

## Desarrollo de software asistido por IA para profesionales con formaciones diversas

Matías Callara<sup>[0009-0007-9521-0674]</sup>

Escuela de Negocios, Universidad Torcuato Di Tella, Av. Figueroa Alcorta 7350  
(1428) Ciudad de Buenos Aires, Argentina

mcallara@gmail.com

**Resumen.** En este trabajo se comparten aspectos relacionados con el diseño e implementación de un curso sobre desarrollo de software asistido por IA, orientado a profesionales con formaciones académicas diversas. El objetivo principal del curso es familiarizar a los alumnos con prácticas y herramientas relevantes en el ámbito industrial que los ayuden a desarrollar software de forma efectiva y eficiente, ya sea colaborativamente o de forma individual.

**Palabras clave:** Desarrollo de software asistido por IA, IA generativa, prácticas de desarrollo de software, código abierto.

## AI-Assisted Software Development for Professionals with Diverse Backgrounds

**Abstract.** This work presents aspects related to the design and implementation of a course on AI-assisted software development, aimed at professionals with diverse academic backgrounds. The main objective of the course is to familiarize students with relevant practices and tools in the industrial field that help them develop software effectively and efficiently, either collaboratively or individually.

**Keywords:** AI-Assisted Software Development, Generative AI, Software Development Practices, Open Source.

### 1 Introducción

Actualmente, múltiples industrias requieren la conformación de equipos multidisciplinarios para el desarrollo de soluciones basadas en software. Esto ha generado una necesidad creciente de cursos que permitan a profesionales sin formación formal en ingeniería de software adquirir competencias en prácticas efectivas de desarrollo de software.

Al mismo tiempo, desde la masificación del uso de los modelos de lenguaje grandes (LLMs) (Minaee et al., 2024), se ha observado una aceleración significativa en la aparición de herramientas basadas en inteligencia artificial (IA) que buscan facilitar tareas asociadas al desarrollo de software (Bilgin Ibryam, 2025). Este fenómeno ha contribuido al surgimiento del desarrollo de software asistido por IA como un tema de creciente relevancia tanto en el ámbito académico como en el industrial, que parece posicionarse como una tendencia clave en la evolución de las prácticas asociadas a la producción de software.

Estos dos fenómenos, la creciente demanda de formación en desarrollo de software para perfiles no técnicos, y el avance de herramientas de IA en este campo, motivaron la creación de un curso que permita a los estudiantes mejorar su capacidad para desarrollar soluciones digitales en contextos diversos (por ejemplo, proyectos personales, industriales, individuales y en equipo) aplicando prácticas modernas y adoptadas ampliamente en la industria.

Este trabajo describe esfuerzos similares de enseñanza de estos temas, los criterios que guiaron el diseño del curso, su contenido, detalles de su implementación, evaluación y una discusión sobre algunos aspectos de este nuevo paradigma de desarrollo de software.

## **2 Antecedentes y trabajos relacionados**

Encuestas indican que un porcentaje significativo de desarrolladores ya utiliza o planea utilizar herramientas de IA para acelerar y mejorar sus flujos de trabajo (Daigle, 2024; McKendrick, 2024; Yepis, 2024). Algunas tendencias en este ámbito incluyen un énfasis creciente en intentar cuantificar el retorno de inversión (ROI) de las herramientas de desarrollo de software basadas en IA, la aplicación de IA generativa para modernizar aplicaciones a gran escala y el uso de soluciones impulsadas por IA para remediar vulnerabilidades (Wald, 2025). Las organizaciones también están intentando utilizar agentes de IA capaces de realizar acciones autónomas a lo largo del ciclo de vida del desarrollo de software (SDLC) (Sommerville, 2016), con el fin de mejorar las capacidades de toma de decisiones proactivas.

Las tecnologías de IA de código abierto están ganando terreno, ofreciendo opciones más rentables para entrenar y operar LLMs dentro de los firewalls corporativos, lo cual es particularmente relevante para industrias con estrictos requisitos de seguridad (Salvador, 2025). La IA generativa promete mejorar la calidad del código, facilitar la curva de aprendizaje asociada con nuevas tecnologías y optimizar la generación de casos de prueba (Daigle, 2024). Los modelos modernos de IA exhiben una sofisticación creciente, que incluye capacidades avanzadas de razonamiento y la habilidad de manejar tareas especializadas, junto con la aparición de agentes de IA diseñados para automatizar diversas actividades relacionadas con el desarrollo de software (Nyhan, 2024). El potencial de la IA para automatizar tareas de programación, mejorar la productividad y transformar el rol de los desarrolladores comienza a concretarse. Líderes de destacadas empresas tecnológicas prevén que, en el corto plazo, una parte signifi-

cativa del código que sustenta sus productos será generada mediante estas tecnologías (Kelly, 2024; Kevin, 2025; Marks, 2025).

A pesar de estos avances, todavía existen barreras para la adopción generalizada de la IA en el desarrollo de software, incluyendo preocupaciones sobre la complejidad de estas herramientas, el nivel de confianza en sus resultados, la precisión del código generado por IA y la ausencia de políticas organizacionales claras que regulen su uso (Yepis, 2024).

Como respuesta al creciente interés en el desarrollo de software asistido por IA, se han desarrollado varios cursos en línea y cursos universitarios. Los cursos masivos abiertos en línea (MOOCs), ofrecidos en plataformas como Coursera (Coursera, s.f.-a), Udemy (Udemy, s.f.) y DeepLearning.AI (DeepLearning.AI, s.f.), brindan a los estudiantes oportunidades para explorar la IA generativa aplicada al desarrollo de software, dominar técnicas de ingeniería de prompts y comprender las aplicaciones de la IA en la ingeniería de software (Coursera, s.f.-b; Generative AI for Software Development, s.f.; Sakhuja, s.f.). Estos cursos suelen enfocarse en habilidades prácticas, utilizando herramientas como GitHub Copilot (GitHub Copilot: Your AI Pair Programmer, s.f.) y ChatGPT (OpenAI, s.f.) para cubrir una variedad de temas, desde la generación básica de código hasta patrones avanzados de diseño de software y metodologías de pruebas (testing) (Generative AI for Software Development, s.f.).

Las universidades también están comenzando a incorporar la IA en sus planes de estudio de ciencias de la computación, como lo demuestran los cursos "Ingeniería de Software con IA Generativa" (original: COMPSCI 1060 Software Engineering with Generative AI) de la Universidad de Harvard y "Ingeniería de Software Asistida por IA" (original: CS59200-ASE: AI-assisted Software Engineering) de la Universidad Purdue (Chen & Im, 2025; Zhang, s.f.). El curso de Harvard enfatiza la aplicación práctica de la IA a lo largo del ciclo de vida del desarrollo de software (Chen & Im, 2025), mientras que el programa de Purdue profundiza en los aspectos teóricos y de investigación de la IA en la ingeniería de software (Zhang, s.f.).

El aumento de cursos centrados en la IA generativa y los LLMs indica un amplio reconocimiento sobre su potencial transformador en el desarrollo de software. Sin embargo, cuando este curso se dictó por primera vez (en una versión preliminar) en marzo de 2024, como materia electiva en la Maestría en Análisis y Gestión de Negocios de la Escuela de Negocios de la Universidad Torcuato Di Tella, la mayoría de las formaciones mencionadas anteriormente no existían o no eran ampliamente conocidas.

### 3 Diseño del curso

El curso fue diseñado considerando el perfil de los participantes, el tiempo disponible y las herramientas más adecuadas para apoyar el proceso. A continuación, se describen los principales aspectos que guiaron su diseño:

### 3.1 Público objetivo y duración del curso

Este curso está dirigido a personas provenientes de campos ajenos a la ingeniería de software, pero con una exposición básica en programación, como haber realizado proyectos simples. La carga horaria del curso, considerando las clases teóricas, las prácticas y las consultas es de aproximadamente 25 horas, distribuidas en cinco módulos de cinco horas cada uno.

### 3.2 Fases del SDLC como puntos de anclaje del contenido

El contenido se estructura en torno a las fases del SDLC, guiando a los estudiantes a través de las etapas de planificación, diseño, implementación, pruebas, despliegue y mantenimiento. Utilizar el SDLC como base permite ofrecer una estructura lógica y coherente para comprender el proceso de desarrollo de software. Aunque las prácticas impulsadas por IA están empezando a influir en el SDLC (Gnanasambandam et al., 2025), sus fases tienden a ser más estables que las herramientas empleadas, las cuales evolucionan rápidamente. Contar con un marco teórico sólido es esencial para mitigar los riesgos asociados con el uso actual de la IA, como vulnerabilidades o problemas de escalabilidad (Daigle, 2024).

### 3.3 Lenguaje y herramientas de desarrollo

**Lenguaje de programación.** Se eligió Python como lenguaje de programación debido a su amplia adopción en ciencia de datos e IA, su facilidad de uso para principiantes, su flexibilidad en diversas áreas y la disponibilidad de herramientas asistidas por IA. Gracias a su sintaxis clara y el fuerte respaldo de la comunidad, Python resulta accesible para quienes se inician en la programación, mientras que sus potentes bibliotecas y frameworks permiten integrar IA y aprendizaje automático en el proceso educativo.

**Herramientas que no requieren IA.** Por un lado, el curso incorpora herramientas que no requieren IA en su funcionamiento actual, como Git (Git, s.f.; Skoulikari, 2023), Virtualenv (Virtualenv, s.f.) o Poetry (Poetry: Python Dependency Management and Packaging Made Easy, s.f.). La selección de estas herramientas se basó en criterios como la accesibilidad (por ejemplo, software de código abierto, licencias permisivas o educativas), la estabilidad, la popularidad, la relevancia en la industria, la utilidad, la flexibilidad y la funcionalidad. Se priorizaron aquellas que pudieran aplicarse a múltiples etapas del desarrollo de software, a diversos casos de uso y que ofrecieran facilidad de integración.

**Herramientas asistidas por IA.** En cuanto a las herramientas asistidas por IA, se aplicaron los mismos criterios para respaldar la elección de Visual Studio Code (Microsoft, s.f.; Speight, 2021) como entorno integrado de desarrollo, junto con GitHub Copilot, utilizado como asistente de IA en la versión inicial del curso. GitHub Copilot

actúa como un asistente de programación que sugiere código, proporciona explicaciones y ofrece soporte en diversas tareas de desarrollo, con integración directa en entornos ampliamente utilizados como Visual Studio Code e IntelliJ IDEA (IntelliJ IDEA: The Capable & Ergonomic Java IDE by JetBrains, s.f.). Su disponibilidad bajo una licencia educativa, junto con condiciones favorables en términos de privacidad, su adopción generalizada tanto por desarrolladores independientes como por la industria y sus mejoras frecuentes motivaron su elección.

Esto no impide que, en futuras versiones del curso, se opte por reemplazar GitHub Copilot por otras soluciones de IA o complementarlo con funcionalidades específicas proporcionadas por otras IA. La capacidad de Visual Studio Code, junto con GitHub Copilot, para asistir en la mayoría de las etapas del SDLC hizo que estas dos herramientas se utilizaran a lo largo de todos los módulos del curso.

## **4 Contenido del curso**

El curso está organizado en cinco módulos que abordan de forma progresiva las etapas clave del desarrollo de software moderno con un enfoque particular en el uso de Github Copilot como asistente de IA. En la Tabla 1 se provee un resumen de los temas y herramientas de cada módulo.

### **4.1 Módulo 1: Introducción al control de versiones**

En este primer módulo se abordan los conceptos fundamentales de Git, su utilización a través de la línea de comandos, el manejo de repositorios locales, la realización de *commits*, la creación y fusión de ramas, y los flujos de trabajo habituales con Git. También se introduce la plataforma GitHub (GitHub, s.f.-b) como servicio de hospedaje de código y se exploran las principales funcionalidades de Visual Studio Code para trabajar con Git.

GitHub Copilot se presenta aquí como asistente en tareas repetitivas o complejas, tales como la sugerencia de mensajes de commit, la explicación de comandos y la generación automática de archivos auxiliares (como archivos de configuración o informativos). Este uso inicial permite familiarizarse con el funcionamiento de Copilot como apoyo contextual en la escritura y mantenimiento de código y documentación relacionada.

### **4.2 Módulo 2: Paquetes de Python, entornos virtuales y gestión de dependencias**

En este módulo se introduce la creación y gestión de paquetes de Python, así como el uso de entornos virtuales para aislar dependencias. Se estudian herramientas como *virtualenv* y *Poetry*, y se enseña a configurar proyectos utilizando buenas prácticas de diseño y plantillas.

GitHub Copilot se integra en este contexto como apoyo para generar y resolver problemas con archivos de configuración, generar scripts de instalación y mantener el

archivo de bloqueo de dependencias. Asimismo, sirve como guía para la utilización adecuada de comandos y en la interpretación de errores relacionados con la configuración del entorno.

#### **4.3 Módulo 3: Desarrollo de software y depuración**

Este módulo profundiza en el entorno de desarrollo con énfasis en la depuración. Se introducen conceptos clave como puntos de interrupción, inspección de variables, ejecución paso a paso y depuración remota, utilizando las herramientas de Visual Studio Code.

Se destaca especialmente el uso de GitHub Copilot como herramienta asistente en este proceso. Copilot puede sugerir soluciones a errores comunes, ayudar a redactar *prompts* efectivos en su chat integrado, y explicar fragmentos de código, contribuyendo significativamente a la comprensión y resolución de problemas. Además, se abordan consideraciones relacionadas con la privacidad de datos durante el uso de herramientas como GitHub Copilot.

#### **4.4 Módulo 4: Pruebas y desarrollo guiado por pruebas (TDD)**

Este módulo introduce las prácticas de pruebas de software, tanto desde una perspectiva tradicional como desde el enfoque de desarrollo guiado por pruebas (TDD) (Beck, 2014). Se trabajan pruebas básicas utilizando Pytest (Okken, 2017; Pytest Documentation, s.f.) y Doctest (Doctest: Test Interactive Python Examples, s.f.), así como estructuras de prueba como fixtures y mocks.

GitHub Copilot apoya el proceso generando automáticamente pruebas a partir del código fuente, sugiriendo posibles casos de prueba y ayudando en la estructuración del entorno de pruebas. En el contexto de TDD, Github Copilot se vuelve especialmente útil para iterar rápidamente sobre el ciclo rojo-verde-refactorizar, al proponer implementaciones mínimas para satisfacer las pruebas escritas por el desarrollador.

#### **4.5 Módulo 5: Ciclo de vida del desarrollo de software, versionado y empaquetado**

El módulo final aborda una visión completa del ciclo de vida del desarrollo, desde la planificación hasta el despliegue continuo. Se introduce el uso de GitHub Actions (GitHub, s.f.-a) para la automatización de flujos de integración y entrega continua, y se explora el versionado semántico con Python Semantic Release (Python-Semantic-Release 9.21.0 Documentation, s.f.).

En esta etapa, GitHub Copilot proporciona un soporte significativo con la redacción de notas de lanzamiento y la configuración de archivos necesarios para la ejecución de procesos CI/CD.

**Tabla 1.** Resumen del contenido de los módulos del curso.

Módulo	Fase del SDLC	Temas principales	Herramientas utilizadas
1. Introducción al control de versiones	Implementación	Fundamentos de Git, uso de la línea de comandos, manejo de repositorios y ramas, flujos de trabajo colaborativos, integración con GitHub y Visual Studio Code, asistencia de IA en tareas repetitivas.	Git GitHub GitHub Copilot Visual Studio Code
2. Paquetes de Python, entornos virtuales y gestión de dependencias	Implementación	Estructuración de proyectos, creación de paquetes, uso de entornos virtuales, manejo de dependencias, asistencia de IA en archivos de configuración y resolución de errores.	Virtualenv Poetry GitHub Copilot Visual Studio Code
3. Desarrollo de software y depuración	Implementación y mantenimiento	Técnicas de depuración en VS Code, puntos de interrupción, inspección de variables, prompting efectivo, privacidad en el uso de asistentes de IA.	GitHub Copilot Visual Studio Code
4. Pruebas y desarrollo guiado por pruebas (TDD)	Pruebas	Tipos de pruebas, Pytest y Doctest, principios de TDD, uso de fixtures y mocks, generación automatizada de pruebas con IA.	Pytest Doctest GitHub Copilot Visual Studio Code
5. Ciclo de vida del desarrollo de software, versionado y empaquetado	Despliegue y mantenimiento	Fases del SDLC, automatización con GitHub Actions (CI/CD), versionado semántico, empaquetado, asistencia de IA en generación de documentación y configuración.	Python Semantic Release Poetry GitHub GitHub Copilot Visual Studio Code

## 5 Implementación

Dada la rápida evolución de los temas abordados en el curso, resultó fundamental adoptar estrategias que favorecieran la adaptabilidad, la actualización constante del material y la participación de los estudiantes.

### 5.1 Libro interactivo de acceso abierto

El material del curso se proporcionó como un libro interactivo utilizando JupyterBook (jupyterbook, s.f.). Esta herramienta permite la integración fluida de texto, ejemplos de código, celdas ejecutables mediante Binder (Binder, s.f.) o JupyterHub (jupyter, s.f.) y recursos multimedia, ofreciendo una experiencia de aprendizaje dinámica y flexible.

El código fuente del libro se puso a disposición en un repositorio público en GitHub. Esta estrategia no solo garantiza transparencia y acceso abierto, sino que también fomenta la colaboración y mejora continua mediante contribuciones de la comu-

nidad (a través de issues y pull requests), al mismo tiempo que permite a los estudiantes practicar los flujos de trabajo de Git y GitHub (cubiertos por el curso).

## **5.2 Redes sociales**

Con el fin de extender el impacto del curso más allá del aula, se creó un grupo dedicado en LinkedIn. Esta plataforma sirve como espacio para el intercambio de experiencias, la resolución colaborativa de dudas y la construcción de redes profesionales, fortaleciendo así el componente comunitario y la continuidad del aprendizaje tras la finalización formal del curso.

# **6 Evaluación**

La evaluación del curso se centra en dos dimensiones: la evaluación de los resultados de aprendizaje alcanzados por los participantes y la recolección de opiniones sobre la calidad y pertinencia del curso.

## **6.1 Evaluación del aprendizaje**

El progreso de los participantes se evalúa mediante cuestionarios al final de cada módulo, tareas prácticas que integran los conocimientos adquiridos y un coloquio opcional que les permite obtener puntos adicionales.

## **6.2 Satisfacción de los estudiantes con el curso**

La percepción de los alumnos se mide principalmente a través de encuestas aplicadas al concluir cada módulo y al finalizar el curso. Las encuestas durante la cursada permiten implementar mejoras en tiempo real, ajustando la experiencia según las necesidades detectadas.

## **6.3 Resultado de una versión inicial del curso**

Con el fin de evaluar la experiencia de los participantes y orientar futuras mejoras, al finalizar una versión inicial del curso se aplicó una encuesta estructurada en tres partes: una centrada en el curso, otra en el desempeño del docente, y una tercera orientada a la reflexión del alumno sobre su propio proceso de aprendizaje.

La sección relativa al curso incluía cuatro preguntas, con una escala de respuesta de 1 a 5, que indagaban si el curso había resultado intelectualmente desafiante y exigente, si la demanda de tiempo había sido adecuada, si el contenido era relevante para su formación en negocios y/o ciencia de datos, y una evaluación global del curso. Además, se incluía una pregunta abierta para comentarios generales, donde los participantes podían expresar apreciaciones personales, sugerencias o críticas constructivas.



La encuesta reflejó altos niveles de satisfacción. Los participantes destacaron el carácter práctico del plan de estudios, indicando que las habilidades y prácticas abordadas eran directamente aplicables a los desafíos que enfrentaban o anticipaban en sus roles profesionales. El enfoque en herramientas estándar de la industria (como, por ejemplo, Git, pytest o plataformas de CI/CD) combinado con el aspecto novedoso de la integración de asistentes de IA fue bien recibido. Muchos expresaron que el curso aumentó significativamente su confianza para contribuir a proyectos de software y mejoró su comprensión de los flujos de trabajo de desarrollo modernos, contribuyendo directamente a su empleabilidad y efectividad percibidas. No obstante, las dificultades relacionadas con la instalación y configuración inicial de las herramientas constituyeron un punto de fricción que impactó negativamente en la experiencia de algunos alumnos, lo que motivó la generación de tutoriales y la búsqueda de soluciones para ediciones posteriores del curso.

## 7 Discusión

La integración de la IA en el desarrollo de software impacta a desarrolladores de todos los niveles de experiencia. Para los desarrolladores junior, el proceso de aprendizaje puede acelerarse gracias a la creación rápida de prototipos que estas nuevas herramientas permiten (Wald, 2025). La capacidad de generar rápidamente ejemplos funcionales permite a los desarrolladores junior orientar mejor su aprendizaje y decidir en qué temas enfocar su atención a partir de resultados tangibles, facilitando así la comprensión de conceptos complejos y la iteración sobre sus ideas. Este aumento en la velocidad de creación de prototipos se evidencia, por ejemplo, en la reducción de la duración de los hackatones, que hoy en día pueden limitarse a apenas un par de horas (Vercel, 2025).

Los desarrolladores senior también pueden beneficiarse de la asistencia de la IA. Al automatizar tareas repetitivas, estas herramientas les permiten enfocarse en consideraciones de diseño de más alto nivel y en decisiones relacionadas con la arquitectura de las soluciones (Wald, 2025), concentrándose en tareas donde su conocimiento y experiencia aportan mayor valor y aumentando su productividad.

Sin embargo, la creciente facilidad para generar código con IA también puede introducir desafíos importantes. En enfoques como el “vibe coding” (TOI Tech Desk, 2025), donde el programador produce código describiendo la solución que desea implementar mediante un diálogo con un LLM, es posible generar soluciones sin comprender los principios subyacentes o sin garantizar que el código sea robusto y mantenible. Aunque la IA puede generar código que, superficialmente, parece funcionar, este puede carecer de estructura, manejo adecuado de errores o consideraciones de seguridad necesarias para aplicaciones del mundo real. Por ello, es fundamental que los desarrolladores posean una base firme en los conceptos teóricos fundamentales de desarrollo de software y habilidades sólidas de pensamiento crítico que les permitirán evaluar las soluciones generadas, comprender sus implicaciones y asegurarse de que estén alineadas con los requerimientos del proyecto y con las mejores prácticas.

## 8 Conclusión

La experiencia positiva con una versión inicial del curso sugiere que su estructura y contenido son relevantes para que los miembros de equipos interdisciplinarios de soluciones digitales puedan adquirir las capacidades necesarias para desempeñarse en la industria. Aunque el contenido se estructura en las fases fundamentales del ciclo de vida del desarrollo de software que probablemente perduren por más tiempo, la rápida evolución de las prácticas y herramientas basadas en IA hacen la continua actualización esencial. Esta necesidad motiva la distribución abierta de los materiales de curso, intentando fomentar una comunidad motivada a intercambiar y contribuir a la mejora del contenido.

En un contexto donde es tan difícil predecir cómo serán afectadas las prácticas y herramientas de desarrollo de software incluso en el corto plazo, la estrategia principal de este curso ha sido familiarizarse con las nuevas tecnologías para identificar aquellas que permiten maximizar el valor generado hoy con soluciones más robustas, seguras e innovadoras.

## Referencias

- Beck, K. (2014). *Test-driven development by example*. Boston Addison-Wesley.
- Bilgin Ibryam. (2024, 11 de diciembre). *AI coding assistants landscape*. Generativeprogrammer.com; Generative Programmer. <https://generativeprogrammer.com/p/ai-coding-assistants-landscape>
- Bilgin Ibryam. (2025, 21 de marzo). *AI trends disrupting software teams*. InfoQ. <https://www.infoq.com/articles/ai-trends-disrupting-software-teams>
- Binder. (s.f.). *The Binder project*. Recuperado el 1 de abril de 2025, de <https://mybinder.org/>
- Chen, X., & Im, D. (2025, 31 de enero). “*The pinnacle of practicality*”: Harvard computer science launches new AI software development course. The Harvard Crimson. <https://www.thecrimson.com/article/2025/1/31/compsci-1060-launch/>
- Coursera. (s.f.-a). *Coursera*. Coursera. <https://www.coursera.org>
- Coursera. (s.f.-b). *Generative AI for software developers*. Coursera. Recuperado el 12 de abril de 2025, de <https://www.coursera.org/specializations/generative-ai-for-software-developers>
- Daigle, K. (2024, 20 de agosto). *Survey: The AI wave continues to grow on software development teams*. The GitHub Blog. <https://github.blog/news-insights/research/survey-ai-wave-grows>
- DeepLearning.AI. (s.f.). DeepLearning.AI. Recuperado el 12 de abril de 2025, de <https://www.deeplearning.ai>
- doctest: Test interactive Python examples*. (s.f.). Docs.python.org. Recuperado el 12 de abril de 2025, de <https://docs.python.org/3/library/doctest.html>
- Generative AI for software development*. (s.f.). DeepLearning.AI. Recuperado el 12 de abril de 2025, de <https://www.deeplearning.ai/courses/generative-ai-for-software-development/>
- Git. (s.f.). *Git*. Git. Recuperado el 12 de abril de 2025, de <https://git-scm.com>
- GitHub. (s.f.-a). *Features - GitHub Actions*. GitHub. Recuperado el 12 de abril de 2025, de <https://github.com/features/actions>
- GitHub. (s.f.-b). *GitHub*. GitHub. <https://github.com/>
- GitHub Copilot: Your AI pair programmer*. (s.f.). GitHub. Recuperado el 12 de abril de 2025,

- de <https://github.com/features/copilot>
- Gnanasambandam, C., Harrysson, M., & Singh, R. (2025, 10 de Febrero). *How an AI-enabled software product development life cycle will fuel innovation*. McKinsey & Company. <https://www.mckinsey.com/industries/technology-media-and-telecommunications/our-insights/how-an-ai-enabled-software-product-development-life-cycle-will-fuel-innovation>
- IntelliJ IDEA: *The capable & ergonomic Java IDE by JetBrains*. (s.f.). JetBrains. Recuperado el 12 de abril de 2025, de <https://www.jetbrains.com/idea>
- jupyter. (s.f.). *jupyterhub*. Jupyter. Recuperado el 12 de abril de 2025, de <https://jupyter.org/hub>
- jupyterbook. (s.f.). *Built with Jupyter Book*. Jupyterbook. Recuperado el 12 de abril de 2025, de <https://jupyterbook.org/en/stable/intro.html>
- Kelly, J. (2024, noviembre). *AI powers 25% of Google's code: What's next for software Engineers?* Forbes. <https://www.forbes.com/sites/jackkelly/2024/11/01/ai-code-and-the-future-of-software-engineers>
- Kevin, W. (2025, 14 de marzo). *Anthropic CEO: AI will be writing 90% of code in 3 to 6 months*. Business Insider. <https://www.businessinsider.com/anthropic-ceo-ai-90-percent-code-3-to-6-months-2025-3>
- Marks, G. (2025, 26 de enero). *Business tech news: Zuckerberg says AI will replace mid-level engineers soon*. Forbes. <https://www.forbes.com/sites/quickerbetteertech/2025/01/26/business-tech-news-zuckerberg-says-ai-will-replace-mid-level-engineers-soon>
- McKendrick, J. (2024, 25 de octubre). *Google survey says more than 75% of developers rely on AI. But there's a catch*. ZDNet. <https://www.zdnet.com/article/draft-theres-good-news-and-bad-news-with-ai-assisted-software-development>
- Microsoft. (s.f.). *Visual Studio Code*. Visual Studio Code. Recuperado el 12 de abril de 2025, de <https://code.visualstudio.com>
- Minaee, S., Mikolov, T., Nikzad, N., Chenaghlu, M., Socher, R., Amatriain, X., & Gao, J. (2024). *Large Language Models: A survey*. *ArXiv*. <https://doi.org/10.48550/arxiv.2402.06196>
- Nyhan, P. (2024, 5 de Diciembre). *6 AI trends you'll see more of in 2025*. Source. <https://news.microsoft.com/source/features/ai/6-ai-trends-youll-see-more-of-in-2025/>
- Okken, B. (2017). *Python testing with pytest : Simple, rapid, effective, and scalable*. The Pragmatic Bookshelf.
- OpenAI. (s.f.). *Introducing ChatGPT*. OpenAI. Recuperado el 12 de abril de 2025, de <https://openai.com/index/chatgpt>
- Poetry: *Python dependency management and packaging made easy*. (s.f.). Python Poetry. Recuperado el 12 de abril de 2025, de <https://python-poetry.org>
- Pytest *documentation*. (s.f.). Pytest. Recuperado el 12 de abril de 2025, de <https://docs.pytest.org/en/stable>
- python-semantic-release 9.21.0 documentation*. (s.f.). Readthedocs.io. Recuperado el 12 de abril de 2025, de <https://python-semantic-release.readthedocs.io/en/latest>
- Sakhuja, R. (s.f.). *Generative AI application design and development*. Udemy. Recuperado el 12 de abril de 2025, de [https://www.udemy.com/course/generative-ai-app-dev/?srsltid=AfmBOop1NCv7vCZ4YNI0-PKNaUFsWep\\_cAKLdcs-uRhyM8fdahmGRUTF](https://www.udemy.com/course/generative-ai-app-dev/?srsltid=AfmBOop1NCv7vCZ4YNI0-PKNaUFsWep_cAKLdcs-uRhyM8fdahmGRUTF)
- Salvador, E. (2025, 7 de marzo). *AI's next chapter: four major shifts in software development for 2025*. GeekWire. <https://www.geekwire.com/sponsor-post/ais-next-chapter-four-major-shifts-in-software-development-for-2025/>
- Skoulikari, A. (2023). *Learning Git*. O'Reilly Media, Inc.
- Sommerville, I. (2016). *Software engineering*. Pearson Education India.
- Speight, A. (2021). *Visual Studio Code for Python programmers*. John Wiley & Sons, Incorporated.

- TOI Tech Desk. (2025, 2 de Marzo). *What is “vibe coding”? Former Tesla AI director Andrej Karpathy defines a new era in AI-driven development*. The Times of India; Times of India. <https://timesofindia.indiatimes.com/technology/tech-news/what-is-vibe-coding-former-tesla-ai-director-andrej-karpathy-defines-a-new-era-in-ai-driven-development/articleshow/118659724.cms>
- Udemy. (s.f.). *Online courses - Learn anything, on your schedule*. Udemy. Recuperado el 12 de abril de 2025, de <https://www.udemy.com>
- Vercel. (2025). *The world's shortest hackathon*. Vercel. <https://shortest-hackathon.vercel.app/virtualenv>. (s.f.). Virtualenv. Recuperado el 12 de abril de 2025, de <https://virtualenv.pypa.io/en/latest>
- Wald, B. (2025, 17 de enero). *Five trends that will drive software development in 2025*. DevOps.com. <https://devops.com/five-trends-that-will-drive-software-development-in-2025/>
- Yepis, E. (2024, 29 de mayo). *Developers get by with a little help from AI: Stack Overflow Knows code assistant pulse survey results*. Stackoverflow Blog. <https://stackoverflow.blog/2024/05/29/developers-get-by-with-a-little-help-from-ai-stack-overflow-knows-code-assistant-pulse-survey-results>
- Zhang, T. (s.f.). Teaching. Tianyi Zhang. Recuperado el 12 de abril de 2025, de <https://tianyi-zhang.github.io/teaching/>