

# Arquitectura de Software

José Ignacio Botero Osorio

**Instructor**

Lyam Acosta Forero

**Aprendiz**

SENA – Servicio Nacional de Aprendizaje  
Tecnología en análisis y Desarrollo de Software

3118526

Bogotá D.C.

## **Contenido**

<b>Arquitectura de Software .....</b>	<b>3</b>
<b>¿Qué entiende por arquitectura de software? .....</b>	<b>3</b>
<b>¿Cuál es su función? .....</b>	<b>3</b>
<b>¿Cómo se elabora la arquitectura de software? .....</b>	<b>4</b>
<b>¿Cómo lograr una buena arquitectura? .....</b>	<b>4</b>
<b>¿Cuáles son los elementos de diseño de una arquitectura de software? .....</b>	<b>4</b>
<b>Aplicación en nuestro proyecto .....</b>	<b>5</b>

# Arquitectura de Software

En esta sección revisarnos la arquitectura de Software de nuestro proyecto, revisando primero algunos conceptos claves.

## ¿Qué entiende por arquitectura de software?

La arquitectura de Software es la estructura y diseño de sistema de un software a alto nivel, define como los componentes del software se organizan, interactúan y cumplen con los requisitos funcionales y no funcionales, como la escalabilidad y el rendimiento.

Es el plano que guía la construcción del sistema y facilita la comprensión y colaboración entre los desarrolladores.

En nuestro proyecto la arquitectura es la estructura organizativa que define como se organizan y comunican los diferentes componentes y subsistemas del juego, como gráficos, físicas, inteligencia artificial y sonidos.

Esta arquitectura permite que el videojuego funcione de manera eficiente, escalable y mantenible durante su desarrollo y evolución.

## ¿Cuál es su función?

La principal función de la arquitectura de Software es:

1. **Definir Estructura:** Establece la organización de componentes y sus relaciones.
2. **Facilitar Comunicaciones:** Permite la interacción entre diferentes módulos del sistema.
3. **Garantizar Escalabilidad:** Asegura que el sistema pueda crecer y adaptarse.
4. **Optimizar Rendimiento:** Mejora del sistema y velocidad del sistema.

## ¿Cómo se elabora la arquitectura de software?

En nuestro videojuego aplicaremos la arquitectura del software así:

1. **Análisis de Requisitos:** Identificar y analizar los requisitos funcionales y no funcionales del sistema.
2. **Identificación de componentes:** Definir los módulos principales y sus responsabilidades.
3. **Diseño de Interacciones:** Establecer como se comunican los componentes entre sí.
4. **Selección de patrones:** Elegir diseño de patrones adecuados para el proyecto.
5. **Documentación:** Crear diagrama y documentación de la arquitectura.

## ¿Cómo lograr una buena arquitectura?

En nuestro juego realizaremos una buena arquitectura con:

1. **Modularidad:** Dividir el sistema en módulos independientes y actualizados.
2. **Reutilización:** Diseñar componentes que puedan ser reutilizados.
3. **Robustez:** Crear un sistema resistente a fallos y errores.
4. **Mantenibilidad:** Facilita la modificación y actualización del código.
5. **Equilibrio:** Balancear rendimiento, escalabilidad y complejidad.
6. **Documentación:** Mantener documentación clara y actualizada

## ¿Cuáles son los elementos de diseño de una arquitectura de software?

Los componentes de diseño de nuestro juego son:

1. **Componentes:** Unidades modulares que encapsulaban funcionalidades específicas.
2. **Conectores:** Mecanismos que permiten la comunicación entre componentes.
3. **Configuraciones:** Arreglo estructural de componentes y conectores.
4. **Restricciones:** Reglas y limitaciones que rigen el diseño.
5. **Atributos de Calidad:** Propiedades no funcionales como rendimiento, seguridad y escalabilidad.

## Aplicación en nuestro proyecto

En nuestro videojuego la arquitectura de software es la estructura y diseño de un sistema de software a nivel alto, nos define componentes del software que se organiza, interactúan y cumplen con los requisitos funcionales y no funcionales, como la escalabilidad y el rendimiento.

Arquitectura de nuestro videojuego:

6. **Capa de Presentación:** Interfaz de usuario, menús, HUD y elementos visuales.
7. **Capa lógica de Juego:** Mecánicas de juego, reglas de negocio y control de flujo.
8. **Capas de renderizado:** Motor gráfico, animaciones y efectos visuales.
9. **Capa de Audio:** Música, efectos de sonido y gestión de audio.
10. **Capa de Datos:** Almacenamiento, guardado de partidas y persistencia.

Es como el plano maestro que nos guía la construcción del sistema.