

# Manual Técnico

---

## GAME-GROUP4

Versión 1.0.0

Julio 2025

---

### 1. Introducción

Este documento describe la arquitectura, diseño, componentes y tecnologías utilizadas en el desarrollo del juego GAME-GROUP4. Está dirigido a desarrolladores, testers y mantenedores del proyecto para facilitar su comprensión y evolución.

---

### 2. Arquitectura del Sistema

- **Lenguaje:** Java (JDK 8+)
  - **Frameworks/Librerías:**
    - Swing para interfaz gráfica
    - Jamepad para integración con mandos de juego (PS4)
    - SQLite para almacenamiento local de puntajes
  - **Patrones:**
    - Modelo-Vista-Controlador (MVC) parcialmente aplicado para separar la lógica de negocio y la interfaz.
    - Singleton para la gestión de la conexión a la base de datos (ScoreDAO).
- 

### 3. Componentes Principales

Clase	Responsabilidad
App	Punto de entrada; inicializa tabla y pantalla.
PantallaInicio	UI para menú principal; opciones iniciar y salir.
SeleccionEscenario	Gestión de selección de escenario (Día/Noche).
CaminoUniversidad	Lógica principal del juego; manejo de juego y físicas.
ControlPS4	Inicialización y manejo del mando PS4.
ScoreDAO	Operaciones sobre la base de datos (crear tabla, guardar puntajes).
VentanaPuntajes	Interfaz para mostrar puntajes almacenados.

---

### 4. Base de Datos

- **Motor:** SQLite
- **Archivo:** `mi_juego.db`
- **Tabla principal:** `scores`
  - `id`: INTEGER PRIMARY KEY AUTOINCREMENT
  - `puntaje`: INTEGER NOT NULL

Operaciones básicas: creación de tabla al iniciar y guardar puntajes tras finalización de partida.

---

## 5. Flujo Principal del Juego

1. Usuario inicia la aplicación (App) → PantallaInicio aparece.
  2. Usuario selecciona iniciar juego → selecciona escenario Día o Noche (SeleccionEscenario).
  3. Se lanza CaminoUniversidad con escenario seleccionado.
  4. CaminoUniversidad maneja el loop del juego, físicas, obstáculos, puntajes y entradas de mando PS4.
  5. Detecta colisiones y termina la partida si es necesario.
  6. Al terminar, guarda puntaje con ScoreDAO y muestra VentanaPuntajes.
  7. Usuario puede reiniciar o volver al menú desde botones o mando.
- 

## 6. Control y Entrada

- La clase **ControlPS4** inicializa el mando mediante Jamepad.
  - CaminoUniversidad lee inputs del mando (botón X para salto, botón Start para menú).
  - También se soportan entradas por teclado (barra espaciadora para saltar, ESC para menú).
- 

## 7. Manejo de Gráficos y UI

- Swing y JPanel usados para renderizado personalizado en CaminoUniversidad.
  - BufferedImage para cargar sprites e imágenes de fondo.
  - Temporizadores Swing (**Timer**) para control de frames (60 FPS) y aparición de obstáculos (cada 1.5 segundos).
  - Pantalla final con overlay semi-transparente y botones interactivos.
- 

## 8. Recomendaciones para Mantenimiento

- Separar lógica de físicas y renderizado en clases distintas para mejorar mantenimiento.
  - Externalizar rutas y configuración (imágenes, puntajes) a archivos de configuración.
  - Implementar manejo de excepciones más robusto, especialmente en carga de recursos y base de datos.
  - Añadir pruebas unitarias para lógica crítica (colisiones, puntajes).
- 

## 9. Herramientas y Entorno de Desarrollo

- **IDE recomendado:** Visual Studio Code con extensiones Java o IntelliJ IDEA
- **Compilación:** JDK 8 o superior
- **Control de versiones:** Git y GitHub
- **Dependencias:** Jamepad para el mando PS4

---

## 10. Créditos Técnicos

- Desarrollo y diseño: Grupo 4 - Programación II
  - Testing y documentación: Equipo GAME-GROUP4
-