

# <Super Mario Bros>

## Arquitectura del Sistema

Versión 1.1

**Producido para:**

Play Store Game

**Presentado por:**

<Anngel Ortiz Meraz>

## Historial de Revisiones

Fecha	Versión	Descripción	Autor
28/08/23	1.0	Versión inicial del proyecto	Anngel Ortiz
14/09/23	1.1	Documentación	Anngel Ortiz

# Tabla de contenido

## 1. Introducción

1.1 Propósito

1.2 Alcance

1.3 Definiciones, Acrónimos y Abreviaciones

## 2. Representación de Arquitectura

2.1 Vista de Casos de Uso

2.2 Vista Lógica

2.3 Vista de Procesos

2.4 Vista de Despliegue

2.5 Vista Física

# 1.Introducción

## 1.1 Propósito

El siguiente documento de Arquitectura de Software tiene como fin presentar la arquitectura del sistema de videojuegos Super Mario Bros a través de distintas vistas, donde se ilustran aspectos en particular del software desarrollado. De esta manera, se pretende que el documento proporcione una vista global y comprensible del diseño general del videojuego.

## 1.2 Alcance

El juego se controlará únicamente con el teclado. Además, jugar es muy sencillo ya que el usuario sólo necesita utilizar las teclas de flecha y la barra espaciadora para enviar bolas de fuego cuando Mario está en forma de fuego.

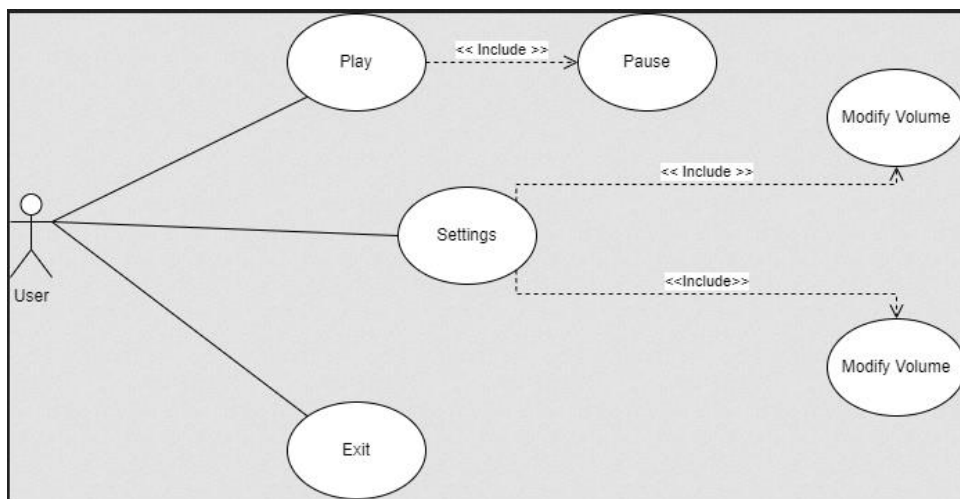
## 1.3 Definiciones, Acrónimos y Abreviaciones

Acrónimo	Definición
MVC	Modelo Vista Controlador; El MVC o Modelo-Vista-Controlador es un patrón de arquitectura de software que, utilizando 3 componentes (Vistas, Models y Controladores) separa la lógica de la aplicación de la lógica de la vista en una aplicación.
Backend	Es la parte o rama del desarrollo de software encargada de que toda la lógica de un sistema funcione.
Frontend	Es la parte de una aplicación que interactúa con los usuarios, es conocida como el lado del cliente. Básicamente es todo lo que vemos en la pantalla cuando accedemos a un sistema software.

## 2. Representación de Arquitectura

### 2.1 Vista de Casos de Uso

Un caso de uso es una técnica de modelado usada para describir lo que debería hacer un sistema nuevo o lo que hace un sistema que ya existe. Los casos de uso describen bajo la forma de acciones y reacciones el comportamiento de un sistema desde el punto de vista de un usuario, permiten definir los límites del sistema y las relaciones entre el sistema y el entorno.



Se realiza una vista general acerca de los casos de uso o escenarios de un usuario o jugador, se define en un panorama funcional del software en alguno de los módulos a desarrollar.

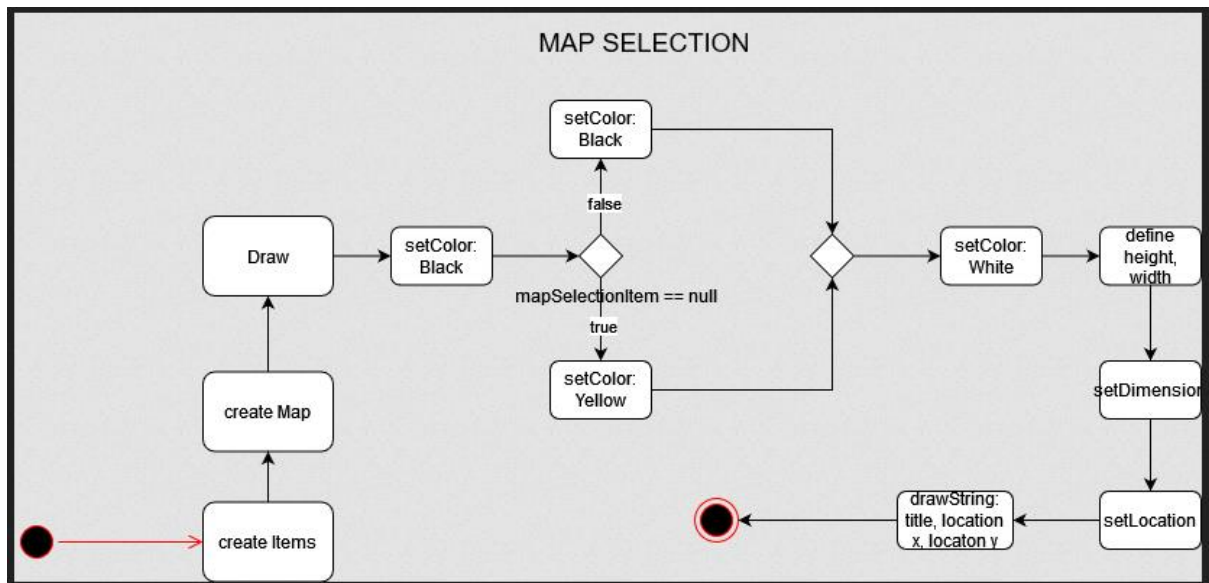
Los casos de uso del videojuego contemplan el siguiente alcance general:

- Jugar el videojuego
- Configuraciones del videojuego
- Salir del juego

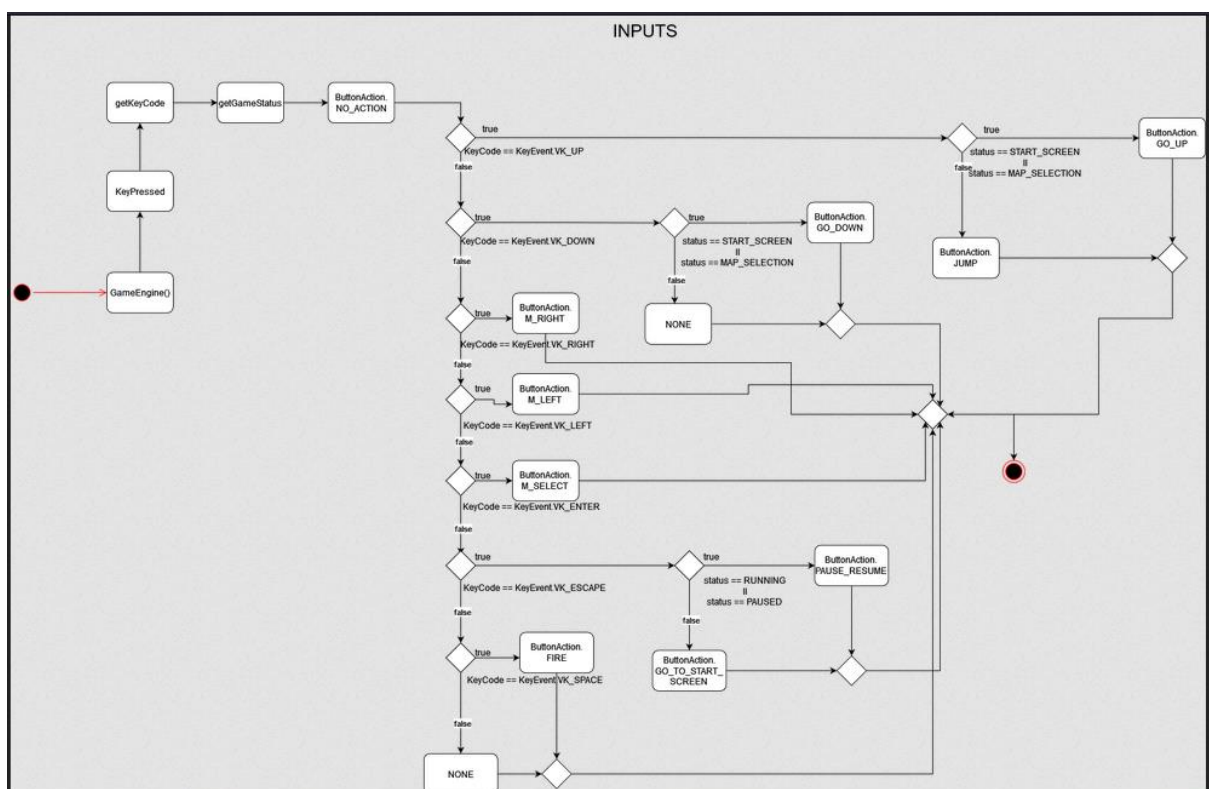
### 2.2 Vista Lógica

Describe las partes de un software relevantes del modelo de diseño, como una vista en capas, subsistemas o paquetes. Una vez presentadas dichas unidades lógicas se profundizan en ellas hasta el nivel considerado adecuado para su comprensión.

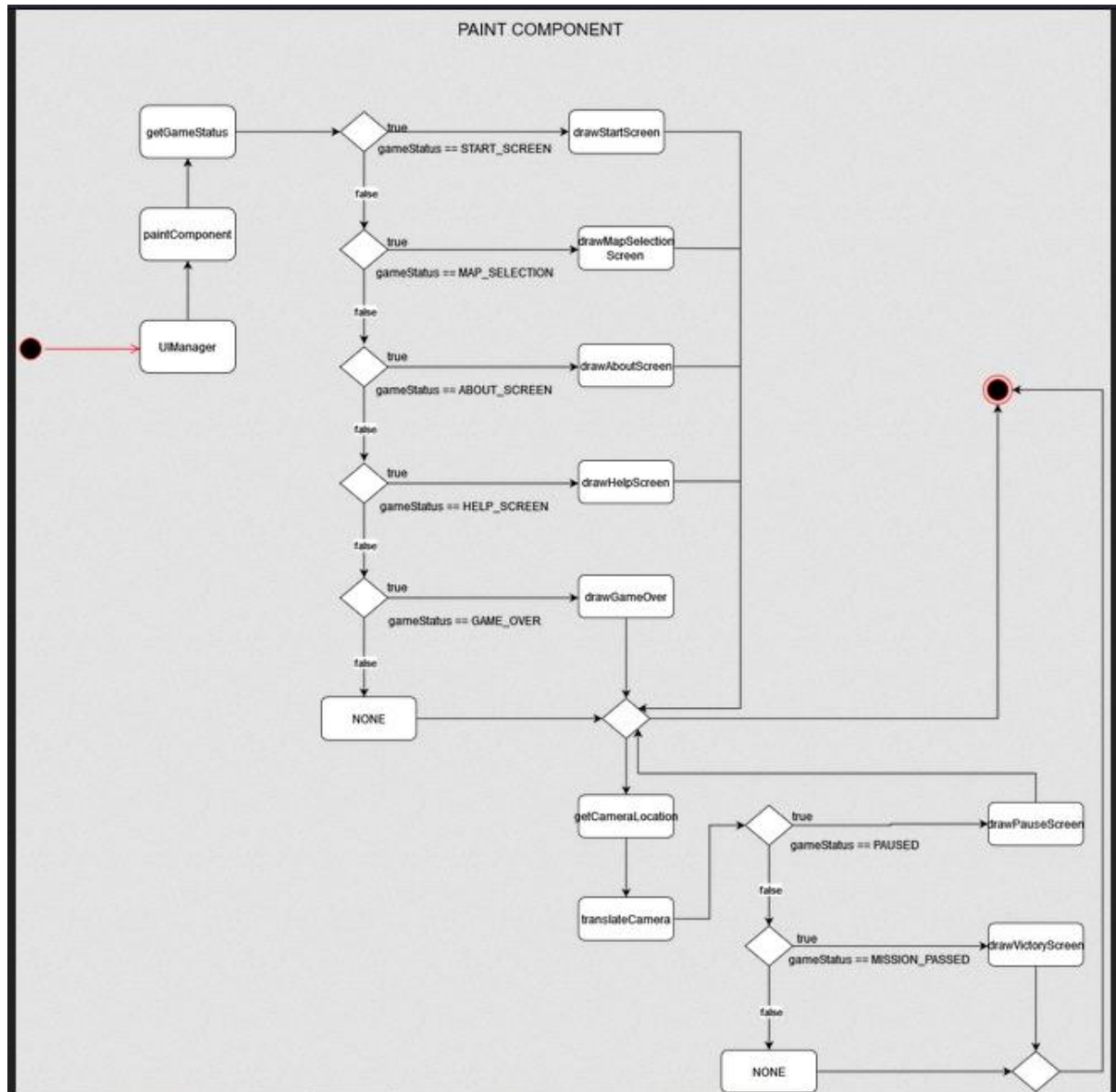
Aquí es donde se documentan los procesos que pertenecen al sistema, especificando su flujo de trabajo paso a paso y las funciones que se cumplen para finalizar algún proceso establecido.



Se presenta el flujo de trabajo que el sistema debe realizar cuando un usuario o jugador desea seleccionar un mapa de juego. Primero deberá instanciarse la clase Createltems donde se cargan algunas imágenes de la interfaz del juego, después se deberá cargar un mapa con la función draw, luego, deberá establecerse un color para la interfaz según se haya creado correctamente la clase MapSelectionItem y por último se establecen las dimensiones y posiciones del juego para dibujarse.



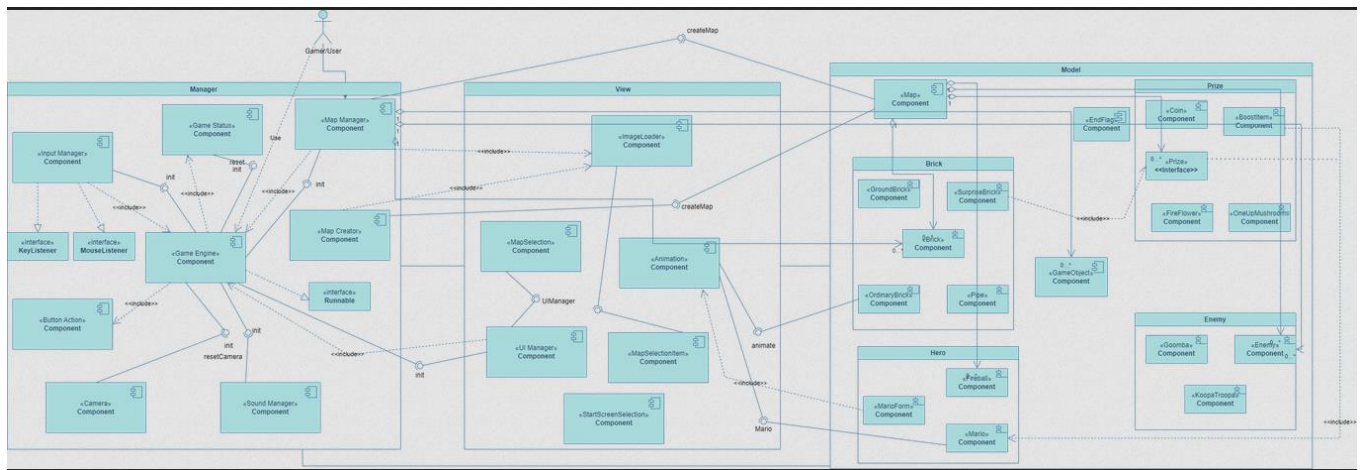
Se presenta el flujo de trabajo que el sistema realiza para detectar los diversos inputs establecidos para el videojuego, donde estos inputs los provee el enumerator ButtonAction y sirve para definir los estados que se tendrán en el sistema.



Se presenta el flujo de trabajo que el sistema realiza para seleccionar algunos inputs establecidos para el sistema, aquí se presentan los inputs de la interfaz gráfica controlada por la clase UIManager para la selección de un estatus de videojuego, como lo sería estar pausado o en game over, dichos estados los provee el enumerator GameStatus.

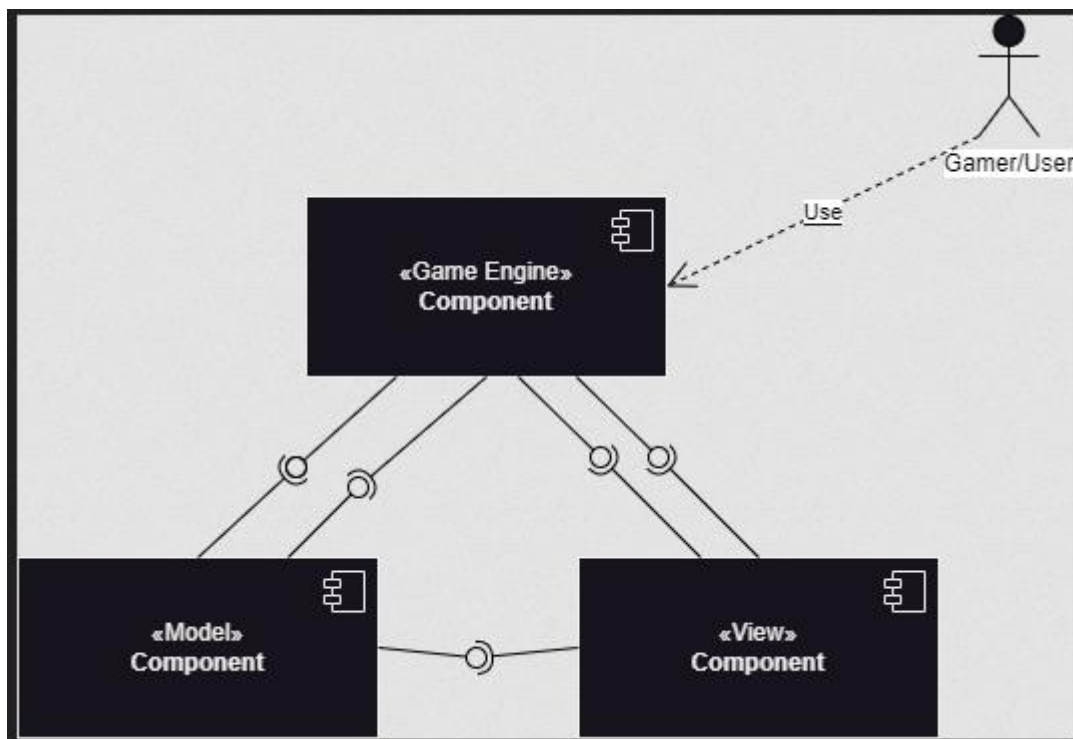


### 2.4.1 Diagrama de Componentes



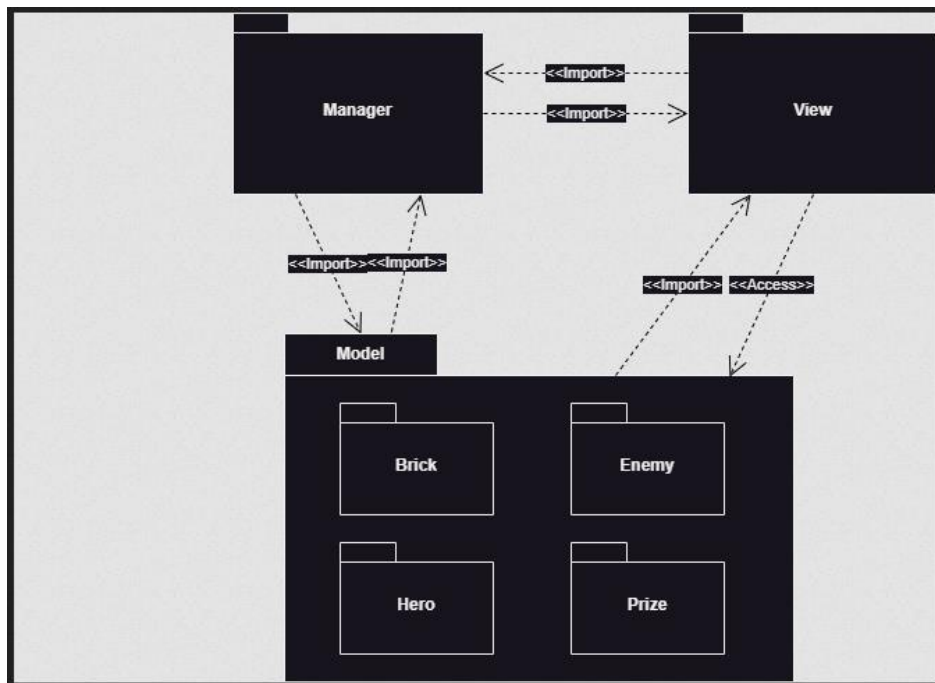
La descomposición del proyecto organiza la arquitectura del sistema de videojuego en un conjunto de subsistemas que interactúan entre sí para cumplir sus objetivos. Los componentes se presentan agrupados en tres contenedores; Manager, View y Model, donde se presentan los componentes principales de cada contenedor y las relaciones existentes con otros componentes de su mismo contenedor y de otros.

### 2.4.2 Diagrama de Componentes MVC



Se realiza una descomposición de arquitectura más general, basado en el patrón MVC (Modelo Vista Controlador), donde el componente Modelo (Modelo en el diagrama) se encarga del backend del sistema, el componente Vista (View en el diagrama) se encarga del frontend e interfaz gráfica del usuario y por último el Controlador (GameEngine en el diagrama) se encarga de controlar los datos a través de View y Model.

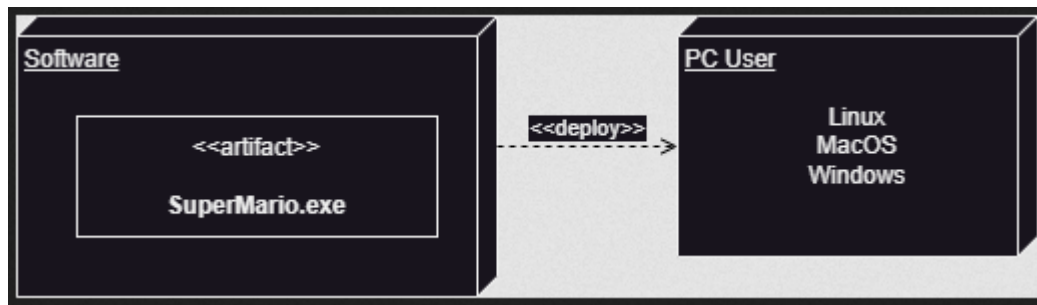
#### 2.4.5 Diagrama de paquetes



Se presentan los paquetes que conforman el sistema y cómo interactúan entre sí, en el videojuego tenemos tres paquetes principales que se encargan de controlar y visualizar los modelos del juego, donde éste último paquete contiene otros cuatro paquetes dentro, los cuales almacenan la información de algunos objetos dentro del juego, personajes y enemigos.

### 2.5 Vista Física

En esta vista se ilustra el hardware y el software del sistema. En general, esta vista ayuda a visualizar los procesadores, los nodos y los dispositivos vinculados. Se basa en los componentes lógicos de un sistema.



El diagrama de despliegue muestra como interactúa el hardware con el software del videojuego, en este caso sólo tenemos como componente de hardware la computadora del usuario o jugador, especificando en qué sistemas operativos es capaz de ejecutarse. Como componente de software tenemos el archivo ejecutable.