



Del prototipado a la construcción: Transformando la enseñanza de Ingeniería de Software con IA y NoCode

Mariela Isabel Camargo Roman, PhD

Docente TIC de la Universidad ESAN

Perú

Premio Internacional Metared TIC 2025

Abril 2025

Del prototipado a la construcción: Transformando la enseñanza de Ingeniería de Software con IA y NoCode

Resumen

Este artículo presenta una experiencia innovadora en la enseñanza de Ingeniería de Software, donde se integró la Inteligencia Artificial (IA) y herramientas NoCode (específicamente Flutter Flow) en un curso introductorio dirigido a estudiantes de Ingeniería de TI y Sistemas. El principal desafío educativo abordado fue la limitación tradicional de alcanzar únicamente la fase de prototipado en proyectos estudiantiles. La metodología implementada incorpora IA en las fases de requisitos y diseño, y utiliza Flutter Flow como plataforma NoCode para la implementación. Los resultados muestran una transformación significativa: los estudiantes ahora completan aplicaciones funcionales integrales, en contraste con interfaces con simples prototipos desarrollados anteriormente. Esta aproximación pedagógica fortalece la comprensión holística del ciclo de desarrollo de software, reduce la brecha entre teoría y práctica, y proporciona experiencia con tecnologías emergentes que están remodelando la industria. Este enfoque no solo optimiza el tiempo de aprendizaje, sino que también mejora significativamente la calidad y alcance de los proyectos estudiantiles.

Introducción

La enseñanza de Ingeniería de Software enfrenta continuamente el desafío de proporcionar experiencias prácticas significativas dentro de las limitaciones temporales de los cursos académicos. En particular, los cursos introductorios suelen presentar dificultades para que los estudiantes experimenten el ciclo completo de desarrollo de software, desde requisitos hasta implementación y pruebas. Esta restricción ha significado históricamente que los estudiantes solo alcanzan a desarrollar prototipos en lugar de aplicaciones completas, limitando su comprensión integral del proceso.

Con la emergencia y maduración de tecnologías como la Inteligencia Artificial generativa y las plataformas NoCode, surge una oportunidad transformadora para repensar la pedagogía en la formación de ingenieros de TI y sistemas. Estas herramientas tienen el potencial de acelerar significativamente aspectos técnicos del desarrollo, permitiendo a los estudiantes enfocarse en el proceso holístico y los principios fundamentales del diseño de software.

Este artículo presenta la experiencia de implementación y los resultados obtenidos al integrar sistemáticamente IA y herramientas NoCode en un curso de Introducción a la Ingeniería de Software para estudiantes de cuarto ciclo de la carrera de Ingeniería de TI y Sistemas. El objetivo principal fue determinar si estas tecnologías permitirían a los estudiantes superar la barrera del prototipado y completar aplicaciones funcionales dentro del mismo período académico.

La hipótesis central que motivó esta investigación fue que la integración estratégica de IA y herramientas NoCode permitiría a los estudiantes experimentar el ciclo completo de desarrollo de software, mejorando significativamente su comprensión de la disciplina y preparándolos mejor para los siguientes cursos, haciéndolos más hábiles y capaces a enfrentarse a los desafíos de la industria actual.

Metodología de Trabajo

La metodología implementada se estructuró en tres fases principales, correspondientes a las etapas fundamentales del desarrollo de software, cada una potenciada por tecnologías específicas:

Enfoque Pedagógico

El enfoque pedagógico adoptado se basa en el principio de "aprender haciendo" (learning by doing), pero con un componente tecnológico potenciador. Tradicionalmente, las limitaciones técnicas y conocimientos respecto a la programación restringían el alcance de lo que los estudiantes podían "hacer" dentro del curso. La integración de IA y NoCode buscó expandir este horizonte.

Se estableció un proyecto continuo a lo largo del semestre donde los estudiantes, organizados en equipos de 4-5 integrantes, debían desarrollar una aplicación móvil basada en requisitos reales proporcionados por stakeholders simulados.

El curso se estructuró en tres sprint principales:

- Sprint 1: Análisis y especificación de requisitos asistido por IA
- Sprint 2: Diseño de la solución potenciado por IA
- Sprint 3: Implementación utilizando Flutter Flow

En cada sprint, se introdujeron primero los conceptos teóricos fundamentales, seguidos de talleres prácticos para aplicar las tecnologías correspondientes, y finalmente sesiones de trabajo guiado donde los equipos aplicaban lo aprendido a sus proyectos específicos.

Tecnologías Utilizadas

La selección de tecnologías se realizó considerando su potencial pedagógico, facilidad de adopción por estudiantes de nivel básico.

IA para requisitos

Para la fase de requisitos se emplearon:

- ChatGPT-4: Utilizado para:
 - Detección de ambigüedades y sugerencias de clarificación
 - Validación de historias de usuario
- Anthropic Claude: Empleado para:
 - Simulación de entrevistas con stakeholders
 - Priorización asistida de requisitos

IA para diseño de software

En la fase de diseño se utilizaron:

- Postgres.New Para:
 - Generación de diagramas ER preliminares basados en requisitos
- Supabase Utilizado para:
 - Generación rápida de la BD y almacenamiento de datos.

IA para la implementación

Flutter Flow como Plataforma NoCode y Diseño de la Interfaz

Flutter Flow constituyó la pieza central para la fase de implementación, seleccionada por:

- Generación de la Interfaz con IA

- Generación de código Flutter (Dart)
- Capacidad para integrar lógica de negocio semi-compleja
- Soporte para autenticación, bases de datos y funcionalidades
- Posibilidad de personalización mediante código cuando fuera necesario

Resultados

La implementación del enfoque descrito produjo resultados significativos en múltiples dimensiones:

- **Transformación del alcance de los proyectos**
El cambio más notable fue la transformación en el alcance y completitud de los proyectos estudiantiles:
 - Antes: En semestres anteriores, los equipos típicamente entregaban prototipos de baja fidelidad que simulaban funcionalidades, pero carecían de implementación real.
 - Después: Todos los equipos completaron aplicaciones móviles funcionales con:
 - Autenticación de usuarios implementada
 - Persistencia de datos reales
 - Navegación completa entre pantallas
 - Lógica de negocio operativa
 - Interfaces de usuario refinadas y responsivas
- **Mejora en la comprensión conceptual**
Las evaluaciones teóricas mostraron una mejora en la comprensión de conceptos fundamentales:
 - Mayor capacidad para identificar interdependencias entre requisitos
 - Comprensión más profunda de las implicaciones arquitectónicas de decisiones de diseño
- **Experiencia de aprendizaje**
El feedback de los estudiantes reveló aspectos positivos sobre la experiencia:
 - Alto nivel de motivación al ver resultados tangibles
 - Reducción de frustración asociada a obstáculos técnicos
 - Mayor confianza en su capacidad para desarrollar aplicaciones completas
 - Apreciación del valor de las herramientas modernas en el proceso de desarrollo
- **Desarrollo de habilidades críticas**
Un resultado no anticipado fue el desarrollo de nuevas habilidades críticas:
 - Capacidad para evaluar críticamente las propuestas generadas por IA
 - Competencia en la adaptación y personalización de plantillas generadas
- **Eficiencia pedagógica**
Desde la perspectiva docente, se observó:
 - Reducción del tiempo dedicado a explicar detalles técnicos de baja relevancia conceptual
 - Más tiempo disponible para discusiones sobre principios, buenas prácticas y aspectos éticos
 - Mayor capacidad para abordar casos de estudio complejos y realistas

Conclusiones

La integración de Inteligencia Artificial y herramientas NoCode en la enseñanza de Ingeniería de Software representa un cambio paradigmático con beneficios sustanciales:

1. **Experiencia integral:** Los estudiantes ahora pueden experimentar el ciclo completo de desarrollo de software, desde la concepción hasta la implementación de aplicaciones funcionales, dentro de las limitaciones temporales de un curso semestral.

2. **Enfoque en lo fundamental:** Al reducir la carga cognitiva asociada a aspectos técnicos específicos, los estudiantes pueden concentrarse en principios fundamentales de la ingeniería de software: arquitectura, diseño, usabilidad, funcionalidad y satisfacción de requisitos.
3. **Preparación para el futuro profesional:** La exposición a herramientas de IA y plataformas NoCode prepara a los estudiantes para un entorno profesional donde estas tecnologías están ganando adopción acelerada.
4. **Democratización del desarrollo:** El enfoque reduce barreras técnicas, permitiendo que estudiantes con diferentes niveles de habilidad en programación puedan contribuir significativamente a proyectos complejos.
5. **Aprendizaje más motivador:** La capacidad de crear productos finales funcionales incrementa la motivación y satisfacción de los estudiantes, potencialmente mejorando la retención y profundidad del aprendizaje.

Esta experiencia demuestra que la adopción estratégica de nuevas tecnologías puede transformar significativamente la educación en ingeniería de software, expandiendo lo que es posible lograr dentro del contexto académico tradicional.

Futuras Investigaciones

- Evaluación longitudinal: Seguimiento de cohortes para determinar el impacto a largo plazo en el desempeño profesional y académico de los estudiantes expuestos a este enfoque.