

Intro Desarrollo de Videojuegos 2024 - UNQ v.I

1 | Concepto de juego y videojuego





Julieta Lombardelli

Lic. Prof. en diseño Multimedial

Doctoranda en Artes UNLP

Artista y docente multimedia

Colaboradores



**Nahuel
Moscatelli**

Estudiante de la
tecnicatura

Desarrollador de
juegos y apps en
Godot

This is a reference.
All craftdwarship is
of the highest
quality.



**Nahuel
Zanier**

Estudiante de
la tecnicatura

Dibujante
(algo)

Fan de old
school d&d



**Fabian
Frangella**

Estudiante de
la tecnicatura

Aqui lo que
desees



**Matías
Janikow**

Estudiante de
la tecnicatura

Desarrollador
Full stack

Tomador Serial
de mate

Antes de comenzar...Por favor completar este breve form

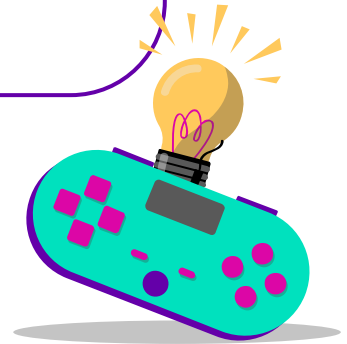
<https://forms.gle/H4DUueGY92dgoFPy9>

Desarrollo de videojuegos

Como dice Cliff Bleszinski, antiguo diseñador jefe de niveles de Epic Games:

"El desarrollo de juegos puede compararse con la construcción de un coche.

Tienes todas estas partes diferentes creadas por gente con talento -programación, modelado, sonido y arte- y en algún momento, el duro trabajo de todos en un coche se une, y los neumáticos salen a la carretera. En un juego, el trabajo de todos se mantiene unido por los niveles que utilizan todo eso, y más vale que sean excepcionales o el juego se tambalea".



Pautas de cursada

Desafíos x clase - sincrónica

Cada clase se desarrollará un desafío, que debe entregarse al final de la misma

1



Proyecto BOSS - asincrónica

En forma asincrónica se diseñará en equipos un videojuego, completo, que se entrega a final de cursada con 2 checkpoints

2



La cursada en juego



Puntos

Cada clase recibirán puntos por los desafíos logrados

Individual

La puntuación sincrónica es individual, por día.

Grupal

Al iniciar el proyecto grupal se establecen puntos para el equipo

Valores

DESAFÍO ALCANZADO
CÓDIGO
PARTICIPACIÓN
BONUS

Nuestro Repo

Juegos de ayer y hoy recomendados por uds para uds

<https://es.padlet.com/yelomba/idvung>

Bosses de cursadas pasadas

https://intropv.github.io/docs/proyectos_previos.html

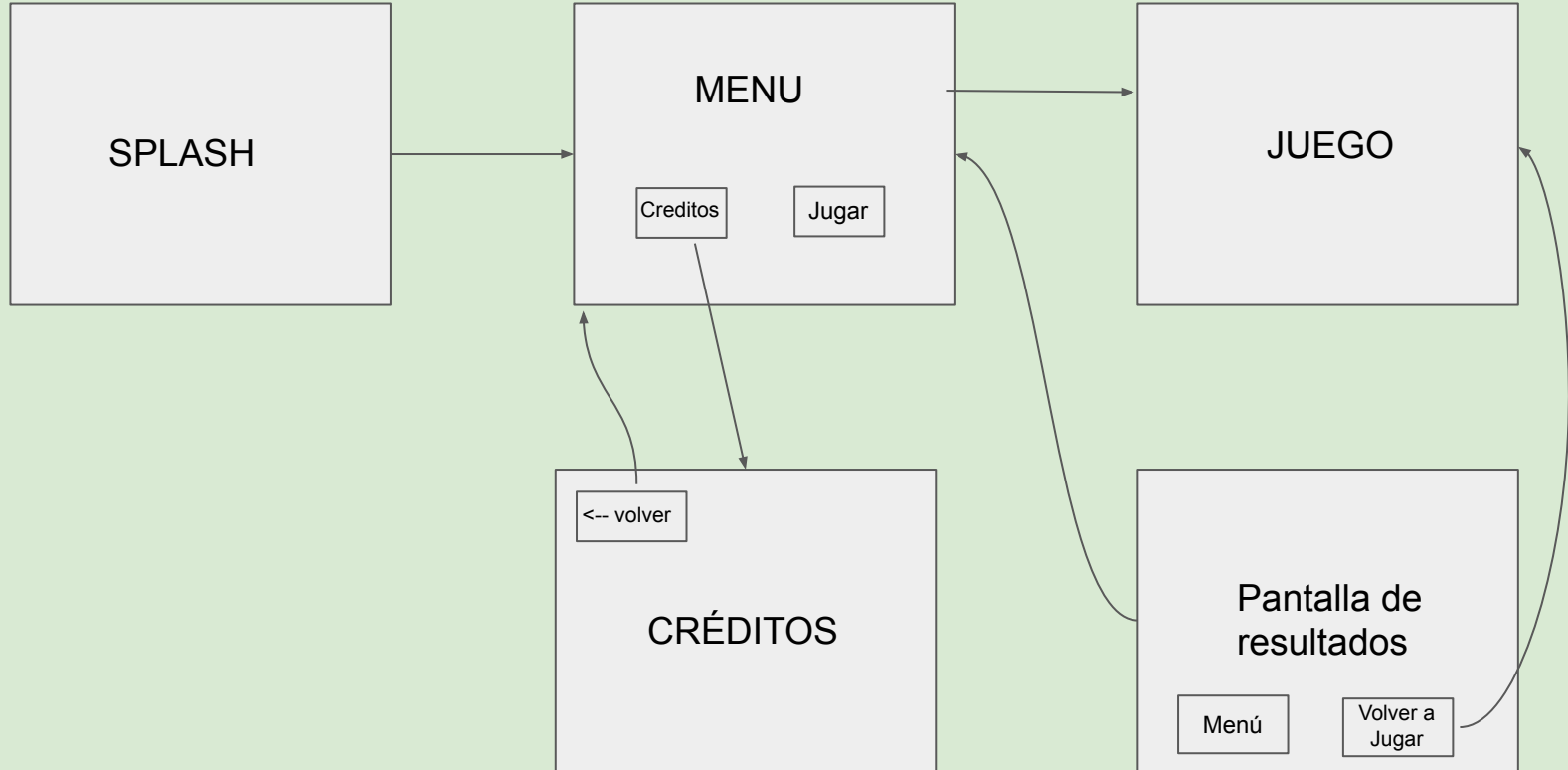


Desafío Bitsy

<https://bitsy.org/>



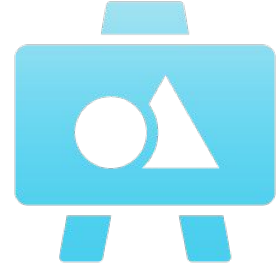
Template basico para jams



Motores de videojuegos

EJEMPLOS - Programación / frameworks

Allegro



EASELJS

gosu



Löve®



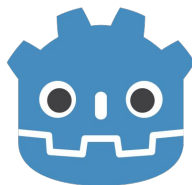
Motores de Videojuegos

https://en.wikipedia.org/wiki/List_of_game_engines



Godot

¿POR QUÉ GODOT?



GODOT

Game engine

GDscript

Es muy similar a Python y es propio de Godot



Documentación

Docs y recursos en línea sólidos



Open Source

Godot es FOSS bajo licencia MIT



Soporte 2d / 3d

Buen soporte en ambos enfoques



Veloz

Curva de aprendizaje simple

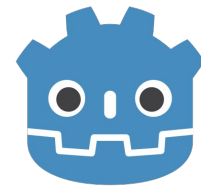


Didáctico

Tiene un enfoque pedagógico



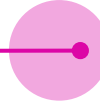
Que es godot



GODOT
Game engine

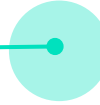
Qué es

Es un motor de videojuegos.
También se utiliza para
hacer aplicaciones y UI



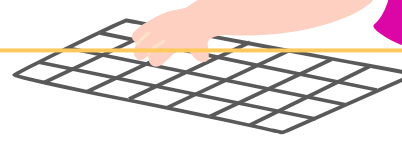
NODOS

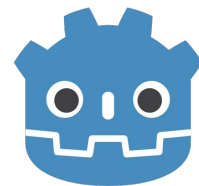
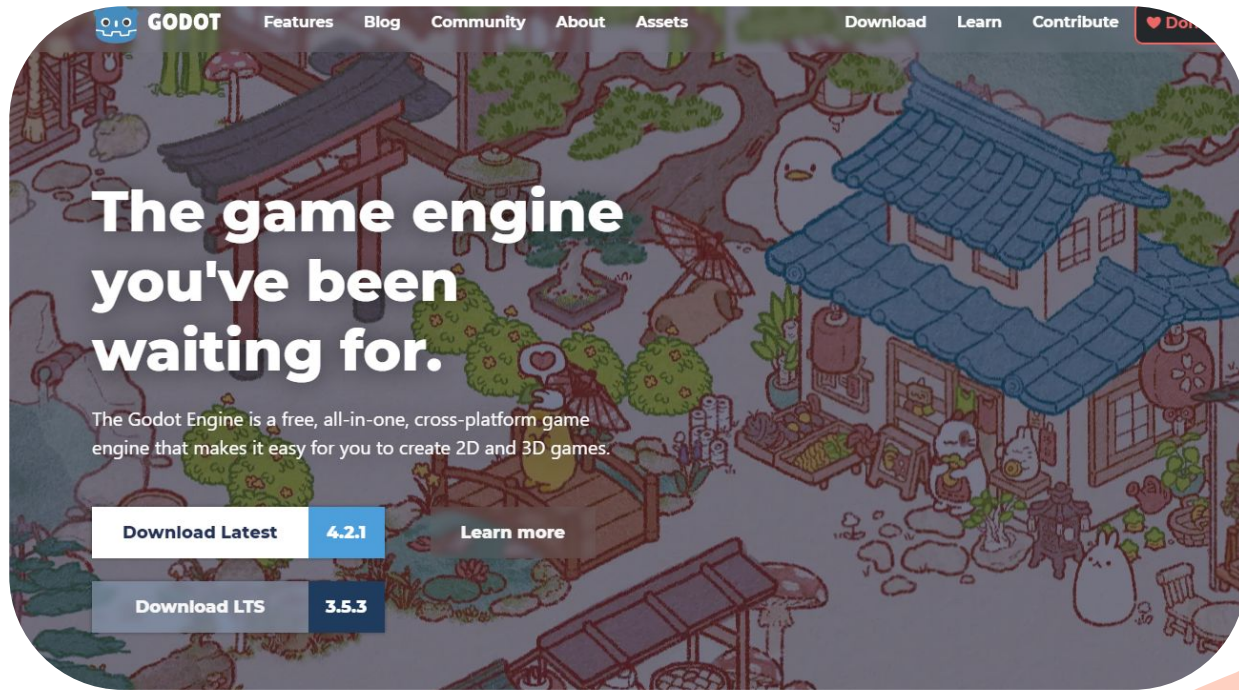
Utiliza un sistema de
escenas y nodos en un árbol
para agilizar el desarrollo



Intuitivo

Una interfaz fácil y amigable
con un fuerte énfasis en las
interacciones visuales

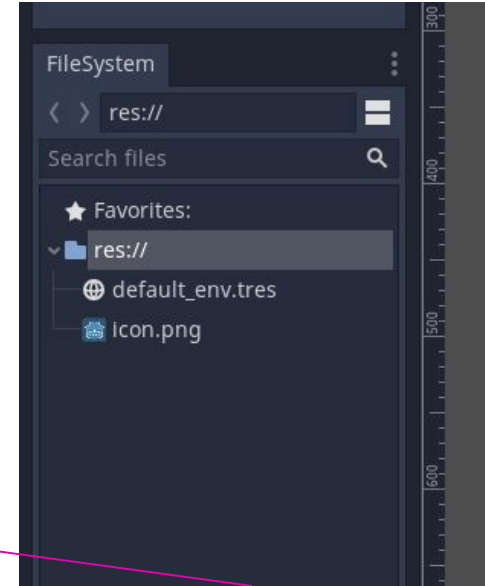
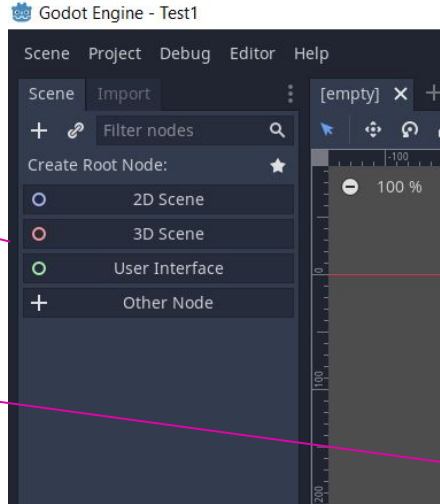
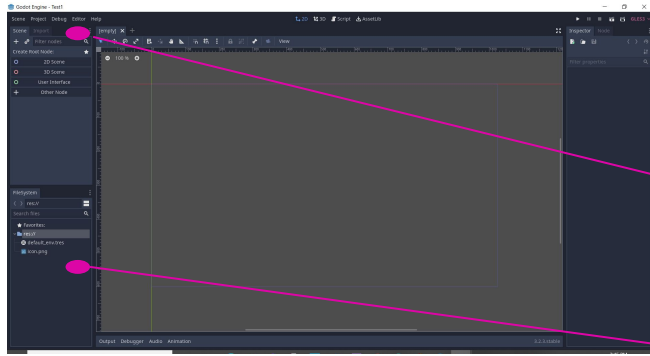




GODOT
Game engine



Interfaz godot



Interfaz

Visualización del
Editor con las
pantallas

Editor de escenas

Seleccionando el tipo
de escena se
visualizan los nodos

Recursos

Sistema de archivos

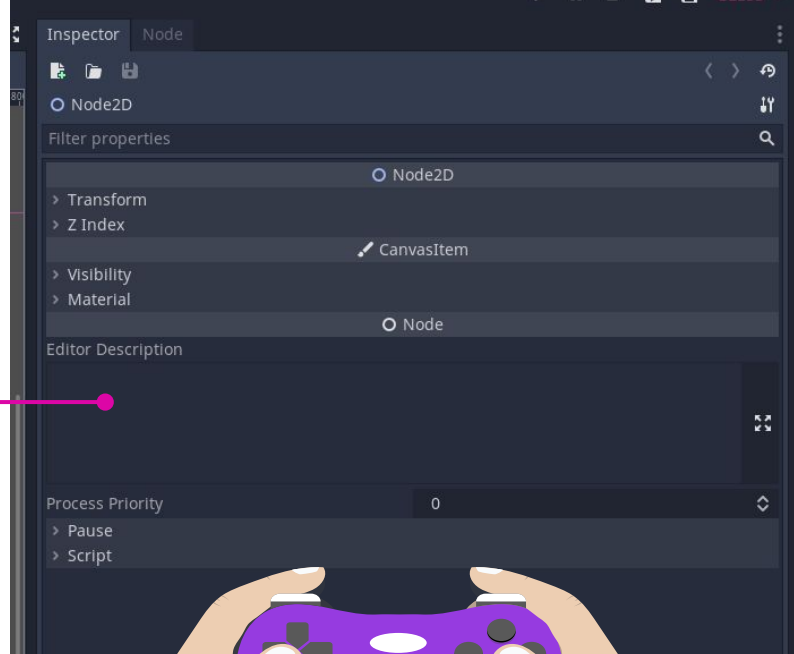
NODOS

Un nodo es el elemento básico para crear un juego, tiene las siguientes características:

- Tiene un nombre.
- Tiene propiedades editables.
- Puede recibir una llamada a procesar en cada frame.
- Puede ser extendido (para tener más funciones).
- Puede ser agregado a otros nodos como hijo.

La última es muy importante. Los nodos pueden tener otros nodos como hijos. Cuando se ordenan de esta manera, los nodos se transforman en un **árbol**.

En Godot, la habilidad para ordenar nodos de esta forma crea una poderosa herramienta para organizar los proyectos. Dado que diferentes nodos tienen diferentes funciones, combinarlos permite crear funciones más complejas.

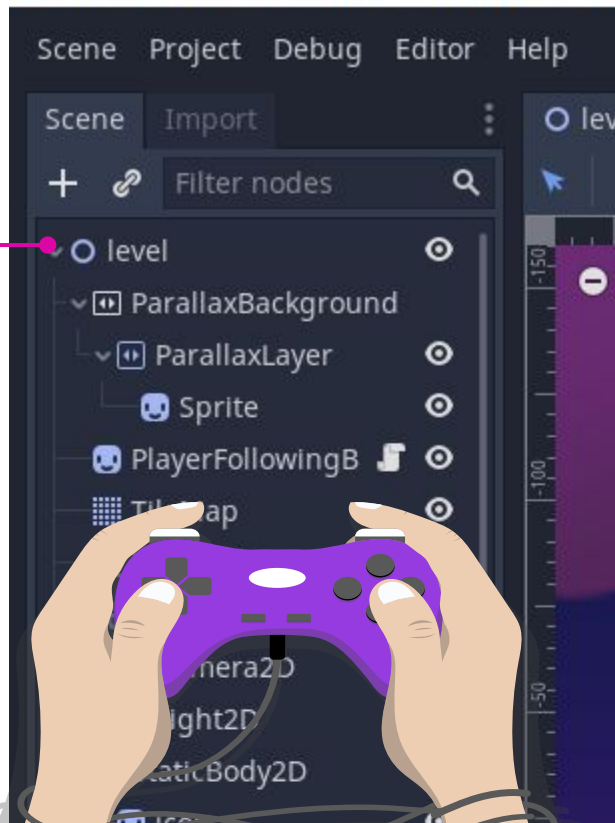


ESCENAS

Una escena está compuesta por un grupo de nodos organizados jerárquicamente (con estilo de árbol). Tiene las siguientes propiedades:

- Una escena siempre tiene un solo nodo raíz. _____
- Las escenas pueden ser guardadas a disco y cargadas nuevamente.
- Las escenas pueden ser *instanciadas* (mas sobre esto después).
- Correr un juego significa ejecutar una escena.
- Puede haber varias escenas en un proyecto, pero para iniciar, una de ellas debe ser seleccionada y cargada primero.

Básicamente, el motor Godot es un **editor de escenas**. Tiene más que suficientes herramientas para editar escenas 2D y 3D así como interfaces de usuario, pero el editor gira entorno al concepto de editar una escena y los nodos que la componen.



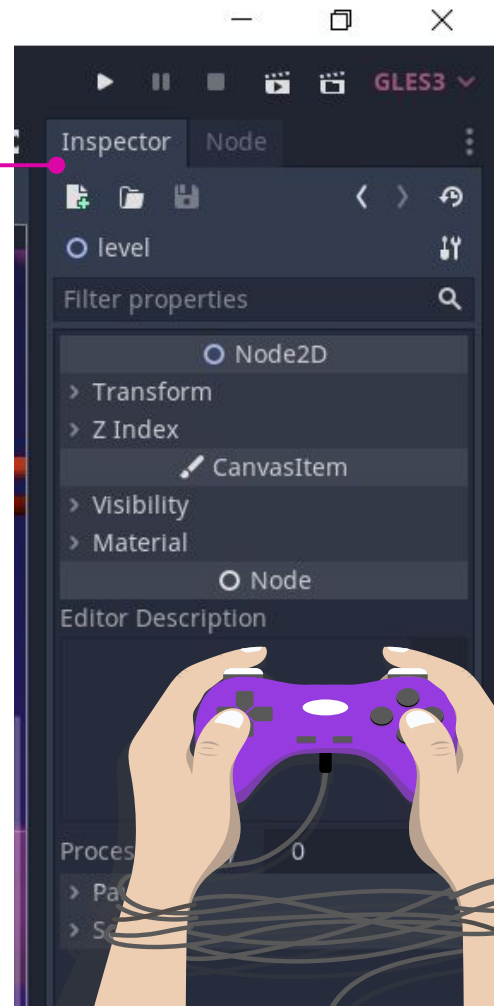
INSPECTOR DE NODOS

Aquí se ven las propiedades de los nodos.

Existe una jerarquía en las propiedades. Siempre veremos último la que se llama “nodo”, porque todos heredan de esa raíz padre.

De abajo hacia arriba en orden jerárquico según se heredan las clases de nodos hasta llegar al tipo de nodo que estaremos seleccionando.

Dentro de cada clases están las propiedades que podemos editar.



SCRIPTS

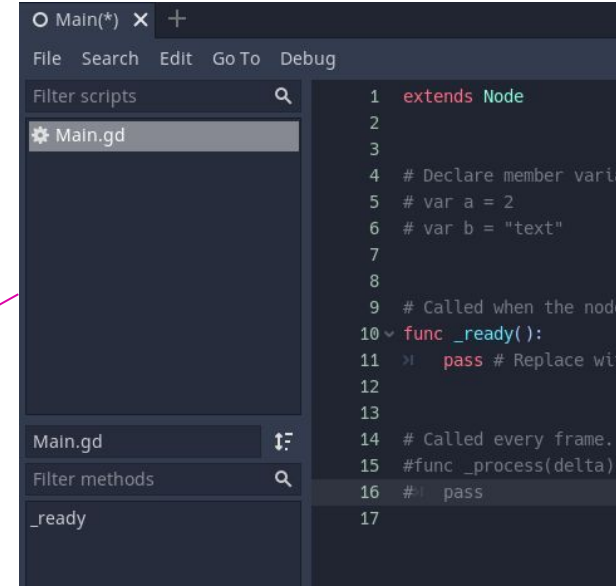
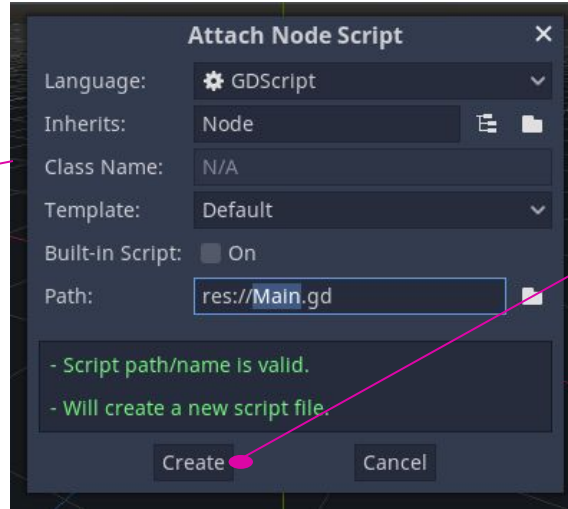
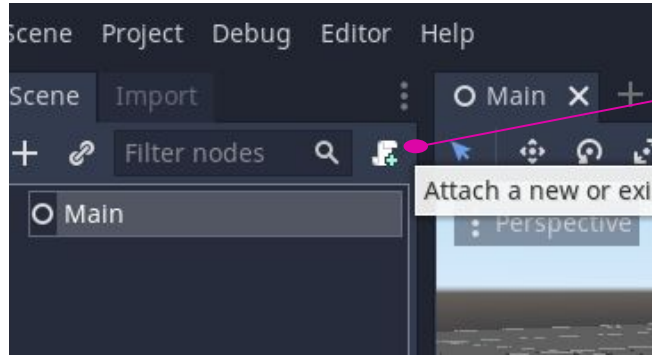
Los nodos por sí mismos poseen un enorme rango de funcionalidades, pero para poder crear interacciones complejas necesitamos poder extender esas funcionalidades mediante código..

Para esto, Godot viene equipado con un robusto lenguaje de scripting llamado GDScript, creado y pensado exclusivamente para hacer juegos con Godot.

Con este lenguaje podemos acceder, alterar, extender y crear sobre las propiedades y métodos que nos brinda la API de Godot, ampliando así el rango de opciones que tenemos al momento de crear sistemas para nuestros juegos.



Creando y asociando scripts a nodos



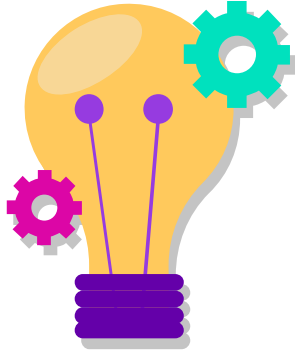
Seleccionamos el
nodo y hacemos click
en el icono



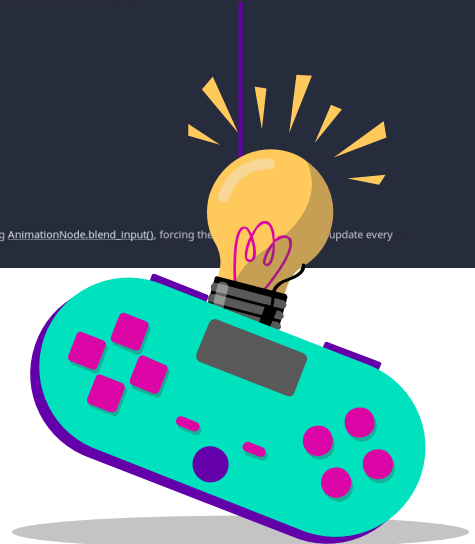
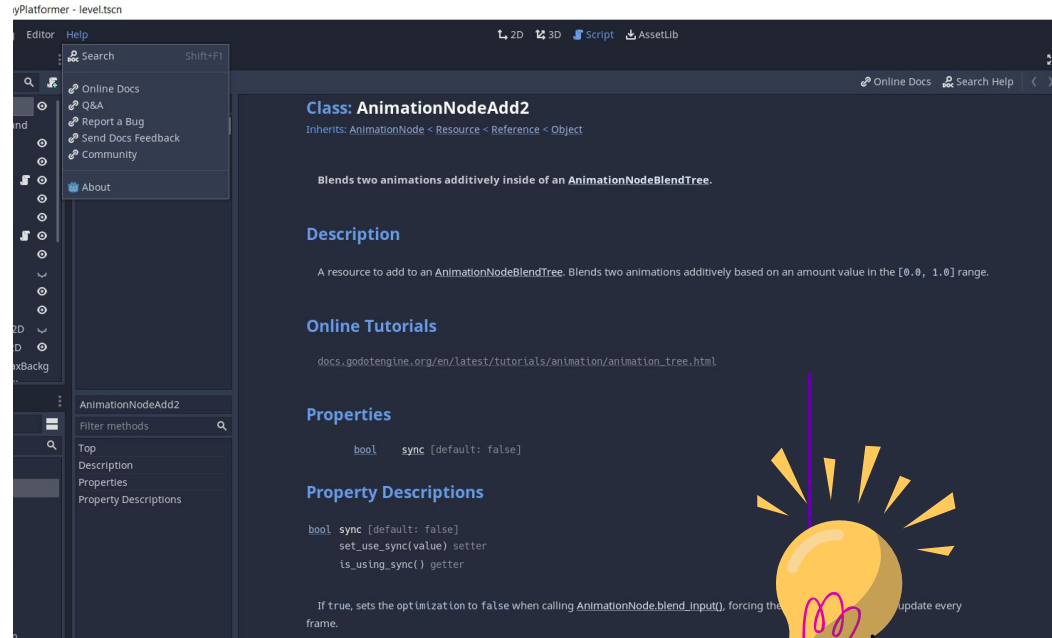
Le damos un nombre y un
path (recomendación,
mantener el mismo
directorio que la escena en
la que se utiliza), y lo
creamos

Se abrirá el visor de
scripts con un editor
donde podremos trabajar

AYUDA INTEGRADA

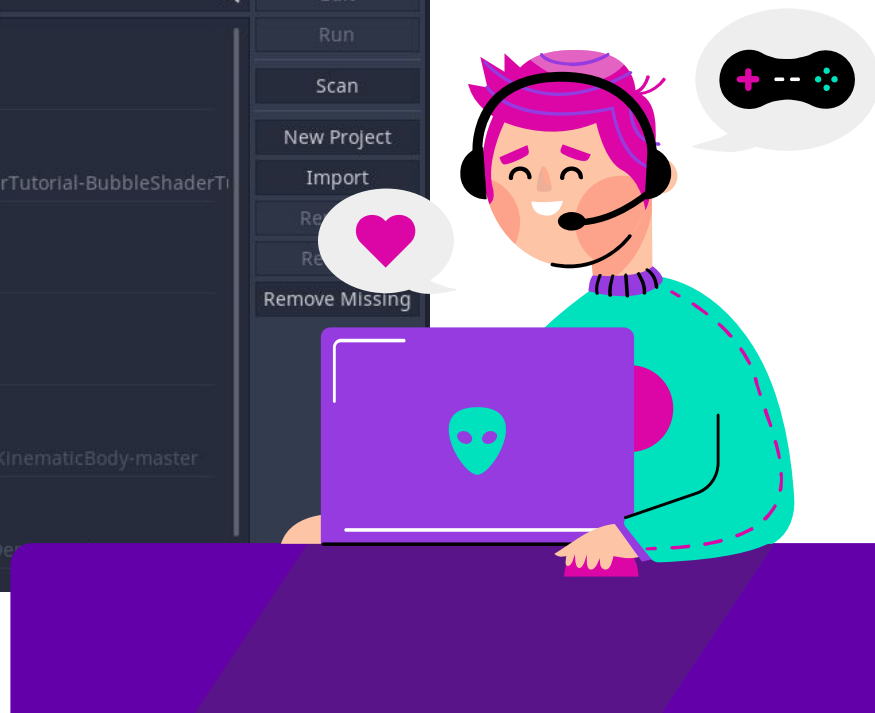
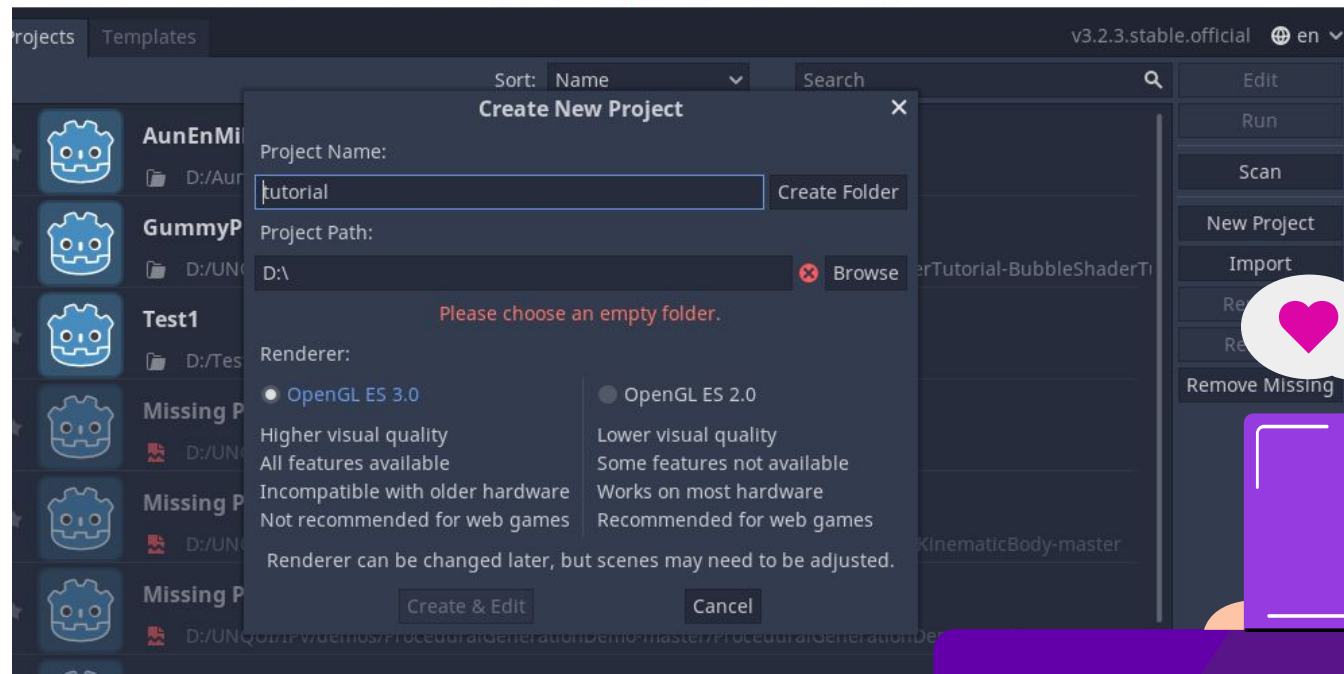


Godot trae una ayuda integrada muy bien documentada. Quizás a revisar por el idioma, pero está muy completa para saber cómo implementar cada uno de los elementos que ofrece el motor.

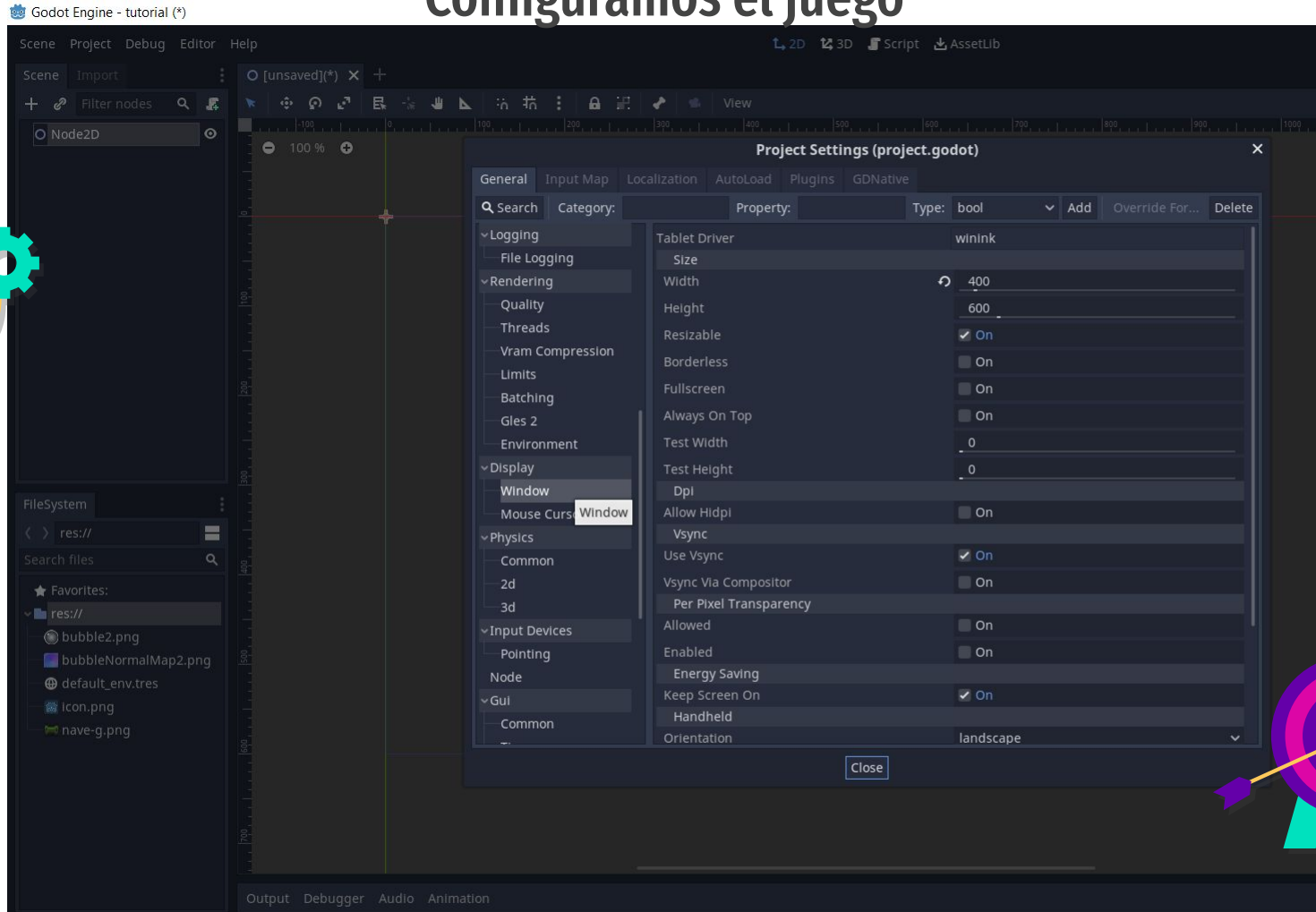


Nuevo Proyecto

Godot Engine - Project Manager - © 2007-2020 Juan Linietsky, Ariel Manzur & Godot Contributors

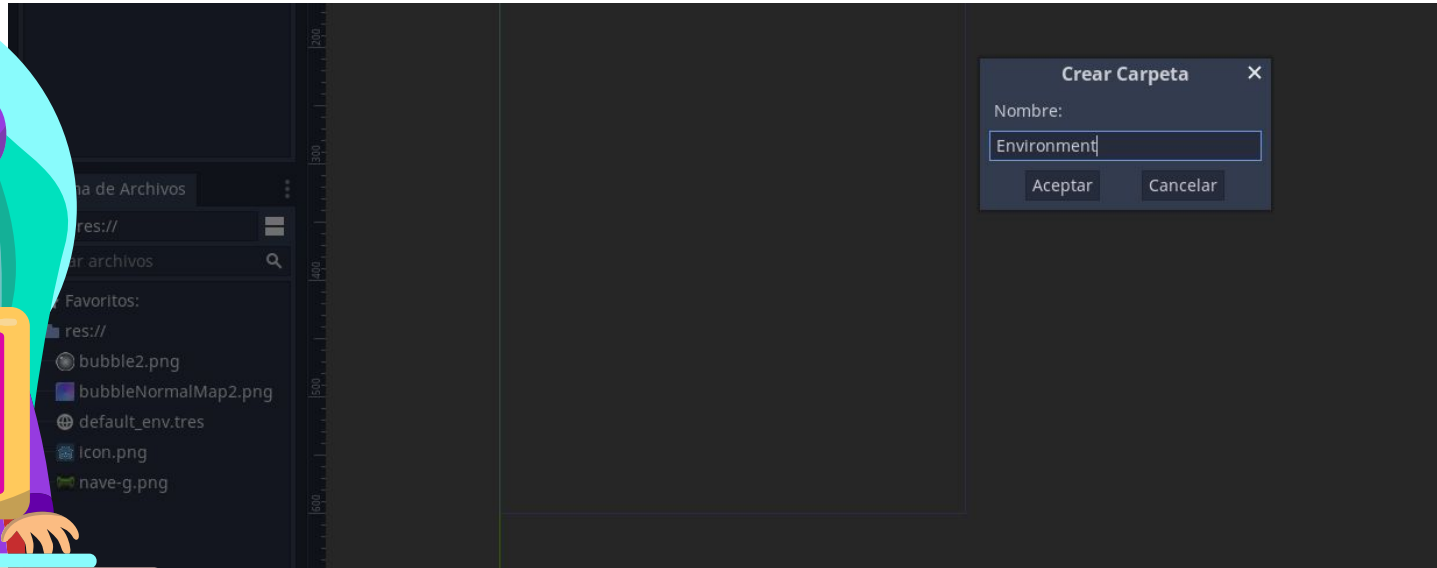
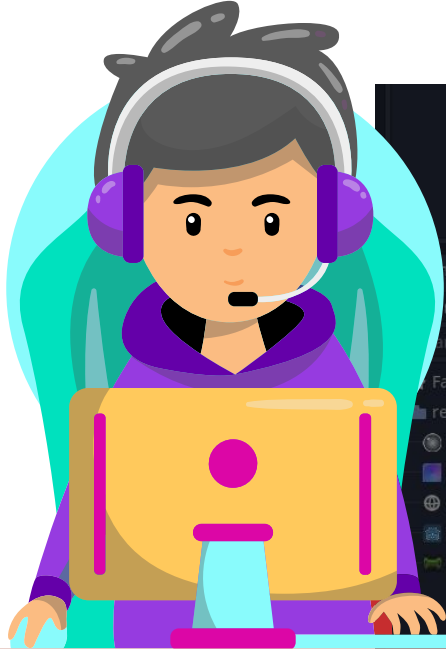


Configuramos el juego



Organización

Es muy importante comenzar los proyectos en forma organizada.
Nombrar las carpetas según los recursos que utilizaremos.



Buenas Prácticas

Godot está pensado para trabajar sobre el paradigma de programación orientada a objetos.

Por ello, los principios SOLID son plenamente aplicables. Principalmente, los principios de responsabilidad única (single responsibility) y encapsulamiento son importantes de considerar.



Además, Godot ofrece herramientas y soluciones propias para solucionar problemas propios de una arquitectura pensada para videojuegos, como las escenas, o posee patrones integrados (built-in) como las señales, que simplifica el uso del patrón Observer

A lo largo de la cursada iremos viendo diferentes problemas y soluciones que hacen a las buenas prácticas

Desafío 1

Tu primer videojuego, por la comunidad de Godot!

Primero, lee la documentación de primeros pasos:

- https://docs.godotengine.org/es/stable/getting_started/introduction/index.html

Y luego, te desafiamos a que completes este tutorial, creado amablemente por la comunidad de Godot:

- https://docs.godotengine.org/es/3.5/getting_started/first_2d_game/index.html



Desafío 2

Elige tu investigación

En Padlet (<https://padlet.com/>)

- Seleccionar un eje de análisis referido a videojuegos (Una temática como inclusividad, un género en particular, una consola, un juego o franquicia).
- Realizar un análisis con línea de tiempo que describa el eje seleccionado.



Pautas de entregas

1. Buenas prácticas - [Referencia](#)
2. Tiempo : antes de las 10 pm -- 10 pts. | límite hasta el viernes 10 pm.
3. Entregas en un solo repositorio con formato “entregas-ipv-apellido-nombre”, con cada desafío en un directorio separado, y cada desafío renombrado con el mismo formato “entrega-x-apellido-nombre”.

🔒 R4nKF1v3 / [entregas-ipv-moscatelli-nahuel](#)

📁 entrega-1-moscatelli-nahuel

📁 entrega-2-moscatelli-nahuel

📁 entrega-3-moscatelli-nahuel

📁 entrega-4-moscatelli-nahuel

📁 entrega-5-moscatelli-nahuel



Gracias!
Nos vemos en la próxima