

COMPARACIÓN ENTRE ENSEÑANZA TRADICIONAL Y APRENDIZAJE ASISTIDO POR INTELIGENCIA ARTIFICIAL: IMPACTO EN EL DESEMPEÑO ACADÉMICO EN GEOMETRÍA

COMPARISON BETWEEN TRADITIONAL TEACHING AND AI-ASSISTED LEARNING: IMPACT ON ACADEMIC PERFORMANCE IN GEOMETRY

¹Janier Andrés Ballesteros Rincón

¹Universidad Americana de Europa (UNADE)

Cómo citar: Ballesteros Rincon, J. A. (2025). Comparación entre enseñanza tradicional y aprendizaje asistido por inteligencia artificial (IA): impacto en el desempeño académico en geometría. Publicaciones E Investigación, 20(1). <https://doi.org/10.22490/25394088.10331>

Recibido: 17 de septiembre 2025 y Aprobado 02 del noviembre de 2025

RESUMEN

GeometricAI es una aplicación móvil que apoya el aprendizaje de la geometría básica en estudiantes de educación secundaria. Esta herramienta integra un modelo de inteligencia artificial (IA) que reconoce automáticamente figuras geométricas y guía al usuario en la resolución de problemas relacionados con el área, el perímetro y otros parámetros esenciales, sin necesidad de conexión a internet para realizar ejercicios o validaciones.

El proyecto se desarrolla bajo un enfoque cuasi experimental con 60 estudiantes de la Institución Educativa Laureano Gómez, ubicada en San Agustín, Huila. Los participantes se dividen en dos grupos: el primero recibe enseñanza tradicional y el segundo utiliza GeometricAI como recurso principal para el aprendizaje.

La aplicación se desarrolló con Ionic y Angular, y emplea un modelo de IA en Python para ofrecer retroalimentación en tiempo real. Además, cuenta con un plan de actualización continua, basado en la experiencia de los usuarios y la incorporación de nuevas figuras geométricas con el fin de optimizar su precisión y funcionalidad.

Durante ocho semanas, el estudio aplica pruebas diagnósticas, encuestas de percepción y análisis estadísticos, como la prueba t de Student, con el objetivo de evaluar el impacto de la IA en el desempeño académico. Se proyecta una mejora significativa en la precisión, la autonomía y la motivación de los estudiantes que utilizan la aplicación.

Esta investigación aporta evidencia científica sobre los beneficios del uso de tecnologías emergentes en el aula y constituye una base para futuras decisiones de política educativa, especialmente en contextos rurales, promoviendo la equidad y la calidad en el aprendizaje geométrico.

Palabras clave: inteligencia artificial; aplicación móvil; geometría; Ionic; Angular; Capacitor.

¹ Ing.janierballesteros@gmail.com / <https://orcid.org/0009-0009-5266-5323>

ABSTRACT

GeometricAI is a mobile application that supports the learning of basic geometry for secondary school students. This tool integrates an artificial intelligence model that automatically recognizes geometric figures and guides the user in solving problems related to area, perimeter, and other essential parameters, without requiring an internet connection to complete exercises or perform validations.

The project is developed using a quasi-experimental approach with 60 students from the Laureano Gómez Educational Institution, located in San Agustín, Huila. Participants are divided into two groups: one receives traditional instruction and the other uses GeometricAI as the primary learning resource.

The application is developed with Ionic and Angular and uses a Python AI model to provide real-time feedback. It also has a continuous update plan based on user experience and the addition of new geometric figures, in order to optimize its accuracy and functionality.

For eight weeks, the study uses diagnostic tests, perception surveys, and statistical analyses such as the Student t-test to evaluate the impact of AI on academic performance. Significant improvements are projected in the accuracy, autonomy, and motivation of students using the application.

This research provides scientific evidence on the benefits of using emerging technologies in the classroom and constitutes a basis for future educational policy decisions, especially in rural contexts, promoting equity and quality in geometric learning.

Keywords: artificial intelligence; mobile app; geometry; Ionic; Angular; Capacitor.



1. INTRODUCCIÓN

El aprendizaje de la geometría en educación secundaria presenta múltiples desafíos, especialmente en la identificación de figuras geométricas, la comprensión de sus propiedades y la aplicación correcta de fórmulas matemáticas. A pesar de los esfuerzos de los docentes, las metodologías tradicionales no siempre logran captar el interés de los estudiantes ni facilitar un aprendizaje significativo. Por lo tanto, las herramientas tecnológicas han demostrado ser un apoyo valioso en la enseñanza de las matemáticas, permitiendo una interacción más dinámica y personalizada con los conceptos. Sin embargo, en muchas instituciones educativas, la integración de estas tecnologías sigue siendo

limitada, lo que motiva la exploración de enfoques innovadores, como la IA aplicada a la educación.

El objetivo principal de este proyecto es comparar la efectividad del aprendizaje tradicional con el aprendizaje asistido por una aplicación móvil Android con IA, evaluando el impacto en el desempeño académico en geometría en estudiantes de secundaria. Para ello, se desarrolla una aplicación en Ionic con Angular, integrada con un modelo de IA en Python, que permite a los estudiantes identificar figuras geométricas y resolver problemas matemáticos de manera autónoma. A través de pruebas diagnósticas, análisis estadísticos

y encuestas de percepción, se medirá el efecto de la tecnología en la comprensión y el rendimiento académico. Los hallazgos de esta investigación contribuyen a la optimización de estrategias pedagógicas y a la implementación de soluciones tecnológicas en el ámbito educativo.

2. MATERIALES Y MÉTODO

La investigación se desarrolla mediante un enfoque cuasi experimental que evalúa los conocimientos de geometría básica, comparando los resultados de aprendizaje de dos grupos de estudiantes:

Grupo de control: recibe enseñanza tradicional basada en la explicación del docente.

Grupo experimental: utiliza la aplicación móvil Android con IA para el reconocimiento y resolución de problemas geométricos.

Las fases metodológicas incluyen:

2.1 Desarrollo de la aplicación móvil

La aplicación móvil Android se desarrolla en Ionic con Angular, integrando un modelo de IA en Python para la detección de figuras geométricas.

2.2 Selección de la muestra

Para garantizar la validez y representatividad de los resultados, se trabaja con una muestra de aproximadamente 60 a 80 estudiantes de educación secundaria de la Institución Educativa Laureano Gómez, ubicada en el municipio de San Agustín, Huila, Colombia. La selección de esta cantidad de participantes responde a criterios metodológicos sustentados en estudios previos que han analizado el impacto de la tecnología en la educación matemática, asegurando que con este tamaño de muestra se pueden detectar diferencias significativas en el aprendizaje.

La determinación del número de participantes se basa en investigaciones como la de Trías Seferian et al. (2024), quienes han demostrado que muestras de entre 50 y 100 estudiantes han sido adecuadas para evaluar intervenciones educativas relacionadas con la incorporación de herramientas tecnológicas en el aula. Además, estudios como los de Ruiz Muñoz (2024) indican que en experimentos educativos con dos grupos comparativos, un mínimo de 30 estudiantes por grupo, son suficientes para realizar análisis estadísticos robustos, como pruebas t de Student o ANOVA, siempre que los grupos sean homogéneos en términos de conocimientos previos y exposición a los contenidos curriculares según la tabla 1.

La Institución Educativa Laureano Gómez cuenta con dos grupos de noveno (9A y 9B) y dos grupos de décimo (10A y 10B). Esta distribución permite que los estudiantes seleccionados tengan características académicas similares, asegurando que la comparación entre grupos sea válida y que las diferencias en el aprendizaje puedan atribuirse al uso de la aplicación GeometricAI y no a otros factores externos. Como afirman Machado Pico et al. (2024), la selección de muestras homogéneas es fundamental en estudios de intervención tecnológica en educación, ya que permite un análisis preciso del impacto de la herramienta.

Para el estudio comparativo, la muestra se divide en dos grupos experimentales con el objetivo de evaluar la efectividad del aprendizaje asistido por IA frente al método tradicional:

Grupo A (aprendizaje tradicional): conformado por entre 30 y 40 estudiantes, quienes reciben enseñanza de geometría mediante una metodología convencional, con clases dirigidas por un docente.

Grupo B (aprendizaje asistido por IA): incluirá entre 30 y 40 estudiantes, quienes utilizarán la aplicación GeometricAI como herramienta principal para la identificación y resolución de ejercicios de geometría básica.

El diseño experimental sigue la metodología recomendada por Parra Pulido (2024), quien argumenta que los estudios comparativos en educación deben incluir grupos de control y experimentales bien equilibrados, para garantizar que las diferencias en los resultados sean atribuibles a la variable independiente, es decir, al uso de la IA en la enseñanza de la geometría.

Tabla 1. Tipos de variables generales del proyecto

Variable	Tipo	Descripción	Instrumento de medición
Desempeño en geometría	Dependiente	Nivel de precisión en la identificación y resolución de problemas geométricos.	Test de evaluación antes y después del experimento.
Motivación del estudiante	Dependiente	Grado de interés y participación en el aprendizaje de geometría.	Encuesta de percepción aplicada antes y después del uso de la aplicación.
Uso de la IA	Independiente	Frecuencia y forma en que los estudiantes utilizan la aplicación GeometricAI.	Registro de uso en la aplicación (interacciones, tiempo de uso, módulos completados).
Método de enseñanza	Independiente	Estrategia utilizada en la enseñanza de geometría (tradicional vs. asistida por IA).	Asignación de grupos experimentales y control.
Percepción del docente	Dependiente	Opinión de los docentes sobre el impacto de la IA en la enseñanza.	Entrevistas y encuestas a docentes después de la implementación.

Fuente: Elaboración propia.

2.3 Implementación y duración del estudio

El estudio tiene una duración de ocho semanas, distribuidas así:

Semana 1: aplicación de la prueba diagnóstica inicial.

Semanas 2 al 7: desarrollo de sesiones de aprendizaje con cada metodología.

Semana 8: aplicación de la prueba final y encuestas de percepción.

Para minimizar la resistencia al cambio por parte de los estudiantes o docentes, se llevan a cabo sesiones de inducción previas al inicio del estudio (una semana), explicando los beneficios y objetivos de la

investigación. Además, se cuenta con asistencia técnica y pedagógica para la aplicación móvil, asegurando que los participantes comprendan su uso y beneficios.

2.4 Criterios de evaluación e indicadores

Se emplean diferentes instrumentos de medición para evaluar la efectividad del aprendizaje en cada grupo.

2.5 Análisis de datos

En la tabla 2 se evidencian las diferentes variables, instrumentos y métodos de análisis que se implementaron en el desarrollo del proyecto.

Tabla 2. Variables, instrumentos y métodos de análisis

Variable	Tipo	Descripción	Instrumento de medición	Método de análisis
Desempeño en geometría	Dependiente	Nivel de precisión en ejercicios de geometría	Test de evaluación antes y después del experimento	Prueba t de Student/ANOVA
Motivación del estudiante	Dependiente	Grado de interés y participación en la clase	Encuesta de percepción	Análisis de contenido (cualitativo)
Uso de la IA	Independiente	Frecuencia y forma en que los estudiantes utilizan la aplicación	Registro de uso en GeometricAI	Correlación de Pearson

Fuente: Elaboración propia.

2.6 Resultados esperados

Se espera que los estudiantes del grupo experimental (aquellos que utilicen la aplicación móvil con IA) muestren una mejora en resolución de problemas de geometría básica, identifiquen más rápido y con precisión las diferentes figuras y apliquen correctamente técnicas de cálculo de lado, perímetro y área. Estas mejoras se evalúan en comparación con el grupo de control (que sigue el método tradicional), que se reflejan en los siguientes aspectos:

- Incremento en los puntajes de las pruebas académicas
- Mayor tasa de aciertos en la resolución de problemas geométricos
- Reducción en el tiempo de resolución de ejercicios

- Mayor autonomía y motivación en el aprendizaje
- Identificación de patrones en la experiencia de aprendizaje

3. DESARROLLO

La aplicación móvil se desarrolló en el entorno de Visual Studio Code, utilizando el *framework* Ionic en conjunto con Angular para el diseño de la interfaz gráfica de usuario (GUI). Este enfoque permitió implementar una arquitectura modular y responsiva, adecuada para dispositivos móviles. Asimismo, se integraron librerías y dependencias de Capacitor, lo que facilitó el acceso a funcionalidades nativas del dispositivo, como la cámara, el almacenamiento y la conectividad. En la figura 1 se visualiza el logotipo utilizado para representar la aplicación móvil GeometricAI.

Figura 1. Logotipo de aplicación móvil GeometricAI.



Fuente: Elaboración propia.

3.1 Fase 1: estructura general del proyecto

La fase inicial del desarrollo de la aplicación móvil educativa basada en IA GeometricAI consistió en la definición y construcción de la estructura general. Esta etapa es importante en el ciclo de vida del *software*, puesto que establece los cimientos en los que se diseñan e implementan las funcionalidades, se integran los modelos de IA y se despliega la aplicación en dispositivos móviles. A través de la selección de tecnologías y *frameworks*, se buscó garantizar tanto la escalabilidad como la compatibilidad multiplataforma de la solución.

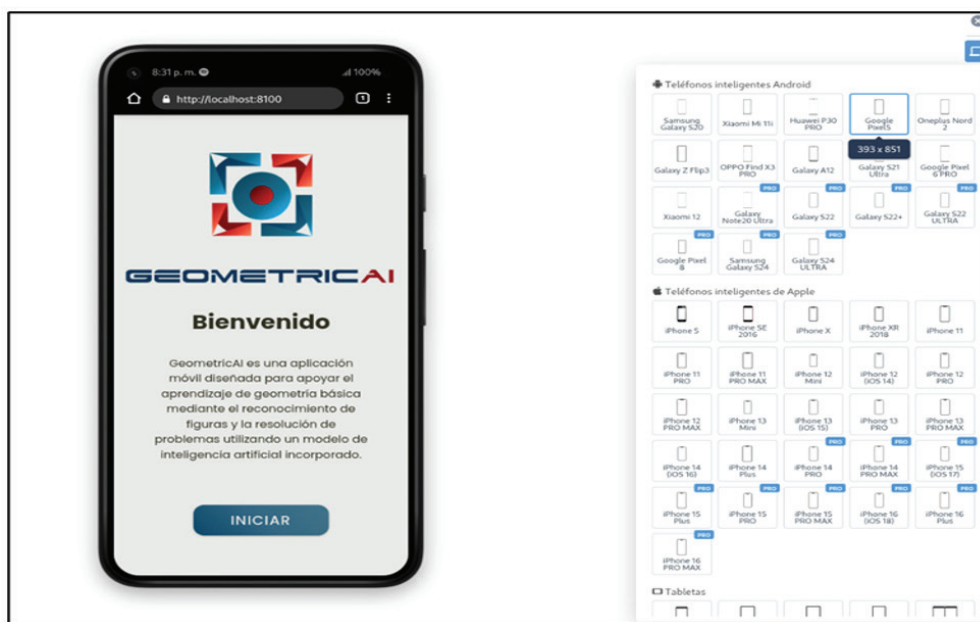
3.2 Frameworks y tecnologías seleccionadas

Para este proyecto se optó por un enfoque híbrido, utilizando herramientas que desarrollan una sola base

de código que se despliega en múltiples plataformas, principalmente Android. En este contexto, se emplearon las siguientes tecnologías principales:

Ionic Framework: Tiene código abierto y ampliamente utilizado para el desarrollo de interfaces móviles híbridas mediante tecnologías web estándar, como HTML, CSS y JavaScript/TypeScript. La fortaleza radica en la capacidad de crear aplicaciones con apariencia nativa, mediante el uso de componentes de UI (Figura 2) predefinidos y adaptativos. En GeometricAI, Ionic proporciona la base visual y estructural de la interfaz, lo que facilita una experiencia de usuario coherente e intuitiva en distintos dispositivos móviles.

Figura 2. Vista inicial de aplicación móvil con emulador en tamaños de pantallas distintos.



Fuente: Elaboración propia.

Angular: se eligió Angular para el desarrollo de la lógica de negocio del proyecto, dada la arquitectura basada en componentes, un sistema robusto de enlace de datos (*data binding*), la inyección de dependencias y la capacidad para estructurar la aplicación de

manera modular. Angular permitió implementar de manera organizada los servicios de comunicación con el modelo de IA, los flujos de navegación entre vistas, la gestión del estado de la aplicación y el control de formularios y validaciones.

Capacitor: es la herramienta de compilación e integración nativa desarrollada por el equipo de Ionic, que reemplaza a Cordova. Esta tecnología actúa como un puente entre el entorno web (Angular e Ionic) y las capacidades nativas del dispositivo (por ejemplo, el acceso al almacenamiento, sensores, cámara o red). En este proyecto, Capacitor fue fundamental para empaquetar la aplicación web en un contenedor nativo, permitiendo la ejecución en Android y facilitando la integración futura de *plugins* específicos para la cámara, redes neuronales embebidas y reconocimiento de imágenes. Permitió, además, efectuar pruebas en emuladores y dispositivos físicos, lo que optimizó el rendimiento y la usabilidad de la aplicación.

Una vez culminada la fase de desarrollo, el proyecto se migró a Android Studio mediante la herramienta Capacitor, para compilar la aplicación y generar el archivo instalador (.apk) compatible con dispositivos Android. En esta etapa, se implementó el control de versiones mediante identificadores incrementales y nombres diferenciados para cada versión, junto con la firma digital de la aplicación, requisito indispensable

para la publicación en Google Play Console. Con este proceso también se aseguró el cumplimiento de los lineamientos exigidos por la plataforma, tales como firma criptográfica de la aplicación, declaración de permisos para el acceso a sensores y recursos del dispositivo móvil y la incorporación de los términos y condiciones de uso de la aplicación GeometricAI. Con ello se garantiza transparencia, seguridad y conformidad legal respecto al tratamiento de datos y funcionalidades ofrecidas.

3.2.1 Organización de los archivos clave del proyecto

Una vez inicializado el proyecto con los comandos “ionic start” e “ionic build”, se creó una estructura estándar de carpetas y archivos que configuraran el entorno de desarrollo. A continuación, se describen los principales archivos y carpetas, explicando el propósito y la relevancia en el proyecto como se evidencia en la tabla 3 archivos utilizados en la implementación de conexiones dentro de la aplicación móvil GeometricAI.

Tabla 3. Archivos y elementos de implementación en la aplicación móvil GeometricAI.

Archivo/carpeteta	Función principal
package.json	Contiene la metadata del proyecto: nombre, versión, scripts, dependencias y scripts de ejecución.
angular.json	Define la estructura del proyecto Angular, especifica las rutas, assets, estilos y configuración de build.
ionic.config.json	Archivo específico de Ionic, que almacena configuraciones como el nombre del proyecto y su tipo (capacitor).
capacitor.config.ts	Archivo central de configuración para Capacitor, define el ID del paquete, nombre de la app y plataforma base.
node_modules/	Carpeta que contiene todas las dependencias descargadas vía npm.
android/	Directorios generados por Capacitor, que contienen el código nativo para la plataforma móvil Android.

Fuente: Elaboración propia.

A nivel funcional, package.json es esencial para la gestión de dependencias mediante npm (*Node Package Manager*). Aquí se declaran todas las bibliotecas necesarias tanto para el desarrollo como para la ejecución, incluyendo librerías como @ionic/angular

y @capacitor/core. También se definen los comandos de inicio (npm start), compilación (ionic build), y pruebas unitarias (ng test), lo que permite una automatización sencilla del ciclo de vida del desarrollo y pruebas desde la terminal de Visual Studio Code.

Por otro lado, `angular.json` define las configuraciones necesarias para compilar la aplicación Angular, que va desde la inclusión de estilos globales hasta la configuración de entornos. Este archivo determina la carpeta de salida que utiliza Capacitor al generar la versión nativa.

`Capacitor.config.ts` es clave en la integración entre el *frontend* y la capa nativa, pues define elementos como el nombre de la aplicación (*appName*), el identificador del paquete (*appId*) y la ruta del directorio web (*webDir*) con los archivos estáticos generados por Angular.

Finalmente, las carpetas `android/` e `ios/` son generadas mediante los comandos `npx cap add android` o `npx cap add ios`. Estas carpetas permiten abrir el proyecto directamente en Android Studio o Xcode, para ser exportado como una aplicación instalable. En el

contexto de este proyecto se priorizó la plataforma Android, debido a la amplia adopción en instituciones educativas colombianas.

3.2.2 Importancia estratégica de esta fase

Establecer una estructura general sólida permite organizar de forma coherente todos los elementos del desarrollo. Además, facilita la escalabilidad, el mantenimiento del código y la integración de tecnologías avanzadas como la IA. Adicionalmente, se adoptan estándares industriales como Angular e Ionic, que garantizan la reutilización de componentes y la integración con bibliotecas especializadas (Figura 3), para reconocimiento de imágenes, gráficos vectoriales y conectividad con modelos de IA desarrollados en Python.

Figura 3. Estructura jerárquica de la aplicación móvil GeometricAI.



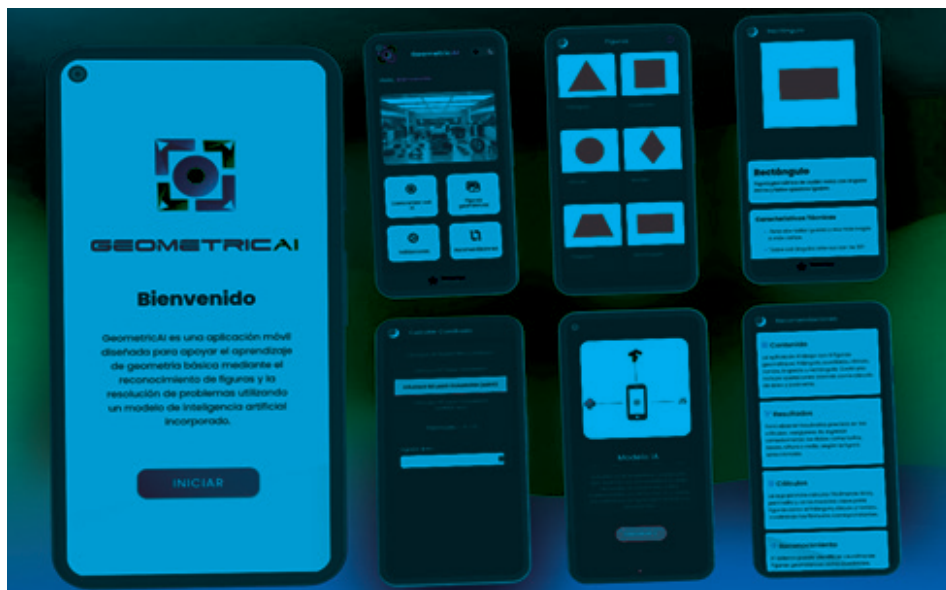
Fuente: Elaboración propia.

3.3 Fase 2: diseño e implementación de la interfaz de usuario y navegación

La segunda fase del desarrollo de la aplicación GeometricAI (Figura 4) se centró en la creación de una interfaz de usuario (UI) intuitiva, accesible y coherente con los principios pedagógicos asociados al uso

de tecnologías emergentes en entornos escolares. Esta etapa fue esencial para garantizar que los estudiantes de básica secundaria pudieran interactuar fácilmente con la aplicación, reconocer figuras geométricas, seleccionar opciones de cálculo y comprender los resultados proporcionados por el sistema de IA.

Figura 4. Imagen de funciones de las pantallas de aplicación móvil GeometricAI



Fuente: Elaboración propia.

3.3.1 Principios de diseño y experiencia de usuario

Entre los principios clave adoptados en el diseño de la interfaz destacan:

Simplicidad visual: Se evitaron elementos distractores, priorizando el uso de íconos reconocibles, botones grandes y colores contrastantes.

Navegación: Todas las vistas comparten una estructura base, con encabezado fijo, botones de navegación lateral y enfoque visual inmediato.

Lenguaje accesible: Se utilizó un lenguaje claro en idioma español y acompañado de ayudas visuales para la mejor comprensión de estudiantes de distintos niveles.

Compatibilidad responsiva: Aunque la *app* está diseñada para dispositivos móviles Android, el diseño adaptativo permite su correcta visualización en distintos tamaños de pantalla.

3.3.2 Estructura de vistas y navegación

La aplicación se estructuró en módulos funcionales independientes que se integran a través del enrutador

de Angular. En la tabla 4 se aprecian las principales vistas.

La navegación se implementó con Angular, lo que permitió transiciones fluidas entre las pantallas sin recargar la aplicación. Además, se integraron animaciones ligeras de Ionic para mejorar la experiencia de usuario.

3.3.3 Componentes personalizados y reusabilidad

Durante esta fase se definieron componentes personalizados reutilizados de manera transversal en la aplicación, como:

- **Boton Accion Component:** Botón grande y estilizado para funciones clave.
- **Tarjeta Resultado Component:** Tarjeta visual con resumen de resultados calculados por IA.
- **Encabezado Component:** Encabezado común con logotipo y nombre de la aplicación.

El enfoque modular facilitó el mantenimiento del código y aseguró una identidad visual uniforme.

Tabla 4. Estructura de vistas y navegación de aplicación móvil GeometricAI.

Vista	Función principal	Componentes clave
home.page.ts/html	Pantalla de inicio, bienvenida al usuario	Botón “Iniciar”, breve descripción, logo de GeometricAI
menu.page.ts/html	Menú principal con acceso a opciones	Lista de opciones: Reconocer la figura, Fichas de figuras geométricas, Validación, Recomendaciones.
figura.page.ts/html	Captura o selección de figura geométrica	Acceso a la cámara
resultado.page.ts/html	Presentación de resultados del análisis	Imagen de la figura detectada, tipo, propiedades y botón calcular para ingresar a calculadora predefinida para la figura geométrica.

Fuente: Elaboración propia.

3.3.4 Consideraciones pedagógicas

Dado el enfoque educativo, la interfaz se diseñó considerando las necesidades cognitivas y tecnológicas del público objetivo. Se incorporaron ayudas contextuales y mensajes explicativos que guían al usuario, permitiéndole realizar las tareas de forma autónoma. La aplicación reconoce figuras geométricas y redirige al estudiante a una sección específica para cálculos de área, perímetro y otras propiedades. También se muestra la fórmula respectiva y el usuario ingresa valores medidos en clase para obtener el resultado de manera inmediata. Esta funcionalidad hace de GeometricAI una herramienta de apoyo para el aprendizaje activo y significativo a través de tecnologías emergentes como la IA.

3.4 Fase 3: integración del modelo de inteligencia artificial y reconocimiento de figuras

En esta fase se incorporó el modelo de IA entrenado en Python, especializado en el reconocimiento de figuras geométricas comunes en la educación básica secundaria. Su integración se realizó mediante la conversión a un formato compatible (.json) y su posterior incorporación en el entorno híbrido de Capacitor e Ionic Angular.

Tiempo de inferencia y despliegue del modelo: Durante las pruebas en un dispositivo móvil de gama media (Samsung Galaxy A32, 4 GB de RAM, Android 12), el tiempo promedio de inferencia fue de 140 milisegundos por imagen, ejecutando el modelo exportado en formato .json (estructura) y .bin (pesos) directamente en la aplicación. Este tiempo garantiza una experiencia fluida y respuestas casi en tiempo real para los estudiantes.

El modelo se entrenó con TensorFlow y exportado a formato compatible con TensorFlow.js. Aunque no se empleó cuantización posentrenamiento, se contempla en futuras versiones migrar a formato .tflite cuantizado con el fin de:

- Reducir el tamaño de modelo (de ~10 MB a menos de 3 MB)
- Mejorar la compatibilidad con dispositivos de baja gama (RAM ≤ 2 GB)
- Aumentar la eficiencia energética durante la inferencia

Preprocesamiento de imágenes en GeometricAI: La ruta de preprocesamiento de imágenes se apoya en

funciones implementadas en el *frontend*, usando capacidades de JavaScript y TensorFlow.js con Capacitor. A continuación, se presentan los siguientes flujos:

- Captura por plugin Camera
- Redimensionamiento a 150×150 píxeles RGB vía elemento <canvas>
- Conversión a tensor en TensorFlow.js (normalización a [0, 1])
- Expansión de dimensión simulando batch [1, 150, 150, 3]

- Inferencia en el navegador con los pesos .bin y la arquitectura .json

El modelo se ejecuta sobre CPU; se planea activar WebGL para aprovechar GPU en navegadores compatibles, lo que reduciría aún más el tiempo de inferencia.

En la práctica, el modelo procesa la imagen, reconoce la figura geométrica y devuelve la clase asignada y un score de predicción para ello se utilizaron elementos y dependencias descritas en la tabla 5. Esta salida redirige al usuario hacia la sección de cálculo correspondiente y fortalece las competencias matemáticas mediante IA.

Tabla 5. Elementos y dependencia utilizadas para creación del modelo de Inteligencia artificial.

Capa	Tipo	Detalles
Conv2D	Convolucional	32 filtros, tamaño 3x3, activación ReLU, entrada (150x150x3)
MaxPooling2D	Pooling	Reducción de dimensiones 2x2
Conv2D	Convolucional	64 filtros, tamaño 3x3, activación ReLU
MaxPooling2D	Pooling	Reducción de dimensiones 2x2
Conv2D	Convolucional	128 filtros, tamaño 3x3, activación ReLU
MaxPooling2D	Pooling	Reducción de dimensiones 2x2
Flatten	Aplanamiento	Pasa de 3D a 1D para Dense
Dense	Capa totalmente conectada	128 unidades, activación ReLU
Dropout	Regularización	30% de unidades apagadas aleatoriamente para evitar sobreajuste
Dense (salida)	Clasificación final	n unidades (1 por clase), activación softmax para clasificación multiclase

Fuente: Elaboración propia.

Dataset y preprocesamiento: Se empleó un *dataset* con 12 060 imágenes, distribuidas así:

- 70 % (9648 imágenes) para entrenamiento
- 15 % (1818 imágenes) para validación
- 15 % (1818 imágenes) para prueba

Las imágenes se redimensionaron a 150×150 píxeles RGB y se normalizaron a [0, 1] (pixel/255). El preprocesamiento se realizó mediante ImageDataGenerator de Keras y no se aplicó aumento de datos, ya que la distribución era adecuada.

Arquitectura del modelo: El modelo consta de las siguientes capas:

- Conv2D (32 filtros, kernel 3×3, activación ReLU)
- MaxPooling2D (*pool size* 2×2)
- Conv2D (64 filtros, kernel 3×3, ReLU)
- MaxPooling2D
- Dropout (*rate* 0.25)
- Flatten
- Dense (128 unidades, ReLU)
- Dropout (*rate* 0.5)

- Dense final (6 unidades, *softmax*)
- Optimizador: Adam
- Tasa de aprendizaje: 0.001
- Función de pérdida: *sparse_categorical_crossentropy*
- Métrica principal: *accuracy*

4.DISCUSIÓN

4.1 Entrenamiento y evaluación

El proceso de entrenamiento del modelo se realizó durante un total de nueve épocas utilizando una política de detención temprana (*early stopping*). Esta técnica permitió restaurar los pesos del mejor modelo obtenido, correspondiente a la época

cuatro.

Estos fueron los principales resultados obtenidos:

- Precisión global de entrenamiento: 99.95 %
- Precisión de validación: 100 %
- Pérdida de validación en la mejor época (época cuatro): 0.000076
- Tiempo por época: entre dos y cinco minutos en GPU
- Gráfico de convergencia (*accuracy* y *loss* por época).

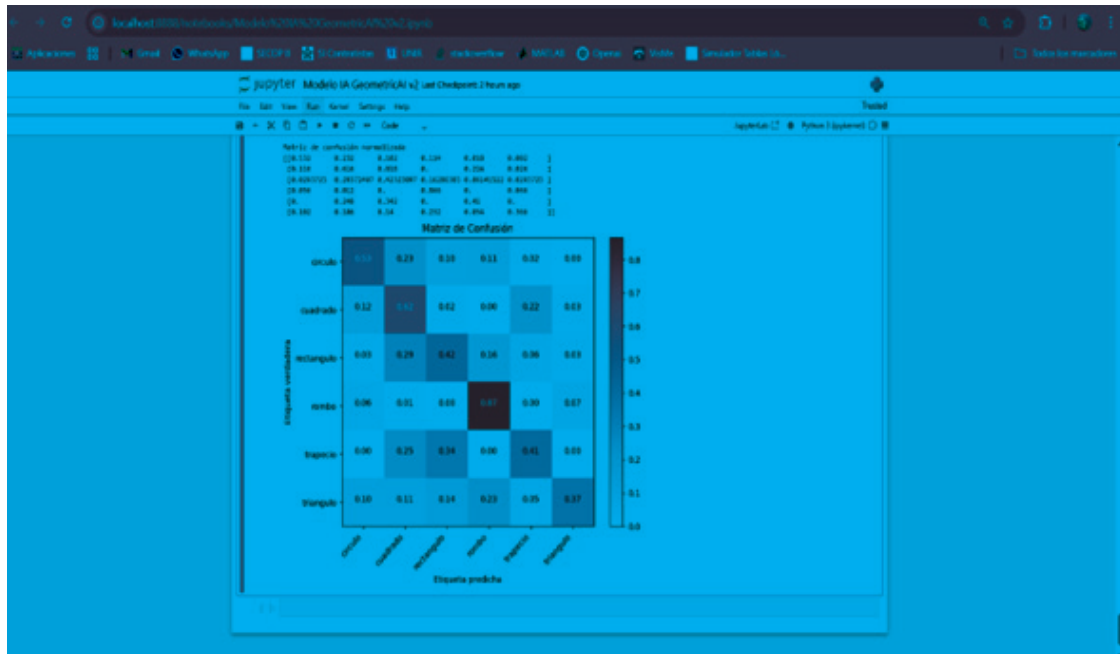
La ausencia de sobreajuste se verificó mediante la gráfica (Figura 5), donde se evidencian la pérdida y precisión por época, lo cual se controló gracias al uso de detención temprana, adicionalmente se puede visualizar la matriz de confusión normalizada (Figura 6), Curva ROC y AUC para clasificación multiclase (Figura 7).

Figura 5. Pérdida y precisión del modelo de inteligencia artificial por cada época.



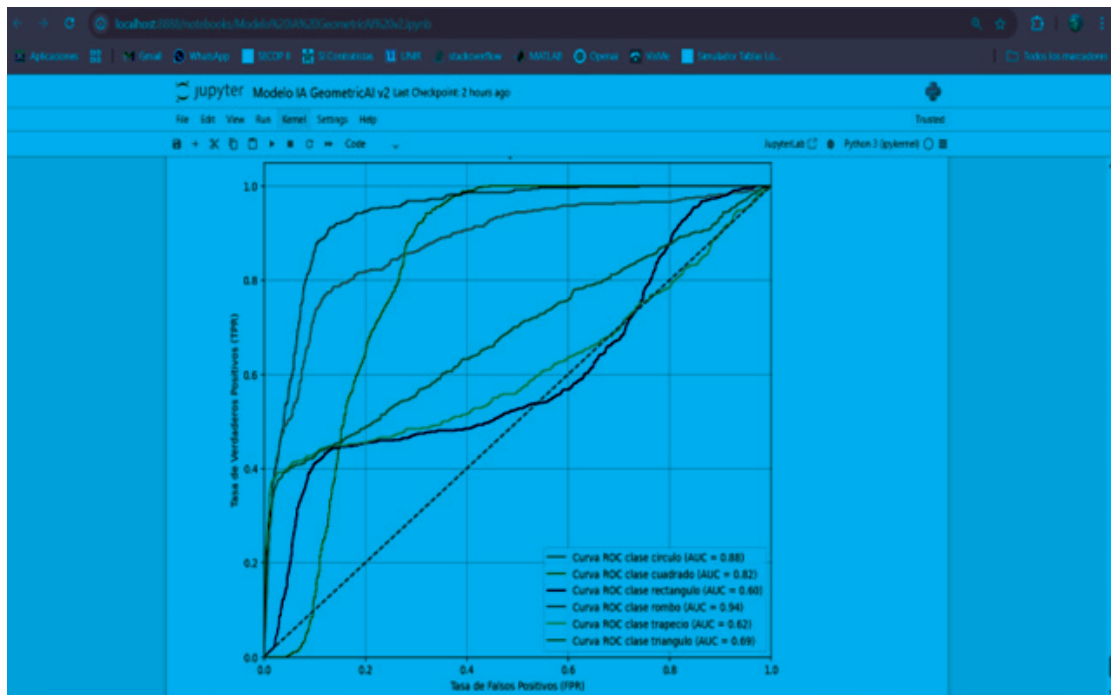
Fuente: Elaboración propia.

Figura 6. Matriz de confusión normalizada



Fuente: Elaboración propia.

Figura 7. Curva ROC y AUC para clasificación multiclase

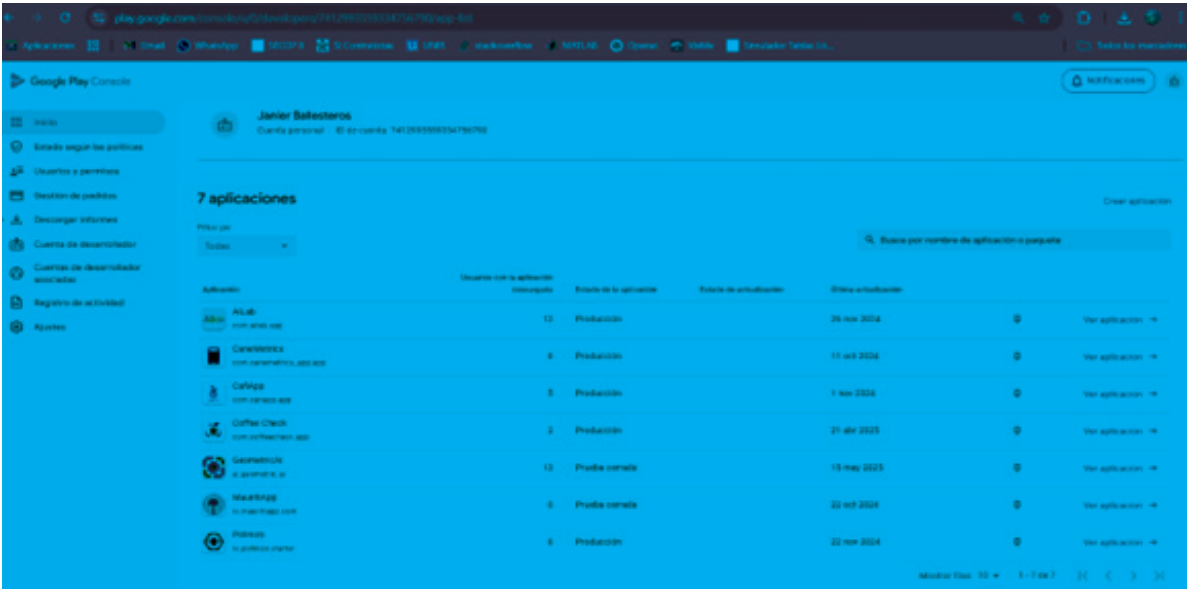


Fuente: Elaboración propia.

Esta fase abordó la puesta en operatividad de la aplicación GeometricAI, enfocándose en el empaquetado (Figura 8), firma digital y posterior lanzamiento en la plataforma oficial de distribución de aplicaciones Android: Google Play Store, mediante el uso de Google Play Console.

4.2 **Empaquetado de la aplicación**

Figura 8. **FALTA EL NOMBRE DE LA FIGURA**



Fuente: Elaboración propia.

Para generar el archivo instalable (.apk o .aab), se utilizó el entorno de Ionic Angular con Capacitor, permitiendo compilar la aplicación híbrida con acceso a funcionalidades nativas.

4.3 **Firma digital y configuración en Google Play Console**

Para que la aplicación se publicara en Google Play Store, fue necesario crear una clave de firma digital, requisito indispensable para validar la autenticidad del desarrollador. Dentro de Google Play Console (Figura 9), se llevó a cabo la siguiente configuración:

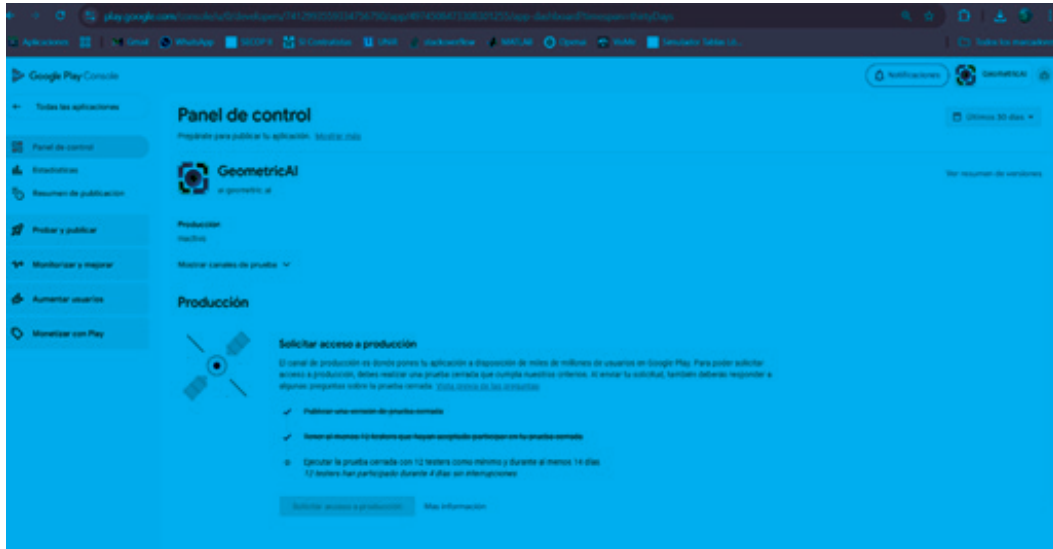
- Creación de un nuevo proyecto en <https://play.google.com/console>
- Configuración del idioma, nombre de la aplicación, tipo de uso (educativo) y categoría (Educación > Herramientas de aprendizaje).

5. **CONCLUSIONES**

El presente estudio permitió identificar que el uso de la aplicación móvil GeometricAI, basada en un modelo de IA para la identificación y resolución de problemas de geometría básica, tuvo un impacto altamente positivo en el desempeño académico de los estudiantes del grado décimo de la Institución Educativa Laureano Gómez.

Los resultados reflejan que el grupo que utilizó la aplicación alcanzó un promedio de acierto del 93 %, mientras que el grupo que trabajó únicamente con el método tradicional obtuvo una media de 81.5 %. Esto muestra una mejora significativa en la comprensión de los conceptos geométricos y la resolución de ejercicios.

Figura 9. Configuración de aplicación móvil GeometricAI en Google Play Console



Fuente: Elaboración propia.

La aplicación favoreció la precisión en los cálculos de áreas, perímetros y propiedades geométricas, y permitió a los estudiantes obtener resultados rápidos e instantáneos, y sin necesidad de conexión a internet. Esto causó un ambiente de satisfacción y confianza, donde los alumnos reconocieron la utilidad de la herramienta para reforzar sus aprendizajes de forma autónoma y práctica.

La percepción de estudiantes, docentes y directivos fue altamente positiva, ya que consideraron que herramientas como GeometricAI representan un gran potencial para fortalecer los procesos de enseñanza-aprendizaje. La facilidad de uso, la precisión en los resultados y la capacidad de apoyo en tiempo real sitúan a la aplicación como un recurso innovador en la evolución de la educación.

Se demuestra que la incorporación de tecnologías basadas en IA puede convertirse en un aliado fundamental para mejorar los indicadores académicos, estimular el aprendizaje autónomo y transformar la

interacción de los estudiantes con la geometría y otras áreas del conocimiento.

REFERENCIAS

- Machado Pico, H. A., Sagnay Alvarado, E., Vera Pisco, D. G., y Sornoza Parrales, D. (2024). Integración Efectiva De Aplicaciones Informáticas Para Potenciar El Aprendizaje De Las Matemáticas En Bachillerato. *Sinapsis: La revista científica del ITSUP*, 25(2), 16.
- Parra Pulido, J. H. (2024). Propiedades psicométricas y estandarización de la subprueba de cálculo matemático del WRAT-4 en adolescentes colombianos. *Acta Colombiana de Psicología*, 27(2), 112-129. <https://doi.org/10.14718/ACP.2024.27.2.7>
- Ruiz Muñoz, G. F. (2024). Enseñanza híbrida y transformación digital en la educación: Integración de tecnología y metodología. *Revista de Investigación en Tecnologías de la Información*, 12(25), 48-55. <https://doi.org/10.36825/RITI.12.25.005>
- Trías Seferian, D., Sastre Abreu, H., y Cuadros-Jiménez, O. E. (2024). Motivación y autorregulación en el desempeño en matemáticas en estudiantes de Educación Secundaria. *Revista Colombiana de Educación*, 92, 209-232. <https://doi.org/10.17227/rce.num92-17121>