

UNIVERSIDAD DE TARAPACÁ
FACULTAD DE INGENIERÍA
INGENIERÍA CIVIL EN COMPUTACIÓN E INFORMÁTICA
ARICA – CHILE



Newenche

Sistema de Visualización y Creación de Maquetas de líneas y subestaciones eléctricas de alta tensión

Equipo de Desarrollo:

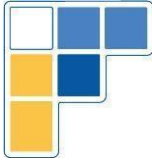
Raiza Ossandón Aravena
Jorge Gutiérrez Montes

Empresa o Unidad: Newenche

Curso: Proyecto IV ICCI

Profesor: Diego Aracena Pizarro

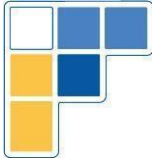
06 de noviembre, 2024



Resumen o introducción

La empresa Newenche ingeniería y servicios reconoce la importancia de garantizar la seguridad y eficiencia en proyectos de gran envergadura, como la implementación de líneas y subestaciones eléctricas de media y alta tensión. Por lo que se busca desarrollar un sistema innovador de maquetado con posiciones y coordenadas geoespaciales con el fin de poder demostrar los posibles riesgos que pueden haber al momento de realizar un proyecto a gran escala como lo es la implementación de dichas líneas y subestaciones eléctricas, de manera innovadora e inmersiva.

Al utilizar tecnología geoespacial, serán capaces de simular escenarios realistas y evaluar el impacto de diversas variables en la seguridad y eficiencia de las infraestructuras eléctricas, además de poder realizar una mejor preparación en conjunto a los trabajadores que realizarán las intervenciones. Esto les brindará una ventaja significativa al anticipar y abordar posibles problemas antes de que ocurran en el mundo real.



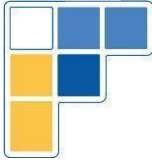
Objetivos

Objetivo General:

- Desarrollar un sistema de visualización y creación de maquetas en 3D para líneas y subestaciones eléctricas de alta tensión, que permita simular de manera geoespacial el entorno de trabajo, con el fin de identificar y mitigar riesgos, mejorando la seguridad y eficiencia en proyectos de infraestructura complejos.

Objetivos Específicos:

- Implementar un sistema que permita la creación de maquetas geoespaciales en 3D utilizando coordenadas reales para visualizar líneas y subestaciones eléctricas.
- Desarrollar herramientas que permitan agregar y gestionar estructuras eléctricas dentro del entorno virtual, facilitando la simulación y detección de colisiones entre componentes.
- Incorporar la funcionalidad de cargar y visualizar terrenos reales mediante coordenadas WGS84 para crear un entorno de trabajo lo más realista posible.
- Optimizar la interfaz de usuario para que sea intuitiva y fácil de usar, garantizando que los operadores puedan interactuar de manera efectiva con las maquetas.
- Utilizar tecnologías avanzadas de renderizado y simulación, como Unreal Engine y Cesium, para asegurar una representación visual de alta calidad y precisión geoespacial.



I. Definición del proyecto

Contexto En proyectos mineros u otros proyectos de infraestructura complejos, es común tener múltiples líneas de alta tensión y otros elementos críticos que deben ser cuidadosamente gestionados durante la construcción. La visualización y gestión de estos elementos es crucial para garantizar la seguridad y la eficiencia del trabajo.

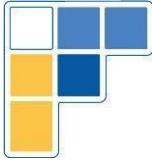
Problema El problema es la dificultad de visualizar y gestionar los riesgos en proyectos de infraestructura complejos, como los mineros, que involucran múltiples cruces de líneas de alta tensión y otros elementos peligrosos. Sin una herramienta efectiva para representar el entorno de trabajo de manera detallada, es complicado identificar y mitigar estos riesgos antes de que surjan en el sitio real, lo que puede comprometer la seguridad y la eficiencia del proyecto. La capacitación del personal también se ve afectada, ya que no cuentan con un medio adecuado para practicar procedimientos y entender los riesgos en un entorno simulado.

Solución Para lograr esto, se propone el uso de motores 3D para crear maquetas geoespaciales. Estas maquetas permiten simular y visualizar el entorno de trabajo en detalle, ayudando a identificar y mitigar riesgos potenciales antes de que se presenten en el sitio.

II. Requisitos del sistema

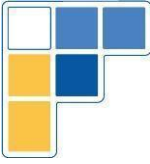
Requisitos de Alto Nivel (opcional)

ID	Definición
RdAN1	El sistema debe permitir generar maquetas en 3D geolocalizables por coordenadas.



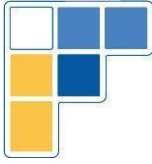
Requisitos funcionales

ID	Definición
Rf1	El sistema debe permitir añadir estructuras a la maqueta.
Rf2	El sistema debe permitir cargar y ubicar terreno geoespacial utilizando coordenadas WGS84.
Rf3	El sistema debe implementar las conexiones entre torres mediante cables.
Rf4	El sistema debe implementar un sistema de colisiones para detectar si existen estructuras o cables que choquen entre sí.
RF5	El sistema debe mostrar los detalles de las estructuras y cables, como el de qué tipo son y sus medidas.
RF6	El sistema debe poder guardar y cargar maquetas hechas en el mismo sistema.
RF7	El sistema debe permitir visualizar la maqueta utilizando una vista libre en primera persona.
RF8	El sistema debe permitir configurar opciones gráficas.



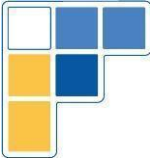
Requisitos no funcionales

ID	Definición
RnF1	El software debe permitir la visualización de las maquetas en un entorno 3D geoespacial
RnF2	La interfaz de usuario del software debe ser intuitiva y fácil de usar.
RnF3	El sistema debe tener una variedad de familias de estructuras (portante, anclaje y remate) para utilizarlas en el sistema, las cuales pueden añadirse más según sea necesario para el proyecto.
RnF4	El sistema debe balancear la calidad gráfica con el rendimiento.



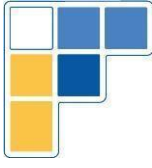
Carta Gantt

	Agosto	Septiembre	Octubre	Noviembre	Diciembre
Definición del problema	■				
Planteamiento de la solución		■	■		
Firma de requisitos Cliente			■		
Implementación Casos de uso			■		
Primera Reunión Prueba casos de usos			■		
Segunda Reunión entrega casos de usos				■	
Pruebas de funcionamiento sistema final				■	
Entrega producto Final					■
Documentación		■	■	■	■



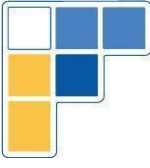
Tecnologías

- **Redmine:** Es una herramienta de gestión de proyectos y seguimiento de incidencias de código abierto. Ofrece funcionalidades como gestión de tareas, control de versiones, seguimiento de tiempo y gestión de recursos, lo que facilita la planificación y ejecución de proyectos. Además, permite la colaboración entre equipos, soporta múltiples proyectos y cuenta con un sistema de notificaciones y foros para discusiones. Es altamente personalizable e integrable con otras herramientas a través de plugins.
- **Unreal Engine 5:** Unreal Engine es un motor de desarrollo en tiempo real, ampliamente utilizado para crear gráficos y simulaciones 3D en videojuegos, aplicaciones de realidad virtual y otras experiencias interactivas. Ofrece potentes herramientas para la creación de entornos realistas, física avanzada, animaciones y programación visual mediante Blueprints. Unreal Engine es altamente valorado por su capacidad de renderizado de alta calidad y su versatilidad para adaptarse a proyectos complejos que requieren interactividad en 3D.
- **Google Drive:** Es una plataforma de almacenamiento en la nube que permite guardar, compartir y colaborar en archivos de manera eficiente. Ofrece aplicaciones como Google Docs, Sheets y Slides, que permiten la creación y edición de documentos en línea de forma colaborativa y en tiempo real. Es ideal para trabajar en equipo, ya que facilita el acceso a los archivos desde cualquier dispositivo con conexión a internet y tiene opciones de control de permisos y versiones de los documentos.
- **Cesium:** Es una plataforma de gráficos 3D basada en la web que se utiliza principalmente para visualizar y analizar datos geoespaciales en tiempo real. Es ideal para crear mapas y modelos 3D interactivos, permitiendo visualizar datos como terrenos, edificios y objetos en movimiento en un contexto global. Cesium ofrece una alta precisión geoespacial y puede manejar grandes conjuntos de datos, como nubes de puntos, modelos 3D de ciudades y terrenos con textura. Su motor es ampliamente usado en aplicaciones de simulación, defensa, realidad aumentada, videojuegos y análisis geoespacial. ✓
- **Github:** Es una plataforma de desarrollo colaborativo que permite a los programadores alojar, gestionar y controlar versiones de proyectos de software. Se basa en Git, un sistema de control de versiones distribuido que facilita el seguimiento de cambios en el código fuente a lo largo del tiempo, permitiendo que varios desarrolladores trabajen en paralelo sin perder el historial de modificaciones.



Departamento de Ingeniería en Computación e
Informática
Facultad de Ingeniería
Universidad de Tarapacá

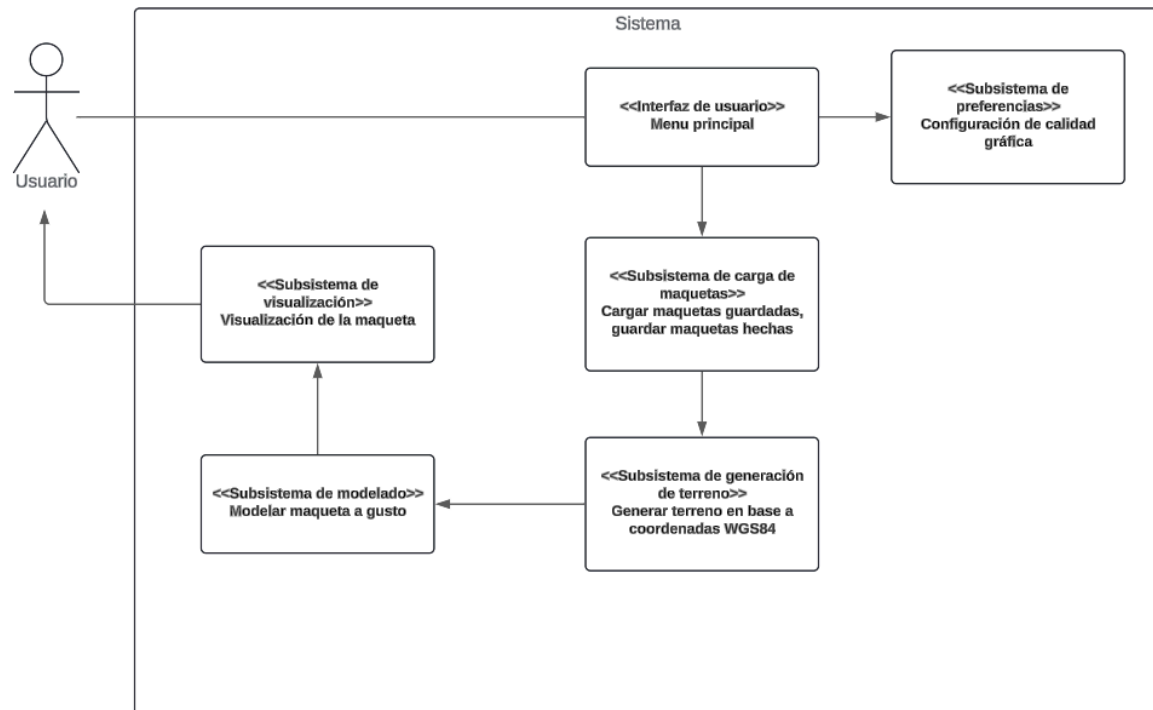




Metodología

La metodología a trabajar es una metodología ágil scrum, basándose en reuniones semanales las cuales entregan una muestra del producto la cual recibirá feedback por parte de los clientes, con el objetivo de poder realizar cambios acordemente a su preferencia.

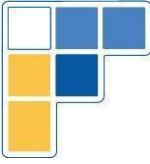
Modelo de contexto



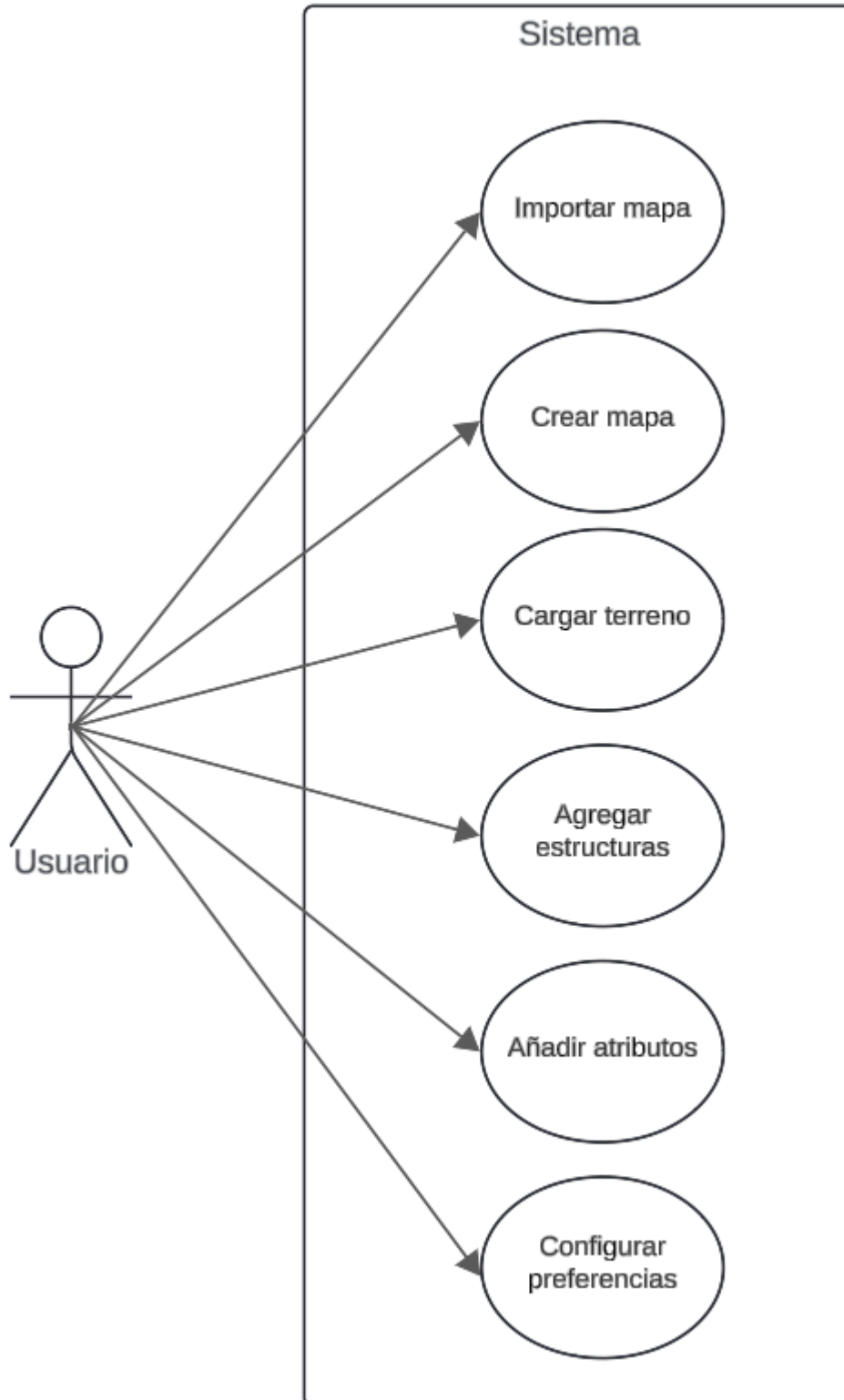
Casos de uso

Para la elaboración del proyecto de Sistema de Visualización y Edición de Maquetas de líneas y subestaciones eléctricas de alta tensión, se procederá a la realización de los siguientes casos de uso:

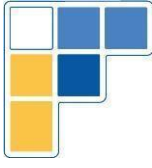
- Importar mapa
- Crear mapa
- Cargar terreno
- Agregar estructuras
- Añadir atributos
- Configurar preferencias



Modelo de casos de uso

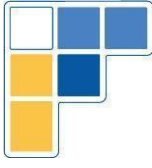


y cuando genero la maqueta y puedo mostrarla ??



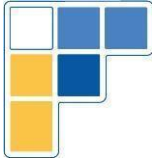
Importar mapa

Nombre: Importar mapa	
Descripción: El usuario selecciona un mapa para ser importado a la aplicación.	
Actores: Usuario	
Precondiciones: Se debe haber iniciado el software.	
Flujo Normal	
Usuario	Sistema
	1.- El sistema muestra la pantalla de inicio, mostrando las opciones.
2.- El usuario selecciona la opción de importar mapa y elige un mapa.	
	3.- El sistema carga el mapa seleccionado.
	4.-El sistema muestra las estructuras y sus atributos.



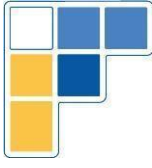
Crear mapa

Nombre: Crear mapa	
Descripción: El sistema permite al usuario crear un mapa, desde una plantilla en blanco.	
Actores: Usuario	
Precondiciones: Se debe haber iniciado el software.	
Flujo Normal	
Usuario	Sistema
2.- El usuario selecciona la opción de crear mapa.	1.- El sistema muestra la pantalla de inicio, mostrando las opciones. 3.- El sistema carga un mapa en blanco, sin estructuras. 4.- El sistema muestra un mapa en blanco.



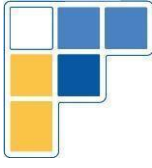
Cargar terreno

Nombre: Cargar terreno	
Descripción: El sistema permite al usuario cargar un terreno geoespacial utilizando el sistema de coordenadas wgs84	
Actores: Usuario	
Precondiciones: Se debe haber iniciado el software. Se debe haber creado un mapa en blanco	
Flujo Normal	
Usuario	Sistema
2.- El usuario ingresa las coordenadas geoespaciales.	1.- El sistema muestra una interfaz con la opción de ingresar coordenadas. 3.- El sistema carga un terreno en 3D del mundo real basado en google earth. 4.- El sistema muestra un terreno en 3D del mundo real basado en google earth.



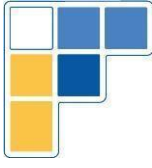
Agregar estructuras

Nombre: Agregar estructuras	
Descripción: El usuario agrega estructuras en el mapa seleccionado.	
Actores: Usuario	
Precondiciones: Se debe haber iniciado el software. Se debe estar en un mapa sea importado o en blanco.	
Flujo Normal	
Usuario	Sistema
	1.- El sistema muestra las estructuras disponibles a colocar en el mapa.
2.- El usuario selecciona qué estructura desea colocar.	
	3.- El sistema carga la estructura seleccionada en el mapa.
	4.- El sistema muestra la estructura en pantalla.



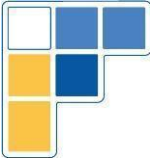
Añadir atributos

Nombre: Añadir atributos	
Descripción: El usuario le añade atributos a la estructura seleccionada.	
Actores: Usuario	
Precondiciones: Se debe haber iniciado el software. Se debe estar en un mapa sea importado o en blanco. Debe haber una estructura agregada.	
Flujo Normal	
Usuario	Sistema
2.- El usuario selecciona la estructura.	1.- El sistema muestra las estructuras en el mapa. 3.- El sistema muestra las opciones de la estructura.
4.- El usuario le agrega o modifica un atributo.	5.- El sistema guarda el atributo ingresado asociado a la estructura.



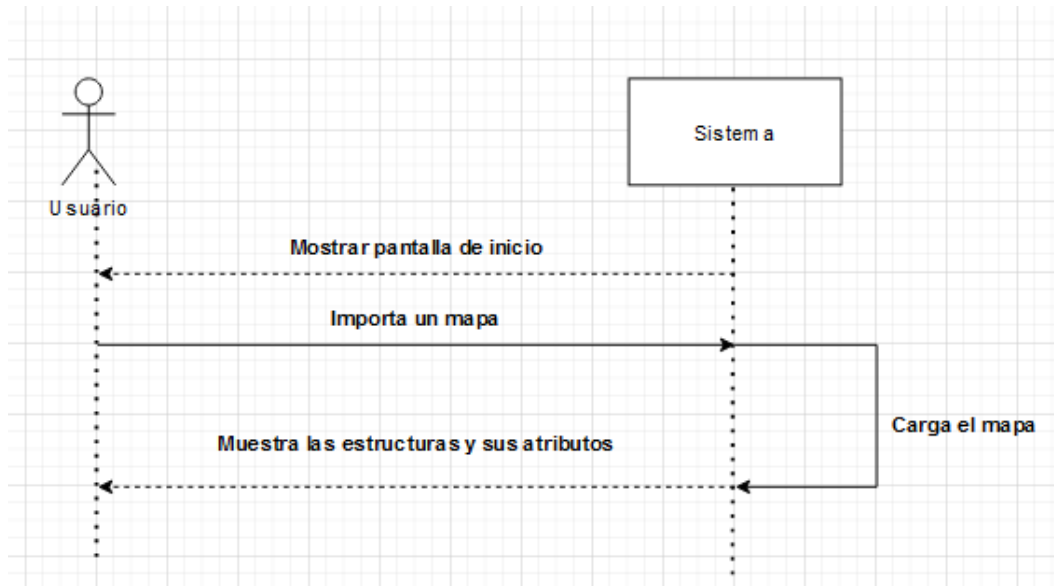
Configurar preferencias

Nombre: Configurar preferencias	
Descripción: El usuario puede configurar preferencias gráficas	
Actores: Usuario	
Precondiciones: Se debe haber iniciado el software.	
Flujo Normal	
Usuario	Sistema
	1.- El sistema muestra la pantalla de inicio, mostrando las opciones.
2.- El usuario selecciona la opción de preferencias.	
	3.- El sistema muestra las opciones gráficas.
4.- El usuario configura las preferencias a su gusto.	
	5.- El sistema guarda las preferencias del usuario

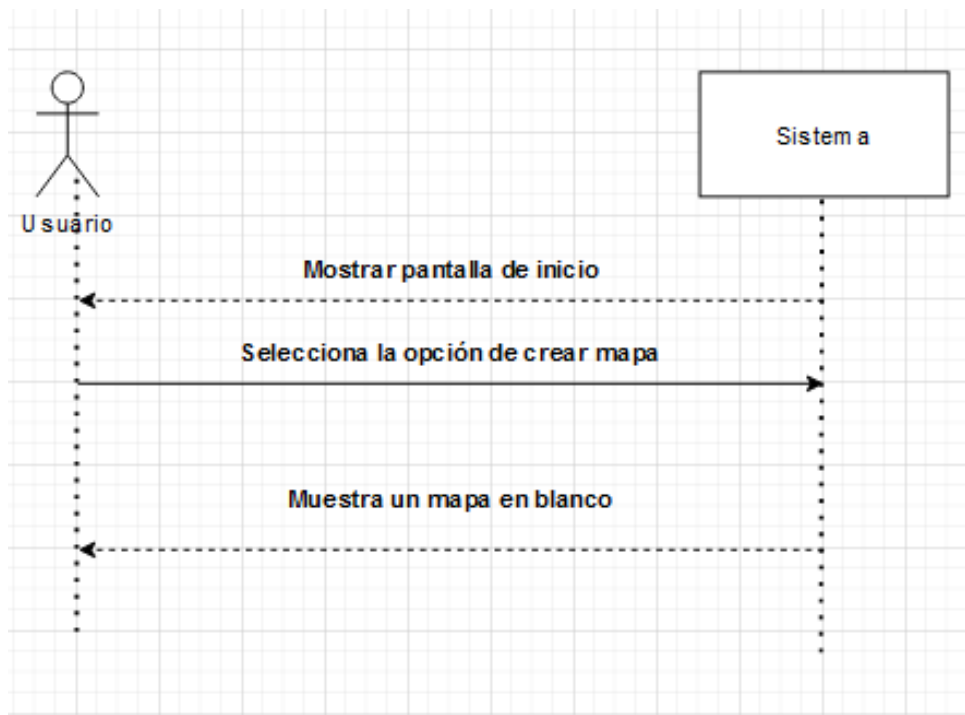


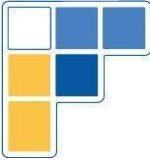
Diagramas de secuencia

Importar mapa

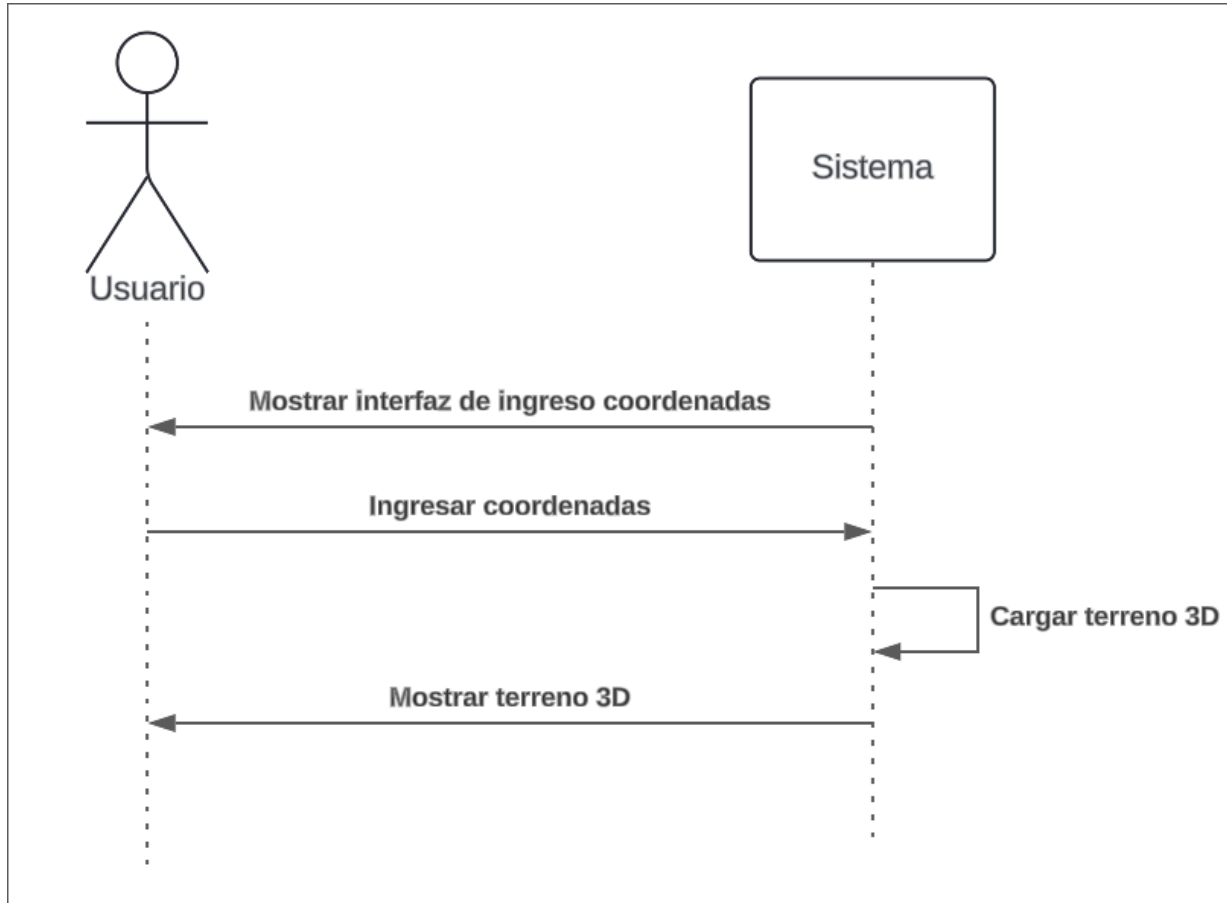


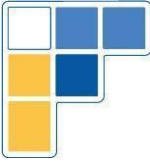
Crear mapa



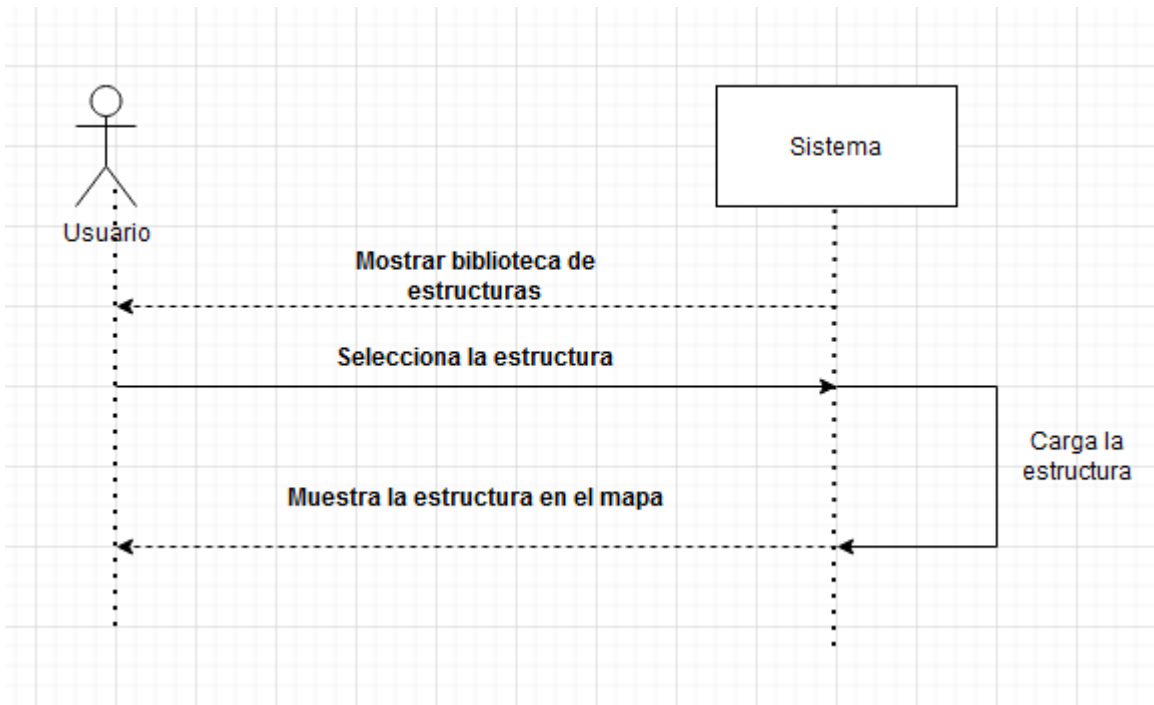


Cargar terreno

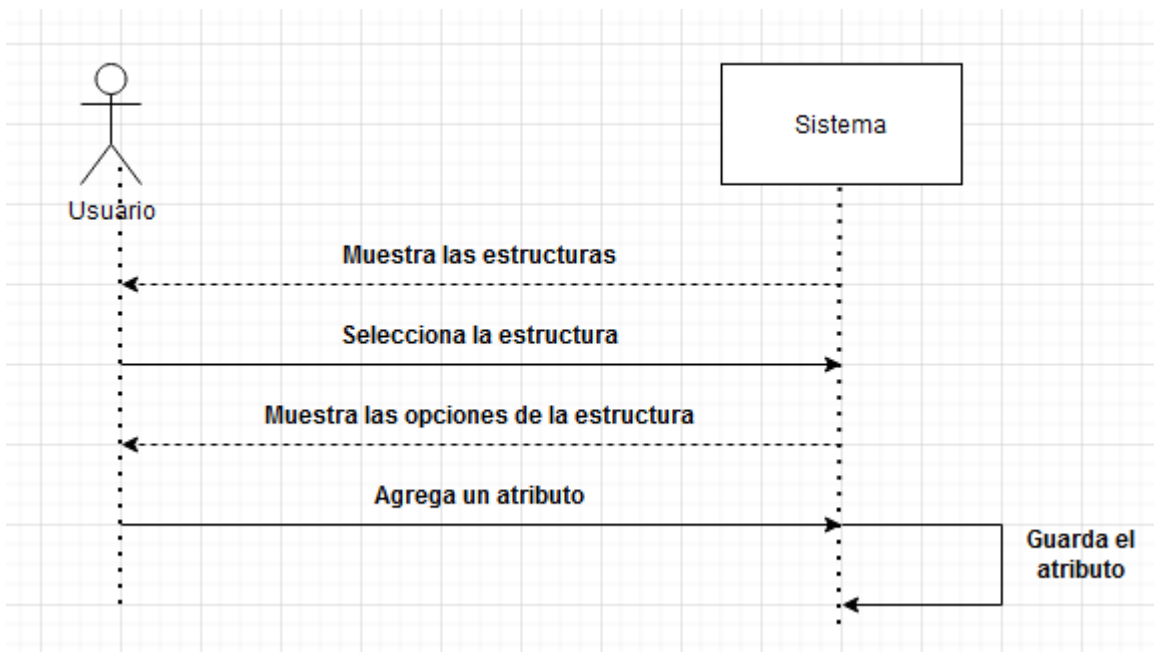


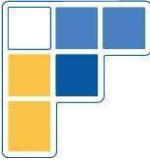


Agregar estructuras

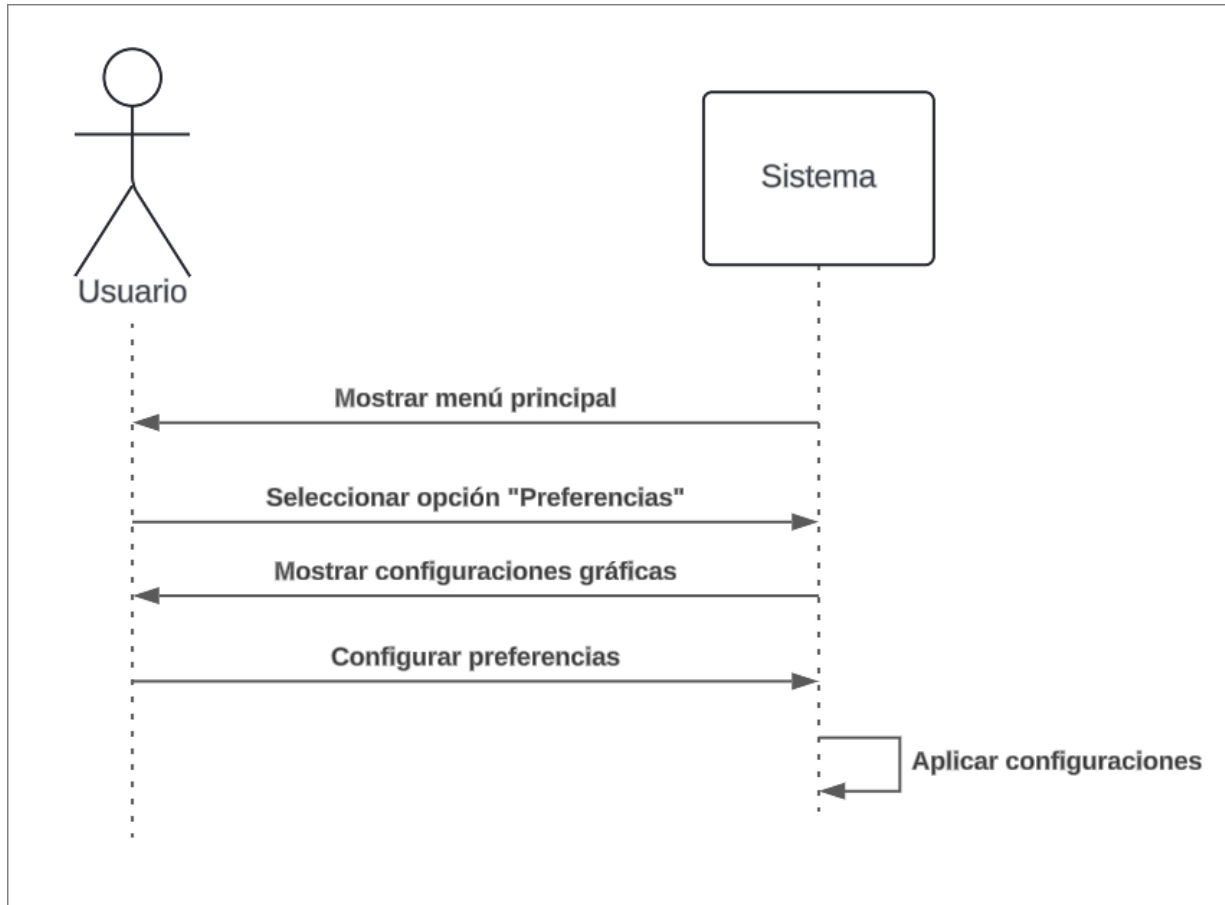


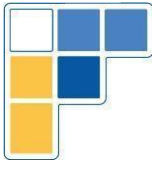
Añadir atributos





Configurar preferencias

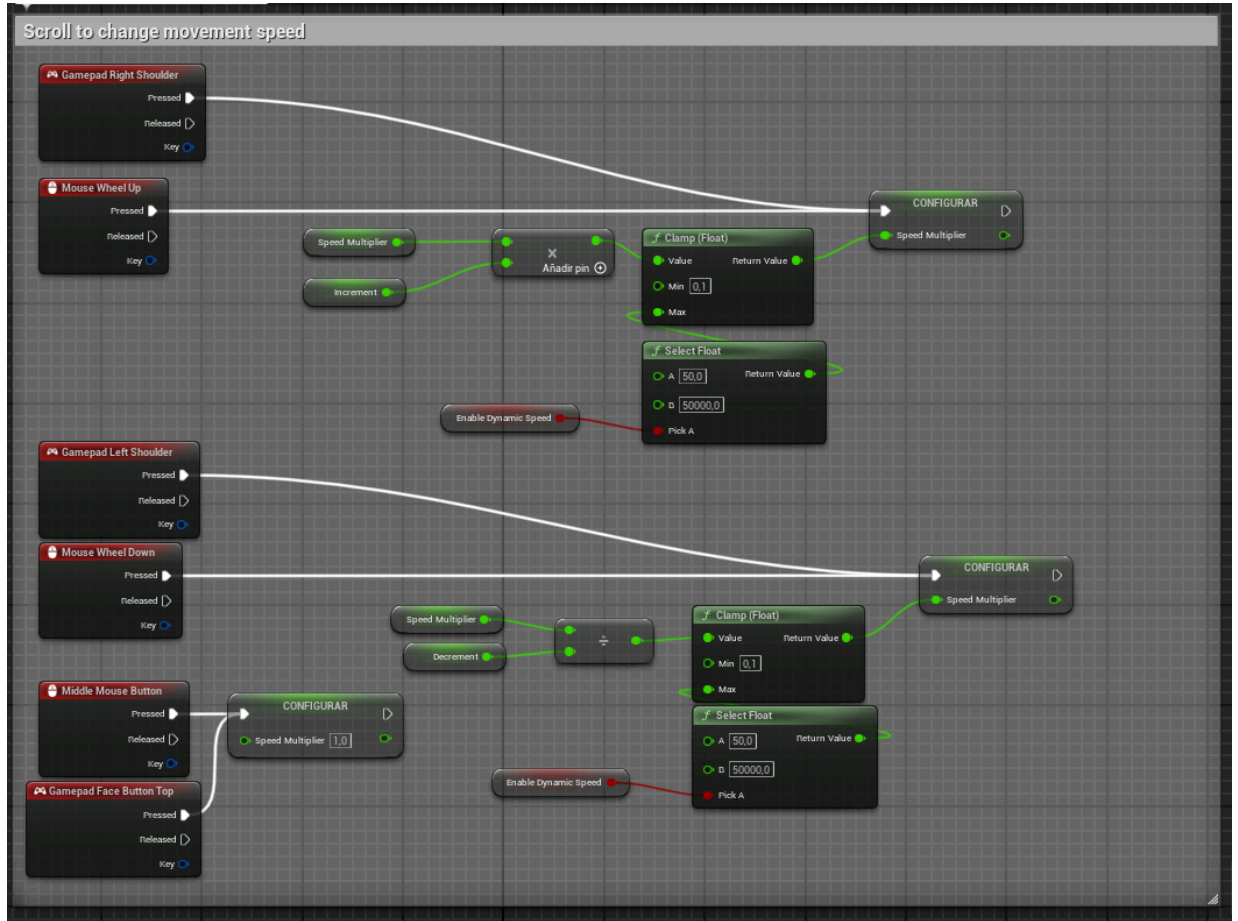




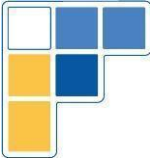
BluePrints

1. Scroll para cambiar la velocidad de movimiento

El siguiente Blueprint de Unreal Engine se diseñó para poder cambiar la velocidad de movimiento usando el mouse o gamepad.

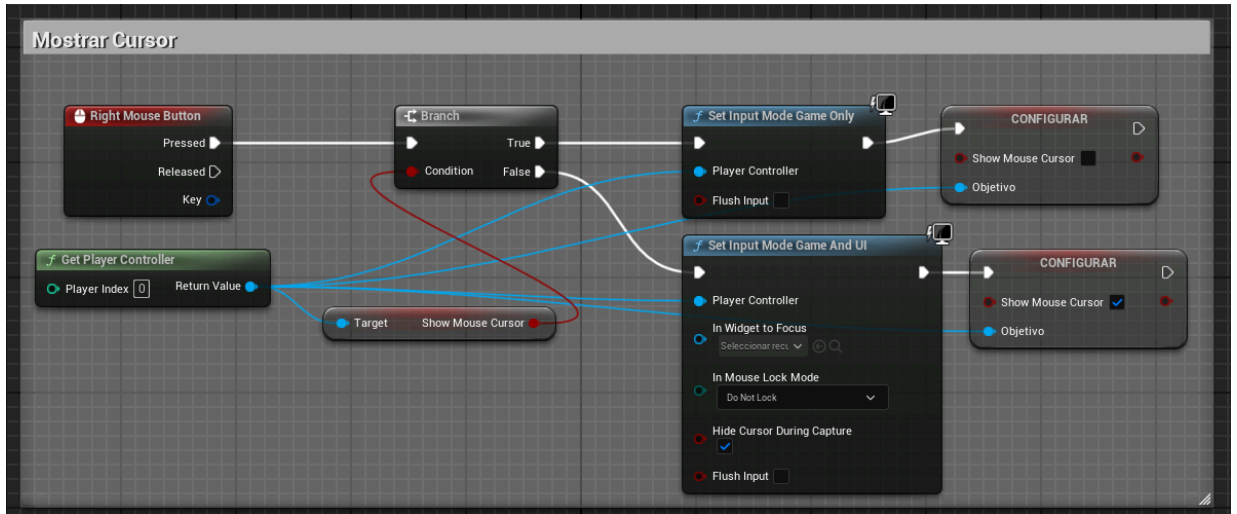


- Mouse Wheel Up / Gamepad Right Shoulder: Incrementan la velocidad de movimiento. Se utiliza una multiplicación con un factor (Speed Multiplier) ajustado por un incremento definido, limitado por un Clamp (Float) para evitar valores extremos.
- Mouse Wheel Down / Gamepad Left Shoulder: Reducen la velocidad del movimiento mediante una resta. También se asegura que no disminuya por debajo del límite establecido.
- Middle Mouse Button: Resetea el multiplicador de velocidad al valor predeterminado de 1.0. Configuración dinámica: Activa o desactiva el cambio dinámico de velocidad. Este enfoque permite control manual y flexible sobre la velocidad de movimiento.

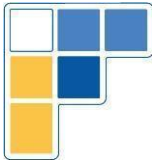


2. Mostrar el cursor

Este Blueprint es el encargado de gestionar si el cursor del mouse debe ser visible durante la interacción.

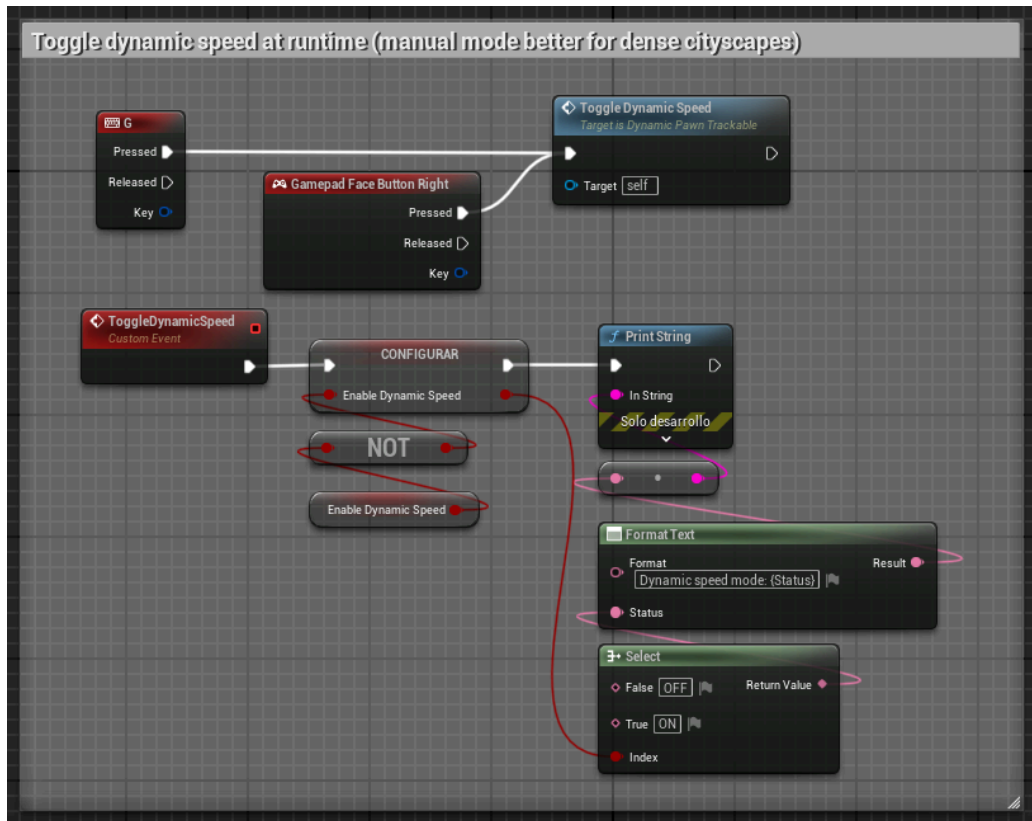


- Botón derecho del ratón (Right Mouse Button): Cambia entre dos modos:
 - Modo de entrada solo para juego (Set Input Mode Game Only): El cursor desaparece, y el control se centra en el juego.
 - Modo de entrada para juego y UI (Set Input Mode Game and UI): El cursor es visible, lo que permite interacción con interfaces gráficas.
- Lógica de cambio: Un nodo de decisión (Branch) determina si el cursor se muestra en función de una condición previa.



3. Alternar velocidad dinámica en tiempo de ejecución

Este Blueprint nos permitirá alternar la velocidad dinámica en tiempo real.



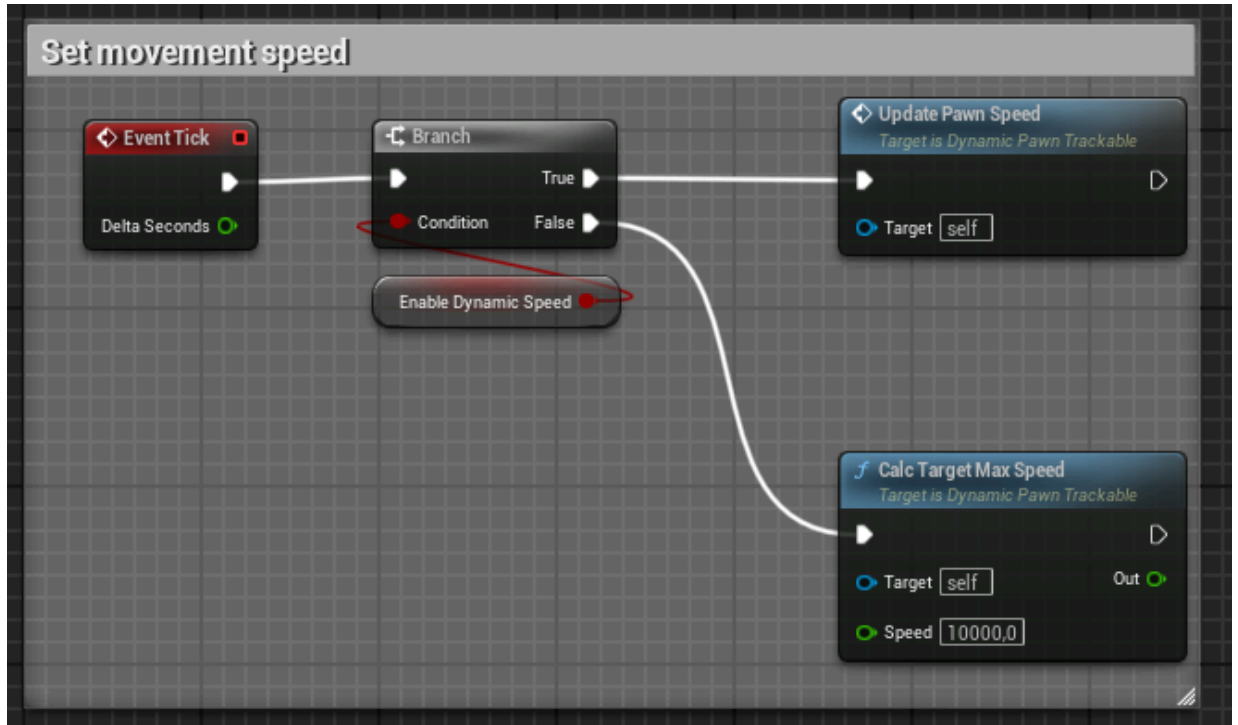
Este blueprint permite alternar la velocidad dinámica en tiempo real.

- Teclas (G / Gamepad Face Button Right): Al presionarlas, ejecutan un evento personalizado (ToggleDynamicSpeed) que cambia el estado de la velocidad dinámica.
- Lógica de alternancia:
 - Utiliza un nodo lógico NOT para invertir el estado actual de la configuración dinámica (Enable Dynamic Speed).
 - Se proporciona retroalimentación al desarrollador con un mensaje (Print String) que muestra el estado actual ("ON" o "OFF") en la consola para depuración.
- Ventaja: Es especialmente útil en contextos como paisajes urbanos densos, donde el movimiento puede requerir ajustes rápidos según la situación.

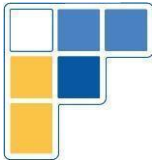


4. Set movement speed

En este BluePrint se muestra una lógica implementada que ajusta dinámicamente la velocidad de movimiento de un actor.

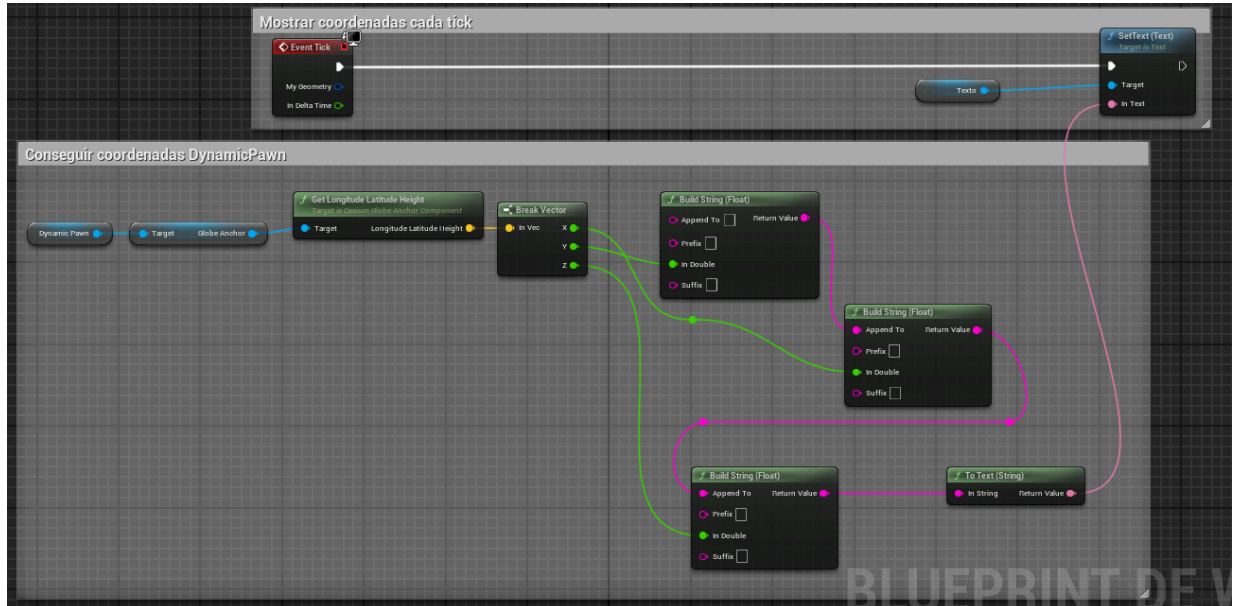


- Event Tick: Se ejecuta cada cuadro para actualizar constantemente el movimiento.
- Branch: Comprueba la condición "Enable Dynamic Speed".
 - Si la condición es verdadera, calcula la velocidad máxima (nodo "Calc Target Max Speed") y la aplica al actor con "Update Pawn Speed".
 - Si es falsa, no se realiza ninguna acción.

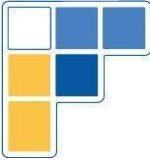


5. Mostrar coordenadas cada tick

Aquí se captura la lógica para mostrar las coordenadas de un actor en pantalla en tiempo real.

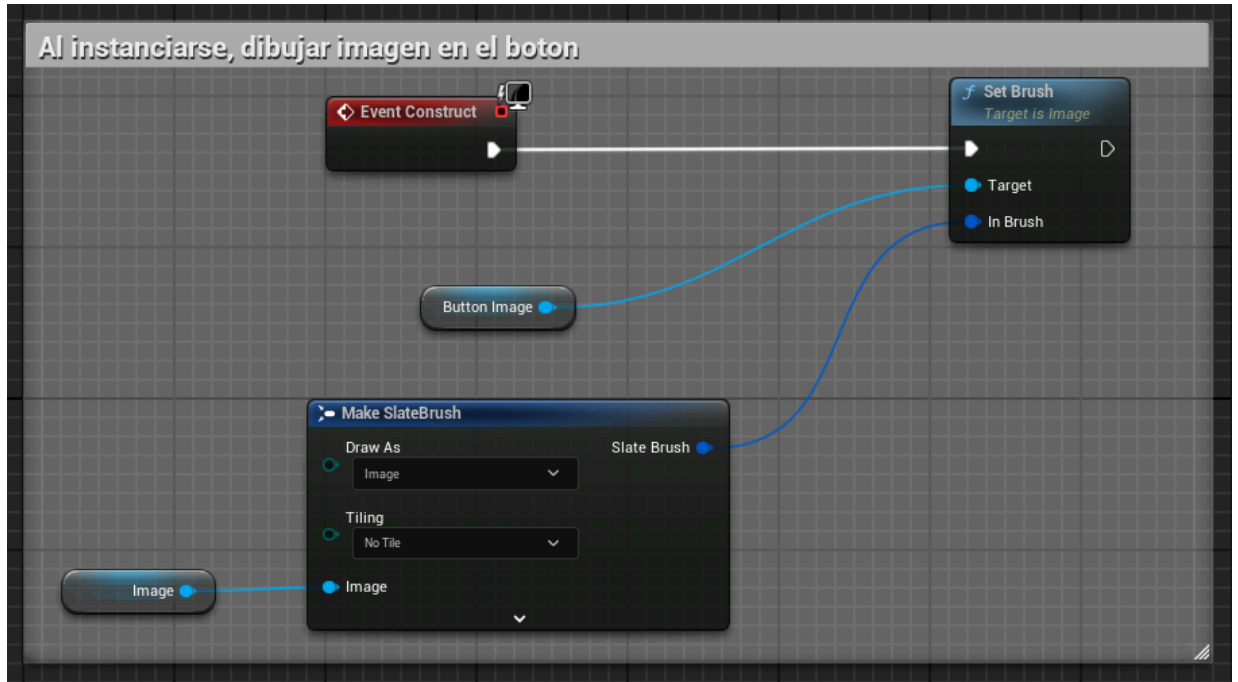


- Event Tick: Se ejecuta constantemente para actualizar la información.
- Get Longitude Latitude Height: Obtiene las coordenadas de un componente anclado a un globo (latitud, longitud y altura).
- Build String (Float): Convierte los valores numéricos (X, Y, Z) en cadenas de texto con formato.
- Set Text: Muestra el resultado en pantalla, posiblemente en un widget.

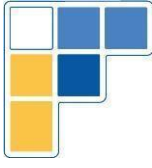


6. Dibujar imagen en el botón

Este gráfico define la lógica para configurar dinámicamente una imagen en un botón al construirse un widget.



- Event Construct: Se ejecuta una vez al inicializar el widget.
- Make SlateBrush: Crea un pincel basado en la textura o imagen que se proporcionó.
- Set Brush: Aplica el pincel al botón para que muestre la imagen deseada.



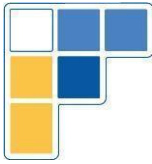
Implementación

1. Interfaz Principal

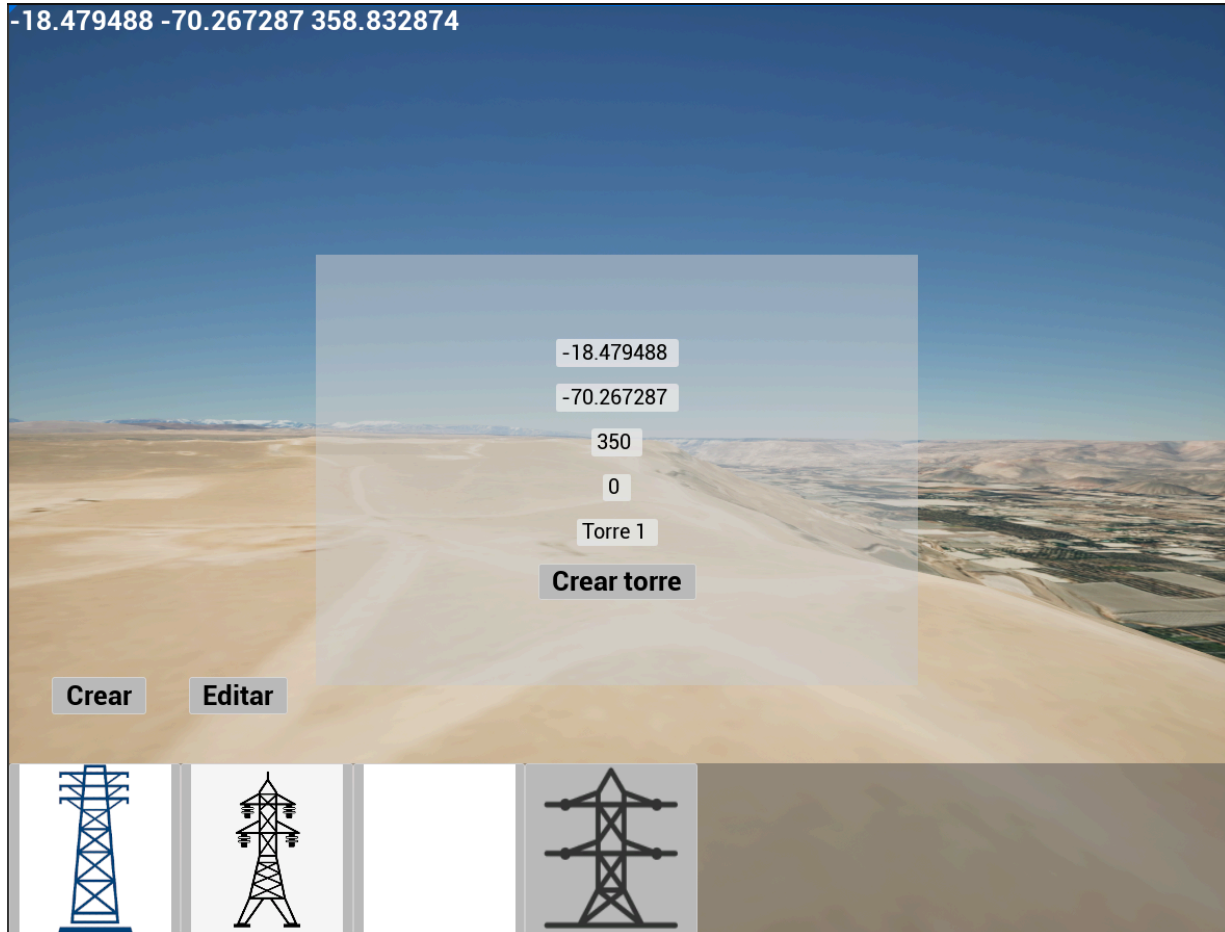


Esta imagen presenta la vista principal de la aplicación, que incluye un modelo 3D del entorno. Desde aquí se tiene acceso a las opciones de creación y edición de objetos. Se destaca la región geográfica con sus características topográficas y urbanas.



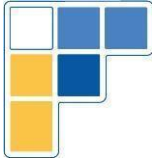


2. Creación de Torre



En esta imagen se muestra la interfaz de usuario utilizada para la creación de torres. Incluye las coordenadas geográficas, la altitud y un campo para personalizar el nombre de la torre. En este caso, se observa un entorno desértico representado en un modelo 3D.

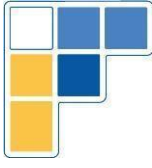




3. Torre Creada



Aquí se muestra una torre ya creada en el entorno virtual. Se observa su posición exacta con coordenadas, además de un diseño tridimensional que incluye el nombre asignado a la torre. En el fondo, se aprecian la ciudad y el océano como parte del modelo geográfico.



Conclusión

En este Segundo avance, se lograron proponer y desarrollar las funcionalidades principales del proyecto, destacando la creación, visualización y edición de torres en un entorno virtual en 3D. Las implementaciones presentadas demuestran cómo los casos de uso cumplen con la principal problemática planteada por el cliente, proporcionando una interfaz intuitiva y precisa que facilita la gestión de los elementos requeridos.

Además, este avance establece una base sólida para el desarrollo completo del proyecto, definiendo una dirección clara para la planificación y ejecución de las actividades necesarias. Con ello, se garantiza que cada etapa esté alineada con la solución de la problemática del cliente, permitiendo un progreso eficiente y efectivo hacia los objetivos finales del proyecto.

referencias y trabajos relacionados