# Level Streaming en un videojuego

## ¿Qué es el Level Streaming en un juego?

El Level Streaming es una técnica para cargar y descargar niveles (o partes del mundo del juego) de manera dinámica, dependiendo de la posición del jugador o ciertos eventos, sin necesidad de pausar el juego o mostrar pantallas de carga.

## ¿Por qué es útil?

- Permite crear mundos más grandes sin que todo esté cargado a la vez.

- Reduce el consumo de memoria y mejora el rendimiento.

- Evita largas pantallas de carga entre zonas.

- Se pueden reutilizar niveles o escenarios como partes de otros niveles.

## ¿Cómo funciona?

Imagina que el mapa de tu juego está dividido en varias zonas: bosque, pueblo, castillo, etc.

- Cuando el jugador se acerca al bosque, se carga (stream) esa zona en segundo plano.

- Al alejarse, el motor puede descargarla para liberar memoria.

- Esto puede basarse en distancia, triggers o límites definidos por el desarrollador.

## En Unreal Engine:

- Los mapas se dividen en niveles (sub-levels).

- Se usa World Composition o World Partition (desde UE5) para organizar el mundo.

- Puedes usar Blueprints o código C++ para cargar/desactivar niveles:

UGameplayStatics::LoadStreamLevel(...)

UGameplayStatics::UnloadStreamLevel(...)

- O en Blueprints: Load Stream Level / Unload Stream Level.

## Tipos de Level Streaming

- Distance-based: Se cargan zonas según lo cerca que esté el jugador.

- Volume-based: Se usan volúmenes invisibles (Streaming Volumes) para decidir qué cargar.

- Manual/Scripted: Se cargan niveles según scripts o lógica del juego.

- Always Loaded: Algunas zonas siempre están cargadas (por ejemplo, HUD o menú).

## Ejemplo visual

Supón que tu mapa es así:

+--------+--------+--------+

| Bosque | Pueblo | Castillo |

+--------+--------+--------+

- Al iniciar, solo el Pueblo está cargado.

- Si el jugador camina hacia el Bosque, se carga esa parte automáticamente.

- Si se aleja del Bosque, este se descarga.

## Ventajas

- Escalabilidad para mundos grandes.

- Tiempo de carga inicial más rápido.

- Mejora de rendimiento general.

- Posibilidad de hacer juegos sin pantallas de carga tradicionales.

## Desventajas

- Puede ser complejo de implementar si no se planifica desde el inicio.

- Mal usado puede causar tirones si los niveles son pesados o se cargan demasiado tarde.