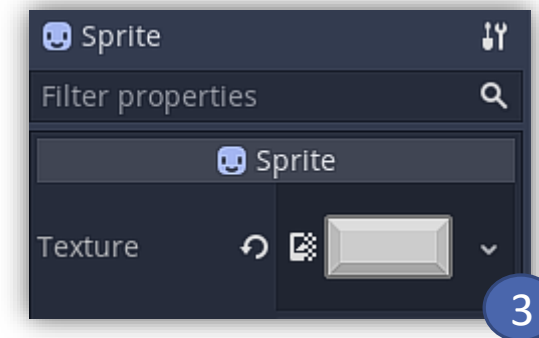
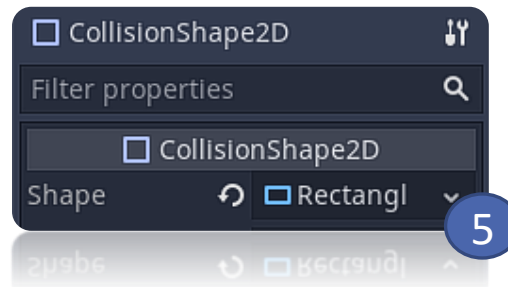
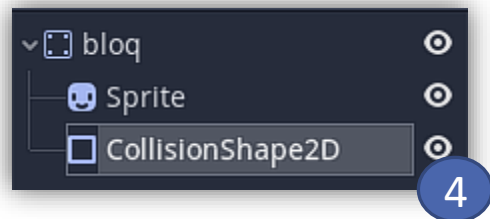
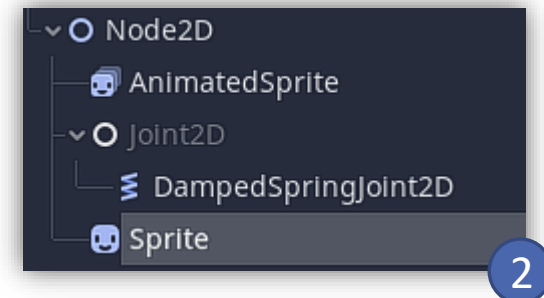
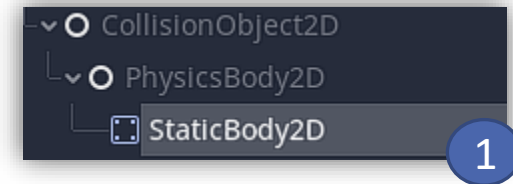
# Creando Escena: Pelota

1. Crear una escena con nodo **RigidBody2D**.
2. Crear un nodo hijo **Sprite**.
3. Asignar textura a la propiedad del nodo Sprite.
4. Crear un nodo hijo **CollisionShape2D**.
5. Asignar forma de Circulo al CollisionShape2D.



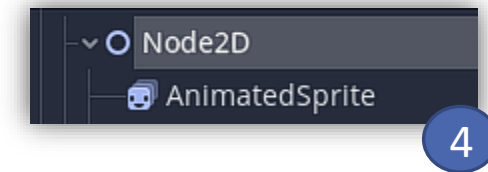
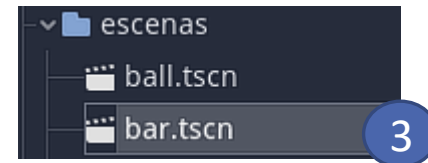
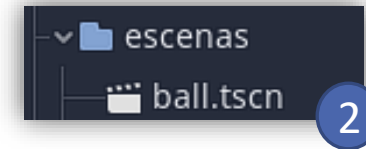
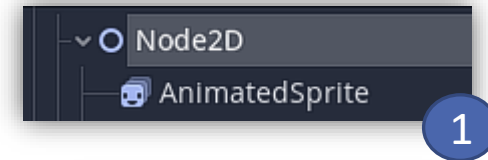
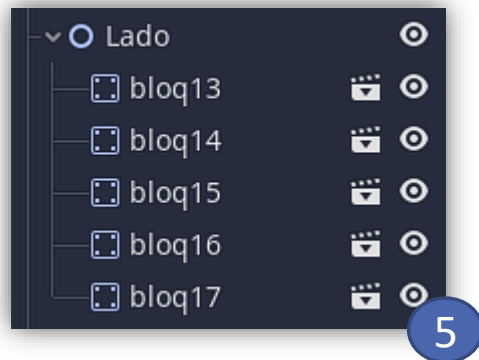
# Creando Escena: Bloque

1. Crear una escena con nodo **StaticBody2D**.
2. Crear un nodo hijo **Sprite**.
3. Asignar textura a la propiedad del nodo Sprite.
4. Crear un nodo hijo **CollisionShape2D**.
5. Asignar forma de Rectángulo al CollisionShape2D.



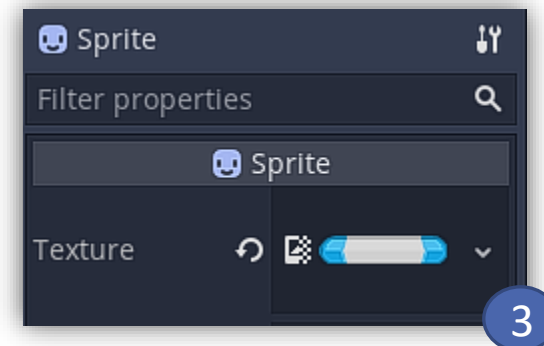
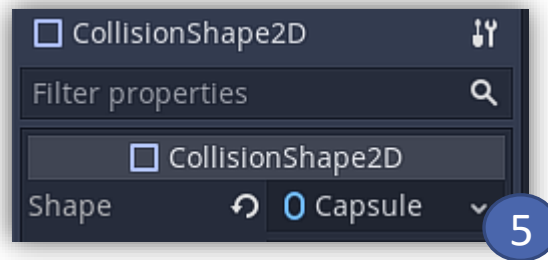
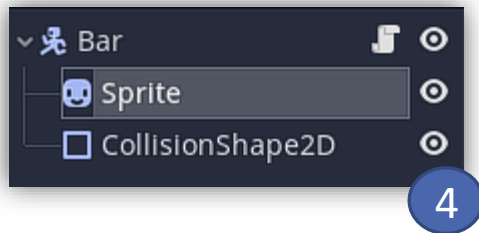
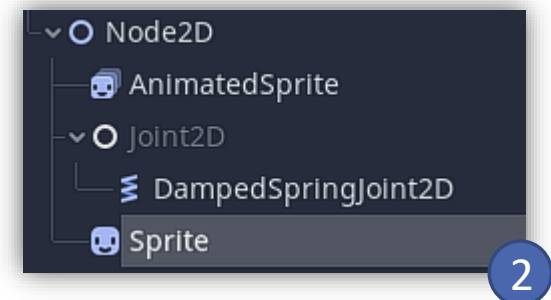
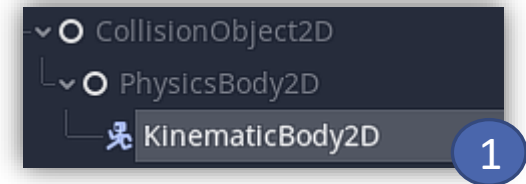
# Creando Escena: Nivel

1. Crear una escena con nodo **Node2D**.
2. Arrastrar escena de la pelota.
3. Arrastrar escena del bloque.
4. Crear un nodo hijo **Node2D**.
5. Duplicar cinco (5) escenas bloque dentro.



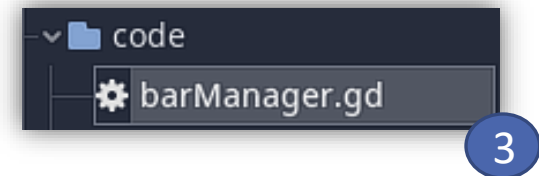
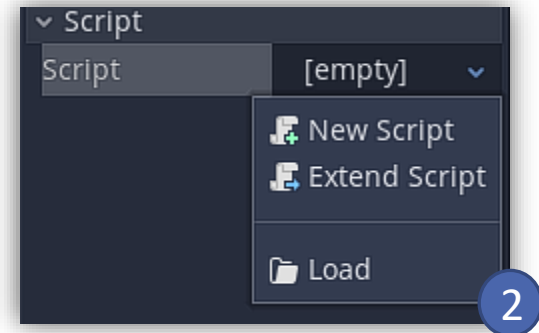
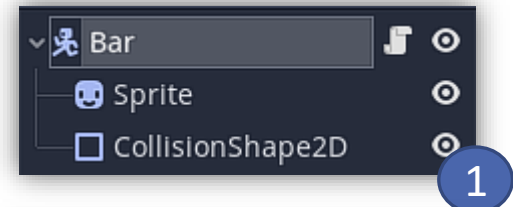
# Creando Escena: Barra

1. Crear una escena con nodo **KinematicBody2D**.
2. Crear un nodo hijo **Sprite**.
3. Asignar textura a la propiedad del nodo Sprite.
4. Crear un nodo hijo **CollisionShape2D**.
5. Asignar forma de Capsula al CollisionShape2D.



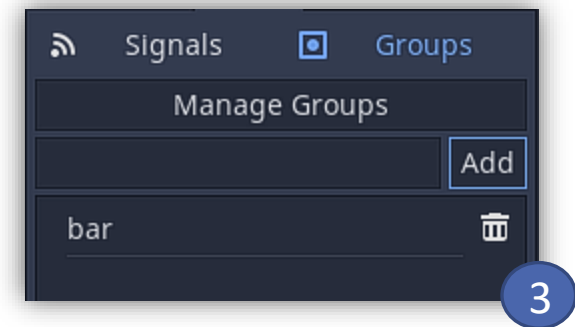
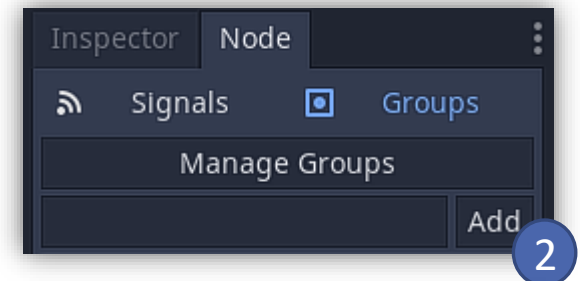
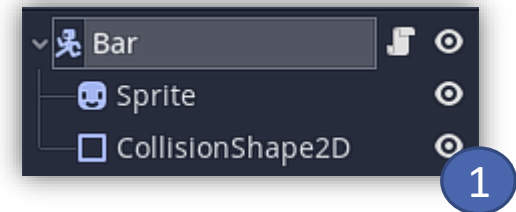
# Asignando Script a la barra

1. Seleccionar nodo padre **KinematicBody2D**.
2. Presionar “Load” en la propiedad Script.
3. Seleccionar el script barManager.gd.



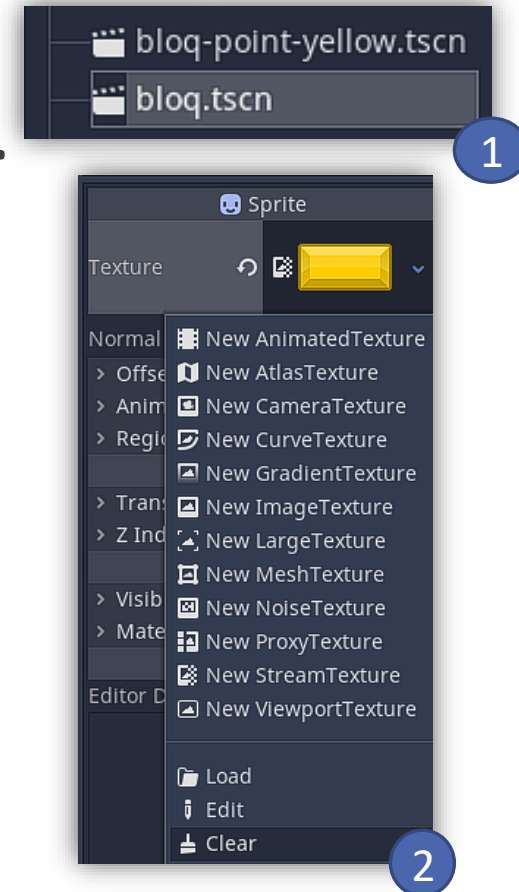
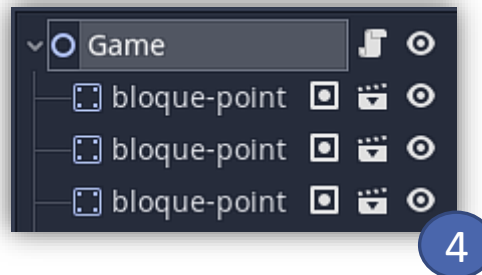
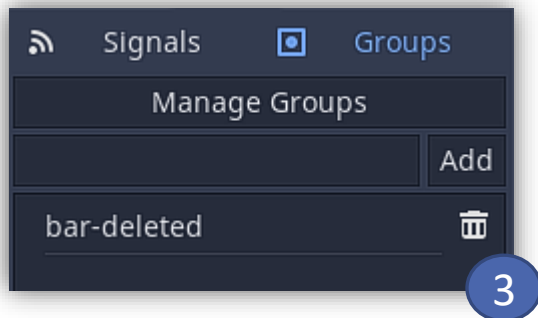
# Asignando Grupo a la barra

1. Seleccionar nodo padre **KinematicBody2D**.
2. Seleccionar pestaña Node -> Groups.
3. Ingresar el nombre del grupo.



# Creando Escena: Bloque a eliminar

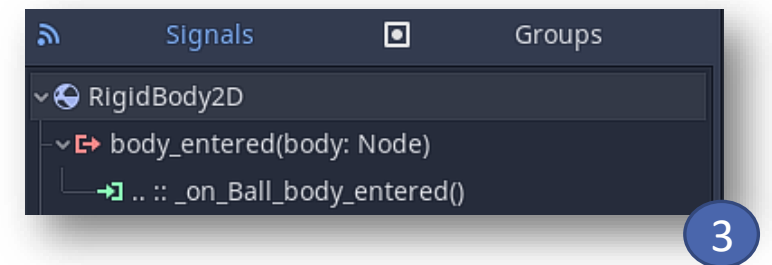
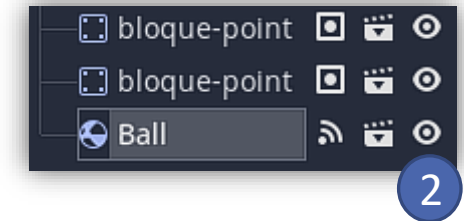
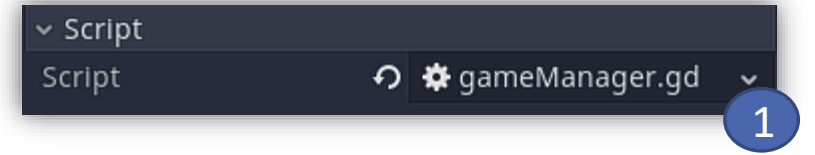
1. Duplicar la escena bloque.
2. Limpiar textura del Sprite existente y asignar nueva.
3. Asignar grupo al nodo padre del bloque a eliminar.
4. Insertar la escena del bloque en la escena de nivel.





# Ensamblando el nivel

1. Asignar script al nodo padre del nivel.
2. Seleccionar pelota en el nivel.
3. Conectar señal **body\_entered()** de la pelota con el nivel.



# Recursos adicionales

---

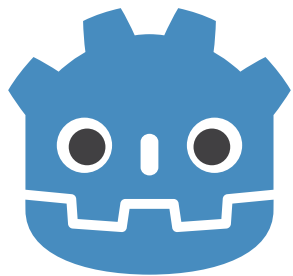
- [Documentación de Godot Engine \(Español\)](#)
- [Godot Engine Series](#)
- [Zamna Atlante – Open 2D Platform game en Godot 3.1](#)
- [Verdandi Norse Legends](#)
- [IEEE GTC Argentina](#)
- [Godot regional communities team](#)



¡Gracias!  
#quedatenecasa

*Prof. Gastón Caminiti. CC by 4.0*

---



**GODOT**



**IEEE**

Computational Intelligence Society

Games Technical  
Committee