# 

**Farming RPG**

**Xx/05/2024**

Vadillo Gutiérrez, Atenea

Informática y Telecomunicaciones

Desarrollo de aplicaciones multiplataforma

CIFP Juan de Colonia

Contenido

[1. Introducción 2](#_Toc165331348)

[1.1 Descripción del proyecto y finalidad del mismo 3](#_Toc165331349)

[2. Análisis y diseño del proyecto 4](#_Toc165331350)

[2.1 Planificación y Especificación de requisitos-descripción de las necesidades del proyecto 4](#_Toc165331351)

[2.2 Construcción 5](#_Toc165331352)

[2.2.1 Diseño 5](#_Toc165331353)

[2.2.2 Implementación 5](#_Toc165331354)

[2.2.3 Pruebas 5](#_Toc165331355)

[3. Recursos 8](#_Toc165331356)

[3.1 Recursos hardware 8](#_Toc165331357)

[3.2 Recursos software 8](#_Toc165331358)

[4. Planificación 9](#_Toc165331359)

[4.1 Planificación temporal 9](#_Toc165331360)

[7. Bibliografía 11](#_Toc165331361)

# Introducción

Este proyecto final de desarrollo de aplicaciones multiplataforma se centra en el mundo de los videojuegos de granjas para ofrecer una propuesta atractiva. Con el objetivo de crear una experiencia de juego compacta pero completa, accesible en escritorio, se centra en la creación de una demostración técnica de un videojuego de granjas.

La demostración técnica presenta un entorno virtual detallado donde los jugadores pueden sumergirse en la gestión y el crecimiento de una granja a través de mecánicas estratégicas y elementos de simulación. A pesar de su alcance limitado, el objetivo es proporcionar una muestra representativa de las funcionalidades clave del juego completo. Para lograrlo, se enfrentará a desafíos técnicos significativos, como la optimización del rendimiento y la adaptación de la interfaz de usuario, asegurando una experiencia fluida y satisfactoria.

Este Trabajo de Fin de Ciclo se centra en el desarrollo de una demostración técnica de un videojuego de granjas. El objetivo es crear una experiencia de juego compacta pero completa, accesible en escritorio. A medida que la demanda de juegos de simulación, como los de granjas, continúa en aumento, esta iniciativa busca ofrecer una experiencia de juego envolvente.

La demostración técnica se basa en un entorno virtual detallado donde los jugadores pueden experimentar la gestión y el crecimiento de una granja a través de mecánicas estratégicas y elementos de simulación, con teclado y mando. A pesar de su limitada extensión, la demo pretende proporcionar una muestra representativa de las funcionalidades clave del juego completo. Por lo tanto, se enfrentará a desafíos técnicos como la optimización de rendimiento y la adaptación de la interfaz de usuario para garantizar una experiencia fluida y satisfactoria en todas las plataformas.

## Descripción del proyecto y finalidad del mismo

El proyecto tiene como objetivo principal explorar el entorno de desarrollo de Godot Engine 4, un motor de juego de código abierto altamente utilizado. Se llevará a cabo a través de la creación de un juego de granjas, lo que permitirá explorar las funcionalidades y herramientas que ofrece este motor, desde su interfaz de usuario hasta su capacidad para renderizar gráficos en 2D, así como otras funciones como el mapeo de entradas.

Sin embargo, más allá de simplemente crear un juego divertido, el propósito último es aprender y perfeccionar habilidades técnicas y de diseño. Se espera que la aplicación práctica de conocimientos en programación, diseño gráfico, animación y música se convierta en un aspecto central del desarrollo. Además, se enfatiza la importancia de la colaboración y comunicación efectiva entre los miembros del equipo para garantizar la cohesión y la calidad del producto final.

Este proyecto se ha inspirado en una amplia gama de juegos, entre los que se destacan títulos como Stardew Valley, Potion Permit y Harvest Moon, todos ellos célebres por su enfoque en la gestión de granjas y la vida rural. La influencia de estos juegos se reflejará en la mecánica y la estética del juego en desarrollo. Además, se explorará la integración de elementos de RPG por turnos, tomando inspiración de otras franquicias de renombre como Pokémon, Undertale y Delta Rune. Estos juegos no solo han sido referentes en sus respectivos géneros, sino que también ofrecen una rica fuente de ideas para enriquecer la experiencia de juego que se busca crear.

En resumen, el proyecto no solo busca ofrecer entretenimiento a los usuarios, sino también servir como plataforma de aprendizaje y desarrollo profesional para los participantes. Se aspira a crear una experiencia de juego sólida y atractiva, respaldada por un enfoque metodológico y una ejecución técnica impecable, lo que guiará cada paso del proceso de desarrollo.

# Análisis y diseño del proyecto

En esta sección se presentará el análisis de requisitos u objetivos y diseño de la aplicación desarrollada.

## Planificación y Especificación de requisitos-descripción de las necesidades del proyecto

Entre los objetivos de este proyecto están:

* Aprender sobre el motor de juegos “Godot Engine”.
* Crear una demo técnica de un juego de granjas.
* Crear un menú con ajustes funcionales.
* Dar al usuario un menu de pausa donde poder salir al menu o al escritorio

Y partiendo de ahí los objetivos de la demo serían

* Mecanismos para de arar, regar y plantar.
* Tiendas funcionales.
* Enemigos.
* Pelea por turnos
* Mecánica día/noche.
* Una casa con funcionalidades como una cama para dormir y un almacén.
* Ajustes de controles, accesibilidad, gráficos y de sonido

## Construcción

Una vez planteados los requisitos que debe cumplir la aplicación en el presente apartado se explicará el planteamiento inicial y cómo fueron abordados.

### Diseño

Inicialmente, este proyecto empezó como un juego de rol por turnos al estilo clásico, inspirado en la mecánica de enfrentar diversos enemigos a lo largo de un camino, siguiendo la dinámica que popularizó la franquicia Pokémon. Sin embargo, durante las primeras fases de desarrollo, cambio la dirección del videojuego al descargar los assets del juego, siendo estos de un juego de granjas.

Al ver los assets de un juego de granjas, cambie de perspectiva a un subgénero completamente diferente: el simulador de granjas. Inspirándome en juegos mencionados anteriormente como Stardew Valley y Potion Permit, decidí adoptar este enfoque y transformar el proyecto en un juego que fusionara elementos de RPG por turnos con la gestión de una granja.

El cambio de dirección no solo amplió mis horizontes creativos, sino que también me ofreció la oportunidad de sumergirme en el diseño y la programación de videojuegos. A través de este proyecto, mi objetivo es adquirir experiencia práctica en el desarrollo de juegos, explorando las complejidades del motor de juego Godot y familiarizándome con sus nodos y funcionalidades.

Tomando en cuenta los contenidos del curso, decidí usar C# en lugar de GDScript. Aunque GDScript está integrado de forma nativa en Godot, opté por C# debido a su amplia adopción en la industria de los videojuegos. Esta elección me permitirá aplicar los conocimientos del curso y adquirir habilidades transferibles valoradas en la industria, aprovechando las ventajas de su sistema de tipos estáticos y su sólido soporte en Godot.

### Implementación

Implementación: Aquí vamos a especificar con código de programación, lo que dije antes que, hacia mi juego, por ejemplo, uno de los objetivos de diseño era que al picar en la tierra esta cambiara de forma, lo cual se logra con (XXXXXXXXXXXXXXXXXXXXX)código de programación, también que antes de que el campesino se moviese de una finca a otra tenía que tener totalmente arada la tierra de esta finca para lo cual tuve que implementar (XXXXXXXXXXXXXXXXX)código, el personaje se diseñó con este código (XXXXXXXXXX), y sus funciones que son esta y esta y esta con este código, la párate más complicada en código fue (XXXXXXXXXXXX) esta que lleva a cabo tal función. Las clases más importantes y demás, como se comunican cual es la principal etc,etc.

### Pruebas

Las pruebas son esenciales en el desarrollo de videojuegos para garantizar calidad y estabilidad. Este apartado detalla los tipos de pruebas, procedimientos, herramientas y resultados obtenidos en cada fase de pruebas. Todas las pruebas se han ejecutado en tiempo real.

|  |  |  |
| --- | --- | --- |
| Prueba | Resultado esperado | Resultado real |
| Funcionamiento de las animaciones | Las animaciones responden a la entrada de teclas | La animación “Idle” se intercala con las animaciones de andar |
| Funcionamiento de las animaciones con el codigo corregido | Las animaciones responden a la entrada de teclas | Las animaciones si se reproducían cuando debían |
| Entrada de mando Xbox 360 | El juego responde a la entrada de mando | El juego responde a la entrada de mando igual que a la de teclas |
| Implementación de las animaciones “plow” (arar) | La animación responde a la entrada de espacio (tecla) y A (Xbox) | No se reproduce la animación/el Sprite se queda en el fin de la animación hasta que se mueve |
| Creación del nodo AnimationTree | Solucion de los problemas con las animaciones “plow” | El Sprite se queda en el fin de la animación hasta que se mueve |
| Implementación de StateMachine en el nodo AnimationTree | Solucion de los problemas con las animaciones “plow” | Se ejecutan todas las animaciones a la vez |
| Cambio de StateMachine a BlendMachine2D | Solucion de los problemas con las animaciones “plow” | El eje Y esta invertido. El sprite andaba de espaldas hacia arriba y abajo |
| Corrección del eje Y | Solucion al movimiento invertido | Las animaciones “plow” se ejecutan cuando deben y el movimiento deja de estar invertido |
| Indicador de posición | Se imprime por consola la posición del Sprite al dar a espacio/A | Referencia nula al Sprite |
| Corrección en codigo de la referencia al Sprite | Se imprime por consola la posición (Vector2i) al dar a espacio/A | Se imprime la posición relativa en integer por consola |
| Cambiar el TileSet indicado | Se cambia el tileset donde está el personaje por uno de tierra | Se cambian tilesets con la posición del personaje anterior |
| Funcionalidad botones menu | Los botones jugar, ajustes y salir funcionan según deben | Jugar abre la escena del juego, ajustes abre el menu de ajustes, pero salir no cierra la aplicación |
| Cambio de evento pressed a button\_down en el boton salir | El boton salir cierra el juego | El boton salir cierra el juego |
| Funcionamiento del combobox de ajustes (modo ventana) | El tamaño de la ventana corresponde a la opcion seleccionada | Excepción por no encontrar el nodo |
| Cambio de script al padre directo y cambio de combobox por botones | El tamaño de la ventana corresponde a la opcion seleccionada | Los botones responden a las opciones seleccionadas |
| Funcionamiento del combobox de ajustes (resolución) | La resolución corresponde a la opcion seleccionada | La resolución corresponde a la opcion seleccionada |
| Funcionamiento del custom data para las plantas | Las plantas solo estan en tilesets de tierra | Las plantas solo estan en tilesets de tierra |
| Colisiones | El sprite no puede pasar por las vallas donde tienen zona de colisión | El sprite no puede pasar por las vallas donde tienen zona de colisión |

# Recursos

A continuación, se procederá a analizar los diferentes recursos de hardware y software que utiliza Farming RPG para su funcionamiento.

## Recursos hardware

Los recursos de hardware necesarios para este proyecto han sido un ordenador con una pantalla de 1920 x 1080px y un controlador de Xbox 360.

El ordenador ha sido esencial tanto para la estructuración de nodos, cambios en sus características y programación de la lógica de algunos, como para la ejecución del juego y sus pruebas.

El controlador de Xbox 360 ha sido crucial para comprobar el mapeado de entrada del usuario. Su uso ha permitido garantizar una experiencia de juego fluida y cómoda, asegurando controles optimizados para aquellos que prefieren jugar con este tipo de dispositivos.

## Recursos software

Entre los recursos software, se destaca:

* Godot 4.2.2: Una plataforma versátil y robusta, fundamental para la creación y desarrollo de elementos clave del videojuego, tales como las animaciones, el diseño del mapa y la configuración de la entrada del usuario.
* Visual Studio 2022 Community: Una herramienta de programación integral, imprescindible para la elaboración de scripts y la implementación de la lógica subyacente en el videojuego, garantizando así su funcionalidad y coherencia interna.
* Git y GitHub: Sistemas de control de versiones y almacenamiento, respectivamente, que han proporcionado un entorno seguro y eficiente para gestionar el desarrollo del proyecto, ofreciendo una solución alternativa de almacenamiento en caso de ser necesario.

La combinación de estos recursos software ha sido esencial para garantizar la calidad y la eficiencia en cada etapa del proceso de creación del videojuego.

# Planificación

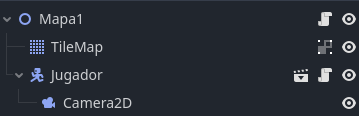
En el siguiente apartado se analizará tanto la forma en la que fue planificado temporalmente el proyecto como el plan económico propuesto.

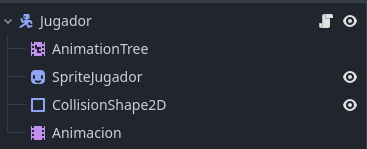
## Planificación temporal

Durante la etapa inicial del proyecto, se priorizó la integración de los activos en el entorno del videojuego. Esta fase implicó la elaboración de los tilesets destinados al tilemap, así como la creación de animaciones y escenas pertinentes. Se diseñaron tilesets para garantizar la coherencia visual y funcionalidad en el tilemap, mientras que las animaciones se elaboraron mediante la disposición secuencial de frames, aprovechando la estructura proporcionada por la hoja de sprites.

La implementación de las animaciones abarcó diversos aspectos del juego, desde los movimientos del personaje hasta las interacciones con el entorno. Para gestionar estas animaciones de manera eficiente, se optó por la creación de un nodo AnimationTree, proporcionando así un enfoque estructurado para la lógica de animación.

Además, se desarrollaron escenas específicas para el jugador (utilizando el nodo CharacterBody2D) y para el mapa (mediante el nodo Node2D), asegurando así una organización coherente y modular del proyecto. La fase inicial culminó con la integración exitosa de estos elementos en el videojuego, sentando así las bases para las etapas subsiguientes del desarrollo.





La segunda fase del proyecto, se concentró en la implementación de la funcionalidad del juego mediante la creación y desarrollo de scripts específicos. Estos scripts fueron diseñados para controlar el movimiento y las acciones del personaje principal, así como para determinar cuándo se reproducen las animaciones correspondientes a cada acción.

En este contexto, se priorizó la creación de scripts que permitieran una interacción fluida y dinámica del jugador con el entorno virtual. Se implementó lógica de programación que gestiona el movimiento del personaje de manera coherente y responsiva, garantizando una experiencia de juego satisfactoria para el usuario.

Además, se dedicó especial atención a la integración de la entrada por mando, lo cual constituye un aspecto crucial para la accesibilidad y la experiencia de juego. Se desarrollaron mecanismos que permiten al jugador controlar el personaje utilizando un mando de forma intuitiva y eficiente, asegurando así una experiencia de juego fluida y adaptable a diferentes preferencias de control.

En la tercera fase del desarrollo, se dedicó esfuerzo significativo a la creación de un menú principal completo y funcional, junto con la implementación de opciones de ajustes para mejorar la experiencia del usuario.

El menú principal se diseñó con el objetivo de ofrecer una interfaz intuitiva y atractiva que permitiera a los jugadores navegar fácilmente por las distintas opciones disponibles. Se incluyeron los botones Jugar, Ajustes y Salir para realzar la experiencia del usuario, así como una disposición clara y coherente de las diferentes secciones del menú.

En cuanto a los ajustes, se implementaron opciones personalizables que permiten a los jugadores modificar diversos aspectos del juego según sus preferencias individuales. Esto incluye ajustes de video, sonido y accesibilidad, contribuyendo a una experiencia de juego más personalizada y satisfactoria para cada jugador.

Se agregaron tres secciones adicionales al proyecto inicial: gráficos, sonido y accesibilidad. En la sección de gráficos, se incluyeron tres opciones de modo de ventana (pantalla completa, maximizado y ventana), junto con varias opciones de resolución para el modo de ventana sin maximizar. Esto permite a los jugadores adaptar la experiencia visual según sus preferencias y la capacidad de su sistema.

Hablar de sonido y accesibilidad

# 7. Bibliografía

05/03/2024 Juego de Plataformas 2D/Godot Tutorial/Introducción Godot/1-Capitulo/Programacion Videojuegos <www.youtube.com/watch?v=F3T_ZhllzJs&t>

05/03/2024 Juego de Plataformas 2D/Godot Tutorial/Crear el Mapa/Godot/2-Capitulo/Programacion videojuegos <www.youtube.com/watch?v=SR7mdh0_i6Q&t>

07/03/2024 GODOT Desde 0/Movimiento Jugador/Primer Script/Tutorial/ 4-Capitulo/Programación Videojuegos <https://www.youtube.com/watch?v=fya91wv1OPI&t>

09/03/2024 Juego de Plataformas 2D/Godot Tutorial/Movimiento Personaje/Godot/3-Cap/Programacion videojuegos <https://www.youtube.com/watch?v=NeUS3Ytjty4>

12/03/2024 Juego de Plataformas 2D/Godot Tutorial/Recoger Monedas/Godot/4-Cap/Programacion videojuegos <www.youtube.com/watch?v=u99myfBJDlc&t>

20/03/2024 How to Create WORKING CROPS in Godot <www.youtube.com/watch?v=QK_uI-m6bpA&t>

20/03/2024 How to Make a 2D FARMING Game in Godot (step by step) <www.youtube.com/watch?v=QnOQNkglXso&t>

27/04/2024 INTRODUCCIÓN A GODOT 4 [10] ANIMATIONPLAYER, Sistema de ANIMACIONES | Indie Game Dev <https://www.youtube.com/watch?v=v5uZuyelKRQ>

27/03/2024 INTRODUCCIÓN A GODOT 4 [11] ANIMATIONTREE, Gestor de Transiciones | Indie Game Dev <https://www.youtube.com/watch?v=hfQkI5zQY3w>

30/03/2024 Hiding the Mouse Cursor in Godot 4 <https://www.youtube.com/watch?v=QIiS5WK2z0Q>

30/03/2024 063 - Viewports - Múltiples Resoluciones - Godot 3 <www.youtube.com/watch?v=ZveIbgizF28&t>

30/03/2024 Godot 4 Making the Game Fullscreen and adding Splash Screen <https://www.youtube.com/watch?v=-iGL2wkARj0>

30/03/2024 Main Menu (Godot 4) <https://www.youtube.com/watch?v=oDtDuwCPasg>

06/04/2024 Animation Tree State Machine Setup w/ Conditions & BlendSpace2D - Godot 4 Resource Gatherer Tutorial <www.youtube.com/watch?v=WrMORzl3g1U&t>

09/04/2024 INTRODUCCIÓN A GODOT 4 [13] TILEMAP, Sistema de Tiles 2D | Indie Game Dev <https://www.youtube.com/watch?v=XVSbjqjJhUQ&t>

16/04/2024 Godot 4 TileMap Tutorial Ep 2 | Terrains / Autotilling <https://www.youtube.com/watch?v=uXZuitdUPP8&t>

16/0/2024 Godot 4 TileMap Tutorial Ep 3 | Placing Tiles with a Mouse Click <https://www.youtube.com/watch?v=PSEPHO8ukjI&list=PLflAYKtRJ7dwtqA0FsZadrQGal8lWp-MM&index=3>

22/04/2024 ¿Como cambiar la pantalla de inicio en Godot? (Splash Screen) | Tutorial Godot 4 <https://www.youtube.com/watch?v=4FyH-wZLuOU>

23/04/2024 How To Change The Window Mode And Resolution In Godot 4 <https://www.youtube.com/watch?v=YsdkcPV0BAo>

23/04/2024 How To Create An Options Menu In Godot! <https://www.youtube.com/watch?v=fFIST_4wmyI>

26/04/2024 Get access to custom tile data in C# in Godot 4 Beta <https://www.reddit.com/r/godot/comments/xjdarh/get_access_to_custom_tile_data_in_c_in_godot_4/>

26/04/2024 Documentación TileMap [https://docs.godotengine.org/en/latest/classes/class\_tilemap.html#class-tilemap-method-get-cell-tile-data](https://docs.godotengine.org/en/latest/classes/class_tilemap.html%23class-tilemap-method-get-cell-tile-data)

27/04/2024 Godot4: Como agregar colisiones a un TileMap (Curso intensivo del TileMap) <https://www.youtube.com/watch?v=0Q2t_oHzZFA>

28/04/2024 Godot 4 TileMap Tutorial Ep 6 | Plant Growth <https://www.youtube.com/watch?v=6h4QSpccqIY&list=PLflAYKtRJ7dwtqA0FsZadrQGal8lWp-MM&index=6>

29/04/2024 ¡5 Tips en Godot que quizás no conozcas! <https://www.youtube.com/watch?v=z76J58ecAFM>

30/04/2024 Godot 4 Timer Node Tutorial <https://www.youtube.com/watch?v=Zf6awHRr7bU>

30/04/2024 Como crear un timer! (Godot Engine 4) <https://www.youtube.com/watch?v=TyPalRr_jpo>

456