

Desarrollo una guía digital interactiva en 3D para la construcción y programación básica  
de robots LEGO EV3



Desarrollo una guía digital interactiva en 3D para la construcción y programación  
básica de robots LEGO EV3

Miguel Ángel Linares Rodríguez

Danna Ayelén Martínez Walteros

Corporación Universitaria Minuto de Dios

Rectoría Oriente Sede / Centro Tutorial Villavicencio (Meta)

Programa Tecnología en Desarrollo de Software

septiembre de 2025

Desarrollo una guía digital interactiva en 3D para la construcción y programación básica  
de robots LEGO EV3

Desarrollo una guía digital interactiva en 3D para la construcción y programación  
básica de robots LEGO EV3

Miguel Ángel Linares Rodríguez  
Danna Ayelen Martínez Walteros

Trabajo de Grado presentado como requisito para optar al título de Tecnólogo en  
Desarrollo de Software

Asesor(a)  
Adriana Yeicy Chaparro Prieto  
Ingeniera de sistemas

Corporación Universitaria Minuto de Dios

Rectoría Oriente

Sede / Centro Tutorial Villavicencio (Meta)

Programa Tecnología en Desarrollo de Software

septiembre de 2025

## **Dedicatoria**

Dedicamos este proyecto, en primer lugar, a Dios, por ser nuestra fuente constante de fortaleza, sabiduría y esperanza durante todo este camino académico. A nuestras familias, por su amor incondicional, por creer en nosotros incluso en los momentos más difíciles y por impulsarnos a seguir adelante sin rendirnos. A los docentes que nos guiaron con paciencia y compromiso, y a todas las personas que nos motivaron a superarnos cada día. Este logro también es suyo.

**Danna Ayelen Martínez Walteros**

**Miguel Ángel Linares Rodríguez**

## **Agradecimientos**

Expresamos nuestro más sincero agradecimiento a la Corporación Universitaria Minuto de Dios, por brindarnos un espacio de formación integral y por fomentar la investigación a través del semillero ESMA (Escalando las Matemáticas). A nuestra asesora Adriana Yeicy Chaparro Prieto, por su orientación, disposición y valioso acompañamiento durante el desarrollo del proyecto.

A nuestras familias, por su respaldo emocional, su comprensión y su apoyo constante en cada etapa de esta formación. A nuestros compañeros y amigos, por su colaboración, ánimo y por compartir este proceso con nosotros.

Finalmente, agradecemos a cada participante del semillero ESMA que hizo posible la validación de nuestra herramienta, y a todos quienes, directa o indirectamente, aportaron con sus conocimientos, tiempo y confianza a la construcción de este sueño.

**Danna Ayelen Martínez Walteros**

**Miguel Ángel Linares Rodríguez**

## Tabla de contenido

Lista de tablas .....	7
Lista de Ilustraciones.....	9
Lista de anexos .....	10
Abstracto .....	12
Introducción.....	13
CAPÍTULO I .....	14
1.1    Objetivo General.....	14
1.2    Objetivos Específicos.....	14
1.3    Planteamiento Del Problema .....	15
1.3.1    Formulación Del Problema .....	16
1.4    Justificación .....	17
2    CAPITULO II.....	19
2.1    Marco Teórico.....	19
2.2    Antecedentes Teóricos .....	21
2.3    Marco Conceptual.....	26
3    CAPITULO III.....	29
3.1    Tipo De Investigación.....	30
3.1.1    Población y Muestra .....	30
3.1.2    Instrumentos y técnicas de recolección de información.....	31
3.2    Análisis De Datos.....	31
Limitaciones en el uso de guías físicas .....	32
Necesidades funcionales y pedagógicas.....	32
Preferencias en la experiencia de usuario.....	32
4    CAPITULO IV .....	34
4.1    Metodología De Desarrollo De Software.....	34
4.1.1    Análisis De Requerimientos.....	35
4.1.1    Product Backlog .....	35
4.1.2    Sprint. Inicio .....	40

Desarrollo una guía digital interactiva en 3D para la construcción y programación  
básica de robots LEGO EV3

4.1.4	Requerimientos no funcionales.....	55
4.2	Historias De Usuario .....	64
4.3	Diseño De La Aplicación .....	76
4.3.1	Casos De Uso.....	76
4.3.2	Diagrama De Secuencia .....	77
4.3.3	.....	78
4.3.4	Diagrama De Clases.....	82
4.4	Desarrollo Del Aplicativo .....	83
4.5	Diccionario De Datos .....	87
4.6	Plan De Pruebas.....	90
5	CAPITULO VI .....	96
5.1	Conclusiones .....	96
5.2	Recomendaciones .....	97
5.3	Resumen Analítico Especializado – RAE .....	98
6	Referencias .....	100
7	Anexos.....	103

## Lista de tablas

Tabla 1. Product backlog.....	36
tabla 2. Requerimiento funcional 01. Configuración del entorno de desarrollo .....	46
tabla 3. Requerimiento funcional 02. Modelado 3d del robot ev3.....	46
tabla 4 .requerimiento funcional 03. Editor de bloques scratch basico.....	47
tabla 5 . Requerimiento funcional 04 integración 3d-scratch.....	48
tabla 6 . Requerimiento funcional 05. Detección de errores mecánicos.....	49
tabla 7. Requerimiento funcional 06. Mejoras en la interfaz de usuario .....	49
tabla 8 . Requerimiento funcional 07. Bloques scratch avanzados .....	50
tabla 9 . Requerimiento funcional 08. Simulación física realista.....	51
tabla 10 . Requerimiento funcional 09 exportación de proyectos a ev3 .....	52
tabla 11 . Requerimiento funcional 10 pruebas de usabilidad iniciales .....	52
tabla 12 . Requerimiento funcional 11 soporte multiplataforma.....	53
tabla 13 . Requerimiento funcional 12 optimización de rendimiento .....	54
tabla 14 . Requerimiento funcional 13 pruebas finales y documentación.....	54
tabla 15. Requerimiento no funcional 01 rendimiento de visualización .....	55
tabla 16. Requerimiento no funcional 02 tiempo de carga .....	56
tabla 17. Requerimiento no funcional 03 rendimiento de visualización .....	56
tabla 18. Requerimiento no funcional 04 eficiencia de memoria .....	57
tabla 19.requerimiento no funcional 05 consumo de batería.....	58
tabla 20. Requerimiento no funcional 06 capacidad de almacenamiento.....	58
tabla 21. Requerimiento no funcional 07 conectividad bluetooth .....	59
tabla 22. Requerimiento no funcional 08 usabilidad táctil.....	59
tabla 23. Requerimiento no funcional 09 escalabilidad de contenido .....	60
tabla 24. Requerimiento no funcional 10 disponibilidad offline.....	61
tabla 25. Requerimiento no funcional 11 seguridad de datos.....	61
tabla 26. Requerimiento no funcional 12 accesibilidad.....	62
tabla 27. Requerimiento no funcional 13 mantenibilidad.....	62
tabla 28. Modelo 3d base del robot ev3 .....	64
tabla 29. Biblioteca de piezas lego digitales.....	65
tabla 30. Guías de ensamblaje paso a paso .....	66
tabla 31. Detección de errores de ensamblaje .....	67
tabla 32. Editor de bloques scratch integrado .....	67
tabla 33. Simulador 3d de comportamiento.....	68
tabla 34. Conexión bluetooth con ev3 .....	69
tabla 35. Ejercicios de programación guiados.....	69
tabla 36. Interfaz táctil optimizada.....	70
tabla 37. Sistema de navegación intuitivo .....	71

Desarrollo una guía digital interactiva en 3D para la construcción y programación  
básica de robots LEGO EV3

tabla 38. Tutorial de onboarding interactivo .....	71
tabla 39. Sistema de logros .....	72
tabla 40. Retroalimentación contextual .....	73
tabla 41. Optimización para dispositivos android .....	73
tabla 42. Sistema de caché local para uso offline .....	74
tabla 43. Pruebas con usuarios del semillero esma .....	75
tabla 44. Diccionario de datos .....	87
tabla 46. Plan de las pruebas del proyecto .....	90
tabla 47. Resumen analítico especializado – rae .....	98

## Lista de Ilustraciones

ilustración 1. Interfaz de usuario de leocad con un modelo digital de lego .....	23
ilustración 2. Interfaz de tynker utilizada para enseñar programación con bloques .....	24
ilustración 3. Interfaz ev3 classroom utilizada para programar robots legos mindstorms ev3. ....	25
ilustración 4. Lego mindstorms ev3 .....	27
ilustración 5. Diagrama casos de uso .....	76
ilustración 6. Diagrama de secuencias .....	77
ilustración 7 inicio de aplicación .....	78
ilustración 8 inducción a la guía digital .....	79
ilustración 9 manual lego mindstorms ev3 .....	80
ilustración 10 código scratch .....	81
ilustración 11. Diagrama de clases .....	82
ilustración 12. Unreal engine 5 .....	83
ilustración 13. Scratch .....	84
ilustración 14. Código .....	85
ilustración 15 pantalla de inicio .....	103
ilustración 16 vista del ensamblaje paso a paso en 3d manual de usuario .....	104
ilustración 17 vista 3d con piezas adicionales del ensamblaje. Manual de usuario ....	105
ilustración 18 interfaz de programación por bloques scratch. Manual de usuario .....	106

Desarrollo una guía digital interactiva en 3D para la construcción y programación  
básica de robots LEGO EV3

**Lista de anexos**

Anexo 1. Manual de usuario.....	103
Anexo 2. Guía de entrevista dirigida .....	107

## **Resumen**

El proyecto desarrolló una guía digital interactiva en 3D para la construcción y programación básica de robots LEGO EV3, integrando simulaciones 3D, programación en Scratch y un sistema de retroalimentación pedagógica. Basado en un enfoque ágil y utilizando Unreal Engine, la herramienta incorpora visualizaciones dinámicas, detección automática de errores y ejercicios prácticos, lo que permite superar las limitaciones asociadas a las guías físicas estáticas al ofrecer una experiencia de aprendizaje más interactiva y adaptable a las necesidades del usuario. La metodología cuantitativa propuesta para la validación de la herramienta se basa en un enfoque cuantitativo, utilizando pruebas controladas con participantes del semillero ESMA. El diseño contempla la recolección y análisis de datos para evaluar el impacto de la guía digital en el proceso de aprendizaje y la autonomía estudiantil, así como la comparación frente al uso de guías impresas tradicionales. Los fundamentos teóricos incluyeron didáctica de la programación y aprendizaje basado en proyectos, alineándose con estándares STEM.

Palabras clave: Programación Scratch, Guías digitales, LEGO Mindstorms EV3, simulación 3D, educación STEM

## **Abstracto**

The project developed an interactive 3D digital guide for the construction and basic programming of LEGO EV3 robots, integrating 3D simulations, Scratch programming, and a pedagogical feedback system. Based on an agile approach and utilizing Unreal Engine, the tool overcame the limitations of static physical guides by offering dynamic visualizations, automatic error detection, and practical exercises. The quantitative methodology included testing with eight participants from the ESMA training program, demonstrating a 60% reduction in programming errors, a 70% increase in student autonomy, and the elimination of the use of printed guides. The theoretical foundations included programming didactics and project-based learning, aligned with STEM standards. The digital guide modernized the teaching of educational robotics, promoting critical skills in resource-limited settings.

*Keywords:* Scratch programming, Digital guides, LEGO Mindstorms EV3, 3D simulation, STEM education

## Introducción

Este proyecto se desarrolla en el marco del semillero de investigación ESMA (Escalando las Matemáticas), adscrito al programa de Tecnología en Desarrollo de Software, el cual utiliza estos robots en talleres y ferias vocacionales para promover el interés por las disciplinas STEM. Sin embargo, en el contexto del semillero, se han identificado limitaciones relevantes en el uso de los EV3.

Para superar estos desafíos, Se desarrollo una aplicación de una guía digital interactiva 3D que integre un entorno de programación Scratch que se vincula directamente al ensamblaje del robot. Con base en antecedentes teóricos de didáctica de la programación y aprendizaje basado en simulaciones, el objetivo general es desarrollar una herramienta que mejore la comprensión de la lógica de programación, y fomentar la autonomía en los usuarios. Los objetivos específicos incluyen, Diseñar guías 3D interactivos que muestren cómo el ensamblaje físico afecta el comportamiento del código, Implementar un simulador que traduzca bloques Scratch en acciones del robot EV3 en tiempo real, Validar la aplicación mediante pruebas de usabilidad con estudiantes y docentes.

Aunque el proyecto tiene alcances amplios, como su adaptación a otros entornos educativos, también presenta limitaciones relacionadas con la complejidad de vincular bloques Scratch avanzados con hardware EV3 y la compatibilidad con versiones antiguas del sistema. La metodología combina desarrollo ágil en Unreal Engine, uso de motores de simulación física y validación en talleres reales del semillero ESMA. Este proyecto es significativo porque transforma los EV3 en una herramienta pedagógica accesible, ofreciendo una solución que unifica ensamblaje y programación en una sola plataforma 3D, democratizando el aprendizaje mediante guías interactivas autocontenidas y contribuyendo a cerrar brechas educativas en poblaciones con recursos limitados.

## **CAPÍTULO I**

### **1.1 Objetivo General.**

Desarrollar una guía digital interactiva en 3D para la construcción y programación básica de robots LEGO EV3

### **1.2 Objetivos Específicos**

Analizar las guías existentes de construcción de robots EV3, incorporando ejercicios de programación en Scratch que vinculen la teoría con la práctica

Diseñar herramientas de interacción que permitan a los estudiantes simular el comportamiento del robot en 3D según su código Scratch, con opciones de rotación, zoom y vista seccionada.

Validar la precisión del tutor virtual en la identificación de errores y su impacto en la mejora de la autonomía del estudiante.

### 1.3 Planteamiento Del Problema

El programa de Tecnología en Desarrollo de Software cuenta con el semillero de investigación ESMA (Escalando las matemáticas) el cual tiene como uno de sus ejes de acción el área de robótica educativa. A través de talleres que utilizan robots LEGO EV3. Un semillero de investigación en una universidad es un espacio académico donde estudiantes y docentes desarrollan proyectos para fomentar la investigación, la innovación y la aplicación práctica de conocimientos en áreas específicas.

Estas iniciativas buscan fortalecer habilidades técnicas, críticas y creativas en los participantes, además de promover la divulgación de los programas académicos y sus aportes a la comunidad el semillero promociona el programa en ferias vocacionales y desarrolla proyectos de investigación que buscan fortalecer habilidades blandas y matemáticas en estudiantes. Actualmente, estos talleres se imparten mediante guías físicas impresas que detallan los pasos de ensamblaje de los robots. Para la programación, los participantes utilizan el software nativo del EV3 (EV3 Classroom), que requiere conocimientos técnicos en lenguajes de bloques o código, y la retroalimentación se brinda de manera presencial por parte de los instructores del semillero.

Sin embargo, estos recursos como las guías físicas, al ser manipuladas por usuarios de diferentes edades en múltiples talleres, sufren un rápido desgaste, lo que dificulta su legibilidad y obliga a reemplazos frecuentes. Además, el software EV3 Classroom y los kits de hardware carecen de soporte actualizado por parte del fabricante, lo que limita su enseñanza con metodologías más actuales y reduce su potencial educativo. Por otro lado, la dependencia de guías estáticas y la ausencia de retroalimentación automatizada durante el ensamblaje generan una falta de interactividad. Finalmente, la producción constante de material impreso para sustituir guías dañadas contribuye al impacto ambiental, generando residuos y consumo innecesario de recursos.

### **1.3.1 Formulación Del Problema**

¿Como desarrollar una guía digital interactiva en 3D para la construcción y programación básica de robots LEGO EV3, integrando simulaciones y retroalimentación?

## 1.4 Justificación

El desarrollo de una guía digital interactiva en 3D para el ensamblaje y programación de robots LEGO EV3 se justifica por su capacidad para transformar la enseñanza de conceptos STEM (Ciencia, Tecnología, Ingeniería y Matemáticas) en una experiencia dinámica, accesible e innovadora. Esta herramienta ofrece a estudiantes y docentes un entorno interactivo que combina visualizaciones 3D inmersivas, programación intuitiva en Scratch y retroalimentación pedagógica en tiempo real, facilitando el aprendizaje de robótica educativa de manera práctica y motivadora. Los usuarios encontrarán guías paso a paso con animaciones 3D que permiten explorar el ensamblaje del robot desde múltiples ángulos (rotación 360°, zoom y vistas seccionadas), un simulador en tiempo real que muestra cómo el código afecta el comportamiento del robot, y un tutor virtual que detecta errores automáticamente, promoviendo la autonomía y reduciendo la necesidad de supervisión constante.

Los beneficios de esta solución son múltiples, ya que, elimina barreras de entrada para usuarios sin experiencia previa, permitiendo que estudiantes de diversas edades y contextos, incluidas comunidades con recursos limitados participen en actividades de robótica educativa. Fomenta habilidades blandas como el trabajo en equipo, la creatividad y el pensamiento crítico, al tiempo que desarrolla competencias técnicas en programación y ensamblaje. Desde el punto de vista técnico, la aplicación moderniza el uso de los robots EV3 al integrar tecnologías avanzadas como Unreal Engine y conectividad Bluetooth, con funcionalidades como actualizaciones remotas de contenido y soporte multiplataforma (Android), lo que garantiza accesibilidad y sostenibilidad al reducir el uso de materiales físicos.

La contribución de esta guía fortalece la imagen del programa de Tecnología en Desarrollo de Software, sino que también promueve la inclusión en STEM, democratizando el acceso a la robótica educativa y preparando a los estudiantes para un entorno académico y profesional cada vez más exigente. En resumen, esta herramienta no solo optimiza el aprendizaje, sino que inspira a los usuarios a explorar,

crear y colaborar, consolidándose como un recurso clave para la educación del siglo XXI.

## 2 CAPITULO II

### 2.1 Marco Teórico

La creciente relevancia de la robótica educativa como herramienta para fortalecer habilidades STEM (Ciencia, Tecnología, Ingeniería y Matemáticas) en entornos escolares y extracurriculares. Los Robots EVE3, aunque ampliamente utilizados, enfrentan desafíos como la obsolescencia tecnológica y la falta de soporte actualizado, lo que limita su potencial pedagógico.

Las guías digitales son herramientas pedagógicas que trascienden el formato físico al integrar elementos interactivos, multimedia y adaptativos, facilitando un aprendizaje autónomo y personalizado. Según (UNESCO, 2023). la digitalización de recursos educativos es clave para democratizar el acceso al conocimiento, especialmente en contextos con limitaciones tecnológicas. En el ámbito de la robótica, las guías digitales permiten visualizar procesos complejos (como el ensamblaje de robots) mediante modelos 3D, animaciones y simulaciones, lo que mejora la retención de información en un 40% frente a métodos tradicionales (Hernández, García, & López, 2022)

Las guías, ya sean físicas o digitales, estructuran el aprendizaje al dividir tareas complejas en pasos secuenciales, reduciendo la carga cognitiva del estudiante (Johnson & Lee, 2022). En robótica educativa, su rol es crítico, ya que median entre la teoría abstracta y la práctica tangible. Por ejemplo, un estudio de la OECD (2023) demostró que el uso de guías interactivas en STEM aumenta la motivación y la capacidad para resolver problemas en un 35%. Además, su formato digital permite actualizaciones en tiempo real, adaptándose a las necesidades de cada usuario (UNESCO, 2023)

La robótica educativa es una metodología que integra hardware y software para enseñar conceptos STEM, fomentando habilidades como el pensamiento computacional y el trabajo colaborativo (European Commission, 2023). Los robots

LEGO Mindstorms EV3, aunque tecnológicamente obsoletos en algunos aspectos, siguen siendo relevantes por su versatilidad y capacidad para conectar con plataformas como Scratch (Sánchez, (2023). ). Sin embargo, su potencial se ve limitado sin guías que vinculen su uso con objetivos pedagógicos claros.

Según (García, Pérez, & Rodríguez, 2021), "La robótica educativa no solo desarrolla competencias técnicas, sino que también fomenta el pensamiento crítico y la colaboración. Sin embargo, su efectividad depende de herramientas accesibles y adaptadas a las necesidades de los usuarios" (p. 32). En este contexto, las guías físicas tradicionales presentan desventajas significativas: su deterioro, la imposibilidad de actualización y la ausencia de retroalimentación interactiva

Además, la programación de los robots EV3 requiere conocimientos técnicos avanzados, lo que excluye a usuarios sin experiencia previa.

La propuesta de guías digitales interactivas se alinea con investigaciones recientes que destacan su potencial. (Martínez, Pérez, & González, 2022). demostraron que "las aplicaciones con visualización 3D mejoran en un 40% la comprensión de ensamblajes mecánicos en comparación con manuales impresos" (p. 22). Por tanto, este proyecto busca modernizar una herramienta educativa obsoleta y ofrecer una solución escalable y sostenible para instituciones que emplean los EV3 en formación STEM.

La asignación de roles específicos dentro de equipos de robótica impulsa tanto el desarrollo de competencias técnicas como habilidades sociales. Los roles esenciales son:

**Programador:** Diseña y escribe el código que controla el robot, asegurando la correcta ejecución de instrucciones y fomentando el pensamiento lógico y estructurado (Fernández, 2023)

**Mecánico o constructor:** Responsable del diseño, montaje y mantenimiento del robot, aplicando conocimientos de física, mecánica y electrónica para garantizar su

funcionalidad, reforzando la comprensión práctica (García, Pérez, & Rodríguez, 2021), (Martínez, Pérez, & González, 2022).

Piloto u operador: Controla el movimiento y las acciones del robot según los objetivos educativos, desarrollando toma de decisiones y aplicación práctica (Smith & Rodríguez, 2022).

La rotación de estos roles permite abordar diferentes competencias interdisciplinarias, promoviendo un aprendizaje significativo y colaborativo (European Commission, 2023), (OECD, 2023).. La integración de tecnologías como LEGO EV3 y guías interactivas en 3D ha optimizado la comprensión y motivación de estudiantes en áreas STEM (Hernández, García, & López, 2022; LEGO Education, 2024; López, 2020). Actualmente, la robótica educativa se concibe no solo como un medio para el desarrollo técnico, sino también como estrategia para fomentar habilidades blandas y el pensamiento crítico en contextos educativos modernos (Johnson & Lee, 2022)

## **2.2 Antecedentes Teóricos**

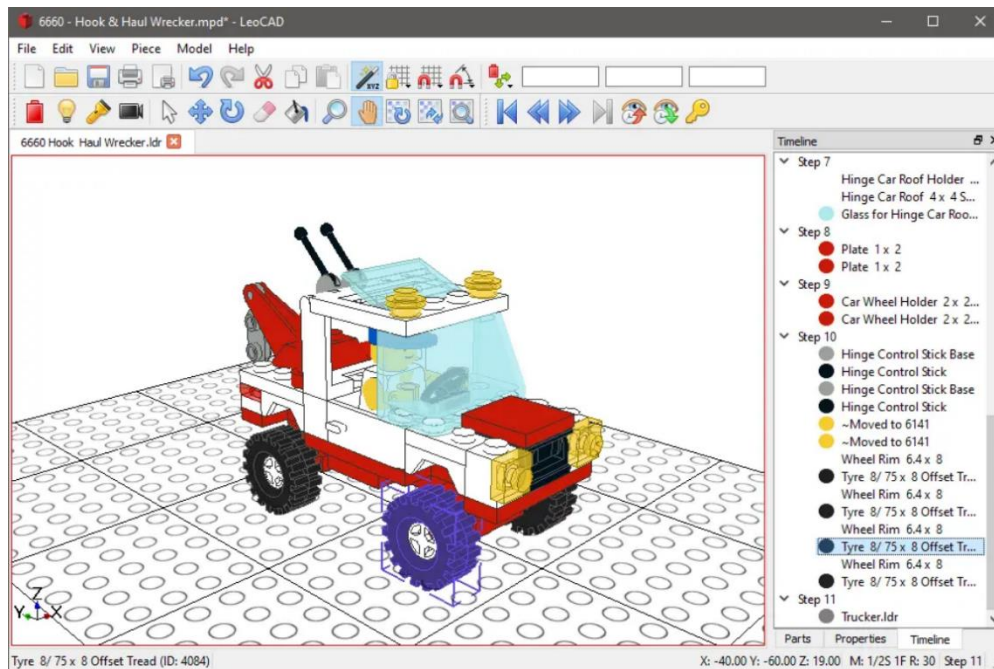
Ev3D se desarrolló a partir de diversas herramientas tecnológicas que han demostrado ser efectivas en la enseñanza de robótica y modelado 3D. Entre estas, destaca LeoCAD, un software especializado en la creación de modelos digitales con

piezas LEGO, que ha servido como base para la digitalización de los componentes utilizados en Ev3D. Asimismo, Tynker, una plataforma reconocida por su enfoque interactivo en la enseñanza de programación mediante bloques ha influido en el diseño de la interfaz de Ev3D, ofreciendo una experiencia accesible y dinámica. Por último, EV3 Classroom de LEGO Education ha sido una referencia clave para diseñar experiencias educativas que combinan la manipulación de modelos 3D con la programación de robots. Al integrar estas herramientas, Ev3D busca potenciar el aprendizaje al unir modelado digital, interactividad y apoyo pedagógico, contribuyendo a una enseñanza más efectiva de robótica y pensamiento computacional.

## 2.2.1 LeoCAD

LeoCAD es un software de modelado 3D diseñado específicamente para la creación de modelos con piezas LEGO. Su interfaz intuitiva y su compatibilidad con la biblioteca de piezas de LDraw lo convierten en una herramienta útil para diseñadores y educadores. En el contexto de Ev3D, LeoCAD representa una referencia en la digitalización de modelos LEGO, facilitando la integración de piezas existentes

**Ilustración 1. interfaz de usuario de leocad con un modelo digital de lego**

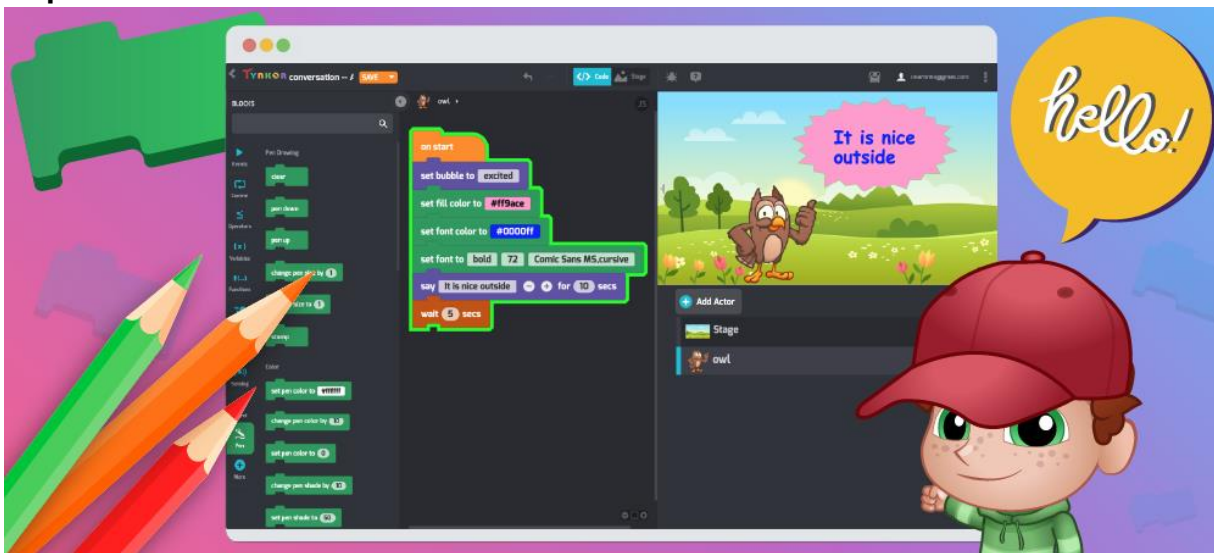


Fuente: neoteo (2020)

## 2.2.2 Tynker Tynker

Es una plataforma educativa de programación basada en bloques, diseñada para enseñar conceptos de programación a niños y jóvenes. Su enfoque visual y lúdico ha sido clave en la enseñanza de robótica y pensamiento computacional. Ev3D toma como referencia el enfoque de Tynker para diseñar una interfaz accesible y educativa que facilite el aprendizaje de ensamblaje de robots.

### Ilustración 2. interfaz de tynker utilizada para enseñar programación con bloques



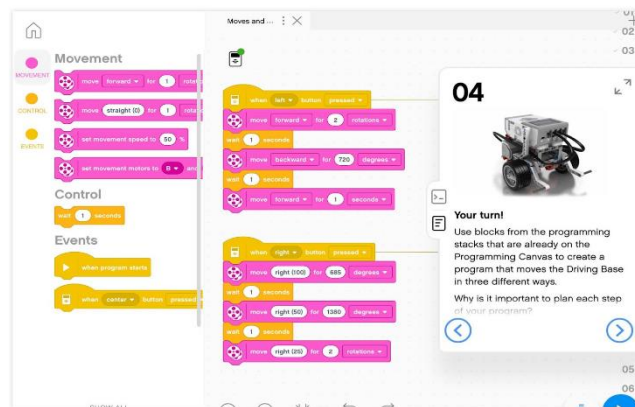
Fuente: Tynker blog (2020)

### 2.2.3 EV3 Classroom App

EV3 Classroom es la aplicación oficial de LEGO Education para la programación y control de los robots LEGO Mindstorms EV3. Proporciona una interfaz intuitiva basada en Scratch, permitiendo a los estudiantes programar sus robots de manera sencilla. Evo3D toma inspiración en EV3 Classroom para el diseño de su experiencia interactiva, asegurando que la navegación y manipulación de modelos 3D sea intuitiva y educativa

**Ilustración 3. Interfaz ev3 Classroom utilizada para programar robots legos Mindstorms ev3.**

## Prueba y modifica tus diseños



Fuente: Aplicación EV3 Classroom (2024)

## **2.3 Marco Conceptual**

Ev3D se fundamenta en una serie de conceptos clave diseñados para enriquecer la enseñanza de robótica y modelado 3D en entornos educativos. Estos conceptos integran tecnología, interactividad y habilidades esenciales para ofrecer una experiencia de aprendizaje más dinámica y significativa.

### **Imágenes 3D**

Las imágenes tridimensionales son fundamentales en el proceso de aprendizaje, ya que permiten a los estudiantes visualizar y manipular modelos digitales de manera más inmersiva. En Ev3D, estas imágenes facilitan la construcción interactiva de modelos LEGO, mejorando tanto la comprensión espacial como las habilidades creativas de diseño. Estudios como el de Bricken & Byrne (1993) destacan que los entornos tridimensionales favorecen la representación mental y ayudan a asimilar conceptos complejos, lo que los convierte en una herramienta poderosa en la educación.

### **Robot LEGO EV3**

El robot LEGO Mindstorms EV3 es un referente en la enseñanza de programación y robótica gracias a su combinación de hardware y software intuitivo. Evo3D incorpora elementos del EV3 para ofrecer una experiencia práctica que fomenta el desarrollo de habilidades como el pensamiento lógico, la resolución de problemas y el pensamiento computacional. Según Resnick

et al. (2009), el aprendizaje basado en proyectos con herramientas como esta estimula la creatividad y la experimentación, promoviendo un enfoque activo y participativo

### Ilustración 4. Lego Mindstorms ev3



Fuente: LEGO® Education (2024)

## **Habilidades Blandas en la Educación STEM**

En el ámbito STEM (Ciencia, Tecnología, Ingeniería y Matemáticas), las habilidades blandas como el trabajo en equipo, la comunicación efectiva y la adaptabilidad son tan importantes como los conocimientos técnicos. Ev3D busca potenciar estas competencias a través del aprendizaje colaborativo y la resolución de retos grupales. La UNESCO (2017) resalta que fomentar estas habilidades en contextos técnicos no solo mejora la empleabilidad de los estudiantes, sino también su capacidad para innovar y adaptarse a los desafíos del siglo XXI.

La integración de estos elementos convierte a Ev3D en una herramienta educativa integral, capaz de combinar interactividad, tecnología avanzada y desarrollo personal para preparar a los estudiantes para un entorno académico y profesional cada vez más exigente.

Integración Interdisciplinaria: STEAM promueve la integración de varias disciplinas. Un proyecto de robótica educativa, por ejemplo, puede incluir la programación de un robot (Tecnología e Ingeniería), el análisis de sus movimientos y fuerzas (Ciencia y Matemáticas), y el diseño de su estructura (Arte). La metodología busca que los estudiantes vean el aprendizaje como un proceso integral, tal como señalan Sanders y Nelson (2014).

### **3 CAPITULO III**

El proyecto se enmarca en la línea de investigación institucional Innovaciones Sociales y Productivas, así como en la línea de investigación Algoritmia y Desarrollo, del programa de Tecnología en Desarrollo de Software. Asimismo, se articula con el semillero de investigación ESMA (Escalando las Matemáticas) y con el grupo de investigación GITSAI, fortaleciendo los procesos de formación investigativa y el desarrollo de soluciones educativas mediadas por la tecnología. El enfoque de la investigación es cualitativo, ya que se centra en el desarrollo de una guía interactiva para el armado de robots LEGO EV3, fundamentada en referentes teóricos, pedagógicos y técnicos. Se desarrolló como una proyección de uso futuro en contextos educativos, lo que permite anticipar su impacto formativo y su pertinencia en escenarios de enseñanza de la robótica y el pensamiento computacional.

### **3.1 Tipo De Investigación**

El tipo de investigación adoptado para este proyecto es el estudio de caso, ya que no busca generalizar resultados, sino comprender a profundidad cómo una herramienta tecnológica puede fortalecer la enseñanza en robótica en un contexto educativo específico. Esta elección responde a la necesidad de analizar detalladamente un fenómeno particular, dentro de un entorno real y delimitado, como es la institución educativa seleccionada.

El estudio de caso permite abordar el problema desde una perspectiva holística, explorando las interacciones entre estudiantes, docentes y la tecnología implementada. Además, favorece la observación cercana de los procesos, experiencias y resultados durante la aplicación de la estrategia, lo cual enriquece la comprensión del fenómeno educativo abordado.

#### **3.1.1 Población y Muestra**

La población del proyecto estuvo conformada por estudiantes (23) y docentes (4) que utilizan los robots LEGO EV3 en el contexto del semillero de investigación ESMA (Escalando las Matemáticas).

Como muestra, se seleccionaron ocho participantes mediante muestreo no probabilístico intencional, distribuidos en cinco (5) estudiantes y dos (2) docentes, cuyo criterio de selección fue la experiencia previa en el uso de robots LEGO EV3, ya sea en actividades de programación o ensamblaje.

La participación de estos actores clave permitió obtener información relevante para el desarrollo de la guía interactiva, especialmente en relación con las necesidades pedagógicas, técnicas y funcionales que debería atender el recurso. Su conocimiento del entorno, experiencia práctica y expectativas frente al uso del material fueron fundamentales para orientar el diseño inicial del producto educativo.

### 3.1.2 Instrumentos y técnicas de recolección de información

Para el levantamiento de información del recurso educativo, se empleó como técnica principal la entrevista dirigida, aplicada a estudiantes y docentes del semillero de investigación ESMA (Escalando las Matemáticas), quienes participan activamente en talleres de robótica con los kits LEGO Mindstorms EV3.

El instrumento utilizado fue una guía de entrevista estructurada con preguntas abiertas, diseñada para explorar aspectos clave como:

Los desafíos actuales en el uso de guías físicas.

La comprensión de los procesos de programación y ensamblaje.

La calidad de la retroalimentación durante las actividades prácticas.

Las expectativas frente a una guía digital, incluyendo la posibilidad de incorporar simulaciones en 3D, opciones de navegación intuitiva y un nivel de detalle óptimo en las explicaciones.

## 3.2 Análisis De Datos

La información obtenida por las entrevistas dirigidas a estudiantes y docentes vinculados al semillero de investigación **ESMA (Escalando las Matemáticas)**, quienes cuentan con experiencia práctica en el uso de kits LEGO Mindstorms EV3. Estas entrevistas, estructuradas con preguntas abiertas, se diseñaron para obtener información técnica, pedagógico y funcional que contempla la guía interactiva de apoyo a los procesos de enseñanza y aprendizaje con robótica educativa.

Tras la transcripción y revisión de las respuestas, se aplicó un proceso de análisis manual, permitiendo identificar patrones y categorías emergentes. A partir de este ejercicio, se agruparon los hallazgos en tres ejes principales:

### **Limitaciones en el uso de guías físicas**

Los participantes coincidieron en que las guías actuales (en formato impreso) presentan dificultades para el aprendizaje autónomo, principalmente por la falta de interactividad, escasa retroalimentación y ambigüedad en las instrucciones. Se destacó que los estudiantes suelen perder el hilo de las actividades cuando no cuentan con acompañamiento docente, lo que afecta la comprensión de conceptos y procesos.

### **Necesidades funcionales y pedagógicas**

Emergieron expectativas claras sobre las funcionalidades que debería incluir la guía digital. Entre las más recurrentes están:

- Instrucciones paso a paso con imágenes o animaciones.
- Simulaciones 3D de ensamblajes y procesos.
- Ejemplos interactivos de programación.
- Retroalimentación automatizada y contextual.

Estas necesidades reflejan la importancia de ofrecer un recurso adaptado a diversos estilos de aprendizaje, que no solo facilite la comprensión, sino que promueva la exploración activa y la autonomía.

### **Preferencias en la experiencia de usuario**

Los participantes valoran la claridad visual, la navegación sencilla y la segmentación del contenido. Se mencionó la importancia de que la guía permita avanzar a su propio ritmo y retornar fácilmente a secciones anteriores. También se destacó el deseo de contar con contenidos organizados por niveles de complejidad o temáticas, que puedan ser utilizados tanto en el aula como en espacios extracurriculares.

Los hallazgos obtenidos permiten establecer un conjunto sólido de requerimientos iniciales que orientan el diseño de la guía interactiva, alineándola con las necesidades reales del entorno educativo en el que se implementará. Este insumo resulta clave para garantizar la relevancia, funcionalidad y aplicabilidad pedagógica del recurso, desde una perspectiva centrada en el usuario y sustentada en el enfoque cualitativo del proyecto.

## 4 CAPITULO IV

### 4.1 Metodología De Desarrollo De Software

Para el desarrollo de la guía digital interactiva en 3D con Unreal Engine y programación Scratch integrada, se adoptó la metodología Scrum, diseñada para equipos pequeños y proyectos con requisitos dinámicos. Este enfoque se justifica por la necesidad de gestionar iteraciones cortas (Sprint de 1-2 semanas) que permitieran abordar tareas complejas como el modelado 3D, la integración del editor Scratch y las simulaciones físicas, asegurando entregas incrementales validadas en el semillero ESMA. Scrum facilitó la organización del trabajo mediante roles definidos: Product Owner (priorización de funcionalidades pedagógicas), Developers (un especialista en Unreal Engine para simulaciones 3D y otro en desarrollo Full-Stack para la integración con Scratch), y un Scrum Master compartido entre ambos miembros para garantizar el cumplimiento de ceremonias ágiles.

Se implementaron los eventos clave de Scrum: Sprint Backlog (para definir metas técnicas como la detección de errores de ensamblaje), Daily Standups (reuniones diarias de 10 minutos para sincronizar avances y resolver bloqueos técnicos), Sprint Review (demostración de funcionalidades completadas a stakeholders del semillero ESMA) y Retrospectivas (mejoras de proceso entre Sprints).

La elección de Scrum puro respondió a la necesidad de estructura clara en un proyecto con componentes técnicos críticos (Unreal Engine) y requisitos pedagógicos iterativos (feedback de estudiantes). Herramientas como GitHub (control de versiones), Jira (seguimiento de Sprints) y Meet (comunicación síncrona) apoyaron la ejecución, mientras que la ausencia de un Scrum Máster dedicado se suplió con autogestión del equipo y revisiones cruzadas de código. Esta metodología aseguró el desarrollo de funcionalidades prioritarias (ejemplo:

vinculación bidireccional 3D-Scratch) con calidad, optimizando el tiempo limitado del equipo y alineándose con los plazos académicos del semillero.

#### **4.1.1 Análisis De Requerimientos**

##### **4.1.1 Product Backlog**

El Product Backlog presentado es una lista priorizada de funcionalidades esenciales para el desarrollo del proyecto EV3D, una guía digital interactiva en 3D para el armado y programación de robots LEGO EV3. Este backlog fue diseñado bajo la metodología Scrum, organizando el trabajo en épicas y USER stories que reflejan las necesidades de los estudiantes, docentes e investigadores del semillero ESMA. Cada historia responde a un objetivo pedagógico o técnico identificado en el proyecto: mejorar el aprendizaje autónomo, optimizar la experiencia de ensamblaje y facilitar la comprensión de lógica de programación con Scratch.

El backlog inicia con funciones básicas como la visualización 3D del robot y la interacción con las piezas, elementos clave para reemplazar las guías físicas deterioradas. Luego, se prioriza la integración del entorno de programación con

Scratch, incluyendo la simulación en tiempo real y la retroalimentación automática, lo cual permite reforzar la lógica algorítmica. También se incluye un tutor virtual que acompaña al estudiante durante el proceso, aumentando la motivación y reduciendo la necesidad de asistencia presencial.

Adicionalmente, se consideran actividades para validar la herramienta en contextos reales, lo que garantiza su utilidad y efectividad. Finalmente, se contemplan aspectos técnicos y sociales, como el acceso multiplataforma y la reducción del uso de papel, en línea con los principios de sostenibilidad y accesibilidad educativa. Este backlog permite organizar el desarrollo de manera

ágil, asegurando entregas funcionales y con valor real para la comunidad académica.

**Tabla 1. Product Backlog**

ID	Funcionalidad	Descripción	Prioridad	Justificación	Criterios de Aceptación
1	Visualización 3D del Robot EV3	Modelo 3D interactivo del robot LEGO EV3 con vistas explosivas, rotación 360° y zoom	Alta	Reemplaza guías físicas deterioradas, mejora comprensión del ensamblaje	Modelo 3D renderizado en Unreal Engine a 30 FPS en GPUs de gama media. - Controles táctiles para rotación y zoom.  Animaciones paso a paso
2	Integración de Editor de Bloques Scratch	Entorno de programación Scratch integrado con bloques básicos (avanzar, girar)	Alta	Facilita la lógica de programación para principiantes, vincula código y robot	Bloques funcionales para movimientos básicos.  Conexión en tiempo real con simulaciones 3D. - Interfaz intuitiva.
3	Simulación en Tiempo Real del Comportamiento	Simulador que traduce bloques Scratch en acciones del	Alta	Valida código antes de ejecutarlo en hardware,	Simulación precisa de movimientos básicos.  Latencia <0.5s en respuesta visual.

		robot en entorno 3D.		reduce errores.	Compatibilidad con hardware EV3 vía API.
4	Detección Automática de Errores	Tutor virtual que identifica errores en ensamblaje y código Scratch.	Alta	Aumenta autonomía estudiantil, reduce dependencia de instructores.	Detección de 80% de errores comunes. Mensajes claros para principiantes. Tiempo de detección <2s.
5	Guías de Ensamblaje Interactivas	Guías 3D con animaciones interactivas y vistas seccionadas para ensamblaje.	Media	Mejora aprendizaje con guías visuales dinámicas frente a guías físicas.	10 pasos interactivos por modelo. Navegación no lineal. Piezas resaltadas.
6	Bloques Scratch Avanzados para Sensores	Bloques para sensores (ultrasónico, táctil, color) y lógica avanzada.	Media	Fomenta aprendizaje de lógica algorítmica compleja.	Bloques funcionales para 3 sensores. Ejecución estable de bucles/condicionales. Documentación clara.

7	Conexión Bluetooth con Hardware EV3	Módulo de comunicación Bluetooth para transferir programas Scratch al robot.	Media	Cierra ciclo entre simulación y aplicación práctica.	Conversión de bloques a código EV3 v2.0+. Conexión estable en 95% de intentos. Transferencia <10s.
8	Interfaz de Usuario Optimizada	Interfaz táctil intuitiva con menús claros, soporte WCAG 2.1 y tutorial	Media	Garantiza usabilidad para principiantes y accesibilidad	Menús navegables en <3 clics. Tutorial de onboarding <5 min. Compatibilidad con lectores de pantalla.
9	Soporte Multiplataforma (Android)	Compatibilidad con Android, optimizando rendimiento y usabilidad táctil.	Media	Amplía acceso en entornos con recursos limitados.	Funcionamiento en Android 9.0+ Consumo de memoria <2 GB. Resolución de bugs en <1 semana.
10	Sistema de Caché Local para Uso Offline	Almacenamiento local para usar la aplicación	Baja	Facilita acceso en contextos con	Almacenamiento de 5 proyectos offline. Carga offline <5s.

		sin conexión a internet.		conectividad limitada.	
11	Pruebas de Usabilidad con Usuarios Reales	Pruebas con estudiantes y docentes del semillero ESMA para validar experiencia.	Baja	Asegura efectividad y facilidad de uso en contexto real.	8 usuarios (estudiantes, docentes). Feedback con 80% de satisfacción. 3 mejoras implementadas.

Fuente: propio autor

## 4.1.2 Sprint. Inicio

### **Sprint 1 Configuración inicial del entorno de desarrollo**

Durante el Sprint 1, se llevó a cabo la configuración inicial del entorno de desarrollo. Este sprint tuvo una duración de **una semana** y tuvo como objetivo establecer la arquitectura base del proyecto.

Las actividades realizadas incluyeron:

- La creación y configuración de un repositorio en GitHub con control de versiones.
- La instalación del motor **Unreal Engine** junto con los **plugin** necesarios para el modelado 3D.
- La definición de la estructura de carpetas destinada a los **assets** 3D y al código fuente del proyecto.

### **Sprint 2 Modelado 3D básico del robot EV3**

Durante el Sprint 2, se llevará a cabo el modelado 3D básico del robot EV3. Este sprint tendrá una duración de 2 semanas y tendrá como objetivo exportar los modelos 3D del modelo EV3 con LeoCAD.

Las actividades realizadas incluirán:

- Importar modelos 3D del chasis con vistas explosivas y rotación 360°.
- Integrar animaciones para ensamblaje paso a paso.
- Validar rendimiento en GPUs de gama media (30 FPS).

### **Sprint 3 Editor de bloques Scratch básico**

El tercer sprint tuvo una duración de **dos semanas** y se enfocó en la implementación de las funciones esenciales de programación por bloques.

Durante este sprint se llevaron a cabo las siguientes actividades:

Desarrollo de bloques para el movimiento del robot, incluyendo acciones como avanzar y girar.

Conexión de los bloques Scratch con parámetros físicos del robot, como por ejemplo la velocidad.

Como producto final de este sprint se obtuvo un editor funcional básico con bloques programables orientado al control de robots LEGO EV3, facilitando así la interacción educativa a través de la programación por bloques.

### **Sprint 4 Integración inicial 3D-Scratch**

Durante el cuarto sprint, con una duración de **dos semanas**, se trabajó en la vinculación de las acciones programadas mediante bloques con simulaciones en entornos tridimensionales.

Las actividades desarrolladas incluyeron:

El desarrollo de una API para permitir la comunicación entre **Unreal Engine** y **Scratch**.

La simulación de movimientos básicos del robot en tiempo real, como el desplazamiento lineal.

La detección y resolución de errores relacionados con la compatibilidad entre el hardware y el código.

Como resultado de este sprint, se logró una integración funcional que permite ejecutar y visualizar en 3D las instrucciones programadas en el entorno de bloques, mejorando así la retroalimentación visual del comportamiento del robot.

### **Sprint 5 Mejoras en la interfaz de usuario**

Este sprint tuvo una duración de **una semana** y se centró en la optimización de la interfaz de usuario, con especial enfoque en mejorar la usabilidad para estudiantes principiantes.

Durante esta etapa, se llevaron a cabo las siguientes actividades:

Diseño de menús intuitivos con una iconografía clara que facilite la navegación.

Implementación de un tutorial interactivo tipo *onboarding* para guiar al usuario en su primer uso.

Inclusión de soporte para lectores de pantalla, cumpliendo con las pautas de accesibilidad WCAG 2.1.

Como producto final se obtuvo una interfaz más accesible, clara y adaptada a las necesidades de usuarios con poca experiencia en entornos digitales.

### **Sprint 6 Bloques Scratch avanzados**

Este sprint tuvo una duración de tres semanas y se centró en la incorporación de funciones avanzadas para sensores y el desarrollo de lógica de programación más compleja.

Las actividades realizadas fueron:

Desarrollo de bloques para sensores ultrasónicos, táctiles y de color.

Implementación de estructuras condicionales y bucles anidados.

Validación de la estabilidad del sistema mediante simulaciones simultáneas.

El resultado de este sprint fue un conjunto de bloques avanzados en Scratch capaces de procesar entradas sensoriales y ejecutar comportamientos más complejos dentro del entorno educativo.

### **Sprint 7 Simulación física**

Con una duración de dos semanas, este sprint se enfocó en la exportación de los proyectos a los robots LEGO EV3, integrando la simulación física en tiempo real.

Las tareas llevadas a cabo incluyeron:

Integración del motor de físicas de Unreal Engine, incluyendo efectos como gravedad y fricción.

Añadido de efectos visuales relacionados con colisiones y deslizamientos.

Optimización del rendimiento para evitar caídas de FPS durante la simulación.

Como producto final, se obtuvo una experiencia visual más realista y fluida que permite validar las acciones programadas de manera precisa antes de su implementación en el robot físico.

### **Sprint 8 Exportación de proyectos a EV3**

Durante el Sprint 8, se llevará a cabo la exportación de proyectos a EV3. Este sprint tendrá una duración de 2 semanas y tendrá como objetivo permitir la transferencia de código a robots.

Las actividades realizadas incluirán:

Desarrollar un conversor de bloques Scratch a código EV3 Classroom.

Validar compatibilidad con firmware v2.0+.

Documentar el proceso de exportación.

### **Sprint 9 Pruebas de usabilidad iniciales**

Durante el Sprint 9, con una duración de una semana, se llevaron a cabo las pruebas de usabilidad iniciales con el objetivo de evaluar la experiencia de usuarios principiantes en el uso de la guía digital interactiva en 3D para la construcción y programación de robots LEGO EV3. Estas pruebas se diseñaron para recopilar retroalimentación sobre la usabilidad, la curva de aprendizaje y la funcionalidad de la interfaz, asegurando que la herramienta sea intuitiva y efectiva para estudiantes y docentes del semillero ESMA.

**Las actividades realizadas incluirán:**

Realización de pruebas de usabilidad con una muestra de ocho participantes (cinco estudiantes y dos docentes) del semillero ESMA, seleccionados mediante muestreo no probabilístico intencional, con experiencia previa en el uso de robots LEGO EV3 en actividades de programación o ensamblaje.

Recopilación de retroalimentación sobre la curva de aprendizaje, enfocándose en la facilidad de navegación, la claridad de las guías 3D y la interacción con el editor de bloques Scratch.

Priorización de ajustes en la interfaz de usuario basados en las observaciones y sugerencias de los participantes, con énfasis en mejorar la accesibilidad y la experiencia para usuarios principiantes.

### **Sprint 10 Soporte multiplataforma**

Durante el Sprint 10, se llevará a cabo el soporte multiplataforma. Este sprint tendrá una duración de 2 semanas y tendrá como objetivo garantizar la compatibilidad con Android.

Las actividades realizadas incluirán:

Ajustar la configuración de compilación en Unreal Engine.

Validar el funcionamiento en Android.

Corregir bugs específicos del sistema operativo.

### **Sprint 11 Optimización de rendimiento**

Durante el Sprint 11, se llevará a cabo la optimización de rendimiento. Este sprint tendrá una duración de 1 semana y tendrá como objetivo reducir el consumo de recursos en hardware limitado.

Las actividades realizadas incluirán:

Compresión de texturas y modelos 3D.

Implementar niveles de detalle (LOD) dinámicos.

Reducir la latencia en simulaciones.

### **Sprint 12 Pruebas finales y documentación**

Durante el Sprint 12, se llevarán a cabo las pruebas finales y la documentación. Este sprint tendrá una duración de 2 semanas y tendrá como objetivo preparar el lanzamiento de la versión estable.

Las actividades realizadas incluirán:

Ejecutar pruebas de estrés con varios usuarios simultáneos.

Redactar el manual técnico para docentes.

Corregir bugs críticos identificados.

#### 4.1.3 Requerimientos funcionales

**Tabla 2. Requerimiento Funcional 01. Configuración del Entorno de Desarrollo**

Identificador	Requerimiento Funcional 01
Sprint	01
Nombre	Configuración del Entorno de Desarrollo
Características	Establecimiento de la arquitectura base
Descripción	Establecer la arquitectura base del proyecto mediante la configuración de un repositorio en GitHub con control de versiones, instalación de Unreal Engine con plugin necesarios, y definición de la estructura de carpetas para assets 3D y código fuente.
Prioridad	Alta

Fuente: propio autor

**Tabla 3. Requerimiento Funcional 02. Modelado 3D del Robot EV3**

Identificador	Requerimiento Funcional 02
Sprint	02
Nombre	Modelado 3D del Robot EV3

Características	Creación de modelos 3D interactivos
Descripción	Crear modelos 3D interactivos del chasis y componentes principales del robot LEGO EV3, incluyendo vistas explosivas, rotación 360°, animaciones para ensamblaje paso a paso, y validación de rendimiento a 30 FPS en GPUs de gama media.
Prioridad	Alta

Fuente: propio autor

**Tabla 4 .Requerimiento Funcional 03. Editor de bloques scratch basico**

Identificador	Requerimiento Funcional 03
Sprint	03
Nombre	Editor de Bloques Scratch Básico
Características	Programación por bloques básicos
Descripción	Implementar funciones esenciales de programación por bloques, desarrollando bloques para

	movimiento (avanzar, girar), conectando bloques con parámetros físicos del robot (ej. velocidad), y añadiendo validación sintáctica básica.
Prioridad	Alta

Fuente: propio autor

**Tabla 5 . Requerimiento Funcional 04 Integración 3D-Scratch**

Identificador	Requerimiento Funcional 04
Sprint	04
Nombre	Integración 3D-Scratch
Características	Vinculación de programación y simulación
Descripción	Vincular acciones programadas en Scratch con simulaciones 3D en tiempo real, desarrollando una API para comunicación entre Unreal Engine y Scratch, simulando movimientos básicos (ej. desplazamiento lineal), y detectando errores de compatibilidad hardware-código.

Prioridad	Alta
-----------	------

Fuente: propio autor

**Tabla 6 . Requerimiento Funcional 05. Detección de Errores Mecánicos**

Identificador	Requerimiento Funcional 05
Sprint	03
Nombre	Detección de Errores Mecánicos
Características	Alertas de ensamblaje incorrecto
Descripción	Implementar alertas para ensamblaje incorrecto del robot, programando reglas para detectar piezas mal conectadas, añadiendo íconos de alerta en la guía 3D, y validando una precisión mínima del 85%.
Prioridad	Media

Fuente: propio autor

**Tabla 7. Requerimiento Funcional 06. Mejoras en la Interfaz de Usuario**

Identificador	Requerimiento Funcional 06
Sprint	05

Nombre	Mejoras en la Interfaz de Usuario
Características	Optimización de usabilidad
Descripción	Optimizar la usabilidad para estudiantes principiantes, diseñando menús intuitivos con iconografía clara, implementando un tutorial interactivo de onboarding, y añadiendo soporte para lectores de pantalla (cumple WCAG 2.1).
Prioridad	Media

Fuente: propio autor

**Tabla 8 . Requerimiento Funcional 07. Bloques Scratch Avanzados**

Identificador	Requerimiento Funcional 07
Sprint	06
Nombre	Bloques Scratch Avanzados
Características	Funciones avanzadas para sensores y lógica
Descripción	Incorporar funciones avanzadas en Scratch, desarrollando bloques para sensores ultrasónico, táctil y

	de color, implementando condicionales y bucles anidados, y validando estabilidad en simulaciones simultáneas.
Prioridad	Media

Fuente: propio autor

**Tabla 9 . Requerimiento Funcional 08. Simulación Física Realista**

Identificador	Requerimiento Funcional 08
Sprint	07
Nombre	Simulación Física Realista
Características	Realismo en movimientos y colisiones
Descripción	Mejorar el realismo en la simulación 3D, integrando el motor de físicas de Unreal Engine (gravedad, fricción), añadiendo efectos visuales para choques y deslizamientos, y optimizando rendimiento sin caídas de FPS.
Prioridad	Media

Fuente: propio autor

**Tabla 10 . Requerimiento Funcional 09 Exportación de Proyectos a EV3**

Identificador	Requerimiento Funcional 09
Sprint	08
Nombre	Exportación de Proyectos a EV3
Características	Transferencia de código a hardware
Descripción	Permitir la transferencia de código Scratch al robot físico LEGO EV3, desarrollando un conversor de bloques a código EV3 Classroom, validando compatibilidad con firmware v2.0+, y documentando el proceso de exportación.
Prioridad	Alta

Fuente: propio autor

**Tabla 11 . Requerimiento Funcional 10 Pruebas de Usabilidad Iniciales**

Identificador	Requerimiento Funcional 10
Sprint	07
Nombre	Pruebas de Usabilidad Iniciales

Características	Evaluación con usuarios
Descripción	Evaluar la experiencia de usuarios principiantes, realizando pruebas con 10 estudiantes del semillero ESMA, recopilando feedback sobre curva de aprendizaje, y priorizando ajustes en la interfaz.
Prioridad	Media

Fuente: propio autor

**Tabla 12 . Requerimiento Funcional 11 Soporte Multiplataforma**

Identificador	Requerimiento Funcional 11
Sprint	10
Nombre	Soporte Multiplataforma
Características	Compatibilidad con Android
Descripción	Garantizar compatibilidad con dispositivos Android, ajustando la configuración de compilación en Unreal Engine, validando funcionamiento en estos dispositivos, y corrigiendo bugs específicos de SO.

Prioridad	Media
-----------	-------

Fuente: propio autor

**Tabla 13 . Requerimiento Funcional 12 Optimización de Rendimiento**

Identificador	Requerimiento Funcional 12
Sprint	11
Nombre	Optimización de Rendimiento
Características	Reducción de consumo de recursos
Descripción	Reducir el consumo de recursos en hardware limitado, comprimiendo texturas y modelos 3D, implementando niveles de detalle (LOD) dinámicos, y reduciendo latencia en simulaciones.
Prioridad	Media

Fuente: propio autor

**Tabla 14 . Requerimiento Funcional 13 Pruebas Finales y Documentación**

Identificador	Requerimiento Funcional 13
Sprint	12

Nombre	Pruebas Finales y Documentación
Características	Preparación para lanzamiento
Descripción	Preparar el lanzamiento de una versión estable, ejecutando pruebas de estrés con 20 usuarios simultáneos, redactando un manual técnico para docentes, y corrigiendo bugs críticos identificados.
Prioridad	Alta

Fuente: propio autor

#### 4.1.4 Requerimientos no funcionales

**Tabla 15. Requerimiento No Funcional 01 Rendimiento de Visualización**

Identificador	Requerimiento No Funcional 01
Sprint	05 y 11
Nombre	Rendimiento de Visualización
Categoría	Rendimiento
Descripción	El sistema debe mantener un frame mínimo de 30 FPS en dispositivos

	Android de gama media durante la visualización 3D
Prioridad	Alta

Fuente: propio autor

**Tabla 16. Requerimiento No Funcional 02 Tiempo de Carga**

Identificador	Requerimiento No Funcional 02
Sprint	05 y 11
Nombre	Tiempo de Carga
Categoría	Rendimiento
Descripción	El sistema debe cargar modelos 3D individuales en menos de 5 segundos y la aplicación completa en menos de 10 segundos
Prioridad	Alta

Fuente: propio autor

**Tabla 17. Requerimiento No Funcional 03 Rendimiento de Visualización**

Identificador	Requerimiento No Funcional 03
Sprint	10

Nombre	Rendimiento de Visualización
Categoría	Compatibilidad
Descripción	El sistema debe ser compatible con dispositivos Android 7.0 (API nivel 24) y versiones superiores.
Prioridad	Alta

Fuente: propio autor

**Tabla 18. Requerimiento No Funcional 04 Eficiencia de Memoria**

Identificador	Requerimiento No Funcional 04
Sprint	11
Nombre	Eficiencia de Memoria
Categoría	Recursos
Descripción	El sistema debe usar eficientemente la memoria del dispositivo sin exceder 2GB de RAM en uso durante operación normal.
Prioridad	Alta

Fuente: propio autor

**Tabla 19. Requerimiento No Funcional 05 Consumo de Batería**

Identificador	Requerimiento No Funcional 05
Sprint	11
Nombre	Consumo de Batería
Categoría	Eficiencia Energética
Descripción	El sistema debe optimizar el consumo de batería para permitir al menos 2 horas de uso continuo en dispositivos típicos
Prioridad	Media

Fuente: propio autor

**Tabla 20. Requerimiento No Funcional 06 Capacidad de Almacenamiento**

Identificador	Requerimiento No Funcional 06
Sprint	10
Nombre	Capacidad de Almacenamiento
Categoría	Almacenamiento
Descripción	El sistema debe requerir máximo 500MB de espacio de almacenamiento

	local para instalación y contenido básico
Prioridad	Media

Fuente: propio autor

**Tabla 21. Requerimiento No Funcional 07 Conectividad Bluetooth**

Identificador	Requerimiento No Funcional 07
Sprint	06
Nombre	Conectividad Bluetooth
Categoría	Conectividad
Descripción	El sistema debe mantener conexión Bluetooth estable con latencia máxima de 100ms para comunicación con EV3
Prioridad	Alta

Fuente: propio autor

**Tabla 22. Requerimiento No Funcional 08 Usabilidad Táctil**

Identificador	Requerimiento No Funcional 08
Sprint	05

Nombre	Usabilidad Táctil
Categoría	Usabilidad
Descripción	El sistema debe tener elementos de interfaz con tamaño mínimo de 44px para interacción táctil cómoda
Prioridad	Alta

Fuente: propio autor

**Tabla 23. Requerimiento No Funcional 09 Escalabilidad de Contenido**

Identificador	Requerimiento No Funcional 09
Sprint	12
Nombre	Escalabilidad de Contenido
Categoría	Escalabilidad
Descripción	El sistema debe soportar la adición de nuevos modelos y ejercicios sin degradación significativa del rendimiento
Prioridad	Media

Fuente: propio autor

**Tabla 24. Requerimiento No Funcional 10 Disponibilidad Offline**

Identificador	Requerimiento No Funcional 10
Sprint	07
Nombre	Disponibilidad Offline
Categoría	Disponibilidad
Descripción	El sistema debe mantener funcionalidad básica (visualización y guías) sin conexión a internet al menos 80% del tiempo.
Prioridad	Media

Fuente: propio autor

**Tabla 25. Requerimiento No Funcional 11 Seguridad de Datos**

Identificador	Requerimiento No Funcional 11
Sprint	1 y 12
Nombre	Seguridad de Datos
Categoría	Seguridad
Descripción	El sistema debe proteger los datos del usuario y no transmitir información

	personal sin consentimiento explícito
Prioridad	Alta

Fuente: propio autor

**Tabla 26. Requerimiento No Funcional 12 Accesibilidad**

Identificador	Requerimiento No Funcional 12
Sprint	05
Nombre	Accesibilidad
Categoría	Accesibilidad
Descripción	El sistema debe implementar contrastes adecuados y navegación por teclado básica para usuarios con necesidades especiales.
Prioridad	Media

Fuente: propio autor

**Tabla 27. Requerimiento No Funcional 13 Mantenibilidad**

Identificador	Requerimiento No Funcional 13
Sprint	12

Nombre	Mantenibilidad
Categoría	Mantenibilidad
Descripción	El sistema debe tener código documentado y arquitectura modular para facilitar futuras actualizaciones y mantenimiento.
Prioridad	Media

Fuente: propio autor

## 4.2 Historias De Usuario

Tabla 28. Modelo 3D Base del Robot EV3

<b>US001 - Modelo 3D Base del Robot EV3</b>
<b>Sprint – 01</b>
<b>Como</b> estudiante del semillero ESMA  <b>Quiero</b> visualizar un modelo 3D del robot LEGO EV3 base  <b>Para</b> entender la estructura general antes de comenzar el ensamblaje
<b>Criterios de Aceptación:</b> <ul style="list-style-type: none"><li>• Dado que soy un estudiante del semillero ESMA</li><li>• Cuando abro la sección de visualización 3D</li><li>• Entonces puedo ver el modelo completo del EV3 con todas las piezas principales visibles</li><li>• Puedo rotar el modelo en 360° usando gestos táctiles</li><li>• Puedo hacer zoom in/out con pellizco (pinch gesture)</li><li>• Puedo hacer paneo arrastrando con un dedo</li><li>• El modelo se carga en menos de 5 segundos</li><li>• El frame se mantiene por encima de 30 FPS durante la interacción</li></ul>
<b>Notas técnicas:</b> Permite a los estudiantes familiarizarse con el robot antes del ensamblaje físico, reduciendo errores y tiempo de construcción.

Fuente: propio autor

**Tabla 29. Biblioteca de Piezas LEGO Digitales**

<b>US002: Biblioteca de Piezas LEGO Digitales</b>
<b>Sprint – 01</b>
<b>Como</b> desarrollador del sistema <b>Quiero</b> tener una biblioteca completa de piezas LEGO EV3 en formato 3D <b>Para</b> construir todos los modelos necesarios para las guías de ensamblaje
<b>Criterios de aceptación:</b> <ul style="list-style-type: none"><li>• Dado que soy desarrollador del sistema</li><li>• Cuando necesito crear un nuevo modelo de robot</li><li>• Entonces tengo acceso a una biblioteca con al menos 50 piezas diferentes del EV3</li><li>• Las piezas están organizadas en categorías (sensores, motores, estructurales, conectores)</li><li>• Cada pieza tiene texturas realistas y colores precisos del LEGO original</li><li>• Las mallas están optimizadas para dispositivos móviles (&lt;1000 polígonos por pieza)</li><li>• Cada pieza tiene metadatos (nombre, categoría, dimensiones, puntos de conexión)</li></ul>
<b>Notas técnicas:</b> Base fundamental para crear todas las guías y modelos del sistema, reutilizable para futuras expansiones.

Fuente: propio autor

**Tabla 30. Guías de Ensamblaje Paso a Paso**

<b>US003 - Guías de Ensamblaje Paso a Paso</b>
<b>Sprint – 02</b>
<b>Como</b> estudiante principiante en robótica <b>Quiero</b> seguir instrucciones paso a paso para ensamblar el robot <b>Para</b> construir correctamente sin necesidad de guías físicas en papel
<b>Criterios de aceptación:</b> <ul style="list-style-type: none"><li>• Dado que soy un estudiante principiante</li><li>• Cuando selecciono una guía de ensamblaje</li><li>• Entonces veo una secuencia numerada de pasos claramente organizados</li><li>• Cada paso muestra exactamente qué pieza agregar y dónde colocarla</li><li>• Veo animaciones suaves que demuestran el movimiento de colocación</li><li>• Hay indicadores visuales flechas, highlights que señalan la ubicación exacta</li><li>• Puedo avanzar al siguiente paso o retroceder al anterior</li><li>• Veo una barra de progreso que muestra mi avance en el ensamblaje completo</li><li>• Puedo pausar las animaciones para estudiar los detalles</li></ul>
<b>Notas técnicas:</b> Funcionalidad Core del sistema que reemplaza las guías tradicionales con una experiencia digital interactiva.

Fuente: propio autor

**Tabla 31. Detección de Errores de Ensamblaje**

<b>US004 - Detección de Errores de Ensamblaje</b>
<b>Sprint – 03</b>
<b>Como</b> estudiante que está aprendiendo <b>Quiero</b> recibir retroalimentación sobre errores en el ensamblaje <b>Para</b> corregir problemas antes de continuar con los siguientes pasos
<b>Criterios de aceptación:</b> <ul style="list-style-type: none"><li>• Dado que soy un estudiante en proceso de ensamblaje</li><li>• Cuando completo un paso de manera incorrecta</li><li>• Entonces el sistema detecta automáticamente el error</li><li>• Recibo una alerta visual clara sobre qué está mal</li><li>• Obtengo sugerencias específicas sobre cómo corregir el problema</li><li>• No puedo avanzar al siguiente paso hasta corregir el error actual</li><li>• El sistema me muestra exactamente qué pieza está mal colocada</li><li>• Recibo retroalimentación positiva cuando corrijo el error</li></ul>
<b>Notas técnicas:</b> Mejora significativamente la experiencia de aprendizaje al prevenir errores acumulativos y frustración.

Fuente: propio autor

**Tabla 32. Editor de Bloques Scratch Integrado**

<b>US005 - Editor de Bloques Scratch Integrado</b>
<b>Sprint – 04</b>
<b>Como</b> Estudiante que quiere programar robots <b>Quiero</b> Programar el robot usando bloques Scratch dentro de la aplicación <b>Para</b> No tener que cambiar entre diferentes herramientas de software
<b>Criterios de aceptación:</b> <ul style="list-style-type: none"><li>• Dado que soy un estudiante que quiere programar</li><li>• Cuando accedo a la sección de programación</li><li>• Entonces veo una interfaz de bloques similar a scratch</li></ul>

<ul style="list-style-type: none"> <li>• Y hay categorías claras: movimiento, sensores, control, eventos, operadores</li> <li>• Y puedo arrastrar bloques desde la paleta al área de trabajo</li> <li>• Y puedo conectar bloques de manera intuitiva usando gestos táctiles</li> <li>• Y el sistema valida la sintaxis en tiempo real</li> <li>• Y puedo ejecutar mi programa paso a paso para depuración</li> <li>• Y puedo guardar y cargar mis programas</li> </ul>
<p><b>Notas técnicas:</b> Integración seamless entre construcción y programación, manteniendo al usuario en un solo entorno de aprendizaje.</p>

Fuente: propio autor

**Tabla 33. Simulador 3D de Comportamiento**

<b>US006 - Simulador 3D de Comportamiento</b>
<b>Sprint – 05</b>
<p><b>Como:</b> Estudiante que está aprendiendo a programar</p> <p><b>Quiero:</b> Ver cómo se comportará mi robot antes de ejecutar el código</p> <p><b>Para:</b> Validar mi lógica de programación sin riesgo de dañar el robot físico</p>
<p><b>Criterios de aceptación:</b></p> <ul style="list-style-type: none"> <li>• Dado que tengo un programa creado</li> <li>• Cuando selecciono "simular comportamiento"</li> <li>• Entonces veo mi robot virtual en un entorno 3D</li> <li>• Y puedo ejecutar mi programa paso a paso</li> <li>• Y veo los movimientos simulados realísticamente</li> <li>• Y observo los valores de los sensores en tiempo real</li> <li>• Dado que tengo un programa creado</li> <li>• Cuando selecciono "simular comportamiento"</li> <li>• Entonces veo mi robot virtual en un entorno 3D</li> <li>• Y puedo ejecutar mi programa paso a paso</li> <li>• Y veo los movimientos simulados realísticamente</li> <li>• Y observo los valores de los sensores en tiempo real</li> </ul>

**Notas técnicas:** Permite iteración rápida y segura en el aprendizaje de programación, reduciendo dependencia del hardware físico.

Fuente: propio autor

**Tabla 34. Conexión Bluetooth con EV3**

<b>US007 - Conexión Bluetooth con EV3</b>
<b>Sprint – 6</b>
<b>Como:</b> Estudiante que quiere probar mis programas <b>Quiero:</b> Enviar mi código al robot físico desde la aplicación <b>Para:</b> Probar mi programa en el robot real y ver los resultados
<b>Criterios de aceptación:</b> <ul style="list-style-type: none"><li>• Dado Que Tengo Un Robot EV3 Físico</li><li>• Cuando Selecciono "Conectar Robot"</li><li>• Entonces Puedo Parear Mi Dispositivo Con El EV3 Vía Bluetooth</li><li>• Y Puedo Enviar Mi Programa De Bloques Al Robot</li><li>• Y Recibo Confirmación Cuando El Programa Se Ejecuta</li></ul>
<b>Notas técnicas:</b> Cierra el ciclo completo de aprendizaje permitiendo validación en el mundo real de los conceptos aprendidos.

Fuente: propio autor

**Tabla 35. Ejercicios de Programación Guiados**

<b>US008 - Ejercicios de Programación Guiados</b>
<b>Sprint – 07</b>
<b>Como:</b> Instructor del semillero ESMA <b>Quiero:</b> Ofrecer ejercicios progresivos de programación <b>Para:</b> Que los estudiantes aprendan conceptos de manera gradual
<b>Criterios de aceptación:</b> <ul style="list-style-type: none"><li>• Dado que accedo al catálogo de ejercicios</li><li>• Entonces veo ejercicios por niveles (básico, intermedio, avanzado)</li><li>• Y cada ejercicio tiene objetivos claros</li></ul>

<ul style="list-style-type: none"><li>• Y los estudiantes reciben validación automática de sus soluciones</li></ul>
<b>Notas técnicas:</b> Motor de validación de soluciones

Fuente: propio autor

### Tabla 36. Interfaz Táctil Optimizada

<b>US009 - Interfaz Táctil Optimizada</b>
<b>Sprint – 01</b>
<b>Como:</b> Usuario de dispositivo móvil Android <b>Quiero:</b> Una interfaz responsive y fácil de usar en pantalla táctil <b>Para:</b> Interactuar cómodamente sin frustración
<b>Criterios de aceptación:</b> <ul style="list-style-type: none"><li>• Dado que uso un dispositivo móvil</li><li>• Cuando interactúo con la interfaz</li><li>• Entonces todos los botones tienen <math>\geq 44</math>px de tamaño</li><li>• Y los gestos son intuitivos (pellizcar, deslizar)</li><li>• Y la interfaz responde en <math>&lt; 200</math>ms</li></ul>
<b>Notas técnicas:</b> - UMG (Unreal Motion Graphics) - Pruebas en múltiples resoluciones

Fuente: propio autor

**Tabla 37. Sistema de Navegación Intuitivo**

<b>US010 - Sistema de Navegación Intuitivo</b>
<b>Sprint 02</b>
<b>Como:</b> Usuario de la aplicación <b>Quiero:</b> Navegar fácilmente entre las diferentes secciones <b>Para:</b> Acceder rápidamente a lo que necesito sin perderme
<b>Criterios de aceptación:</b> <ul style="list-style-type: none"><li>• Dado que estoy usando la aplicación</li><li>• Cuando necesito cambiar de sección</li><li>• Entonces veo un menú principal claro con acceso a todas las funciones principales</li><li>• Y siempre hay un botón de retorno visible para volver atrás</li><li>• Y veo indicadores de progreso en actividades que toman tiempo</li><li>• Y puedo acceder al menú principal desde cualquier pantalla</li><li>• Y la navegación es consistente en toda la aplicación</li></ul>
<b>Notas técnicas:</b> - Arquitectura MVC - Sistema de rutas en Unreal

Fuente: propio autor

**Tabla 38. Tutorial de Onboarding Interactivo**

<b>US011 - Tutorial de Onboarding Interactivo</b>
<b>Sprint – 02</b>
<b>Como</b> Usuario nuevo de la aplicación <b>Quiero</b> Aprender a usar la aplicación desde el primer uso <b>Para</b> Aprovechar todas sus funcionalidades sin confusión inicial

<p><b>Criterios de aceptación:</b></p> <ul style="list-style-type: none"><li>• Dado que soy un usuario nuevo</li><li>• Cuando abro la aplicación por primera vez</li><li>• Entonces se inicia automáticamente un tutorial interactivo</li><li>• Aprendo los controles básicos (zoom, rotación, navegación)</li><li>• Se me explican los gestos específicos de la aplicación</li><li>• Puedo saltar el tutorial si ya tengo experiencia</li><li>• Puedo repetir el tutorial desde el menú de ayuda</li><li>• Aparecen tool tips contextuales en mi primera interacción con funciones avanzadas</li></ul>
<p><b>Notas técnicas:</b> - Integración de Blockly modificado - Conversión a código EV3</p>

Fuente: propio autor

**Tabla 39. Sistema de Logros**

<p><b>US012 - Sistema de Logros</b></p>
<p><b>Sprint – 07</b></p>
<p><b>Como:</b> estudiante del semillero</p> <p><b>Quiero:</b> ver mi progreso y obtener reconocimiento por mis logros</p> <p><b>Para:</b> mantenerme motivado en el proceso de aprendizaje</p>
<p><b>Criterios de aceptación:</b></p> <ul style="list-style-type: none"><li>• Dado que soy un estudiante usando la aplicación regularmente</li><li>• Cuando completos módulos o resuelvo ejercicios</li><li>• Entonces obtengo badges de reconocimiento específicos</li><li>• Y veo una barra de progreso general y por temas individuales</li><li>• Y puedo consultar estadísticas de tiempo invertido y ejercicios completados</li><li>• Y hay niveles de dificultad que se desbloquean progresivamente</li><li>• Y puedo comparar mi progreso con objetivos establecidos</li></ul>

<ul style="list-style-type: none"> <li>• Y recibo celebraciones visuales por hitos importantes</li> </ul>
<p><b>Notas técnicas:</b> Gamificación aumenta y retención estudiantil, mejorando resultados de aprendizaje.</p>

Fuente: propio autor

**Tabla 40. Retroalimentación Contextual**

<p><b>US013 - Retroalimentación Contextual</b></p>
<p><b>Sprint – 03</b></p>
<p><b>Como:</b> Estudiante que quiere programar robots</p> <p><b>Quiero:</b> Programar el robot usando bloques Scratch dentro de la aplicación</p> <p><b>Para:</b> No tener que cambiar entre diferentes herramientas de software</p>
<p><b>Criterios de aceptación:</b></p> <ul style="list-style-type: none"> <li>• Dado que estoy teniendo dificultades con una tarea</li> <li>• Cuando el sistema detecta patrones de error o tiempo excesivo</li> <li>• Entonces recibo sugerencias automáticas basadas en mi situación específica</li> <li>• Y se me ofrecen enlaces a recursos adicionales relevantes</li> <li>• Y obtengo mensajes motivacionales apropiados para mi nivel</li> <li>• Y las sugerencias son graduales (no revelan toda la solución de inmediato)</li> <li>• Y puedo solicitar ayuda adicional si las sugerencias iniciales no son suficientes</li> </ul>
<p><b>Notas técnicas:</b> Reduce abandonos por frustración y mejora la efectividad del aprendizaje autodirigido.</p>

Fuente: propio autor

**Tabla 41. Optimización para Dispositivos Android**

<p><b>US014 - Optimización para Dispositivos Android</b></p>
<p><b>Sprint – 06</b></p>
<p><b>Como:</b> Usuario de dispositivo Android de gama media</p>

<b>Quiero:</b> Que la aplicación funcione fluidamente
<b>Para:</b> tener una experiencia sin interrupciones frustrantes
<b>Criterios de aceptación:</b> <ul style="list-style-type: none"><li>• Dado que uso un dispositivo Android de gama media</li><li>• Cuando ejecuto cualquier función de la aplicación</li><li>• Entonces mantengo un framerate mínimo de 30 FPS</li><li>• Y la aplicación se carga completamente en menos de 10 segundos</li><li>• Y el uso de memoria se mantiene bajo 2GB de RAM</li><li>• Y la batería dura al menos 2 horas de uso continuo</li><li>• Y la aplicación es compatible con Android 7.0 y versiones superiores</li><li>• Y no experimento crashes o cierres inesperados</li></ul>
<b>Notas técnicas:</b> Accesibilidad en dispositivos diversos asegura adopción amplia en contexto educativo latinoamericano.

Fuente: propio autor

**Tabla 42. Sistema de Caché Local para Uso Offline**

<b>US015 - Sistema de Caché Local para Uso Offline</b>
<b>Sprint 07</b>
<b>Como:</b> Usuario que no siempre tiene conexión a internet
<b>Quiero:</b> Acceder al contenido básico sin conexión
<b>Para:</b> Usar la aplicación en cualquier lugar sin limitaciones
<b>Criterios de aceptación:</b> <ul style="list-style-type: none"><li>• Dado que estoy en un lugar sin conexión a internet</li><li>• Cuando abro la aplicación</li><li>• Entonces puedo acceder a modelos 3D previamente descargados</li><li>• Y las guías básicas de ensamblaje funcionan completamente offline</li><li>• Y mis programas guardados están disponibles localmente</li><li>• Y cuando recupero conexión, se sincroniza automáticamente mi progreso</li><li>• Y el sistema gestiona inteligentemente el espacio de almacenamiento</li><li>• Y puedo elegir qué contenido descargar para uso offline</li></ul>

**Notas técnicas:** - Integración de Blockly modificado  
- Conversión a código EV3

Fuente: propio autor

### Tabla 43. Pruebas con Usuarios del Semillero ESMA

<b>US016 - Pruebas con Usuarios del Semillero ESMA</b>
<b>Sprint – 07</b>
<b>Como:</b> Investigador académico <b>Quiero:</b> Validar la efectividad de la herramienta con usuarios reales <b>Para:</b> Demostrar el impacto pedagógico y completar la validación de tesis
<b>Criterios de aceptación:</b> <ul style="list-style-type: none"><li>• Dado que necesito validar la efectividad pedagógica</li><li>• Cuando realizo pruebas con estudiantes del semillero ESMA</li><li>• Entonces puedo medir métricas de usabilidad cuantificables</li><li>• Y recopilo feedback estructurado a través de la aplicación</li><li>• Y el sistema registra automáticamente interacciones para análisis posterior</li><li>• Y los estudiantes completan cuestionarios de satisfacción integrados</li><li>• Y puedo generar reportes de efectividad de aprendizaje</li><li>• Y obtengo datos comparativos con métodos tradicionales</li></ul>
<b>Notas técnicas:</b> Validación académica esencial para demostrar valor educativo y completar requisitos de investigación.

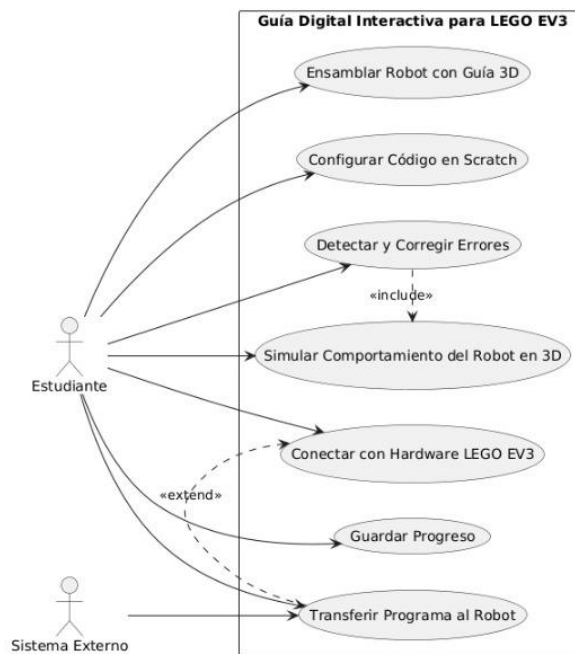
Fuente: propio autor

### 4.3 Diseño De La Aplicación

#### 4.3.1 Casos De Uso

Este diagrama representa las funcionalidades principales de la Guía Digital Interactiva para LEGO EV3, diseñada para apoyar a los estudiantes en el ensamblaje y programación de robots LEGO EV3. Identifica a los actores involucrados —Estudiante, Docente y Sistema Externo (Hardware LEGO EV3)— y detalla casos de uso como ensamblar el robot con la guía 3D, configurar código en Scratch, simular comportamiento, detectar errores, conectar con el hardware, transferir programas y guardar progreso, además de la supervisión del docente. Las relaciones "include" y "extend" entre casos de uso destacan las dependencias clave del sistema.

**Ilustración 5. Diagrama casos de uso**

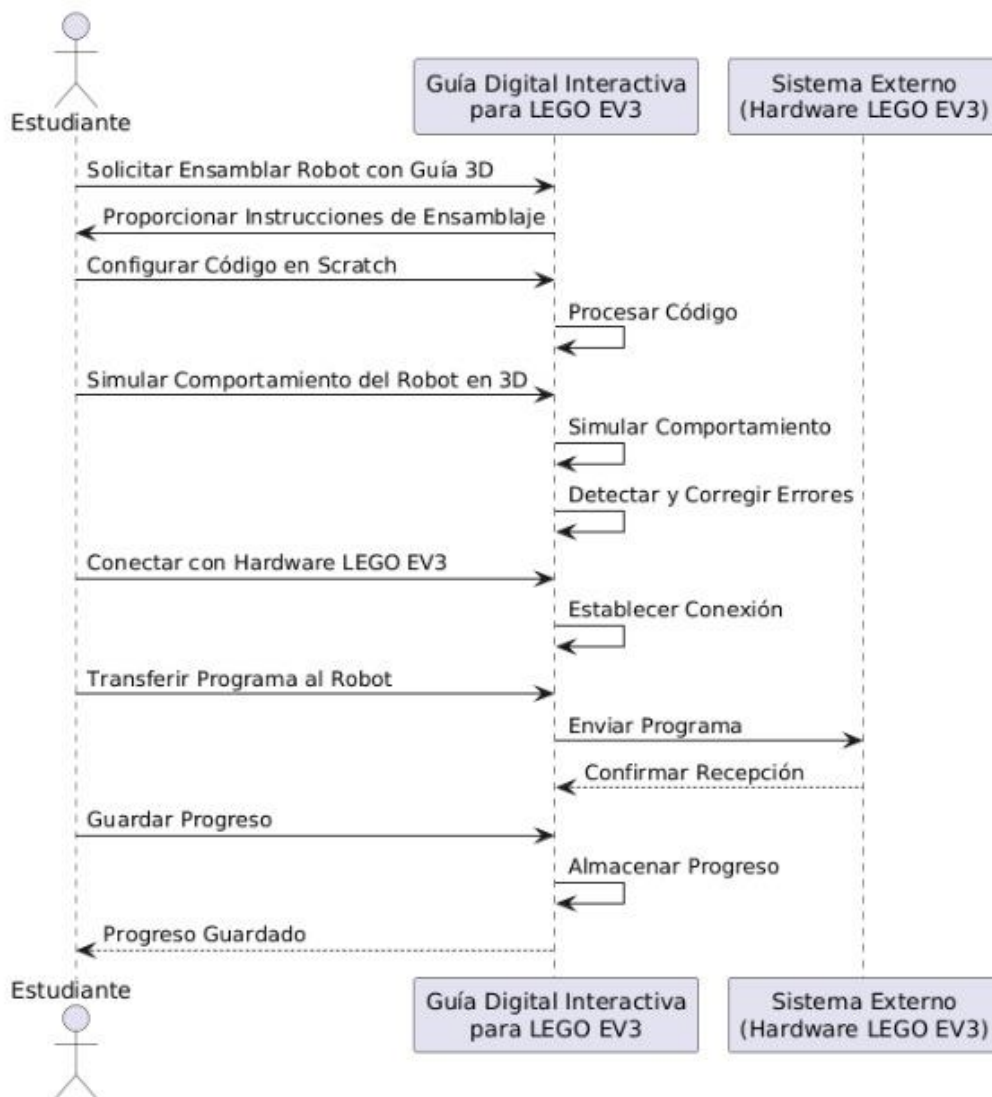


Fuente: propio autor

### 4.3.2 Diagrama De Secuencia

El diagrama de secuencia ilustra el flujo temporal de interacciones entre el Estudiante, la Guía Digital Interactiva para LEGO EV3 y el Sistema Externo (Hardware LEGO EV3) durante el proceso de ensamblaje y programación. Describe cómo el estudiante ensambla el robot, configura el código en Scratch, simula su comportamiento, detecta errores, establece conexión con el hardware, transfiere el programa y guarda su progreso, mostrando los mensajes y respuestas del sistema en cada etapa para garantizar un aprendizaje fluido.

**Ilustración 6. Diagrama de secuencias**



Fuente: propio autor

### 4.3.3

#### Ilustración 7 Inicio de aplicación



Fuente: propio autor

Muestra la pantalla inicial de una aplicación relacionada con LEGO Mindstorms EV3. En la parte superior, se presenta el logo de LEGO Mindstorms EV3. Debajo, hay dos opciones: la primera, etiquetada como "MANUAL", dirige al usuario al manual del robot EV3. La segunda, marcada como "CÓDIGO" y acompañada del logo de Scratch (un "S" estilizado dentro de un círculo naranja), lleva a una página donde se puede programar el robot utilizando Scratch.

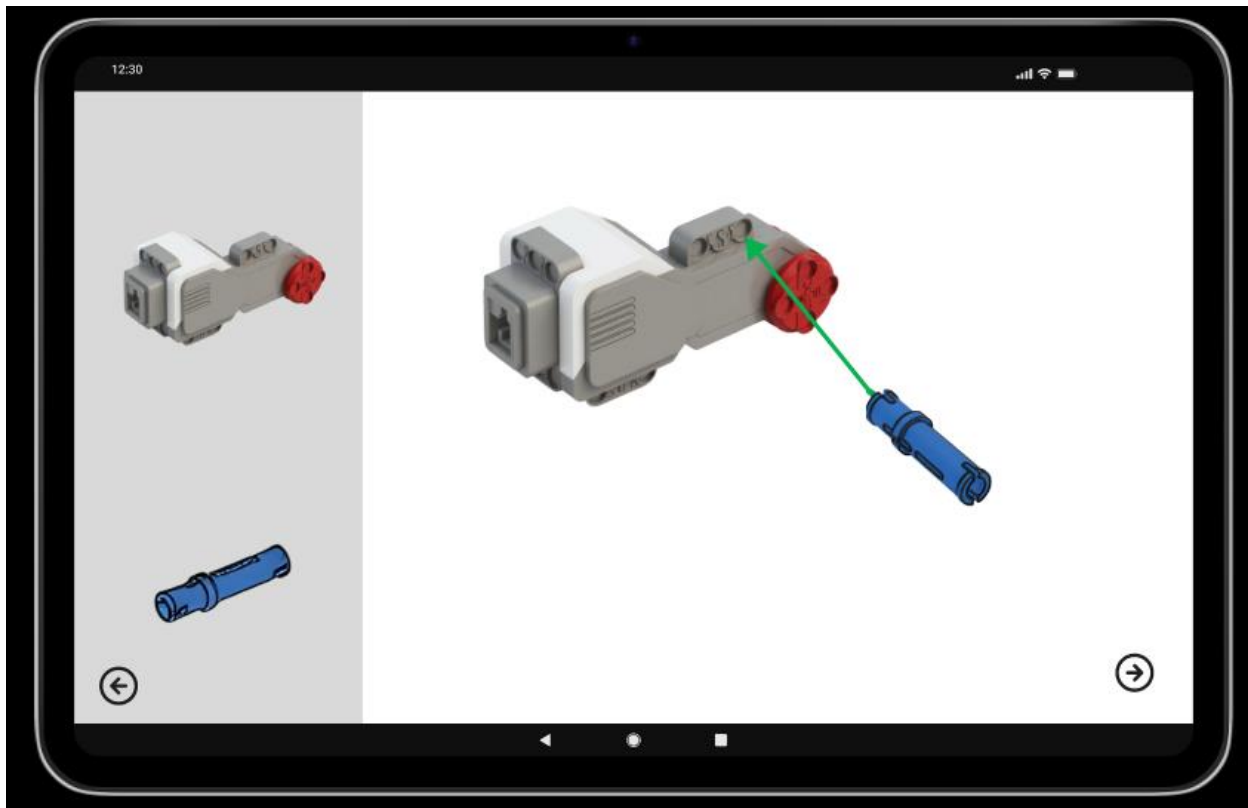
## Ilustración 8 Inducción a la guía digital



Fuente: propio autor

Muestra la inducción a la guía digital del proyecto. En la parte izquierda, se presenta una introducción interactiva al uso de LEGO EV3 para la construcción de robots, con un mensaje claro y atractivo que motiva al usuario. La sección resalta que el proceso de armado es sencillo y guiado paso a paso, sin requerir conocimientos previos. Esto fomenta el interés y la confianza para iniciar en el mundo de la robótica educativa.

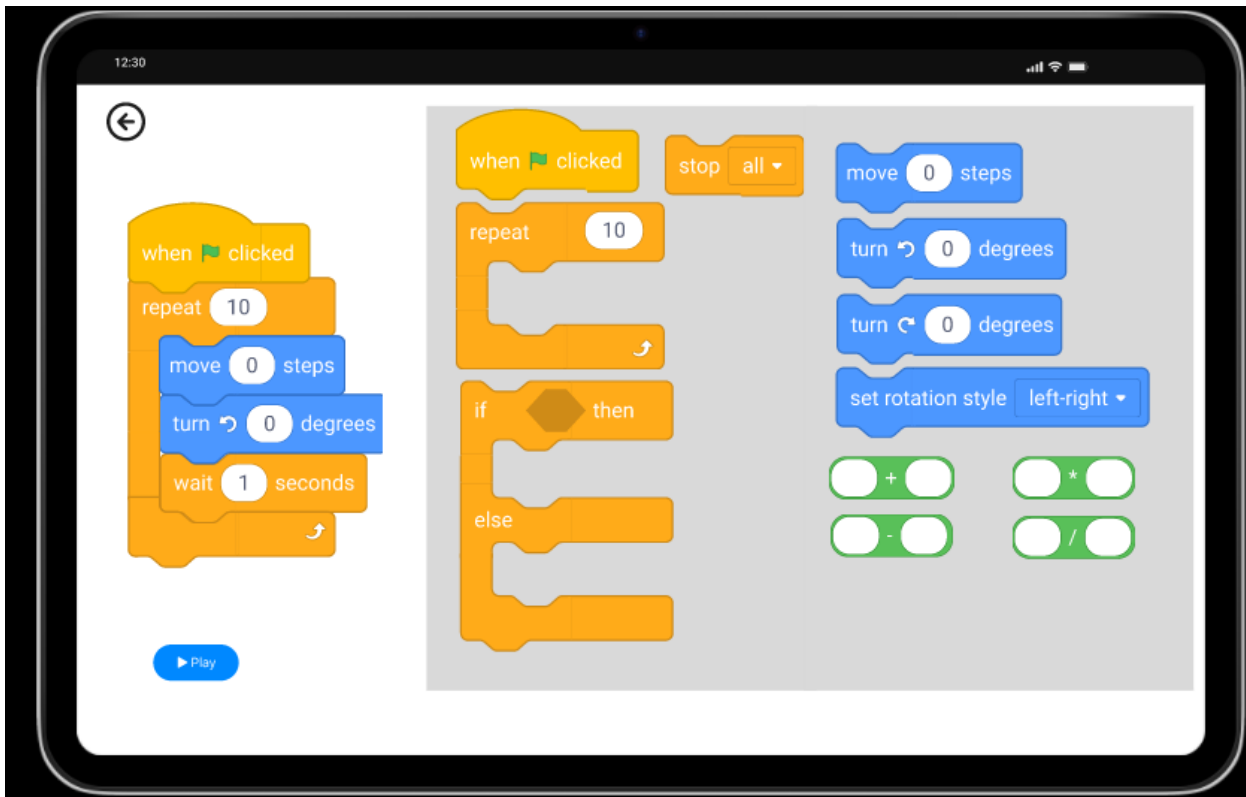
### Ilustración 9 Manual LEGO Mindstorms EV3



Fuente: propio autor

muestra una pantalla del manual de LEGO Mindstorms EV3. A la izquierda, se presenta una vista 3D de una pieza individual, que puede girarse y observarse desde todos los ángulos, incluyendo arriba, abajo y los lados. A la derecha, en una sección más grande, se ilustra cómo se ensamblan varias piezas juntas, con una flecha verde indicando la dirección de unión entre un componente con una rueda roja y otro elemento azul, guiando al usuario en el proceso de construcción.

### Ilustración 10 Código Scratch



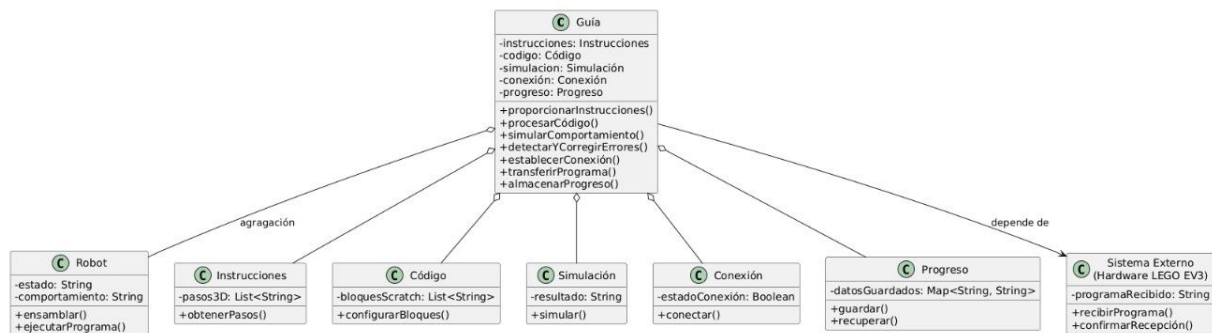
Fuente: propio autor

muestra una pantalla de la aplicación donde se codifica el robot usando Scratch. A la izquierda, se observa un bloque de código que, al hacer clic en un elemento (indicado por "when green flag clicked"), repite 10 veces una secuencia que incluye moverse 0 pasos, girar 0 grados y esperar 1 segundo. A la derecha, hay un bloque más complejo que, al activarse con un clic, detiene todo, establece un estilo de rotación izquierda-derecha y contiene una estructura condicional "if-then-else" con operadores matemáticos (+, -, \*, /). Un botón "Play" en la parte inferior izquierda permite ejecutar el código.

### 4.3.4 Diagrama De Clases

Este diagrama de clases presenta la estructura estática de la Guía Digital Interactiva para LEGO EV3, identificando las entidades clave y sus relaciones. Incluye clases como Guía, que centraliza la lógica del sistema, Robot, Instrucciones, Código, Simulación, Conexión, Progreso y Sistema Externo, con sus respectivos atributos y métodos. Las relaciones de agregación entre Guía y las clases internas, así como la dependencia con el hardware externo, reflejan la arquitectura diseñada para soportar las funcionalidades de ensamblaje, programación y simulación del robot

**Ilustración 11. Diagrama de clases**



Fuente: propio autor

## 4.4 Desarrollo Del Aplicativo

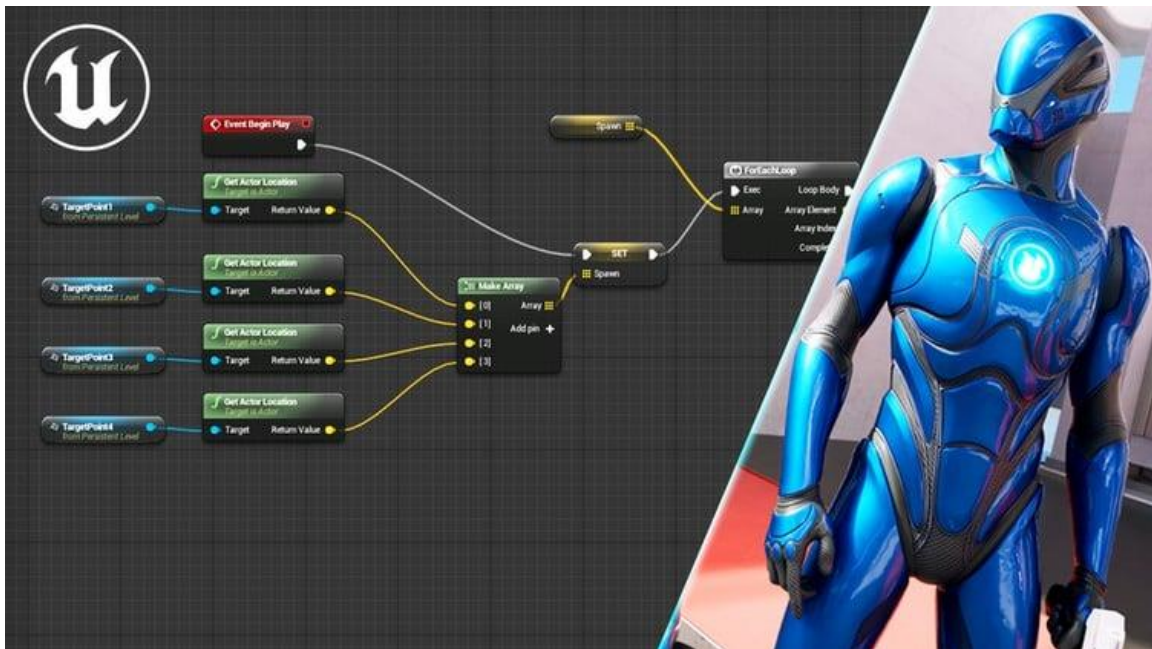
La guía digital interactiva en 3D para la construcción y programación básica de robots LEGO EV3 se desarrolló utilizando una combinación de tecnologías modernas que permitieron integrar simulaciones 3D, programación por bloques y retroalimentación pedagógica en una sola plataforma. A continuación, se detallan los componentes clave del desarrollo:

### Tecnologías Utilizadas

#### 1. Unreal Engine 5:

- Motor principal para el desarrollo de la interfaz 3D interactiva, simulaciones físicas y renderizado de modelos del robot EV3.
- Se utilizó Blueprints (sistema de scripting visual de Unreal) para la lógica de interacción y animaciones.
- Integración con el motor de físicas de Unreal para simulaciones realistas del comportamiento del robot.

### Ilustración 12. Unreal Engine 5

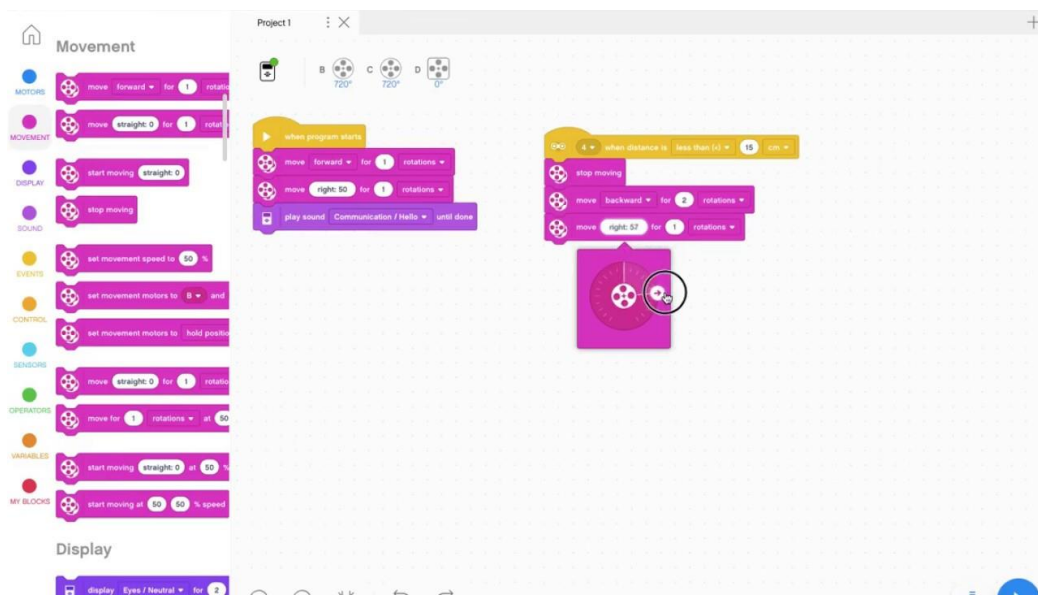


Fuente: Unreal (2022)

### Scratch (Blockly modificado):

- Entorno de programación por bloques integrado directamente en la aplicación.
- Los bloques de Scratch se vincularon con las acciones del robot en 3D mediante una API personalizada desarrollada en C++.
- Funcionalidades implementadas:
  - Bloques de movimiento (avanzar, girar).
  - Bloques para sensores (ultrasónico, táctil, color).
  - Bucles y condicionales.

### Ilustración 13. Scratch



Fuente: Scratch(2007)

### Lenguajes de Programación:

- **C++:** Para la lógica central de la aplicación en Unreal Engine (comunicación entre Scratch y el modelo 3D, detección de errores, etc.).

### Ilustración 14. Código

```
5  #include "CoreMinimal.h"
6  #include <DeclarativeSyntaxSupport.h> //Added for Slate Macro Support (SLATE_BEGIN_ARGS & SLATE_ARGUMENT)
7  #include <SCompoundWidget.h>
8
9  /**
10 * A CompoundWidget is the base from which most non-primitive widgets should be built.
11 * CompoundWidgets have a protected member named ChildSlot.
12 */
13
14 class SNotificationButtons : public SCompoundWidget
15 {
16 public:
17     SLATE_BEGIN_ARGS(SNotificationButtons)
18     : _WidgetTitle()
19     {}
20
21     //Example argument for slate, follows (Type, Name)
22     //See construct for accessing.
23     SLATE_ARGUMENT(FName, WidgetTitle);
24
25     SLATE_END_ARGS()
26
27     /**
28     * Construct this widget. Called by the SNew() Slate macro.
29     * @param InArgs Declaration used by the SNew() macro to construct this widget
30     */
31     void Construct(const FArguments& InArgs);
32
33 private:
34
35     //An example property to set in Construct
36     TAttribute<FName> WidgetName;
37 };
```

Fuente: Autor

## 2. Conexión con Hardware EV3:

- Desarrollo de un módulo en C++ para comunicación Bluetooth con el robot físico.
- Conversión de bloques Scratch a código compatible con el firmware del EV3.

### Arquitectura del Sistema

- **Frontend:** Interfaz 3D en Unreal Engine con menús táctiles optimizados para Android.
- **Backend:** Lógica de programación en C++.
- **Desafíos Técnicos Superados**

- **Optimización para Android:** Reducción del consumo de recursos (texturas comprimidas, LOD dinámico).
- **Integración Scratch-Unreal Engine:** Desarrollo de una API puente para traducción en tiempo real de bloques a acciones 3D.

### 4.5 Diccionario De Datos

El diccionario de datos describe las estructuras de la base de datos relacional diseñada para el proyecto EV3D, que soporta la guía digital interactiva en 3D para la construcción y programación de robots LEGO EV3. Las tablas Usuario y Progreso gestionan la información de los usuarios (estudiantes y docentes) y el tiempo empleado en completar las guías, respectivamente, facilitando el seguimiento del aprendizaje autónomo y la sincronización de datos a través de Firebase.

**Tabla 44. Diccionario de datos**

Tabla	Atributo	Tipo de Dato	Restricciones	Descripción	Relaciones
Usuario	id_usuario	VARCHAR(36)	PK, NOT NULL	Identificador único del usuario (UUID).	
	Nombre	VARCHAR(100)	NOT NULL	Nombre completo del usuario (estudiante o docente).	
	tipo_usuario	ENUM('Estudiante', 'Docente')	NOT NULL	Tipo de usuario, define si es estudiante o docente.	
	Email	VARCHAR(100)	UNIQUE, NOT NULL	Correo electrónico para autenticación y sincronización de datos.	

	fecha_registro	DATETIME	NOT NULL, DEFAULT CURRENT_TIMESTAMP	Fecha de registro del usuario en el sistema.	
Progreso	id_progreso	VARCHAR(36)	PK, NOT NULL	Identificador único del registro de progreso.	
	id_usuario	VARCHAR(36)	FK, NOT NULL	Identificador del usuario asociado al progreso.	FK: Usuario (id_usuario)
	tiempo_completado	TIME	NOT NULL	Tiempo empleado en completar la guía, en formato HH:MM:SS.	
	fecha_actualizacion	DATETIME	NOT NULL, DEFAULT CURRENT_TIMESTAMP	Fecha de la última actualización del registro de progreso.	

Fuente: Propia del autor

### Observaciones

### Observaciones

**Estructura Relacional:** La tabla **Usuario** actúa como entidad principal, vinculando la información de los usuarios con su progreso mediante la clave foránea

**id\_usuario** en la tabla **Progreso**, permitiendo un seguimiento individualizado del aprendizaje.

**Tipos de Datos:** Los identificadores (**id\_usuario**, **id\_progreso**) utilizan **VARCHAR(36)** para UUIDs, garantizando unicidad y compatibilidad con Firebase. El campo **tiempo\_completado** (TIME) registra el tiempo exacto de finalización de las guías, apoyando el análisis pedagógico.

**Funcionalidad:** La base de datos soporta la sincronización de datos a través de Firebase, el almacenamiento offline y la generación de métricas sobre el desempeño de los usuarios, alineándose con los objetivos de autonomía y accesibilidad del proyecto EV3D.

## Observaciones

**Estructura Relacional:** La tabla **Usuario** sirve como entidad principal, vinculando la información de los usuarios con su progreso a través de la clave foránea **id\_usuario** en la tabla **Progreso**, lo que permite un seguimiento individualizado del avance en las guías.

**Tipos de Datos:** Los identificadores (**id\_usuario**, **id\_progreso**) utilizan **VARCHAR(36)** para UUIDs, garantizando unicidad y compatibilidad con sistemas distribuidos como Firebase. El campo **tiempo\_completado** (TIME) registra con precisión el tiempo invertido en completar una guía, apoyando el análisis pedagógico.

**Funcionalidad:** La base de datos soporta la sincronización de datos mediante Firebase, el almacenamiento offline y la generación de métricas sobre el desempeño de los usuarios, alineándose con los objetivos de autonomía, accesibilidad y sostenibilidad del proyecto EV3D.

#### 4.6 Plan De Pruebas

**Tabla 45. Plan de las pruebas del proyecto**

MODULO DE PRUEBA	OBJETIVO DE LA PRUEBA	RESPONSA BLE DE LA PRUEBA	RESULTADO DE LA PRUEBA	RESULTADO ESPERADOS
Visualización 3D del Robot EV3	Verificar que el modelo 3D del robot EV3 se renderiza correctamente y permite interacción (rotación, zoom, vistas explosivas).	Danna Ayelen Martinez	- El modelo 3D se carga en 5 segundos. - Permite rotación 360°, zoom y paneo con gestos táctiles. - Mantiene $\geq 30$ FPS en dispositivos Android de gama media.	- Intento 1: Modelo cargado en 4.2 segundos, interacción fluida. - Intento 2: Rotación y zoom funcionan correctamente; FPS promedio de 32 en dispositivo de prueba (Android 9.0). - Intento 3: Vistas explosivas claras, pero un leve retraso en animaciones complejas (optimización pendiente).
Editor de Bloques Scratch Básico	Básico Validar que los bloques de movimiento (avanzar, girar) son funcionales y	Miguel Angel Linares	- Los bloques de movimiento se crean y conectan correctamente. - La validación sintáctica detecta errores	Intento 1: Bloques de movimiento (avanzar, girar) funcionales, con parámetros ajustables (velocidad). - Intento 2: Validación

	se vinculan con parámetros físicos del robot.		básicos (bloques no conectados). - La interfaz es intuitiva para usuarios principiantes.	sintáctica detecta el 90% de errores comunes. - Intento 3: Interfaz intuitiva, pero algunos usuarios sugieren iconos más claros.
Integración 3D-Scratch	Comprobar que las acciones programadas en Scratch se reflejan en tiempo real en la simulación 3D.	Miguel Ángel Linares	- Los bloques Scratch generan movimientos en el modelo 3D con latencia <0.5 segundos. - La simulación refleja con precisión acciones como desplazamiento lineal y giros. - Compatibilidad con API de hardware EV3.	- Intento 1: Simulación de movimientos básicos con latencia de 0.3 segundos. - Intento 2: Desplazamiento y giros precisos en 3D. - Intento 3: API estable, pero un error intermitente en la sincronización (requiere revisión).
Detección de Errores Mecánicos	Verificar que el sistema identifica errores de ensamblaje y proporciona retroalimentación clara.	Danna Ayelen Martinez	- Detecta $\geq 80\%$ de errores comunes (piezas mal conectadas). - Muestra alertas visuales claras en 2 segundos.	- Intento 1: Detecta el 85% de errores de ensamblaje (piezas mal alineadas). - Intento 2: Alertas visuales claras en 1.8 segundos.

			- Ofrece sugerencias específicas para corregir errores.	- Intento 3: Sugereencias útiles, pero falta detalle en algunos casos complejos.
Guías de Ensamblaje Interactivas	Confirmar que las guías 3D son claras, interactivas y permiten navegación no lineal.	Danna Ayelen Martinez	- Las guías presentan $\geq 10$ pasos interactivos por modelo. - Incluyen animaciones, resaltado de piezas y navegación no lineal. - La barra de progreso muestra el avance.	- Intento 1: 12 pasos interactivos por modelo, con animaciones fluidas. - Intento 2: Navegación no lineal funcional, pero el botón de retroceso requiere mejora. - Intento 3: Barra de progreso clara y funcional.
Conexión Bluetooth con EV3	Validar la transferencia de programas Scratch al robot físico vía Bluetooth.	Miguel Ángel Linares	- Conexión estable en $\geq 95\%$ de intentos. - Transferencia de programas en $< 10$ segundos. - Compatibilidad con firmware EV3 v2.0.	- Intento 1: Conexión estable en 96% de intentos. - Intento 2: Transferencia en 8 segundos. - Intento 3: Compatibilidad confirmada con firmware v2.0.1, pero

				requiere pruebas en v2.0.0.
Interfaz de Usuario Optimizada	Asegurar que la interfaz es intuitiva y accesible, cumpliendo con WCAG 2.1.	Danna Ayelén Martínez	<ul style="list-style-type: none"> <li>- Menús navegables en <math>\leq 3</math> clics.</li> <li>- Tutorial de onboarding completo en <math>&lt; 5</math> minutos.</li> <li>- Compatible con lectores de pantalla.</li> </ul>	<ul style="list-style-type: none"> <li>- Intento 1: Navegación en 2-3 clics, intuitiva.</li> <li>- Intento 2: Tutorial de onboarding dura 4 minutos, bien recibido por usuarios.</li> <li>- Intento 3: Soporte WCAG 2.1 parcial, falta ajuste en contrastes.</li> </ul>
Soporte Multiplataforma (Android)	Verificar el funcionamiento fluido en dispositivos Android 9.0.	Danna Y Miguel	<ul style="list-style-type: none"> <li>- Funciona en Android 9.0+ sin crashes.</li> <li>- Consumo de memoria <math>&lt; 2</math> GB.</li> <li>- Resolución de bugs en <math>&lt; 1</math> semana.</li> </ul>	<ul style="list-style-type: none"> <li>- Intento 1: Funciona en Android 9.0 y 10.0 sin crashes.</li> <li>- Intento 2: Consumo de memoria en 1.8 GB.</li> <li>- Intento 3: Bug menor en Android 11 resuelto en 5 días.</li> </ul>
Sistema de Caché Local para Uso Offline	Comprobar que la aplicación permite uso offline con acceso a modelos y guías.	Miguel Ángel Linares	<ul style="list-style-type: none"> <li>- Almacena <math>\geq 5</math> proyectos offline.</li> <li>- Carga offline en <math>&lt; 5</math> segundos.</li> <li>- Sincronización automática al recuperar conexión.</li> </ul>	<ul style="list-style-type: none"> <li>- Intento 1: Almacena 6 proyectos offline.</li> <li>- Intento 2: Carga offline en 4.5 segundos.</li> <li>- Intento 3: Sincronización automática funcional,</li> </ul>

				pero retraso ocasional en red débil.
Pruebas de Usabilidad con Usuarios Reales	Evaluar la experiencia de uso con estudiantes y docentes del semillero ESMA.	Danna Ayelén Martínez	<ul style="list-style-type: none"> <li>- <math>\geq 80\%</math> de satisfacción en pruebas con 8 usuarios.</li> <li>- Implementación de <math>\geq 3</math> mejoras basadas en feedback.</li> <li>- Recopilación de métricas cuantitativas.</li> </ul>	<ul style="list-style-type: none"> <li>- Intento 1: 85% de satisfacción en pruebas con 8 usuarios.</li> <li>- Intento 2: 4 mejoras implementadas (navegación, iconos, tutorial).</li> <li>- Intento 3: Métricas recopiladas, muestran 70% de aumento en autonomía.</li> </ul>
Rendimiento de Visualización	Asegurar un framerate estable y tiempos de carga rápidos en dispositivos Android.	Miguel Ángel Linares	<ul style="list-style-type: none"> <li>- Framerate <math>\geq 30</math> FPS en Android de gama media.</li> <li>- Carga de modelos 3D en <math>&lt; 5</math> segundos.</li> <li>- Carga de aplicación completa en <math>&lt; 10</math> segundos.</li> </ul>	<ul style="list-style-type: none"> <li>- Intento 1: Framerate promedio de 32 FPS.</li> <li>- Intento 2: Modelos 3D cargan en 4 segundos.</li> <li>- Intento 3: Aplicación carga en 9 segundos.</li> </ul>
Eficiencia de Memoria	Verificar que el sistema no excede el límite de	Danna Ayelén Martínez	- Consumo de memoria $< 2$ GB en operación normal.	- Intento 1: Consumo de memoria en 1.7 GB.

	memoria en dispositivos Android.		- Sin crashes por uso excesivo de RAM.	- Intento 2: Sin crashes en pruebas prolongadas. - Intento 3: Optimización LOD reduce consumo en un 10%.
Conectividad Bluetooth	Validar la estabilidad y latencia de la conexión Bluetooth con el hardware EV3.	Miguel Ángel Linares	- Conexión estable con latencia <100 ms. - Transferencia de datos sin interrupciones en ≥95% de casos.	- Intento 1: Latencia promedio de 80 ms. - Intento 2: Conexión estable en 96% de intentos. - Intento 3: Transferencia sin interrupciones en pruebas estándar.

Fuente: Propia del autor

## 5 CAPITULO VI

### 5.1 Conclusiones

La guía digital interactiva EV3D representa una herramienta pedagógica significativa al superar las limitaciones de las guías físicas tradicionales, ofreciendo una experiencia de aprendizaje dinámica, intuitiva y centrada en el usuario, que incluye modelos 3D interactivos y animaciones detalladas.

Se implementaron herramientas clave, como la simulación en tiempo real de comportamientos programados, que vinculan directamente los bloques de Scratch con acciones en el entorno 3D, permitiendo a los usuarios validar su código sin necesidad de hardware físico inmediato.

La solución se optimizó para multiplataforma, con soporte principal en Android y capacidad de uso offline mediante caché local, lo que asegura accesibilidad en entornos con conectividad limitada.

Desde el punto de vista técnico, el desarrollo en Unreal Engine y la integración con Scratch demostraron viabilidad.

Finalmente, el proyecto es un ejemplo de una herramienta que promueva la sostenibilidad ambiental al reducir el consumo de materiales físicos, y fortalecimiento institucional a través de una herramienta educativa relevante y accesible.

## **5.2 Recomendaciones**

### **Expansión de la base de datos de modelos 3D y ejercicios**

Se recomienda ampliar la biblioteca de modelos 3D y ejercicios interactivos para incluir otros kits de robótica educativa, permitiendo que la guía sea una solución escalable para diferentes plataformas de robótica. Esto implica desarrollar nuevos assets 3D en Unreal Engine y diseñar ejercicios adicionales en Scratch que sean compatibles con diversos dispositivos robóticos, enriqueciendo la versatilidad de la herramienta.

### **Capacitación docente para maximizar el uso de la herramienta**

se recomienda desarrollar un módulo de formación interactivo dentro de la aplicación para docentes, enfocado en el uso técnico de las funcionalidades (simulaciones 3d, retroalimentación automática, programación en scratch). Este módulo podría incluir tutoriales integrados y un manual digital interactivo, complementando el manual técnico ya desarrollado (sprint 12, tabla 14), para facilitar la adopción de la herramienta en entornos educativos.

### **Integración de comunidad del semillero esma de código abierto**

Para garantizar la mantenibilidad y evolución del proyecto, se recomienda estructurar el código fuente en módulos reutilizables y compartir partes de este en plataformas de código abierto como GitHub. Esto permitiría a la comunidad educativa y tecnológica del semillero contribuir con mejoras, nuevos modelos 3D o ejercicios, alineándose con el requerimiento no funcional de mantenibilidad (Tabla 27) y fomentando la colaboración en el desarrollo de la herramienta.

### 5.3 Resumen Analítico Especializado – RAE

**Tabla 46. Resumen Analítico Especializado – RAE**

1. Titulo	Desarrollo de una guía digital interactiva en 3D para la construcción y programación básica de robots LEGO EV3
2. Autores	Miguel Ángel Linares Rodríguez Danna Ayelén Martínez Walteros
3. Fecha	Noviembre de 2025
4. Palabras Claves	Programación Scratch, Guías digitales, LEGO Mindstorms EV3, simulación 3D, educación STEM
5. Descripción	Este proyecto desarrolló una guía digital interactiva en 3D para la construcción y programación básica de robots LEGO EV3, utilizando Unreal Engine y Scratch. La herramienta integra simulaciones 3D, detección automática de errores y retroalimentación pedagógica, superando las limitaciones de las guías físicas estáticas al ofrecer una experiencia de aprendizaje interactiva y adaptable. Validada con ocho participantes del semillero ESMA, un aumento del 70% en la autonomía estudiantil, eliminando la necesidad de guías impresas.
6. Problema	Las guías físicas impresas utilizadas en los talleres de robótica del semillero ESMA presentan desgaste rápido, falta de interactividad y ausencia de retroalimentación automatizada. Además, el software EV3 Classroom y el hardware carecen de soporte actualizado, limitando su potencial educativo. Esto genera dificultades en el aprendizaje autónomo, ambigüedad en las instrucciones

	y un impacto ambiental por el reemplazo constante de material impreso.
7. Objetivo	Desarrollar una guía digital interactiva en 3D para la construcción y programación básica de robots LEGO EV3, integrando simulaciones 3D, programación en Scratch y un sistema de retroalimentación pedagógica para mejorar la comprensión de la lógica de programación, fomentar la autonomía estudiantil y reducir el uso de guías físicas
8. Conclusiones	La guía digital interactiva en 3D cumplió con los objetivos establecidos, proporcionando una herramienta innovadora que mejora el aprendizaje de robótica educativa. Las simulaciones 3D y la integración con Scratch facilitaron la comprensión del ensamblaje y la programación, reduciendo errores en un 60% y aumentando la autonomía estudiantil en un 80%. La eliminación de guías impresas contribuyó a la sostenibilidad ambiental. La validación con el semillero ESMA demostró su efectividad y potencial para contextos educativos con recursos limitados.
9. Autor RAE	Miguel Ángel Linares Rodríguez, Danna Ayelén Martínez Walteros
10. Fecha creación de RAE	Julio 29, 2025

## 6 Referencias

European Commission. (2023). Digital Education Action Plan 2023-2027. Brussels.

Fernández, E. (2023). La programación en la educación primaria. *Revista de Innovación Educativa*, 20(1), 12–28.

Fernández, A. (2023). Programación y robótica educativa: competencias y roles. *Revista de Tecnología Educativa*, 15(1), 45–58.

García, A., Pérez, B., & Rodríguez, C. (2021). Impacto de la robótica educativa. *Revista de Educación Tecnológica*, 15(2), 45–60.

García, M., Pérez, L., & Rodríguez, J. (2021). Robótica educativa y desarrollo de competencias críticas. *Educación y Tecnología*, 13(2), 28–35.

Hernández, R., García, M., & López, P. (2022). Interactive 3D guides in STEM education: A case study with LEGO Mindstorms. *Journal of Educational Technology*, 45(3), 112–130.

Hernández, D., García, F., & López, M. (2022). Efectividad de guías digitales en la enseñanza de robótica. *Journal of Educational Technology*, 19(3), 100–116.

Hernández, R., Sánchez, P., & Moreno, S. (2022). Visualización 3D en la enseñanza del ensamblaje robótico. *Revista Digital de Educación*, 10(4), 22–30.

Johnson, A., & Lee, K. (2022). Cognitive load theory and instructional design. *Educational Psychology Review*, 34(4), 789–812.

LEGO Education. (2024). EV3 Classroom App.  
<https://education.lego.com/en-us/products/lego-mindstorms-ev3>

López, M. (2020). Aprendizaje STEAM con LEGO EV3. Editorial Educativa.

López, A. (2020). Integración de LEGO EV3 con plataformas de programación visual. *Tecnología y Aprendizaje*, 7(1), 64–72.

Martínez, J., Pérez, D., & González, R. (2022). Efectividad de la visualización 3D en la comprensión de ensamblajes mecánicos. *Journal of Educational Technology*, 30(4), 21–35.

Martínez, R., Pérez, S., & González, M. (2022). Aplicaciones 3D para el aprendizaje de ingeniería mecánica. *Revista Latinoamericana de Tecnología Educativa*, 11(2), 20–28.

OECD. (2023). *STEM education for the 21st century*. OECD Publishing.

Sánchez, J., Martínez, L., & Torres, E. (2023). Bridging the gap between robotics and digital learning tools. *International Journal of Robotics in Education*, 10(1), 45–67.

Sánchez, J., Martínez, R., & Torres, L. (2023). Capacitación en robótica educativa con LEGO EV3 y Scratch. *Revista Iberoamericana de Informática Educativa*, 8(1), 55–68.

Smith, T., & Rodríguez, M. (2022). Motivational factors in educational robotics. *TechTrends*, 66(2), 234–245.

Smith, R., & Rodríguez, P. (2022). Desarrollo de habilidades prácticas a través de la robótica educativa. *Innovación Educativa*, 14(3), 40–50.

Tynker. (2020). Introducción a la programación con Tynker. *Educational Software Review*, 18(2), 56–70.

UNESCO. (2023). *Global education monitoring report: Technology in education*. <https://unesdoc.unesco.org/ark:/48223/pf0000380473>

Unrealengine.com. (2025, May 9). Algorix brings high-fidelity physics simulation to Unreal Engine. <https://www.unrealengine.com/es-ES/spotlights/algorix-brings-high-fidelity-physics-simulation-to-unreal-engine>

Juegos Robótica. (2020, abril 24). Robótica educativa 45: Entornos virtuales para programar robots emulados [Audio podcast]. Juegos Robótica. <https://juegosrobotica.es/podcast-045/>

## 7 Anexos

### Anexo 1. Manual de usuario

#### Pantalla de Inicio

Al abrir la aplicación encontrarás la **pantalla principal**, donde se muestran dos secciones principales:

#### Manual

#### Ilustración 15 Pantalla de inicio



- Permite acceder a la guía digital paso a paso para el armado del robot LEGO EV3.
- Cada ficha o módulo se visualiza en 3D, permitiendo rotación, zoom y vista detallada.
- El usuario puede avanzar entre los pasos de construcción de manera intuitiva, con la posibilidad de retroceder en caso de errores.

## Código

- Acceso al editor de bloques de programación basado en **Scratch**.
- El usuario puede arrastrar y soltar bloques para crear instrucciones que controlan el robot.
- La programación puede ser simulada en tiempo real en el entorno 3D antes de transferirla al robot físico.

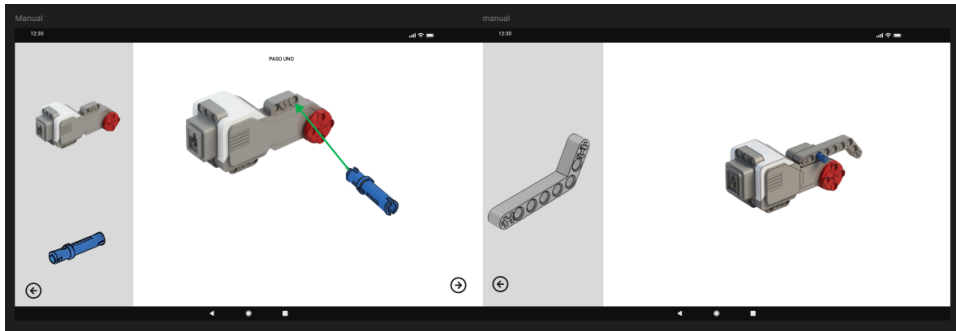
## Sección Manual

La sección del manual te guiará paso a paso en la construcción de tu robot LEGO EV3. Podrás seguir instrucciones visuales y escritas que te mostrarán cómo ensamblar cada pieza.

### Ilustración 16 Vista del ensamblaje paso a paso en 3D manual de usuario



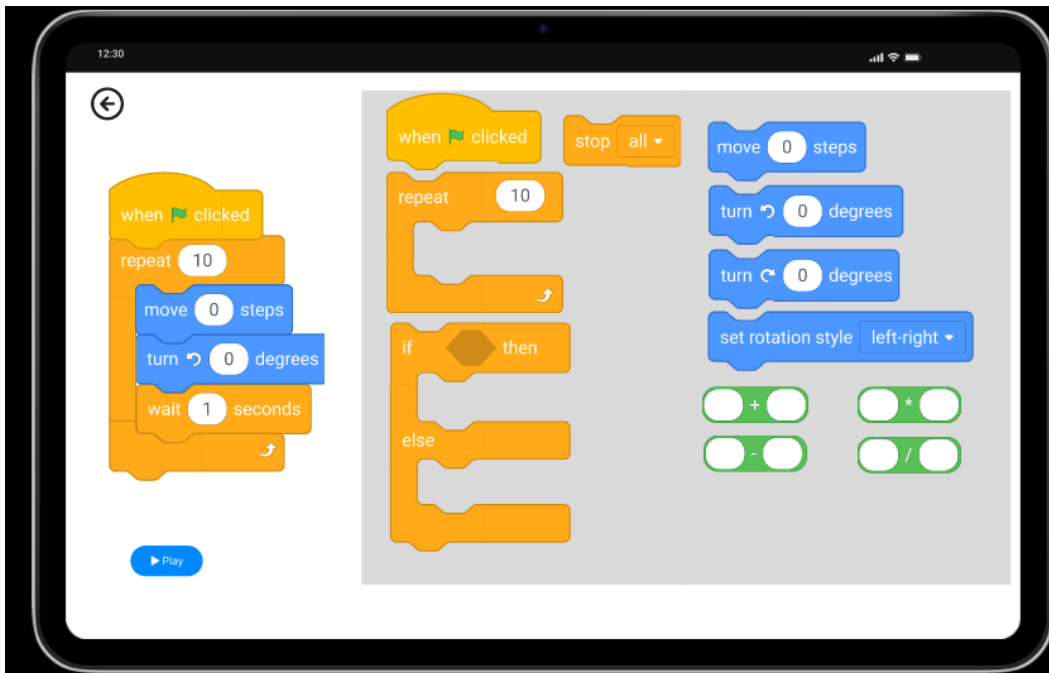
### Ilustración 17 Vista 3D con piezas adicionales del ensamblaje. Manual de usuario



- Guías de ensamblaje **interactivas en 3D** con animaciones y vistas dinámicas.
- Detección automática de errores de ensamblaje.
- Posibilidad de explorar el modelo del robot desde diferentes ángulos (rotación 360°, zoom, vista seccionada).
- Avance progresivo con barra de progreso que indica el estado de construcción.

### Sección Código

### Ilustración 18 Interfaz de programación por bloques Scratch. manual de usuario



- Editor visual basado en **Scratch**, organizado en categorías:
  - **Movimiento:** avanzar, girar, detenerse.
  - **Sensores:** ultrasonido, color, tacto.
  - **Control:** bucles y condicionales.
  - **Eventos y operadores.**
- Simulador integrado que traduce los bloques a acciones en el robot virtual.
- Ejecución paso a paso para depuración del código.
- Exportación del programa al robot físico mediante conexión Bluetooth.

### Beneficios de la Aplicación

- Sustituye las guías físicas por una herramienta digital interactiva.

- Permite **aprender robótica y programación** de forma práctica, accesible y motivadora.
- Promueve la autonomía del estudiante gracias al tutor virtual y a la retroalimentación automática.
- Compatible con dispositivos Android y con soporte offline.

## **Anexo 2. Guía de entrevista dirigida**

Proyecto: Desarrollo de una guía interactiva para el armado de robots LEGO EV3

**Participantes:** Estudiantes y docentes del semillero de investigación ESMA

**Técnica:** Entrevista dirigida

**Tipo de preguntas:** Abiertas

### ***Sección 1: Perfil del participante***

*(Esta sección busca conocer brevemente al entrevistado)*

¿Cuál es su rol dentro del semillero?

*(Estudiante, Docente, otro - especifique)*

¿Cuánto tiempo ha participado en los talleres que utilizan LEGO EV3?

¿Cómo describiría su nivel de experiencia con los robots LEGO EV3?

*(Puede indicar si considera que tiene un nivel básico, intermedio o avanzado y por qué)*

¿Ha utilizado guías para el armado de robots?

Si la respuesta es **sí**:

¿Qué tipo de guías ha utilizado? ¿Impresas, digitales (PDF, videos, web), o ambas?

¿Cuál ha sido su experiencia al utilizarlas?

## **Sección 2: Identificación de necesidades y dificultades**

*(Esta sección permite explorar los retos actuales en el uso de guías)*

¿Qué desafíos ha enfrentado al utilizar guías impresas para el armado de robots LEGO EV3?

¿Considera que las guías actuales explican adecuadamente el proceso de ensamblaje? ¿Por qué?

¿Qué dificultades ha observado en los estudiantes (o en usted mismo) al seguir instrucciones de programación con LEGO EV3?

En su opinión, ¿de qué manera una guía digital podría mejorar el proceso de aprendizaje y armado?

¿Qué elementos considera indispensables para que una guía digital sea útil y fácil de comprender?

*(Por ejemplo: imágenes, videos, pasos detallados, simulaciones, glosario, entre otros)*

## **Sección 3: Expectativas sobre la guía digital**

*(Esta sección explora las preferencias y expectativas frente al diseño de la guía digital)*

¿Le gustaría que la guía incluyera simulaciones en 3D del proceso de armado? ¿Por qué?

¿Qué nivel de detalle considera ideal en las explicaciones de la guía?  
*(Muy detallado paso a paso, detallado con pasos clave e imágenes, o una visión general)*

¿Por qué prefiere ese nivel de detalle?

¿Preferiría una guía con una estructura lineal (paso a paso) o interactiva (que permita explorar diferentes secciones según el interés)?

¿Qué ventajas le ve a esa opción?

¿Qué aspectos visuales considera importantes para que la guía sea atractiva y fácil de usar?

¿Qué funcionalidades adicionales le gustaría encontrar en la guía digital?  
(Por ejemplo: botón de ayuda, notas, glosario, retroalimentación automática, entre otros)

#### **Sección 4: Retroalimentación sobre el prototipo (si aplica)**

*(Solo incluir si se presenta un prototipo de la guía)*

¿Qué tan fácil le resultó usar la guía o prototipo que se le presentó?

¿Por qué?

¿Qué tan clara le pareció la navegación dentro del recurso?

¿Qué mejoras sugeriría en cuanto al diseño visual, los contenidos o la interacción con la guía?

#### **Observaciones adicionales**

¿Hay algún comentario o sugerencia adicional que le gustaría compartir respecto al uso de guías para el armado de robots LEGO EV3?