**UNIVERSIDAD TECNOLÓGICA DE CIUDAD JUÁREZ**



SEP CGUT

**“DESARROLLO DE APLICACION PARA JUEGOS EN UNITY 3D Y REALIDAD AUMENTADA CON VUFORIA”**

MEMORIA DE ESTADÍA QUE PRESENTA:

**ANGEL DANIEL HERNANDEZ MARTINEZ**

PARA OBTENER EL TÍTULO DE

TÉCNICO SUPERIOR UNIVERSITARIO EN:

**TECNOLOGÍAS DE INFORMACIÓN Y COMUNICACIÓN, ÁREA DESARROLLO DE SOFTWARE MULTIPLATAFORMA**

CD. JUÁREZ, CHIH. AGOSTO DE 2025

# AGRADECIMIENTOS

Durante mi estadía profesional en **Mesh Robotix**, tuve la oportunidad de desarrollarme en un entorno técnico enriquecedor y desafiante. Agradezco profundamente al director **José Roberto Carlos** por brindarme la oportunidad de integrarme a la empresa, así como su disposición y orientación durante el proceso.

Mi reconocimiento también va dirigido al equipo del área de desarrollo, cuyo acompañamiento fue fundamental para llevar a cabo las tareas asignadas y lograr un aprendizaje significativo. Esta experiencia ha representado una etapa clave en mi formación, permitiéndome crecer tanto a nivel técnico como profesional.

# ÍNDICE

[AGRADECIMIENTOS i](#_Toc207272699)

[ÍNDICE ii](#_Toc207272700)

[RESUMEN vi](#_Toc207272701)

[CAPÍTULO I 1](#_Toc207272702)

[1.- INTRODUCCIÓN 1](#_Toc207272703)

[1.1.- ANTECEDENTES DE LA EMPRESA 1](#_Toc207272704)

[1.2.- Definición del problema 2](#_Toc207272705)

[1.3.- Justificación 2](#_Toc207272706)

[1.4.- Objetivo y metas 2](#_Toc207272707)

[CAPÍTULO II 4](#_Toc207272708)

[2.- Marco de referencia 4](#_Toc207272709)

[CAPÍTULO III 13](#_Toc207272710)

[3.- Desarrollo 13](#_Toc207272711)

[3.1 Metodología Cascada (Waterfall) Seleccionada 13](#_Toc207272712)

[3.1.1Justificación de la Metodología: 13](#_Toc207272713)

[3.1.2 Fases de la Metodología Cascada 14](#_Toc207272714)

[3.2 Tabla de Calendarización de Actividades del Proyecto 16](#_Toc207272715)

[3.3 Diagrama de Gantt de la tabla de actividades. 17](#_Toc207272716)

[3.4 Recursos necesarios para desarrollo del proyecto (materiales, humanos y financieros) con descripción y costo. 18](#_Toc207272717)

[3.5 Análisis de Requerimientos funcionales y no funcionales del proyecto. 19](#_Toc207272718)

[3.6 Diagramas Casos de Uso general y especifico con su tabla de descripción. 26](#_Toc207272719)

[3.7 Diagrama Clases del sistema. 27](#_Toc207272720)

[3.8 Diagrama entidad relación de Base de datos. 29](#_Toc207272721)

[3.9 Diseño Tablas relacionadas de la base de datos, mínimo tres tablas. 29](#_Toc207272722)

[3.10 Diseño lógico de interfaces de la aplicación (menús, formularios). 30](#_Toc207272723)

[Diagrama Grantt explicado 31](#_Toc207272724)

[PRIMERA FASE: INVESTIGACIÓN (SEMANAS 1-2) 31](#_Toc207272725)

[SEGUNDA FASE: CREAR LA APLICACIÓN (SEMANAS 3-10) 32](#_Toc207272726)

[TERCERA FASE: PROBAR Y VERIFICAR (SEMANAS 11-12) 33](#_Toc207272727)

[CAPÍTULO IV 35](#_Toc207272728)

[4.- Resultados y recomendaciones 35](#_Toc207272729)

[4.1 Pantallas lógicas, resultado del programa, con descripción del mismo 35](#_Toc207272730)

[4.2 Sugerencias de Mejora 38](#_Toc207272731)

[6.-Fuentes consultadas 39](#_Toc207272732)

Tabla de ilustraciones

[Ilustración 1 Logo de Unity (1000MARCAS, 2025) 4](#_Toc207272662)

[Ilustración 2 Escena en Unity (0k\_Drive\_9247, 2025) 4](#_Toc207272663)

[Ilustración 3 GameObject (Documentation, 2025) 4](#_Toc207272664)

[Ilustración 4 Prefabs (Documentation, Unity Documentation, 2016) 5](#_Toc207272665)

[Ilustración 5 Muestar de Rigidbody (blog, 20) 5](#_Toc207272666)

[Ilustración 6 Box Collider (Perez, 2020) 5](#_Toc207272667)

[Ilustración 7 Diagrama de joint (Documentation, Unity Documentation, 2025) 6](#_Toc207272668)

[Ilustración 8 Angular Drag (French, 2024) 6](#_Toc207272669)

[Ilustración 9 Inercia y fuerza de un objecto en especifico (French, 2024) 6](#_Toc207272670)

[Ilustración 10 Medidas de un Capsule Collider (Documentation, Unity Documentation, 2025) 7](#_Toc207272671)

[Ilustración 11 La sphere collider agrega 3 lados x, y,z (Documentation, Unity Documentation, 2025) 7](#_Toc207272672)

[Ilustración 12 Logo blender (StickyPNG, 2025) 7](#_Toc207272673)

[Ilustración 13 Modelado 3d en blender (BLENDER, 2025) 8](#_Toc207272674)

[Ilustración 14 Materiales en blender (BLENDER, 2025) 8](#_Toc207272675)

[Ilustración 15 UV MAPPING (BLENDER, 2025) 8](#_Toc207272676)

[Ilustración 16 Ejemplo de Realidad aumentada Unity (Documentation, Unity Documentation, 2025) 9](#_Toc207272677)

[Ilustración 17 andoid vs ios (techtudo, 2023) 9](#_Toc207272678)

[Ilustración 18 Logo de c# (concepto, 2024) 9](#_Toc207272679)

[Ilustración 19 Logo de C++ (concepto, 2024) 10](#_Toc207272680)

[Ilustración 20 Logo de python (concepto, 2024) 10](#_Toc207272681)

[Ilustración 21 NPC ejemplo (movil, 2024) 10](#_Toc207272682)

[Ilustración 22 Paradigma (ACADEMIC, 2023) 11](#_Toc207272683)

[Ilustración 23 Logo NET (concepto, 2024) 11](#_Toc207272684)

[Ilustración 24 Logo de Visual Studio code (CODE, 2025) 12](#_Toc207272685)

[Ilustración 25 Ejemplo de Script (concepto, 2024) 12](#_Toc207272686)

[Ilustración 26 Diagrama Casos de Uso general 26](#_Toc207272687)

[Ilustración 27 Diagrama de Capa Vista 27](#_Toc207272688)

[Ilustración 28 Diagrama de Capa controlador 27](#_Toc207272689)

[Ilustración 29 Diagrama de Capa Modelo 28](#_Toc207272690)

[Ilustración 30 Diagrama de Entidad relacion 29](#_Toc207272691)

[Ilustración 31 DIagramas de base de datos 29](#_Toc207272692)

[Ilustración 32 Diseño logico de Interfaces de la aplicacion 30](#_Toc207272693)

[Ilustración 33 Menu de la aplicacion RoboJam 35](#_Toc207272694)

[Ilustración 34 Permiso de la camara de RoboJam 36](#_Toc207272695)

[Ilustración 35 Codigo Escaneado adecuadamente de RoboJam 36](#_Toc207272696)

[Ilustración 36 Campo visualizado de RoboJam 37](#_Toc207272697)

[Ilustración 37 Gestion de codigo QR de RoboJam 37](#_Toc207272698)

Índice de tablas

[Tabla 1 Tabla de Calendarización de Actividades 16](#_Toc207272658)

[Tabla 2 Diagrama de Grantt 17](#_Toc207272659)

[Tabla 3 Requerimientos y costos 18](#_Toc207272660)

# RESUMEN

El presente documento expone las actividades realizadas durante mi estancia en la empresa Mesh Robotix, donde me desempeñe como Desarrollador de Juegos en Unity 3D y Realidad Aumentada con Vuforia.

Mi trabajo se centró en el desarrollo de aplicaciones interactivas mediante el uso de herramientas como UNITY, BLENDER, Autodesk Inventor, Python, C#, C++ y visual Studio. Las tareas incluyeron el diseño y modelado de entornos tridimensionales, la integración de modelos en aplicaciones de realidad aumentada, y la programación de funcionalidades para mejorar la experiencia del usuario.

Uno de los logros más destacados fue la implementación de un proyecto en realidad aumentada, donde pude aplicar conocimientos en programación y diseño digital, desarrollando soluciones funcionales dentro de un entorno profesional. Así mismo, reforcé el uso de BLENDER y Python, ampliando mis habilidades en diseño y automatización.

Esta experiencia me permitió comprender mejor el flujo de trabajo en el desarrollo de tecnologías interactivas, al mismo tiempo que fortalecí mis capacidades de adaptación, análisis y resolución de problemas en contextos reales del sector tecnológico.

# CAPÍTULO I

## 1.- INTRODUCCIÓN

El presente informe detallara las actividades que se realizaran durante el periodo de estancia profesional en la empresa Mesh Robotix, específicamente dentro del área enfocada al desarrollo de aplicaciones educativas mediante tecnologías emergentes como la realidad aumentada y el modelado 3D. Esta experiencia significara una valiosa oportunidad para aplicar los conocimientos adquiridos durante la formación académica, enfrentando situaciones reales que exigirán soluciones fundamentales y el dominio de herramientas tecnológicas.

Durante la estancia, se participará en el uso de plataformas como Unity y BLENDER. Además, se iniciará el aprendizaje practico del lenguaje Python, lo cual será utilizado en algunos procesos de automatización y lógica de programación dentro de los proyectos de la empresa.

A través de esta experiencia, será posible integrarse a un equipo multidisciplinario y conocer el fuljo de trabajo en una empresa tecnológica que impulsará la enseñanza STEM mediante videojuegos, simuladores y soluciones digitales adaptadas al entorno educativo.

### 1.1.- ANTECEDENTES DE LA EMPRESA

Mesh Robotix es una empresa ubicada en Ciudad Juárez, cuyo enfoque principal es promover la ciencia y la tecnología a través de la robótica educativa y la creación de contenidos digitales para el aprendizaje. Su oferta incluye talleres, capacitaciones y desarrollo de software, combinando metodologías innovadoras con herramientas interactivas.

El público objetivo de la empresa abarca desde niños y adolescentes hasta jóvenes universitarios interesados en áreas como programación, robótica y diseño 3D. Así mismo, Mesh Robotix colabora como instituciones educativas y organizaciones privadas para brindar soluciones personalizada que integran tecnología en entornos formativos.

La empresa se ha posicionado como un referente regional en la integración de tecnologías emergentes, gracia a su enfoque practico y a su capacidad de crear entornos interactivos para la enseñanza de conceptos científicos y técnicos. (Carlos, 2025)

### 1.2.- Definición del problema

La falta de un entorno de simulación para que los estudiantes puedan realizar prácticas, aunque no se cuente con el equipo físico para realizarse, además de la falta de equipo físico o simulado para realizar pruebas de funcionamiento de los robots genera que los usuarios no puedan ir preparados a sus competencias.

### 1.3.- Justificación

El uso de videojuegos simulados mediante UNITY con BLENDER para practicar el uso de equipos robóticos permite a los usuarios interactuar de manera más fácil y rápida.

### 1.4.- Objetivo y metas

#### 1.4.1.- Objetivo general

Desarrollar y mejorar las aplicaciones interactivas en Mesh Robotix, mediante la programación de videojuegos en Unity 3D y proyectos de realidad aumentada, integrando recursos 3D y optimizando procesos mediante automatización

#### 1.4.2.- Objetivos específicos

* Identificar las necesidades del usuario
* Diseñar un plan de trabajo para mejorar el entorno que se tiene
* Desarrollar videojuegos y aplicaciones en UNITY 3D con integración de realidad aumentada utilizando VUFORIA
* Realizar pruebas de funcionamiento
* Integrar módulos al proyecto ya establecido

#### 1.4.3.- Metas

* Establecer un sistema eficiente para las prácticas para los usuarios
* Mejorar el proceso de aprendizaje mediante la virtualización

# CAPÍTULO II

## 2.- Marco de referencia

**2.1 Unity 3D**: Es un motor de videojuegos y desarrollo de experiencias interactivas en 2D/3D. Soporta múltiples plataformas (PC, móvil, consolas, AR/VR). (Alarcón, 2006)



Ilustración Logo de Unity (1000MARCAS, 2025)

* 1. **Escena (Scene):** Espacio donde se colocan objetos (GameObjects) para construir niveles. (Learning, 2025)

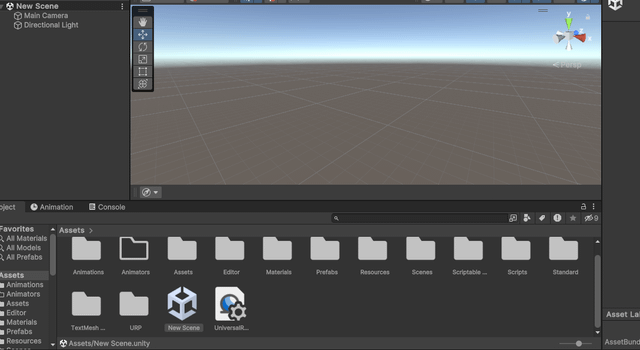


Ilustración Escena en Unity (0k\_Drive\_9247, 2025)

* 1. **GameObject:** Elemento básico de UNITY (personajes, luces, cámaras). (Solutions, 2025)

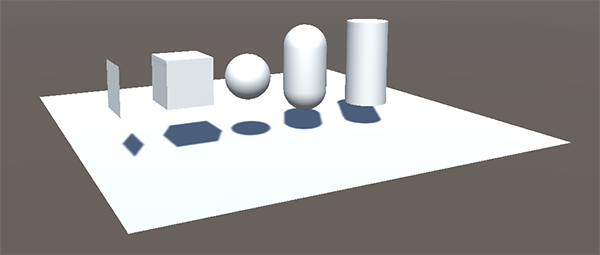


Ilustración GameObject (Documentation, 2025)

* 1. **Prefabs:** Plantillas reutilizables de GameObjects. (Documentation, Unity Documentation, 2016)

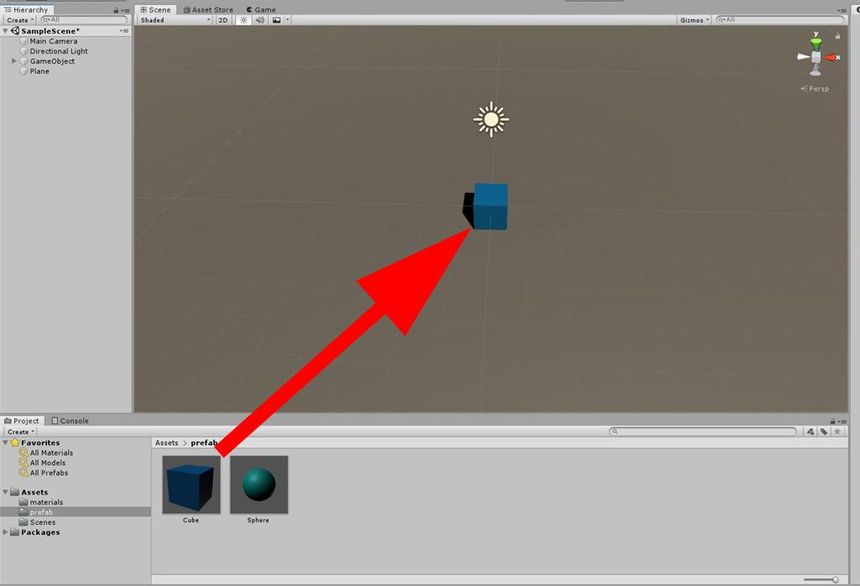


Ilustración Prefabs (Documentation, Unity Documentation, 2016)

* 1. **Rigidbody:** Componente que permite que un objeto sea afectado por gravedad y fuerzas. (Documentation, Unity Documentation, 2018)

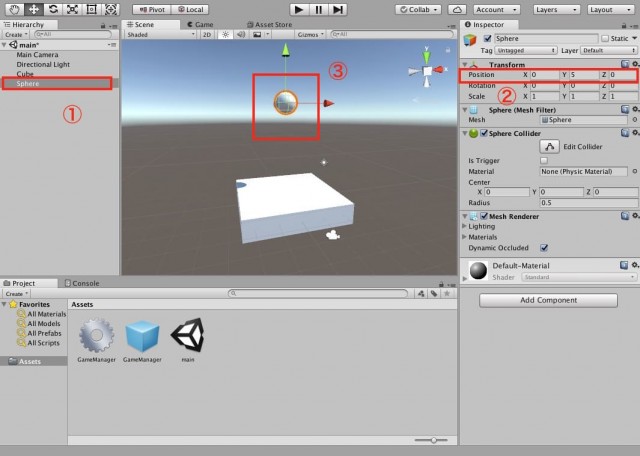


Ilustración Muestar de Rigidbody (blog, 20)

* 1. **Collider:** Formas de colisión (Box, Sphere, Mesh) que detectan contactos. (Perez, 2020)

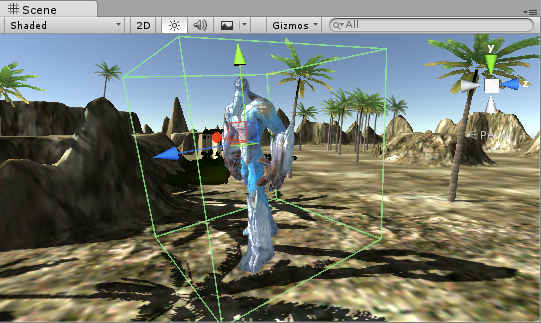


Ilustración Box Collider (Perez, 2020)

* 1. **Joint:** Restricciones físicas (ej: bisagras, resortes). (Documentation, Unity Documentation, 2025)

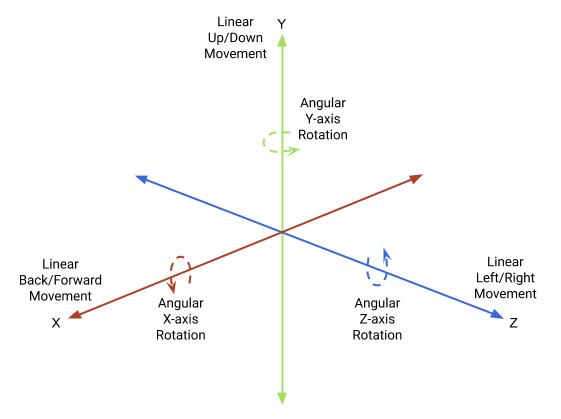


Ilustración Diagrama de joint (Documentation, Unity Documentation, 2025)

* 1. **Box Collider**: cubo (óptimo para paredes, suelos). (Documentation, Unity Documentation, 2025)
  2. **Angular Drag:** Frena giros. (Documentation, Unity Documentation, 2025)

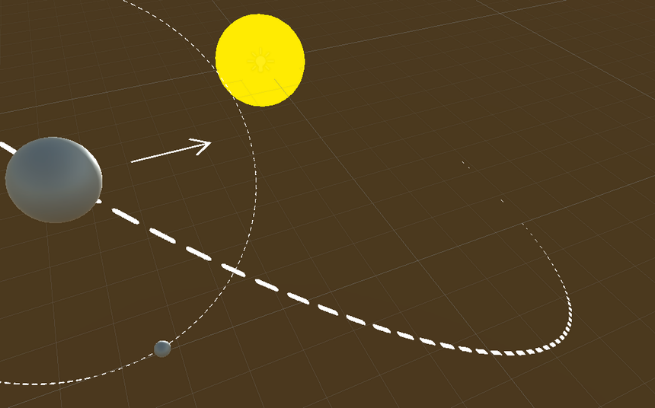


Ilustración Angular Drag (French, 2024)

* 1. **Mass:** Afecta inercia y fuerza (Documentation, Unity Documentation, 2025).

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

Ilustración Inercia y fuerza de un objecto en especifico (French, 2024)

* 1. **Capsule Collider:** Capsula (para personajes humanos). (Documentation, Unity Documentation, 2025)

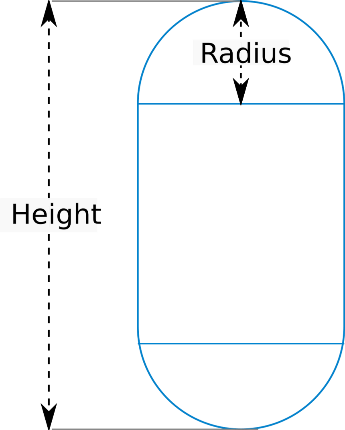


Ilustración Medidas de un Capsule Collider (Documentation, Unity Documentation, 2025)

* 1. **Sphere Collider:** física para ruedas. (Documentation, Unity Documentation, 2025)



Ilustración La sphere collider agrega 3 lados x, y,z (Documentation, Unity Documentation, 2025)

* 1. **Blender:** es un software gratuito y de código abierto para modelado 3D, animación, escultura, renderizado y más. Se usa en videojuegos, cine, y diseño 3D. (BLENDER, 2025)



Ilustración Logo blender (StickyPNG, 2025)

* 1. **Modelado 3D**: Crear objetos con vértices, aristas, y caras (ej. Personajes, edificios). (BLENDER, 2025)



Ilustración Modelado 3d en blender (BLENDER, 2025)

* 1. **Texturizar y Materiales**: Pintar superficies, y ajustar brillo/transparencia. (BLENDER, 2025)

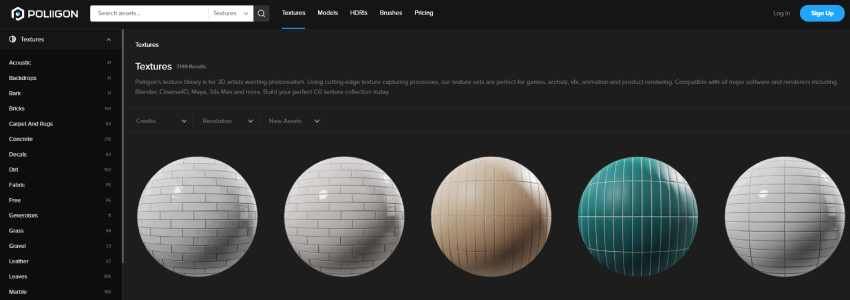


Ilustración Materiales en blender (BLENDER, 2025)

* 1. **UV Mapping:** Desplegar la malla en 2D para aplicar texturas. (BLENDER, 2025)

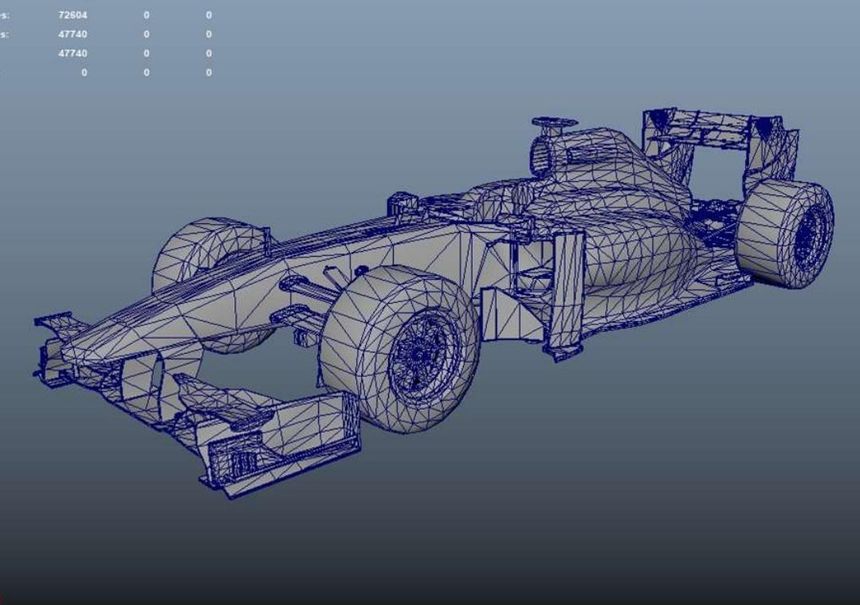


Ilustración UV MAPPING (BLENDER, 2025)

* 1. **Realidad aumentada (AR):** en unity permite superponer objetos virtuales en el mundo real usando la cámara de un dispositivo móvil o gafas AR. Unity ofrece herramientas para desarrollar experiencias AR multiplataforma. (Documentation, Unity Documentation, 2025)

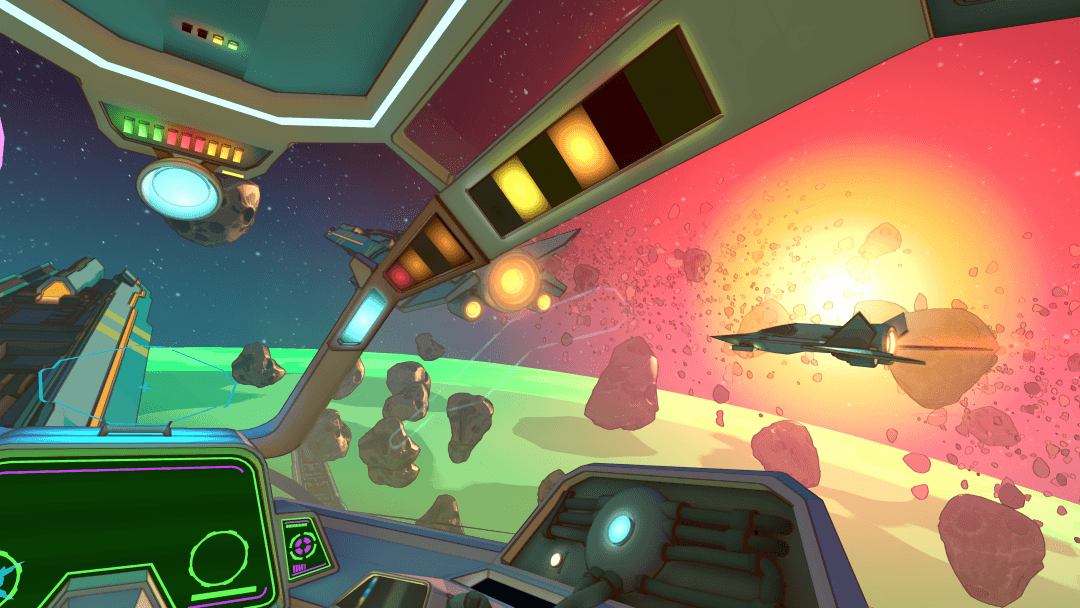


Ilustración Ejemplo de Realidad aumentada Unity (Documentation, Unity Documentation, 2025)

* 1. **SDKS Nativos**: ARKit (iOS): seguimientos de superficies, oclusión, face tracking (REDDIT, 2022)



Ilustración andoid vs ios (techtudo, 2023)

* 1. **C#:** Orientado a objetos, es similar a java, con una sintaxis clara. Integrado en unity se usa para scripts que controlan la lógica del juego. (concepto, 2024)

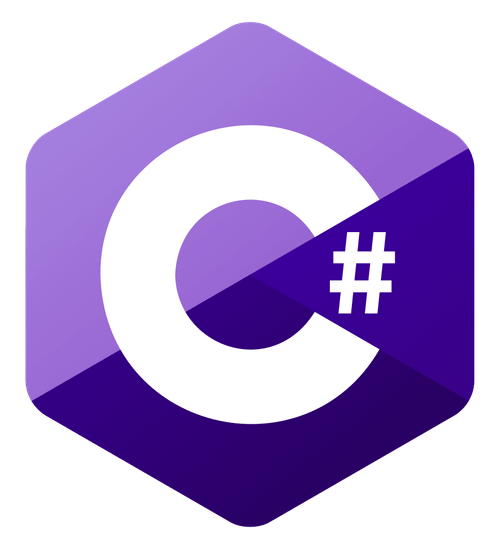


Ilustración Logo de c# (concepto, 2024)

* 1. **C++:** Usado en motores avanzados y plugin, ideal para motores gráficos como unreal engine, gestiona manual de memoria (punteros), en unity se usa para pugins de rendimiento que es para cálculos intensivos como simulaciones físicas personalizadas. (concepto, 2024)

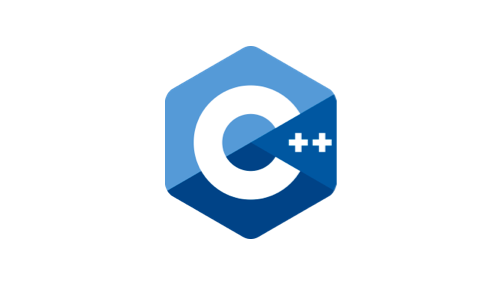


Ilustración Logo de C++ (concepto, 2024)

* 1. **Python:** automatización e IA, en blender el scripting automatiza tareas repetitivas como la generación de terrenos procedurales, en machine learning se una para entrenar modelos de IA para integraros en unity como NPCs con comportamiento inteligente. (concepto, 2024)

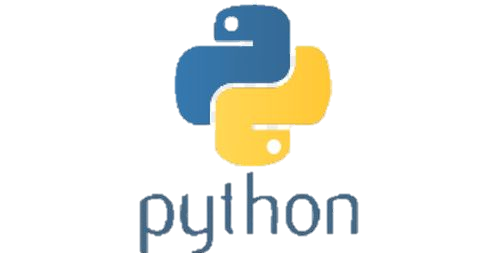


Ilustración Logo de python (concepto, 2024)

* 1. **NPC:** siglas de "Non Playable Character" o "Personaje No Jugable" en español, es un personaje dentro de un videojuego que no es controlado directamente por el jugador. (movil, 2024)



Ilustración NPC ejemplo (movil, 2024)

* 1. **Multiparadigma:** es un enfoque flexible para el desarrollo de software que enfatiza el uso de diferentes paradigmas, técnicas y conceptos de programación para resolver eficientemente una amplia gama de problemas en ingeniería de software. (ACADEMIC, 2023)



Ilustración Paradigma (ACADEMIC, 2023)

* 1. **NET:** es una plataforma de desarrollo de software gratuita, de código abierto y multiplataforma que permite crear una amplia variedad de aplicaciones, incluyendo web, móviles, de escritorio, servicios, microservicios, juegos e Internet de las Cosas (IoT). (concepto, 2024)

Icono

El contenido generado por IA puede ser incorrecto.

Ilustración Logo NET (concepto, 2024)

* 1. **Visual Studio:** es un entorno de desarrollo integrado (IDE) desarrollado por Microsoft que permite a los programadores crear aplicaciones para diversas plataformas, incluyendo Windows, Android, iOS, Linux y macOS. (CODE, 2025)

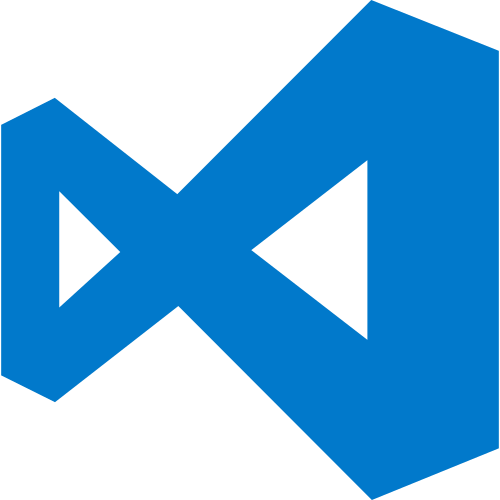


Ilustración Logo de Visual Studio code (CODE, 2025)

* 1. **Scripts:** es un conjunto de instrucciones o comandos escritos en un lenguaje de programación que se utiliza para automatizar tareas, manipular sistemas o añadir funcionalidad a aplicaciones y páginas web (concepto, 2024)

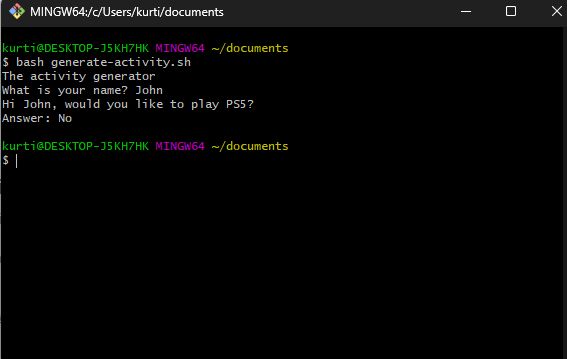


Ilustración Ejemplo de Script (concepto, 2024)

# CAPÍTULO III

## 3.- Desarrollo

### 3.1 Metodología Cascada (Waterfall) Seleccionada

Para el desarrollo de la aplicación móvil de entrenamiento RoboJam utilizando Unity 3D y Vuforia, se ha seleccionado la metodología de cascada (Waterfall) debido a las características específicas del proyecto académico.

### 3.1.1Justificación de la Metodología:

#### Ventajas para el Proyecto:

* Documentación Exhaustiva: Requerida para memoria de estadía universitaria
* Planificación Detallada: Timeline fijo de 13 semanas con entregables específicos
* Control de Calidad: Validación completa de cada fase antes de continuar
* Presupuesto Predecible: Recursos y costos definidos desde el inicio
* Requerimientos Estables: Competencias RoboJam bien establecidas
* Equipo Pequeño: Un desarrollador principal facilita el control secuencial

#### Características del Proyecto que Favorecen Cascada:

* Proyecto académico con fechas fijas de entrega
* Requerimientos claros y bien definidos (reglas RoboJam)
* Entregable único al final del período
* Necesidad de documentación detallada para evaluación académica
* Recursos limitados y presupuesto fijo

### 3.1.2 Fases de la Metodología Cascada

#### Fase 1: Análisis de Requerimientos (2 semanas)

**Objetivos:**

* Investigar a fondo las competencias RoboJam oficiales
* Identificar necesidades de entrenamiento de estudiantes
* Definir requerimientos funcionales y no funcionales
* Analizar tecnologías AR disponibles (Vuforia vs ARCore vs ARKit)

**Actividades Específicas:**

* Entrevistas con instructores de robótica
* Estudio de documentación oficial de competencias RoboJam
* Definición de alcance y limitaciones del proyecto

#### Fase 2: Diseño del Sistema (2 semanas)

**Objetivos:**

* Diseñar arquitectura completa del sistema
* Crear modelos de base de datos
* Definir interfaces de usuario para AR
* Planificar integración Unity-Vuforia

**Actividades Específicas:**

* Diseño de base de datos relacional
* Mockups de interfaces AR
* Arquitectura de componentes Unity

#### Fase 3: Implementación (6 semanas)

**Objetivos:**

* Integrar componentes Unity con Vuforia
* Implementar lógica de ejercicios de entrenamiento
* Crear sistema de puntuación y progreso

**Subdivisión por Sprints:**

* **Sprint 1 (Sem 1-2)**: Configuración Unity + módulos básicos
* **Sprint 2 (Sem 3-4)**: Integración AR + modelos 3D
* **Sprint 3 (Sem 5-6)**: Sistema de ejercicios + puntuación

#### Fase 4: Testing y Verificación (2 semanas)

**Objetivos:**

* Validar requerimientos funcionales y no funcionales
* Optimizar rendimiento AR
* Corregir errores críticos

**Tipos de Pruebas:**

* Pruebas unitarias de componentes
* Pruebas de integración AR
* Pruebas de usabilidad con usuarios reales
* Pruebas de rendimiento en múltiples dispositivos

#### Fase 5: Deployment y Documentación (1 semana)

**Objetivos:**

* Generar build final para Android
* Crear documentación completa del usuario
* Documentar proceso de instalación

### 3.2 Tabla de Calendarización de Actividades del Proyecto

Texto

El contenido generado por IA puede ser incorrecto.

Tabla Tabla de Calendarización de Actividades

### 3.3 Diagrama de Gantt de la tabla de actividades.

Gráfico, Gráfico en cascada

El contenido generado por IA puede ser incorrecto.

Tabla Diagrama de Grantt

### 3.4 Recursos necesarios para desarrollo del proyecto (materiales, humanos y financieros) con descripción y costo.

|  |  |
| --- | --- |
| REQUERIMIENTOS | COSTOS |
| Laptop o PC:  **Procesador:**   * Mínimo: Intel Core i5 o equivalente. * Recomendado: Intel Core i7 o superior.   **Memoria RAM:**   * Mínimo: 8 GB * Recomendado: 16 GB para proyectos más grandes o escenas complejas.   **Almacenamiento:**   * Mínimo: 40 GB libres para Unity + Vuforia + assets. * SSD recomendado para mejorar tiempos de carga y compilación. | $8,379 MXN a 13,219 MXN para recursos aceptables |
| Unity, inventor, blender | $0 MXN si eres estudiante, si no eres solo cobre Inventor $4,315 MXN |
| Programación y modelos | $1,000 MXN |

Tabla Requerimientos y costos

### 3.5 Análisis de Requerimientos funcionales y no funcionales del proyecto.

#### 3.5.1 Requerimientos Funcionales Detallados

##### RF-001: Sistema de Detección AR de Marcadores

Prioridad: Crítica (Alta)

Descripción: El sistema debe detectar y rastrear marcadores físicos usando la cámara del dispositivo móvil para anclar objetos virtuales en el mundo real.

**Criterios de Aceptación:**

* Detectar mínimo 5 marcadores simultáneamente
* Tiempo detección inicial: < 3 segundos
* Precisión tracking: > 90% en condiciones normales
* Distancia operación: 50cm - 2 metros
* Tolerancia rotación: hasta 60 grados
* Recuperación automática tras pérdida temporal

**Casos de Prueba:**

* CP-001: Detección bajo diferentes condiciones de luz
* CP-002: Tracking con movimiento rápido del dispositivo
* CP-003: Recuperación tras oclusión parcial del marcador

##### RF-002: Simulación 3D de Robots Virtuales

Prioridad: Crítica (Alta)

Descripción: Renderizar modelos 3D realistas de robots con animaciones, física y comportamientos programables sobre marcadores AR.

**Especificaciones Técnicas:**

* Modelos optimizados: máximo 8,000 polígonos
* Animaciones suaves 30+ FPS
* Física de colisiones básica
* Audio espacial 3D

**Funcionalidades:**

* Carga dinámica de modelos según ejercicio
* Sistema de animaciones (idle, movimiento, error, éxito)
* Interacción táctil para selección/manipulación
* Efectos visuales de estado (LED, pantallas)

##### RF-003: Sistema Integral de Ejercicios de Entrenamiento

Prioridad: Crítica (Alta)  
Descripción: Biblioteca de ejercicios estructurados que simulen competencias RoboJam con progresión adaptativa.

**Categorías de Ejercicios:**

1. **Navegación Básica (10 ejercicios)**
   * Movimiento lineal con obstáculos
   * Giros de precisión (90°, 180°, 360°)
   * Seguimiento de líneas y trayectorias
   * Navegación por laberintos simples
2. **Manipulación de Objetos (8 ejercicios)**
   * Recolección de cubos por color
   * Apilamiento y construcción
   * Clasificación por forma/tamaño
   * Transporte colaborativo
3. **Programación Visual (6 ejercicios)**
   * Secuencias de comandos básicos
   * Bucles y repeticiones
   * Condicionales simples (if/else)
   * Uso de sensores (distancia, color)
4. **Competencias Simuladas (4 ejercicios)**
   * Recreación competencias oficiales
   * Desafíos cronometrados
   * Competencias multirobot
   * Resolución de problemas abiertos

##### RF-004: Sistema de Seguimiento de Progreso

Prioridad: Media

Descripción: Plataforma analytics que registre, analice y visualice el progreso del usuario en múltiples dimensiones.

**Métricas Capturadas:**

* Tiempo promedio por ejercicio
* Tasa de éxito por categoría
* Intentos antes del éxito
* Patrones de error más frecuentes
* Progresión de habilidades específicas
* Engagement (tiempo de sesión, frecuencia)

**Visualizaciones:**

* Gráficas de progreso temporal
* Radar charts de habilidades
* Rankings
* Recomendaciones personalizadas

##### RF-005: Sistema Avanzado de Puntuación

Prioridad: Media

Descripción: Algoritmo de scoring multifactorial que evalúe el desempeño considerando múltiples variables.

**Factores de Puntuación:**

* **Completitud (40%)**: Porcentaje de ejercicio completado
* **Eficiencia temporal (25%)**: Bonus por completar rápido
* **Precisión (20%)**: Exactitud en movimientos/comandos
* **Elegancia código (10%)**: Simplicidad de la solución
* **Creatividad (5%)**: Soluciones alternativas innovadoras

**Fórmula Base:**

PuntajeFinal = (Completitud × 0.4) + (BonusTiempo × 0.25) +

(Precisión × 0.2) + (Elegancia × 0.1) + (Creatividad × 0.05)

#### 3.5.2 Requerimientos No Funcionales Especificados

##### RNF-001: Rendimiento AR Optimizado

Métrica: Frame Rate ≥ 30 FPS constantes

Descripción: El sistema debe mantener fluidez visual durante toda la experiencia AR sin drops significativos.

**Especificaciones:**

* FPS mínimo: 30 (objetivo: 60 FPS)
* Latencia tracking: < 50ms
* Tiempo carga modelos: < 5 segundos
* Uso RAM: < 2GB en dispositivos 4GB
* Uso CPU: < 70% sostenido
* Temperatura: incremento < 8°C tras 30min

**Estrategias Optimización:**

* Object pooling para elementos recurrentes
* Compresión texturas adaptativa
* Profiling continuo con Unity Profiler

##### RNF-002: Tiempo de Respuesta del Sistema

Métrica: Interacciones < 200ms

Descripción: Respuesta inmediata a todas las interacciones del usuario para mantener sensación de control.

**Tiempos Objetivo:**

* Touch/tap response: < 100ms
* Menu transitions: < 300ms
* Exercise loading: < 8 segundos
* AR initialization: < 15 segundos
* Database queries: < 2 segundos
* Score calculation: < 1 segundo

##### RNF-003: Compatibilidad Multiplataforma

Especificación: Soporte completo Android

Descripción: Funcionalidad equivalente en ambas plataformas principales.

**Android Compatibility:**

* Versión mínima: Android 8.0 (API 26)
* Versión objetivo: Android 13 (API 33)
* Arquitecturas: ARM64-v8a, ARMv7
* RAM mínima: 4GB (recomendado: 6GB)
* Almacenamiento: 3GB libres
* GPU: Adreno 530+, Mali-G71+, PowerVR GX6450+

##### RNF-004: Usabilidad y Accesibilidad

Métrica: Curva aprendizaje < 10 minutos

Descripción: Interfaz intuitiva que minimice fricción de adopción.

**Principios Diseño:**

* Accesibilidad WCAG 2.1 AA compliance
* Soporte múltiples idiomas (ES/EN inicial)
* Onboarding interactivo < 5 minutos
* Navegación consistente entre pantallas
* Modo ayuda contextual

**Métricas Usabilidad:**

* Task success rate: > 85% primer uso
* Time to first success: < 3 minutos
* Error recovery rate: > 95%
* User satisfaction: > 4.2/5.0

##### RNF-005: Confiabilidad y Disponibilidad

Métrica: Uptime > 95% durante uso

Descripción: Sistema robusto que funcione consistentemente.

**Especificaciones:**

* Crash rate: < 0.5% de sesiones
* Data integrity: 100% consistencia progreso
* Offline functionality: 70% características sin red
* Auto-recovery: < 10 segundos tras error
* Backup automático: cada 5 minutos de uso

### 3.6 Diagramas Casos de Uso general y especifico con su tabla de descripción.

Diagrama

El contenido generado por IA puede ser incorrecto.

Ilustración Diagrama Casos de Uso general

**Flujo Principal:**

* Usuario selecciona "Iniciar Entrenamiento AR" desde menú principal
* Sistema verifica permisos de cámara del dispositivo
* Sistema inicializa Vuforia Engine y configura tracking
* Sistema muestra pantalla de calibración con instrucciones visuales
* Usuario posiciona dispositivo enfocando marcador AR
* Sistema detecta marcador y valida su integridad
* Sistema carga modelo 3D del robot correspondiente al marcador
* Sistema renderiza robot virtual anclado al marcador
* Sistema muestra overlay con controles y menú de ejercicios
* Usuario selecciona ejercicio de entrenamiento específico
* Sistema configura parámetros del ejercicio seleccionado
* Sistema inicia cronómetro y activa sesión de entrenamiento

### 3.7 Diagrama Clases del sistema.

Diagrama

El contenido generado por IA puede ser incorrecto.

Ilustración Diagrama de Capa Vista

Diagrama

El contenido generado por IA puede ser incorrecto.

Ilustración Diagrama de Capa controlador

Diagrama

El contenido generado por IA puede ser incorrecto.

Ilustración Diagrama de Capa Modelo

### 3.8 Diagrama entidad relación de Base de datos.

Diagrama

El contenido generado por IA puede ser incorrecto.

Ilustración Diagrama de Entidad relacion

### 3.9 Diseño Tablas relacionadas de la base de datos, mínimo tres tablas.

Tabla

El contenido generado por IA puede ser incorrecto.Tabla

El contenido generado por IA puede ser incorrecto.Tabla

El contenido generado por IA puede ser incorrecto.

Ilustración DIagramas de base de datos

### 3.10 Diseño lógico de interfaces de la aplicación (menús, formularios).

Diagrama

El contenido generado por IA puede ser incorrecto.

Ilustración Diseño logico de Interfaces de la aplicacion

## Diagrama Grantt explicado

El estudiante creó una aplicación para teléfonos celulares donde los estudiantes practican para competencias de robots. La aplicación usa la cámara del celular para mostrar robots que no existen realmente (como Pokemon Go) sobre papeles impresos especiales.

El desarrollador escogió trabajar paso a paso porque necesitaba escribir mucha documentación para la universidad, tenía 13 semanas fijas para terminarlo, las reglas no iban a cambiar, y trabajaba solo.

### PRIMERA FASE: INVESTIGACIÓN (SEMANAS 1-2)

#### SEMANA 1: ESTUDIAR LAS COMPETENCIAS DE ROBOTS

El estudiante investigó a fondo las competencias RoboJam estudiando las reglas oficiales del 2024, encontrando tres grupos de edad (niños de 7-12 años, adolescentes de 13-15 años, jóvenes de 16-18 años). Estudió qué robots de LEGO se pueden usar y qué partes están permitidas. Vio más de 20 videos de competencias para ver qué estrategias funcionan mejor y qué errores son comunes.

Hizo entrevistas con dos maestros con experiencia y habló con 8 estudiantes de 13-16 años para entender qué problemas tienen actualmente, qué métodos de enseñanza funcionan mejor, y qué tecnología prefieren usar.

Probó otras aplicaciones educativas similares para ver qué funciona bien y qué no. Comparó tres tecnologías diferentes para crear realidad aumentada y eligió la mejor opción para su proyecto.

#### SEMANA 2: PLANEAR CÓMO HACER LA APLICACIÓN

El desarrollador decidió cómo organizar todo el código del programa, creando cuatro secciones principales: lo que ve el usuario (pantallas y botones), la lógica del juego (las reglas y puntuaciones), donde se guardan los datos (información de usuarios), y la parte de realidad aumentada (cámara y robots virtuales).

Identificó seis tipos principales de información que necesita guardar: datos de usuarios, ejercicios, puntuaciones, sesiones de uso, configuraciones personales, y logros obtenidos. Organizó la base de datos de manera eficiente para que las búsquedas sean rápidas.

### SEGUNDA FASE: CREAR LA APLICACIÓN (SEMANAS 3-10)

#### SEMANA 3: PREPARAR LAS HERRAMIENTAS

El estudiante instaló Unity (programa para hacer juegos) con licencia profesional y agregó Vuforia (tecnología para realidad aumentada) configurándolo para que funcione bien en celulares. Organizó todos los archivos del proyecto de manera profesional.

Creó los sistemas principales: un coordinador general que controla todo, un manejador de pantallas y navegación, y un sistema para guardar información de usuarios con seguridad.

#### SEMANAS 4-5: CREAR LAS PANTALLAS

El desarrollador hizo la navegación principal con tres pestañas (Entrenar/Ver Progreso/Perfil) con animaciones suaves. La aplicación se adapta automáticamente a diferentes tamaños de pantalla desde celulares pequeños hasta tabletas grandes.

Creó una biblioteca de elementos reutilizables: botones de diferentes tipos, tarjetas para mostrar ejercicios, barras de progreso. Desarrolló la pantalla de realidad aumentada que ocupa toda la pantalla, un panel de control con gráficas, y un sistema donde se arrastran bloques para programar robots.

Se aseguró de que la aplicación sea fácil de usar con colores que se ven bien, botones suficientemente grandes para tocar con el dedo.

#### SEMANAS 6-7: PROGRAMAR LA LÓGICA DEL JUEGO

El estudiante creó un sistema flexible para diferentes tipos de ejercicios: navegación (mover el robot), manipulación (agarrar objetos), programación (escribir código), y competencias simuladas. El sistema carga ejercicios desde la base de datos y se adapta automáticamente al nivel de cada usuario.

Creó formas de evaluar las soluciones considerando si están correctas, qué tan rápido se hicieron, qué tan elegante es el código, y qué tan creativas son. Implementó un sistema de puntuación que considera varios factores (40% completar la tarea, 25% velocidad, 20% precisión, 10% elegancia del código, 5% creatividad) y elementos de juego como logros y recompensas.

#### SEMANAS 8-9: REALIDAD AUMENTADA

El desarrollador optimizó la tecnología de realidad aumentada para que funcione de manera estable, con ajuste automático según la iluminación y recuperación automática cuando se pierde el seguimiento. El sistema puede manejar hasta 5 papeles impresos especiales al mismo tiempo.

Creó una colección de 2 tipos diferentes de robots optimizados para celulares (máximo 8,000 puntos en 3D) con un sistema que reduce automáticamente la calidad cuando es necesario. Implementó animaciones fluidas que mantienen más de 30 cuadros por segundo y efectos visuales realistas.

**SEMANA 10: PUNTUACIÓN Y ANÁLISIS**

Desarrolló sistemas de análisis educativo: seguimiento del progreso a lo largo del tiempo, análisis de tipos de errores, identificación de estilos de aprendizaje usando inteligencia artificial básica.

Implementó un motor de recomendaciones que identifica áreas donde el estudiante necesita practicar más y selecciona automáticamente los próximos ejercicios más útiles considerando el nivel actual y las preferencias detectadas.

### TERCERA FASE: PROBAR Y VERIFICAR (SEMANAS 11-12)

#### SEMANA 11: PRUEBAS INDIVIDUALES

El desarrollador configuró un sistema de pruebas con diferentes categorías, datos de prueba reutilizables, y versiones falsas de componentes para probar otras partes del sistema. Implementó pruebas exhaustivas para algoritmos críticos, lógica de negocio, sistemas de realidad aumentada, y operaciones de base de datos logrando más del 95% de cobertura.

Creó pruebas de integración simulando viajes completos de usuario y validando flujos de datos entre módulos. Implementó pruebas de rendimiento automatizadas midiendo velocidad de cuadros, manejo de memoria, y uso de batería en diferentes situaciones.

#### SEMANA 12: INTEGRACIÓN Y MEJORAS

El estudiante realizó pruebas completas en condiciones reales con situaciones completas de aprendizaje, pruebas en múltiples dispositivos en todo el rango de hardware objetivo, y pruebas de estrés ambiental bajo condiciones desafiantes.

Realizó pruebas de seguridad simulando ataques contra el sistema de autenticación, validando protección de datos, y probando la limpieza de datos de entrada.

CUARTA FASE: TERMINAR Y ENTREGAR (SEMANA 13)

#### SEMANA 13: FINALIZACIÓN

El desarrollador configuró versiones de producción optimizadas para ambas plataformas con configuraciones de compilador, paquetes de recursos, compresión balanceada, y protección de código. Implementó optimizaciones específicas para Android.

Realizó pruebas finales completas: pruebas básicas, pruebas de regresión, revisión de seguridad, medición de rendimiento, y pruebas de aceptación de usuario. Completó validación formal con lista de funcionalidades y aprobación de interesados.

# CAPÍTULO IV

## 4.- Resultados y recomendaciones

### 4.1 Pantallas lógicas, resultado del programa, con descripción del mismo

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

Ilustración Menu de la aplicacion RoboJam

El menú de la aplicación, consta de 3 opciones, el de Escanear Codigo QR, sirve para sincronizar nueva cancha, el de arena de entrenamiento nos da acceso directo a nuestras canchas, y por ultimo el de Gestionar QR, podemos borrar o renombrar

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

El contenido generado por IA puede ser incorrecto.

Ilustración Permiso de la camara de RoboJam

El entrar en la primera opción nos pedirá acceso a la cámara del teléfono, este con el fin de poder escanear el código.

Imagen que contiene Código QR

El contenido generado por IA puede ser incorrecto.

Ilustración Codigo Escaneado adecuadamente de RoboJam

Al escanear el código nos aparecerá una legenda que diga el Codigo QR Escaneado, nos da opciones de Cancelar y de ir a la arena

Imagen de la pantalla de un video juego

El contenido generado por IA puede ser incorrecto.

Ilustración Campo visualizado de RoboJam

En el apartado de arena de entrenamiento aparecerá la cancha, esta cancha dependerá del código QR que hayamos escaneado con anterioridad en la programación.

Imagen que contiene Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

Ilustración Gestion de codigo QR de RoboJam

Y por ultimo la pestaña de Gestionar QR, sirve para mostrar historial de los códigos QR que hemos escaneado, así mismo poder eliminarlo o renombrarlo

### 4.2 Sugerencias de Mejora

La aplicación podría mejorar significativamente implementando un sistema de administración más avanzado que permita a los maestros crear clases virtuales, asignar tareas específicas a cada estudiante, y monitorear el progreso de toda la clase en tiempo real. También sería valioso expandir las categorías de competencias para incluir no solo RoboJam sino también otras competencias internacionales como World Robot Olympiad y FIRST Robotics, permitiendo que los estudiantes se preparen para diferentes tipos de concursos según sus intereses. Una característica muy solicitada sería la posibilidad de que los usuarios diseñen y personalicen sus propios robots virtuales, eligiendo diferentes partes, sensores, y colores, e incluso importar modelos 3D que hayan creado en otros programas. Adicionalmente, la aplicación se beneficiaría de modo multijugador donde varios estudiantes puedan trabajar juntos en el mismo ejercicio simulando el trabajo en equipo real, conexión directa con robots físicos para ejecutar los programas creados en la app, sistema de certificaciones oficiales que den valor académico al progreso logrado, y inteligencia artificial más avanzada que detecte automáticamente las dificultades específicas de cada estudiante y genere ejercicios personalizados para ayudarlos a mejorar en esas áreas particulares.

## 6.-Fuentes consultadas

MasterD. (2019, 8 noviembre). *Qué es Unity y para qué sirve*. MasterD. <https://www.masterd.es/blog/que-es-unity-3d-tutorial>

Baez, C. D. A. (2020, November 12). Físicas y colisiones en Unity – Niixer Niixer. . .. *Portal De Noticias De Tecnología, Realidad Virtual, Aumentada Y Mixta, Videojuegos*. <https://niixer.com/index.php/2020/11/10/fisicas-y-colisiones-en-unity/>

School, C. (2024, November 29). ¿Qué es blender? | The Core School. *The Core*. <https://www.thecoreschool.com/blog/que-es-blender-y-como-puedes-sacarle-partido-para-la-animacion-3d/>

*¿Qué es la realidad aumentada o AR? | Microsoft Dynamics 365*. (n.d.). <https://www.microsoft.com/es-mx/dynamics-365/topics/augmented-reality/what-is-augmented-reality>

Chakray. (2025, February 24). *Lenguajes de programación: tipos, características y diferencias*. Chakray. <https://chakray.com/es/lenguajes-programacion-tipos-caracteristicas/>

*Autodesk Inventor: qué es y sus ventajas*. (n.d.). NTI. <https://www.nti-group.com/es/blog/es/autodesk-inventor-que-es-y-ventajas/>

Staff, C. (2023, November 29). *¿Qué es Python y para qué se usa? Guía para principiantes*. Coursera. <https://www.coursera.org/mx/articles/what-is-python-used-for-a-beginners-guide-to-using-python>

Delgado, D. O. (2017, March 29). Qué es C#: Introducción. *OpenWebinars.net*. <https://openwebinars.net/blog/que-es-c-introduccion/>

*¿Qué significa que Python es multiparadigma?* (n.d.). Quora. <https://es.quora.com/Qu%C3%A9-significa-que-Python-es-multiparadigma>

Fontoura, A. (2023, February 23). *Estandarización: ¿qué es? ¿Para qué sirve? ¿Cómo aplicarla?* FM2S. <https://www.fm2s.com.es/padronizacion/>

*¿Qué es .NET? Una plataforma para desarrolladores de código abierto | .NET*. (n.d.). Microsoft. <https://dotnet.microsoft.com/es-es/learn/dotnet/what-is-dotnet>

*El lenguaje C++ — Fundamentos de Programación en C++*. (n.d.). <https://www2.eii.uva.es/fund_inf/cpp/temas/1_introduccion/introduccion.html>

Urrutia, D. (2024, October 21). *Qué es Visual Studio | Definición, historia y ventajas*. Arimetrics. <https://www.arimetrics.com/glosario-digital/visual-studio>

*¿Qué es la virtualización? - Explicación de la virtualización de la computación en la nube - AWS*. (n.d.). Amazon Web Services, Inc. <https://aws.amazon.com/es/what-is/virtualization/>