



UNIVERSIDAD ESTATAL DE BOLÍVAR

**FACULTAD DE CIENCIAS ADMINISTRATIVAS, GESTIÓN
EMPRESARIAL E INFORMÁTICA**

CARRERA DE SOFTWARE

**TRABAJO DE INTEGRACIÓN CURRICULAR
PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO EN SOFTWARE**

FORMA: PROYECTO DE INVESTIGACIÓN

TEMA:

**MANUAL DE BUENAS PRÁCTICAS PARA EL DESARROLLO DE
SOFTWARE APOYADO EN INTELIGENCIA ARTIFICIAL PARA
ESTUDIANTES DE LA CARRERA DE SOFTWARE DE LA UNIVERSIDAD
ESTATAL DE BOLÍVAR, PERÍODO 2023-2024.**

AUTOR:

ANGELO OCTAVIO FRANCO ORTEGA

DIRECTOR:

ING. HENRY ALBÁN

GUARANDA – ECUADOR

2024

TEMA DEL PROYECTO DE INVESTIGACIÓN

MANUAL DE BUENAS PRÁCTICAS PARA EL DESARROLLO DE SOFTWARE APOYADO EN INTELIGENCIA ARTIFICIAL PARA ESTUDIANTES DE LA CARRERA DE SOFTWARE DE LA UNIVERSIDAD ESTATAL DE BOLÍVAR, PERÍODO 2023-2024.

AGRADECIMIENTO

Quiero expresar mi profundo agradecimiento a todas las personas que contribuyeron al éxito de este proyecto. Sin su apoyo, dedicación y sabiduría, este estudio no hubiera sido posible.

Doy las gracias a todos aquellos compañeros estudiantes que participaron en este proyecto. Gracias por compartir sus experiencias, pues sin ellas este estudio no se hubiera realizado.

Me gustaría agradecer a mi director, el Ing. Henry Albán, cuya paciencia infinita y guía constante han sido un faro en este extenso viaje. Sus valiosas correcciones y recomendaciones han sido cruciales para dirigir mis esfuerzos y asegurar el éxito de este proyecto.

Agradezco a mis estimados pares académicos que supieron aconsejarme sobre el rumbo que debería tomar esta investigación. Al Doc. Carlos Taco por tan grata ayuda respecto al trato de los datos estadísticos y al Ing. Edgar Rivadeneira que me encaminó de manera excepcional en la metodología de la investigación.

Gracias a ustedes, profesionales que han dedicado su vida a formar, hoy puedo decir con orgullo que he completado mi investigación y estoy a punto de convertirme en Ingeniero en Software. Su influencia perdurará en mi carrera y en mi vida.

Finalmente, quiero agradecer a mi familia por su apoyo y comprensión. Sin ustedes este logro no tendría sentido. Gracias por estar siempre a mi lado.

DEDICATORIA

A mis amados padres y hermanos, que siempre han sido el pilar en el que me apoyo, la inspiración que me impulsa y el amor que me fortalece. A ustedes les atribuyo todo lo que he llegado a ser y todo lo que he conseguido.

A mis amigos, que han caminado junto a mí en cada etapa de este recorrido, compartiendo alegrías, retos y momentos inolvidables que guardaré siempre en mi memoria.

A mis profesores, que no solo me enseñaron lecciones académicas, sino también lecciones de vida. Su sabiduría y paciencia han sido la guía que necesitaba para alcanzar este logro.

CERTIFICADO DE VALIDACIÓN



FACULTAD DE CIENCIAS
ADMINISTRATIVAS,
GESTIÓN EMPRESARIAL
E INFORMÁTICA

CERTIFICADO DE VALIDACIÓN

Ing. Henry Albán, Ing. Edgar Rivadeneira y Dr. Carlos Taco, en su orden Director y Pares Académicos del Trabajo de Integración Curricular “MANUAL DE BUENAS PRÁCTICAS PARA EL DESARROLLO DE SOFTWARE APOYADO EN INTELIGENCIA ARTIFICIAL PARA ESTUDIANTES DE LA CARRERA DE SOFTWARE DE LA UNIVERSIDAD ESTATAL DE BOLÍVAR, PERÍODO 2023-2024” desarrollado por el señor Franco Ortega Angelo Octavio.

CERTIFICAN

Que, luego de revisado el Trabajo de Integración Curricular en su totalidad, cumple con las exigencias académicas de la carrera SOFTWARE, por lo tanto, autorizamos su presentación y defensa.

Guaranda, 16 de abril del 2024



Ing. Henry Albán
Director



Ing. Edgar Rivadeneira
Par Académico



Dr. Carlos Taco
Par Académico

DERECHOS DE AUTOR**DERECHOS DE AUTOR**

Yo **Angelo Octavio Franco Ortega** portador de la Cédula de Identidad N° **0503447781** en calidad de autor y titular de los derechos morales y patrimoniales del Trabajo de Titulación: **“Manual de buenas prácticas para el desarrollo de software apoyado en inteligencia artificial para estudiantes de la carrera de software de la Universidad Estatal de Bolívar, período 2023-2024”**, modalidad **Trabajo de Integración Curricular**, de conformidad con el Art. 114 del CÓDIGO ORGÁNICO DE LA ECONOMÍA SOCIAL DE LOS CONOCIMIENTOS, CREATIVIDAD E INNOVACIÓN, concedemos a favor de la Universidad Estatal de Bolívar, una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos. Conservamos a mi/nuestro favor todos los derechos de autor sobre la obra, establecidos en la normativa citada.

Así mismo, autorizo a la Universidad Estatal de Bolívar, para que realice la digitalización y publicación de este trabajo de titulación en el Repositorio Digital, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

El autor declara que la obra objeto de la presente autorización es original en su forma de expresión y no infringe el derecho de autor de terceros, asumiendo la responsabilidad por cualquier reclamación que pudiera presentarse por esta causa y liberando a la Universidad de toda responsabilidad.

A handwritten signature in blue ink, appearing to read 'ANGELO', is written over a horizontal line. The signature is stylized and includes a large, sweeping flourish that extends to the right.

Angelo Octavio Franco Ortega
C.I: 0503447781

ÍNDICE DE CONTENIDO

TEMA DEL PROYECTO DE INVESTIGACIÓN	i
AGRADECIMIENTO	ii
DEDICATORIA	iii
CERTIFICADO DE VALIDACIÓN	iv
DERECHOS DE AUTOR	v
INTRODUCCIÓN	1
RESUMEN.....	3
ABSTRACT.....	4
CAPÍTULO I.....	5
FORMULACIÓN GENERAL DEL PROYECTO.....	5
1.1 Descripción del Problema.....	5
1.2 Formulación del Problema	5
1.3 Preguntas de Investigación.....	5
1.4 Justificación.....	6
1.5 Objetivos: General y Específicos	7
1.6 Hipótesis.....	8
1.7 Variables (Operacionalización).....	8
CAPÍTULO II	10
MARCO TEÓRICO.....	10
2.1 Antecedentes	10
2.2 Científico	11
Buenas Prácticas de Desarrollo de Software	11
COBIT	12
ITIL 4 Foundation.....	13
Inteligencia Artificial en el desarrollo de Software.....	15
Aplicación de la IA en el desarrollo de Software.	16
2.3 Conceptual.....	17
2.4 Legal.....	18
CAPITULO III.....	20
METODOLOGÍA	20
3.1 Tipo de Investigación	20
3.2 Enfoque de la investigación	20
3.3 Métodos de Investigación.....	20
3.4 Técnicas e Instrumentos de Recopilación de Datos	20

3.5	Universo, Población y Muestra	21
3.6	Procesamiento de la Información	22
CAPITULO IV		23
RESULTADOS Y DISCUSIÓN		23
4.1	Análisis, Interpretación y Discusión de Resultados	23
4.1.1	Encuesta establecida para identificar las debilidades al desarrollar software por parte de los estudiantes.	23
4.1.2	Encuesta para evaluar la utilización del manual y su eficacia.	38
4.1.3	Discusión de resultados	50
CONCLUSIONES.....		51
RECOMENDACIONES		53
BIBLIOGRAFÍA.....		54
Anexo I.....		57
	Cronograma (Gantt)	57
Anexo II		59
	Presupuesto Ejecutado.....	59
Anexo III		61
	Encuesta 1.....	61
Anexo IV		65
	Encuesta 2.....	65
Anexo V		68
	Entrevista (guion)	68
Anexo VI.....		71
	Fotografías.....	71
Anexo VII.....		74
	Certificado Antiplagio	74
Anexo VIII		77
	Manual.....	77

ÍNDICE DE TABLAS

Tabla 1 Operacionalización de variables.	8
Tabla 2 Prácticas definidas por ITIL.....	14
Tabla 3 Distribución de estudiantes	21
Tabla 4	23
Tabla 5	24
Tabla 6	25
Tabla 7	26
Tabla 8	27
Tabla 9	28
Tabla 10	29
Tabla 11	30
Tabla 12	31
Tabla 13	32
Tabla 14	33
Tabla 15	34
Tabla 16	35
Tabla 17	36
Tabla 18	37
Tabla 19	38
Tabla 20	39
Tabla 21	40
Tabla 22	41
Tabla 23	42
Tabla 24	43
Tabla 25	44
Tabla 26	45
Tabla 27	46
Tabla 28	47
Tabla 29	48
Tabla 30	49
Tabla 31 Resumen de debilidades identificadas.	51

ÍNDICE DE FIGURAS

Figura 1	23
Figura 2	24
Figura 3	25
Figura 4	26
Figura 5	27
Figura 6	28
Figura 7	29
Figura 8	30
Figura 9	31
Figura 10	32
Figura 11	33
Figura 12	34
Figura 13	35
Figura 14	36
Figura 15	37
Figura 16	38
Figura 17	39
Figura 18	40
Figura 19	41
Figura 20	42
Figura 21	43
Figura 22	44
Figura 23	45
Figura 24	46
Figura 25	47
Figura 26	48
Figura 27	49

INTRODUCCIÓN

La industria del software está siempre en un estado de cambio y crecimiento, con nuevos lenguajes, modelos y tecnologías emergentes que buscan revolucionar la forma en que se construyen los sistemas. Por tanto, la inteligencia artificial (IA) está cambiando la manera de desarrollar software (Bodemer, 2023), pudiendo integrarse en todas las etapas del proceso, optimizando tareas y reduciendo los tiempos de desarrollo.

En este contexto, los estudiantes de los ciclos 5to a 8vo de la carrera de Software de la Universidad Estatal de Bolívar han incorporado la IA como un “copiloto” en sus proyectos, con el objetivo de aprovechar sus ventajas. No obstante, muchos han encontrado dificultades asociadas a la implementación de esta nueva tecnología, lo que ha limitado su uso eficiente.

Frente a esta circunstancia, el propósito de este estudio es abordar los problemas detectados, recogiendo y proponiendo técnicas que asistan al programador en el desarrollo de software de alta calidad. Se plasmarán soluciones aplicables en cualquier contexto en un manual titulado “Buenas prácticas para el desarrollo de software con apoyo de inteligencia artificial”.

El proyecto se estructura de la siguiente manera:

Capítulo I: Se aborda la problemática relacionada con la utilización de inteligencia artificial en el desarrollo de software. Además, se presenta la justificación que manifiesta los aportes que este proyecto brindará. Asimismo, se enuncian los objetivos generales y específicos que encaminarán la investigación.

Capítulo II: Concierno al marco teórico. En él, se presentan antecedentes de proyectos relacionados a la temática. Además, se investigan las bases científicas sobre el cual se apoyará el desarrollo. Por último, se exponen los conceptos relevantes y las normativas legales.

Capítulo III: Este apartado se centra en la metodología elegida para la investigación. En él, se establece el tipo de investigación, el enfoque seleccionado,

así como las técnicas de recopilación de datos que en este caso se optó por encuestas y entrevistas. Además, se describe el universo, población y muestra utilizada. Por último, se explica cómo se procesará la información.

Capítulo IV: Este capítulo aborda los resultados y discusión. Primero, se representan los resultados mediante tablas y gráficos, para después interpretarlos. A partir de esto, se ha elaborado el manual de buenas prácticas para el desarrollo de software apoyado en inteligencia artificial, anexo al final del documento.

Para finalizar, se presentan las deducciones alcanzadas al culminar con la investigación, así como también las recomendaciones y los anexos.

RESUMEN

El objetivo de este trabajo de titulación es elaborar un manual de buenas prácticas para el desarrollo de software apoyado en inteligencia artificial, con el propósito de ayudar a los estudiantes a optimizar la creación de sistemas. Para lograr este cometido, se utilizaron encuestas y entrevistas, lo que permitió identificar las debilidades que los estudiantes presentan en las etapas de construcción de un proyecto. Se empleó una metodología con enfoque mixto, complementando los datos cuantitativos con las experiencias de los estudiantes para generar una visión más detallada del problema y abordarlo de la manera más adecuada. Posteriormente, se brindó una solución basada en estándares. El manual guiará a los programadores a lo largo de todo el ciclo de vida del software, indicando las maneras en que la IA puede ser implementada en cada etapa para optimizar todo el proceso. Los resultados muestran que aquellos que aplicaron el manual experimentaron un aumento en la eficiencia y calidad del software desarrollado. Sin embargo, para obtener estos resultados se requieren dos factores clave: un alto grado de conocimientos técnicos y la comprensión de los principios de la inteligencia artificial.

Palabras clave: Inteligencia artificial, Optimización, Desarrollo de software, Manual de buenas prácticas.

ABSTRACT

The objective of this degree work is to elaborate a manual of good practices for software development supported by artificial intelligence, with the purpose of helping students to optimize the creation of systems. To achieve this goal, surveys and interviews were used to identify the weaknesses that students present in the construction stages of a project. A mixed approach methodology was used, complementing the quantitative data with the students' experiences to generate a more detailed vision of the problem and address it in the most appropriate way. Subsequently, a standards-based solution was provided. The manual will guide programmers through the entire software lifecycle, indicating the ways in which AI can be implemented at each stage to optimize the entire process. The results show that those who applied the manual experienced an increase in the efficiency and quality of the developed software. However, achieving these results requires two key factors: a high degree of technical knowledge and a comprehensive understanding of artificial intelligence principles.

Keywords: Artificial intelligence, Optimization, Software development, Best practices manual.

CAPÍTULO I

FORMULACIÓN GENERAL DEL PROYECTO

1.1 Descripción del Problema

La comunidad universitaria de la carrera de Software de la Universidad Estatal de Bolívar ha introducido la inteligencia artificial en el desarrollo de software sin la existencia de guías específicas que aborden correctamente su uso. La situación es preocupante, pues la IA tiene la capacidad de cambiar la manera en que el software es construido, pero sin el lineamiento adecuado, su implementación puede verse perjudicado, siendo ineficiente e incluso perjudicial.

Los desarrolladores de software nacientes en la institución carecen de guías de buenas prácticas que aborden el uso de la IA en el proceso de desarrollo, las guías existentes se centran en panoramas generales y no profundizan en la singularidad que suscita al crear software mediante IA. Esta falencia puede conducir a equivocaciones y a una implementación incorrecta, lo que resultará en un software de baja calidad y el desperdicio de tiempo y recursos.

Además, la falta de esta guía puede dificultar la capacidad de los desarrolladores para mejorar y optimizar sus proyectos, entorpeciendo la identificación de los aspectos del uso de la IA que funcionan bien y cuáles deben ser mejorados. Es así que la falta de directrices plantea un problema que requiere una solución efectiva e inmediata.

1.2 Formulación del Problema

¿Cómo beneficiaría la elaboración de un manual de buenas prácticas para el desarrollo de software apoyado en la inteligencia artificial a los estudiantes de la carrera de software de la Universidad Estatal de Bolívar, período 2023-2024?

1.3 Preguntas de Investigación

¿Cuáles son las dificultades más comunes que enfrentan los estudiantes de la carrera de software al desarrollar aplicaciones?

¿Cómo puede la inteligencia artificial mejorar la eficiencia y la calidad en cada fase del desarrollo de software?

¿Cómo afecta la aplicación de la inteligencia artificial a la calidad del software?

1.4 Justificación

En los últimos años el desarrollo de la inteligencia artificial ha avanzado rápidamente, consolidándose como una herramienta poderosa utilizada en varios campos, indica Ruohonen (2018) . El impacto de la IA ha sido tal, que se ha llegado a integrar en diversas áreas como medicina, astronomía, física, ciencias sociales, ciencias ambientales, ciencia de datos, ciencias computacionales, entre otras disciplinas científicas, para interceder en la toma de decisiones y desempeño de tareas.

En cuanto al desarrollo de software, gran parte de programadores han adoptado a la IA como soporte en el desarrollo de sistemas de escritorio, móvil o web. Según una encuesta realizada por el equipo de Stack Overflow (2023) a 90.000 desarrolladores y tecnólogos, indica que el 70% ya está usando o planea usar herramientas de IA para el proceso de desarrollo. Esto deja claro que los desarrolladores abrazan la idea de introducir a la IA como un asistente en sus actividades de trabajo. La integración de la IA en este contexto ha significado la reducción en los tiempos de desarrollo y un aumento en la productividad.

La acogida de la comunidad universitaria a la inteligencia artificial va en ascenso y con ello el surgimiento de nuevos problemas que deberán ser resueltos tarde o temprano. Uno de los desafíos que enfrentan los estudiantes de la carrera de software de la Universidad Estatal de Bolívar es la falta de normas para desarrollar sistemas con IA.

La importancia de esta investigación radica en la necesidad de brindar una solución al desafío que enfrentan los desarrolladores de software debido a la carencia de guías de buenas prácticas para el desarrollo apoyado en la IA. Los manuales existentes son genéricos y no tratan las particularidades y desafíos que surgen al crear software con IA. Esta carencia genera un problema significativo, pues los

desarrolladores encuentran obstáculos en la aplicación efectiva de la IA en sus proyectos, desaprovechando así todo su potencial.

La investigación se enmarca en la línea de Ingeniería de Software, redes y Telecomunicaciones, sublínea de Pruebas y aseguramiento de la calidad del Software, redes y telecomunicaciones. Al plantear una guía de buenas prácticas que aborde los aspectos de la introducción de la inteligencia artificial en el ciclo de vida del software, se proporcionará instrucciones que un desarrollador podrá seguir para aprovechar al máximo el potencial de la IA, ayudando a evitar las malas prácticas de programación y aumentando la calidad del software. Los beneficiarios directos serán los estudiantes de la carrera de Software, quienes podrán desarrollar aplicaciones de forma efectiva. De manera indirectamente los usuarios de los sistemas también se verán beneficiados, pues recibirán un software de mayor calidad.

1.5 Objetivos: General y Específicos

General

Elaborar un Manual de buenas prácticas con normas ISO que garanticen la calidad y eficiencia en el desarrollo de software, apoyado en inteligencia artificial.

Específicos

Identificar las debilidades que presentan los estudiantes de la carrera de software al desarrollar aplicaciones.

Investigar la relación entre la aplicación de la inteligencia artificial y la calidad del software.

Determinar la contribución de la inteligencia artificial en la creación del Software.

Evaluar la funcionalidad del Manual desarrollado.

1.6 Hipótesis

El manual de buenas prácticas para el desarrollo de software con inteligencia artificial permitirá optimizar el proceso de creación de aplicaciones a los estudiantes de la carrera de Software.

1.7 Variables (Operacionalización)

Tabla 1

Operacionalización de variables.

Variable	Definición	Dimensiones	Indicadores
Desarrollo de software con inteligencia artificial	Proceso que utiliza instrumentos de inteligencia artificial para la creación de programas informáticos que realizan tareas específicas.	Inteligencia artificial	Adopción de la IA
		Precisión del software	Tasa de errores
			Tasa de aciertos
		Funcionalidad del software	Cobertura de los requisitos
			Tiempo de respuesta
Mantenibilidad	Facilidad para modificar, mejorar y adaptar el software		
Tipo: Independiente			

Optimización del proceso de creación del software	Mejora continua del proceso de desarrollo de software para aumentar la eficiencia y calidad del producto.	Tiempo de desarrollo	Tiempo de documentación
			Tiempo de diseño
			Tiempo de codificación
		Calidad del software	Reducción de errores.
			Tiempo de inactividad
			Rendimiento
Satisfacción del cliente	Puntuación de satisfacción del cliente		
	Net Promoter Score		
Tipo: Dependiente			
Manual de buenas prácticas	Documento que recopila las estrategias y técnicas diseñadas para optimizar el proceso de desarrollo de software, con un enfoque en la integración de la inteligencia artificial.	Buenas prácticas para el desarrollo de Software	Guías existentes
Tipo: Interviniente			

CAPÍTULO II

MARCO TEÓRICO

2.1 Antecedentes

Acosta Bravo, E. A. (2019) realizó un trabajo acerca de la documentación y adopción de buenas prácticas para el desarrollo de software basado en los fundamentos DevOps. En el documento presentó una propuesta de marco de trabajo que facilita la adopción y documentación de las mejores prácticas en el desarrollo de software. Este enfoque combina los fundamentos de DevOps con los principios de las metodologías tradicionales. El propósito es establecer procesos que evolucionen con el tiempo y permitan crear aplicativos de calidad. El autor describe el concepto, la historia, las ventajas, el ciclo y los componentes de DevOps, así como las mejores prácticas, los casos de éxito y los motivos de fracaso. También sugiere un enfoque para evaluar el grado de madurez de las organizaciones antes de emprender la implementación de DevOps, basada en la caracterización de diferentes categorías y la asignación de pesos por preguntas. Finalmente, presenta un estudio de caso de una compañía de desarrollo de software que aplicó la metodología propuesta y obtuvo un informe de análisis con recomendaciones para mejorar su proceso DevOps.

Soraluz et al. (2021) concretaron una revisión bibliográfica sobre el desarrollo guiado por comportamiento (BDD) y sus buenas prácticas para la calidad de software. Ellos identificaron algunas prácticas como el uso de escenarios, etiquetas, carpetas y lenguaje natural para describir y probar las funcionalidades del software. Ellos concluyeron que BDD permite una mejor comunicación, diseño, ejecución e integridad del software, así como un menor riesgo de pérdida de recursos y un mayor margen de éxito.

Sarco et al. (2022) presenta una propuesta de un modelo para integrar la inteligencia artificial (IA) y la ingeniería de software en el desarrollo de soluciones de programación. El objetivo es aprovechar la capacidad de la IA para mejorar la eficiencia y calidad del software. El artículo describe los conceptos, tipos, aplicaciones y beneficios de la IA en el desarrollo de software, así como los desafíos y limitaciones que se presentan. También plantea una estrategia para medir el grado

de adopción de la IA en el ciclo de vida del software, basada en una revisión sistemática de literatura y un caso de estudio de la plataforma científica EcuCiencia. Finalmente, presenta un modelo que determina los elementos clave par lograr el éxito de la incorporación de la IA y la ingeniería de software, utilizando modelos y algoritmos de última generación.

En base a los resultados de los casos expuestos se puede afirmar que la adopción de las mejores prácticas en el desarrollo de software es fundamental para la mejora de calidad y eficiencia. En el primer caso destaca la importancia de adoptar y documentar prácticas para el desarrollo de software basado en fundamentos DevOps, lo que permite definir y establecer procesos que evolucionan con el tiempo. En el segundo caso se identificaron prácticas clave del desarrollo guiado por comportamiento que permiten tener una mejor comunicación, diseño, ejecución e integridad del software. En el tercer caso propusieron un modelo para integrar IA en el desarrollo de software, lo que puede aprovechar su potencial para mejorar la eficiencia y calidad del software. Todos estos trabajos subrayan la importancia de la adopción de buenas prácticas y la integración de tecnologías emergentes.

2.2 Científico

Buenas Prácticas de Desarrollo de Software

Shyamal Mitra (2014), en su artículo “Best Practices in Developing Programs” de la Universidad de Texas en Austin, propone las siguientes prácticas óptimas para el desarrollo de software:

- **Iteración en el desarrollo:** Se resuelven los riesgos críticos antes de realizar inversiones significativas. Las primeras iteraciones permiten recibir retroalimentación temprana de los usuarios. Las pruebas y la integración son procesos continuos. Los hitos objetivos proporcionan un enfoque a corto plazo.
- **Manejo de requisitos:** Los requisitos son dinámicos y se espera que cambien durante el desarrollo del software. La comprensión del usuario sobre los requisitos evoluciona con el tiempo. Se debe llegar a un acuerdo

con el usuario sobre lo que el sistema debe hacer y no cómo. Se debe mantener la trazabilidad de los requisitos hacia adelante y hacia atrás.

- **Uso de una arquitectura basada en componentes:** El uso de componentes permite la reutilización. Existen miles de componentes disponibles comercialmente. Mejora la mantenibilidad y la extensibilidad. Promueve una división clara del trabajo entre los equipos de desarrolladores.
- **Modelado visual del software:** El modelado visual mejora nuestra capacidad para manejar la complejidad del software. Captura la organización y conducta de los componentes. Oculta o expone detalles según sea apropiado para la tarea. Promueve una comunicación no ambigua.
- **Verificación de la calidad del software:** ¿Qué es la calidad? Cumple o supera los requisitos acordados por algunas medidas objetivas acordadas.
- **Control del cambio:** Las mejores prácticas son un conjunto de enfoques empíricamente probados para el desarrollo de software. Cuando se utilizan en combinación, atacan a las raíces de las dificultades del desarrollo de software. Se les llama “mejores prácticas” no porque podamos cuantificar precisamente su valor, sino porque se observa que son comúnmente utilizadas en la industria por organizaciones exitosas.

COBIT

ISACA (2019) en su marco referencial sobre COBIT (Objetivos de Control para Información y Tecnología), se establece como un marco de trabajo que sirve como guía para la administración y supervisión de la información y la tecnología, y es aplicable a todas las organizaciones. Este marco ha crecido y se ha transformado en un manual completo para la administración y supervisión de la tecnología y la información.

La versión más reciente, COBIT 2019, ofrece una serie de mejores prácticas. Aunque no proporciona un análisis exhaustivo de todo el ambiente tecnológico de una entidad, ni un esquema para estructurar procesos comerciales, ni un marco experto para administrar toda la tecnología, COBIT proporciona un enfoque de ciclo de vida que ayuda a las empresas a manejar la complejidad y los desafíos que suelen encontrarse durante las implementaciones.

El marco COBIT distingue claramente entre la gobernanza y la gestión. La gobernanza se encarga de evaluar las necesidades, condiciones y opciones de los stakeholders para determinar objetivos empresariales equilibrados y consensuados. Por otro lado, la gestión se encarga de planificar, construir, ejecutar y supervisar acciones en concordancia con la orientación fijada por el ente de supervisión para lograr las metas de la empresa.

Relacionado al ciclo de vida del desarrollo de software (SDLC), aunque la lista y las delimitaciones de las fases o flujos de trabajo del SDLC pueden variar entre los enfoques, generalmente se acuerda que se incluyen los siguientes siete flujos de trabajo dentro del SDLC: Planificación, Recopilación y análisis de requisitos, Diseño, Implementación, Pruebas, Despliegue, Mantenimiento

ITIL 4 Foundation

ITIL, que significa Biblioteca de Infraestructura de Tecnologías de la Información, es un grupo de prácticas destacadas destinadas a la administración de servicios de tecnología informática. El libro ITIL 4 de AXELOS (2019) aborda una variedad de temas, incluyendo el Sistema de Valor del Servicio, los principios guía, las cuatro perspectivas del enfoque holístico, la secuencia de valor del servicio y los procedimientos.

El Sistema de Valor del Servicio es un arquetipo operativo que describe cómo las organizaciones pueden crear valor para sus clientes y usuarios a través de servicios habilitados por TI. Este sistema se compone de varios elementos, incluyendo los principios guía, las cuatro dimensiones, la secuencia de valor del servicio, las prácticas y el mejoramiento continuo.

Los principios guía son siete recomendaciones que ayudan a las entidades a asumir y ajustar la administración de servicios de TI de acuerdo con su contexto y necesidades. Estos principios incluyen enfocarse en el valor, empezar por lo más simple, avanzar de manera iterativa con feedback, cooperar y fomentar la visibilidad, reflexionar y actuar de manera integral, preservar la flexibilidad, optimizar y automatizar.

Las cuatro dimensiones son aspectos que deben considerarse para garantizar un enfoque holístico de la gobernanza de servicios. Estas dimensiones son: entidades e individuos, datos y tecnología, colaboradores y proveedores, y secuencias de valor y procesos.

La secuencia de valor del servicio consiste en una serie de tareas vinculadas que posibilitan atender a la necesidad y facilitar la generación de valor mediante servicios. Las actividades incluyen planificar, progresar, involucrar, diseñar y hacer el cambio, obtener y cimentar, entregar y dar soporte.

Las prácticas son conjuntos de recursos institucionales que se utilizan para realizar la labor o alcanzar una finalidad. ITIL 4 define 34 prácticas agrupadas en tres categorías: generales, de servicio y técnicas. El libro presenta siete prácticas con profundidad: perfeccionamiento constante, control de modificaciones, administración de incidentes, manejo de problemas, gestión de peticiones de servicio, mesa de ayuda y administración de niveles de servicio.

Tabla 2
Prácticas definidas por ITIL

Prácticas de Gestión General	Prácticas de Gestión de Servicios	Prácticas de Gestión Técnica
<ul style="list-style-type: none"> • Administración de la arquitectura • Perfeccionamiento constante • Manejo de seguridad de la información • Administración del saber • Cuantificación y reporte • Manejo del cambio institucional 	<ul style="list-style-type: none"> • Administración de la disponibilidad • Evaluación del negocio • Manejo de la capacidad y rendimiento • Supervisión de cambios • Manejo de incidentes • Administración de recursos de TI 	<ul style="list-style-type: none"> • Administración de la implementación. • Gestión de infraestructuras y plataformas. • Desarrollo y manejo de software.

<ul style="list-style-type: none"> • Administración del portafolio • Administración de proyectos • Manejo de relaciones • Manejo de riesgos • Administración financiera del servicio • Manejo de estrategias • Gestión de proveedores • Manejo de personal y talento 	<ul style="list-style-type: none"> • Gestión de eventos y seguimiento • Resolución de problemas • Manejo de lanzamientos • Manejo de la lista del servicio • Gestión de la configuración del servicio • Continuidad del servicio • Diseño de servicio • Mesa de ayuda • Administración de niveles de servicio • Manejo de solicitudes de servicio • Verificación y prueba del servicio 	
--	---	--

Nota. Freshworks (s.f.). ITIL v4: marco y mejores prácticas.

Inteligencia Artificial en el desarrollo de Software.

La introducción de la inteligencia artificial (IA) en el desarrollo de software ha revolucionado la forma en que se crean, mantienen y mejoran los sistemas. La IA en conjunto con la labor de los programadores ha permitido un mejoramiento significativo en la eficiencia y calidad del desarrollo de Software. Según Vergadia (2023), la IA permite:

1. **Automatizar tareas repetitivas:** Como las revisiones de código, pruebas y depuración, lo que minimiza el tiempo que los programadores dedican a estas tareas.

2. **Optimizar el tiempo:** La automatización de tareas repetitivas permite a los programadores emplear su tiempo en tareas más significativas e innovadoras.
3. **Generar código:** Mediante herramientas impulsadas por IA como GitHub Copilot, Amazon CodeWhisperer, entre otras, se puede facilitar la escritura de código.
4. **Modernizar aplicaciones:** La IA puede traducir el código de software de un lenguaje a otro, facilitando su transformación y modernización.
5. **Mejorar la productividad del desarrollador:** Al utilizar todas estas herramientas, se facilita la escritura de código y se recomiendan fragmentos de código para cumplir con los requerimientos.

Aplicación de la IA en el desarrollo de Software.

El informe titulado “The State of AI in Software Development”, publicado por GitLab (2023), evalúa el estado actual de la IA en el desarrollo de software. Este informe indica que la adopción de la IA es la dirección que se tomará para el futuro, pues el 83% de los encuestados considera que es esencial implementar la IA en sus procesos de desarrollo de software para no quedarse atrás. El 23% de los encuestados dice que su organización ya está usando la IA en el ciclo de vida del desarrollo de software, y el 67% dice que tiene planes de hacerlo.

Los principales beneficios que los encuestados asocian con el uso de la IA en el desarrollo de software son la mejora de la eficiencia (55%), la disminución de los tiempos de ciclo (44%) y el aumento de la innovación (41%). Los encuestados también esperan beneficios personales como el aumento de la productividad (51%), la rapidez en las implementaciones (44%) y la mayor precisión (40%).

En lo que respecta a los casos de uso de la IA en el desarrollo de software es la generación y sugerencia de código (55%), seguido de la predicción de métricas de productividad y la identificación de anomalías (54%), los resúmenes de cambios de código (53%) y las explicaciones de cómo funciona un código (53%). Sin embargo, el uso actual de la IA se centra más en los chatbots de lenguaje natural en la

documentación (41%), la generación automática de pruebas (41%) y los resúmenes de cambios de código (39%).

Los principales obstáculos que los encuestados han encontrado o esperan encontrar al implementar la IA en el desarrollo de software son las preocupaciones sobre la privacidad y la protección de los datos (37%), las vulnerabilidades de seguridad en el software construido con IA (35%) y la falta de habilidades adecuadas para emplear la IA o interpretar su salida (34%). Los encuestados también expresaron preocupaciones sobre la propiedad intelectual, la confianza en la salida generada por la IA y la necesidad de formación y recursos para usar la IA.

Además, los encuestados reconocen las limitaciones inherentes de la IA y la necesidad de revisión humana de la salida generada por la IA. El 57% de los encuestados cree que la IA reemplazará su rol en los próximos cinco años, pero el 90% de los encuestados se siente seguro usando la IA en sus tareas diarias. Los encuestados más experimentados son menos propensos a asociar la IA con ventajas como la mejora de la productividad y la reducción de los tiempos de ciclo, lo que sugiere que valoran más la experiencia, el conocimiento y la resolución de problemas humanos

2.3 Conceptual

Buenas prácticas en el desarrollo de Software. - Es la colección de métodos o técnicas que facilitan la realización eficiente de todas las tareas involucradas en la creación de un sistema. (García et al., 2015)

Eficiencia. - Capacidad de un proyecto para transformar recursos, tales como tiempo, trabajo y dinero, en funcionalidades de software de manera efectiva. (Chávez et al., 2014)

Inteligencia artificial. - La IA es definida como la capacidad de las computadoras y máquinas de imitar las habilidades de resolución de problemas y toma de decisiones de la mente humana (Ocaña et al., 2019). También se plantea que la IA en su forma más simple es un campo que combina la informática y sólidos conjuntos de datos para permitir la resolución de problemas (IBM, 2023).

Tecnologías de la información y comunicación. - Tecnologías que constituyen un grupo de herramientas que facilita la adquisición, elaboración, procesamiento y difusión de información en diversas formas y formatos. (Cruz et al., 2019)

Calidad del software. - Cumplimiento de los requisitos funcionales y desempeño claramente definidos, alineados con estándares de desarrollo, y con propiedades no explícitas que se anticipan a cualquier software creado de manera profesional. (Pressman, 2010)

Stakeholder. – Cualquier actor, interno o externo, que afecta o se ve afectado por el proyecto en desarrollo. (Caballero et al., 2007)

Ciclo de vida del desarrollo de software. – Según Amazon Web Services, se trata de un marco de trabajo compuesto por varias fases, aplicado por equipos de desarrollo para producir software de calidad de manera eficiente y rentable.

Automatización. – Término utilizado frecuentemente en el sector industrial para describir un sistema que desarrolla tareas sin la necesidad de intervención humana. (Agudelo et al., 2020)

Ingeniería de Software. – Aplicación metódica de conocimientos, técnicas y experiencias científicos y tecnológicos al diseño, implementación, prueba y documentación de software. (IEEE, 2017)

2.4 Legal

El art. 145 de la Ley Orgánica de Educación Superior insta el “Principio de autodeterminación para la producción del pensamiento y conocimiento”, en el cual se menciona que las entidades de enseñanza avanzada deben tener la libertad y el autogobierno para la producción de pensamiento y conocimiento, valorar todas las formas de pensamiento y conocimiento sin importar su origen, y adaptarse a los avances científicos y tecnológicos tanto a nivel local como global.

El art. 23 de la Ley Orgánica de Protección de Datos manifiesta el derecho a la educación digital. Lo que garantiza el acceso a la información en el uso sensato de las tecnologías de la información y comunicación (TIC), respetando la dignidad humana, derechos fundamentales y libertades individuales. Se resalta la salvaguarda de la privacidad de los usuarios, autodeterminación informativa, la

identidad y reputación online, y la defensa de los datos personales. Promoviendo una cultura que valora la repercusión de la custodia de los datos personales.

La normativa institucional vigente en la Universidad Estatal de Bolívar no establece impedimentos legales para el desarrollo del Manual de Buenas Prácticas apoyado en Inteligencia Artificial. Lo que abre un camino prometedor para la integración de tecnologías emergentes en nuestras prácticas académicas y administrativas.

CAPITULO III

METODOLOGÍA

3.1 Tipo de Investigación

La investigación será de tipo aplicada, documental

- Aplicada porque se brindará una solución al problema, la falta de una guía de buenas prácticas para la integración de la inteligencia artificial en el desarrollo de software.
- Documental porque se basa en la revisión y análisis de fuentes secundarias de datos para fundamentar teóricamente el estudio.

3.2 Enfoque de la investigación

La investigación tendrá un enfoque mixto, ya que se requiere combinar tanto los datos cuantitativos como cualitativos para lograr una perspectiva más completa. Por un lado, se usarán los datos numéricos para medir cuáles son las debilidades más comunes de los estudiantes de Software al desarrollar aplicaciones. Por otro lado, se utilizarán datos verbales para comprender esas debilidades desde un aspecto más específico.

3.3 Métodos de Investigación

Se optó por un método hipotético - deductivo el cual Karl Popper (2008) estatuye como el método científico se puede representar mediante un modelo que consiste en un ciclo de tres fases: inducción, deducción e inducción nuevamente, que sirve para plantear y verificar o rechazar hipótesis. Además, afirmó que las hipótesis científicas no provienen de la observación, sino que son el resultado de la inventiva humana, que busca resolver un problema a través de ellas.

3.4 Técnicas e Instrumentos de Recopilación de Datos

En un esfuerzo por entender mejor las dificultades que enfrentan los estudiantes de Software al desarrollar aplicaciones, se empleará dos métodos clave de recopilación de datos: encuestas y entrevistas.

La encuesta permitirá recoger información cuantitativa y dará una idea de las dificultades más comunes que los estudiantes encuentran. Esta herramienta proporcionará una visión panorámica de los desafíos más habituales y ayudará a comprender la magnitud de estos problemas. La encuesta será aplicada sobre aproximadamente 50 estudiantes de la carrera de Software que cursan el quinto semestre en adelante. (Ver Anexo III / Anexo IV)

En cuanto a la entrevista, permitirá profundizar en estas dificultades, pues se tendrá la oportunidad de explorar las experiencias particulares de los estudiantes, lo que dará una perspectiva más matizada de los problemas que enfrentan. Se prevé que la entrevista se aplicará a 2 estudiantes de cada ciclo, un total de 8. Estos serán seleccionados de manera aleatoria. (Ver Anexo V)

Al combinar tanto el método cuantitativo y cualitativo, se obtendrá una visión completa de las dificultades que los estudiantes de Software enfrentan al desarrollar aplicaciones. Esta combinación de técnicas permitirá tratar el problema desde diferentes ángulos y ayudará a desarrollar soluciones más efectivas.

3.5 Universo, Población y Muestra

El universo está compuesto por 64 estudiantes de la carrera de software que cursan quinto, sexto, séptimo y octavo semestre. No se tomará muestra.

Tabla 3
Distribución de estudiantes

Semestre	Número de estudiantes
Quinto	11
Sexto	13
Séptimo	22
Octavo	18
Total	64

Nota. Semestres inferiores no se incluyeron en el estudio debido a que aún no han recibido formación sobre las buenas prácticas para el desarrollo de software.

3.6 Procesamiento de la Información

Se empleará un software de hoja de cálculo para la organización de datos, que facilite la ordenación y manipulación de grandes volúmenes de información de forma eficaz.

Se usó un editor de texto para integrar los datos tabulados, lo que facilita la actualización de estos en el documento cuando se modifiquen en la hoja de cálculo.

CAPITULO IV

RESULTADOS Y DISCUSIÓN

4.1 Análisis, Interpretación y Discusión de Resultados

4.1.1 Encuesta establecida para identificar las debilidades al desarrollar software por parte de los estudiantes.

Encuesta 1

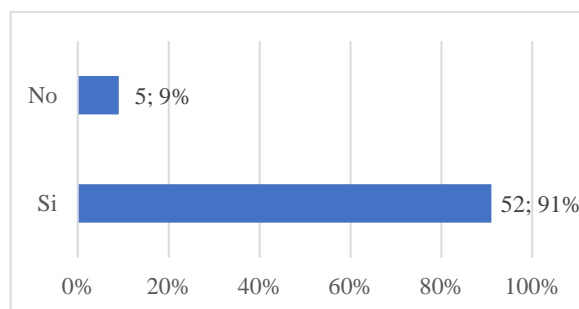
1. ¿Utilizas inteligencia artificial?

Tabla 4

Opciones	Frecuencia	Porcentaje
Si	52	91%
No	5	9%
Total general	57	

Fuente: Encuesta 1 realizada a estudiantes de la carrera de Software
Autor: Angelo Franco

Figura 1



Fuente: Encuesta 1 realizada a estudiantes de la carrera de Software
Autor: Angelo Franco

Interpretación:

El 91% de estudiantes responden que sí utilizan IA, el restante no.

Se concluye que la IA ha tenido una gran acogida por parte de los estudiantes de la carrera de Software, sin embargo, hay un 9% que la rechaza.

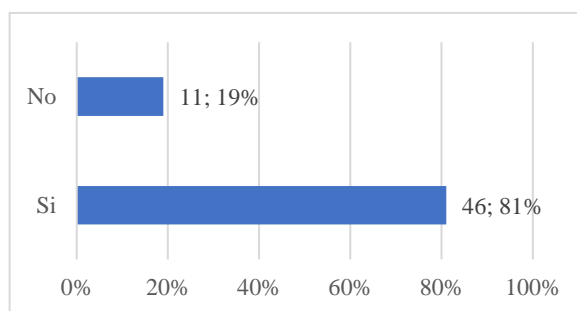
2. ¿Has utilizado la inteligencia artificial para desarrollar software (en cualquiera de sus fases)?

Tabla 5

Opciones	Frecuencia	Porcentaje
Si	46	81%
No	11	19%
Total general	57	

Fuente: Encuesta 1 realizada a estudiantes de la carrera de Software
Autor: Angelo Franco

Figura 2



Fuente: Encuesta 1 realizada a estudiantes de la carrera de Software
Autor: Angelo Franco

Interpretación:

El 81% de estudiantes indican que sí utilizan IA para desarrollar software. El resto de los estudiantes no lo utilizan.

Los estudiantes manifiestan que utilizan la IA en el desarrollo de software para la automatización de tareas. Por otro lado, la no utilización de la herramienta puede deberse a la falta de formación en este aspecto o a la preferencia por otros métodos.

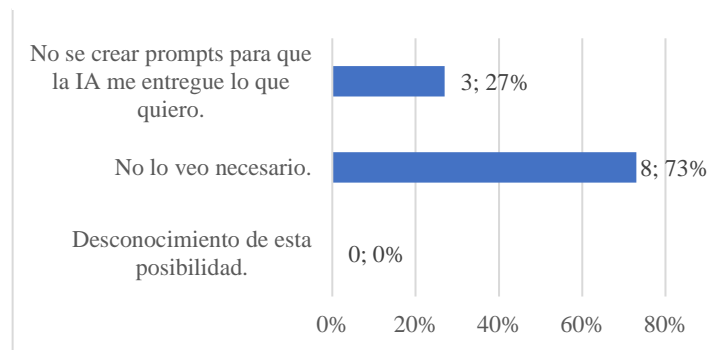
3. Indica el motivo por el cual no utiliza inteligencia artificial al desarrollar software.

Tabla 6

Opciones	Frecuencia	Porcentaje
Desconocimiento de esta posibilidad.	0	0%
No lo veo necesario.	8	73%
No se crear prompts para que la IA me entregue lo que quiero.	3	27%
Total de respuestas	11	

Fuente: Encuesta 1 realizada a estudiantes de la carrera de Software
Autor: Angelo Franco

Figura 3



Fuente: Encuesta 1 realizada a estudiantes de la carrera de Software
Autor: Angelo Franco

Interpretación:

El 19% de los estudiantes que no utilizan IA indicaron sus motivos. El 73% no lo ven necesario y el 27% tienen complicaciones para crear prompts.

La negativa se debe a que no creen que sea necesario utilizar IA y a la complicación para formular prompts adecuados.

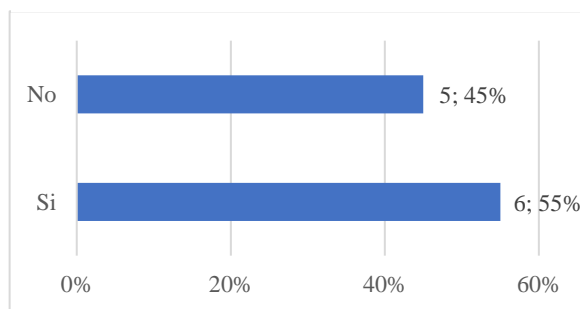
4. ¿Estarías dispuesto a utilizar la inteligencia artificial en tus proyectos de desarrollo?

Tabla 7

Opciones	Frecuencia	Porcentaje
Si	6	55%
No	5	45%
Total general	11	

Fuente: Encuesta 1 realizada a estudiantes de la carrera de Software
Autor: Angelo Franco

Figura 4



Fuente: Encuesta 1 realizada a estudiantes de la carrera de Software
Autor: Angelo Franco

Interpretación:

Del 19% de estudiantes que no utilizan IA, el 55% indican que sí están dispuestos a utilizar IA en sus proyectos de desarrollo, el 45% no.

Los resultados arrojan un porcentaje no muy distintivo, lo cual indica que algunos estudiantes ven el potencial de la IA, pero todavía hay reservas o desconfianza en su uso.

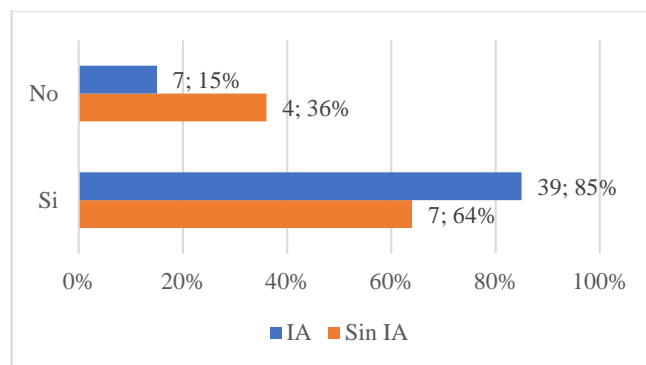
5 – 6. ¿Has tenido dificultades al desarrollar software con/sin inteligencia artificial?

Tabla 8

Opciones	IA		Sin IA	
	Frecuencia	Porcentaje	Frecuencia	Porcentaje
Si	39	85%	7	64%
No	7	15%	4	36%
Total general	46		11	

Fuente: Encuesta 1 realizada a estudiantes de la carrera de Software
 Autor: Angelo Franco

Figura 5



Fuente: Encuesta 1 realizada a estudiantes de la carrera de Software
 Autor: Angelo Franco

Interpretación:

El 81% de la totalidad de estudiantes presenta dificultades al desarrollar software, que incluye el 85% que utilizan IA y 64% que no la utilizan.

Se evidencia que los estudiantes que usan IA en el proceso de desarrollo se encuentran con mayores inconvenientes que aquellos que no la utilizan. Si bien la IA tiene la capacidad de mejorar los procesos, también introduce nuevos desafíos que los estudiantes deberán enfrentar.

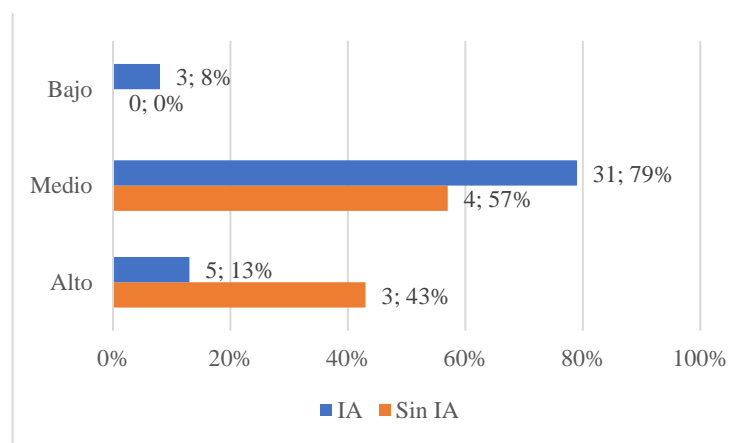
7. Indica cuál es el grado de la dificultad.

Tabla 9

Opciones	IA		Sin IA	
	Frecuencia	Porcentaje	Frecuencia	Porcentaje
Alto	5	13%	3	43%
Medio	31	79%	4	57%
Bajo	3	8%	0	0%
Total general	39		7	

Fuente: Encuesta 1 realizada a estudiantes de la carrera de Software
Autor: Angelo Franco

Figura 6



Fuente: Encuesta 1 realizada a estudiantes de la carrera de Software
Autor: Angelo Franco

Interpretación:

Del 81% de estudiantes que tienen dificultades, el 76% experimentan un grado de dificultad medio, incluyendo el 79% de quienes usan IA y el 57% que no. Mientras que un grado de dificultad alto está presente en el 13% de estudiantes que usan IA y el 43% que no lo hacen.

Los datos muestran que, quienes no utilizan IA tienden a presentar un mayor grado de dificultad a comparación de quienes la utilizan.

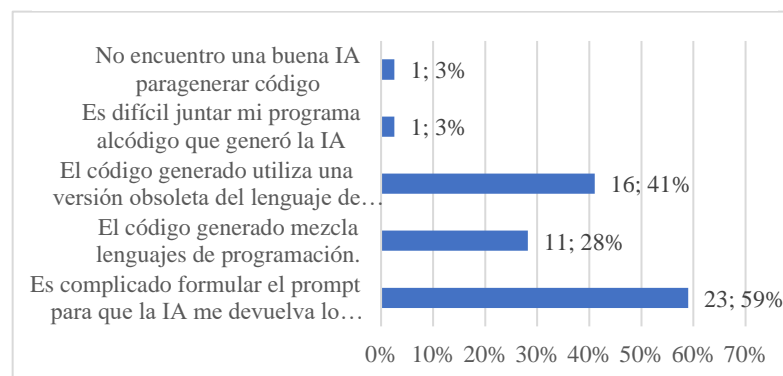
8. ¿Te ha ocurrido alguno de estos problemas al utilizar la inteligencia artificial?

Tabla 10

Opciones	Frecuencia	Porcentaje
Es complicado formular el prompt para que la IA me devuelva lo que busco.	23	59%
El código generado mezcla lenguajes de programación.	11	28%
El código generado utiliza una versión obsoleta del lenguaje de programación.	16	41%
Es difícil juntar mi programa al código que generó la IA	1	3%
No encuentro una buena IA para generar código	1	3%
Total general	52	

Fuente: Encuesta 1 realizada a estudiantes de la carrera de Software
 Autor: Angelo Franco

Figura 7



Fuente: Encuesta 1 realizada a estudiantes de la carrera de Software
 Autor: Angelo Franco

Interpretación:

De los estudiantes que utilizan IA, el 59% tiene problemas para generar un prompt efectivo, el 41% manifiestan que el código generado utiliza una versión obsoleta del lenguaje y el 28% indican que el código generado mezcla lenguajes de programación.

Se concluye que los tres principales problemas están ligados a la complicación de formular prompts, pues la orden es la que genera la respuesta. Esto indica que la habilidad para formular prompts efectivos es crucial para el uso de la IA en el desarrollo del software.

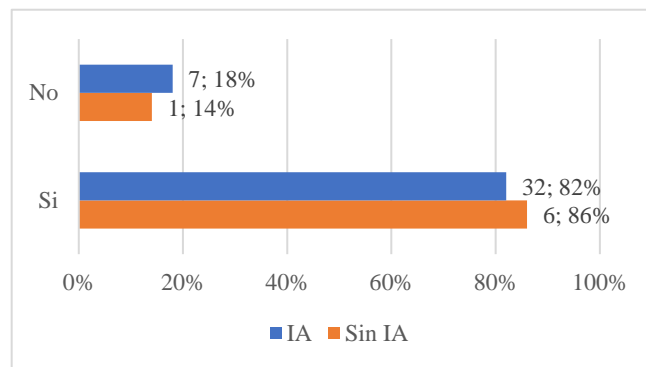
9. ¿Has tenido dificultades en la fase de planificación?

Tabla 11

Opciones	IA		Sin IA	
	Frecuencia	Porcentaje	Frecuencia	Porcentaje
Si	32	82%	6	86%
No	7	18%	1	14%
Total general	39		7	

Fuente: Encuesta 1 realizada a estudiantes de la carrera de Software
Autor: Angelo Franco

Figura 8



Fuente: Encuesta 1 realizada a estudiantes de la carrera de Software
Autor: Angelo Franco

Interpretación:

El 83% de estudiantes manifiestan que sí presentan complicaciones en la fase de planificación, esto incluye el 82% que utilizan IA y 86% que no.

Los resultados muestran que, quienes adoptan la IA en las actividades que comprende esta fase tienen menos complicaciones. El porcentaje no es muy dispar, pero muestra un indicio de que la IA puede tener impacto positivo en esta fase.

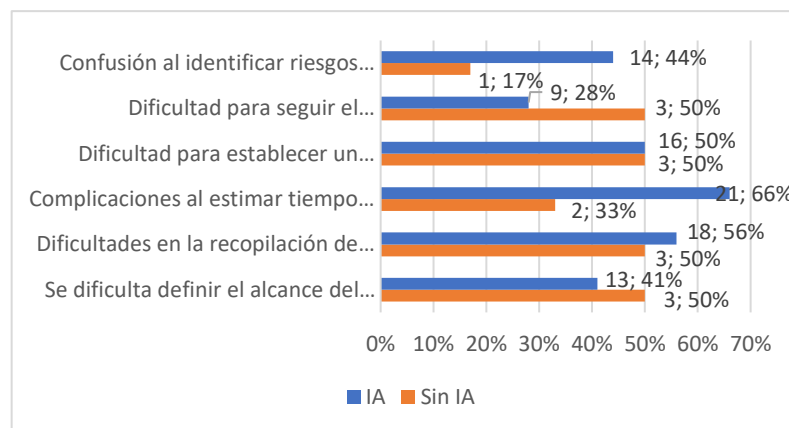
10. ¿Alguna de las siguientes frases describe tu problema?

Tabla 12

Opciones	IA		Sin IA	
	Frecuencia	Porcentaje	Frecuencia	Porcentaje
Se dificulta definir el alcance del proyecto.	13	41%	3	50%
Dificultades en la recopilación de requisitos.	18	56%	3	50%
Complicaciones al estimar tiempo y recursos.	21	66%	2	33%
Dificultad para establecer un cronograma.	16	50%	3	50%
Dificultad para seguir el cronograma.	9	28%	3	50%
Confusión al identificar riesgos del proyecto.	14	44%	1	17%
Total de respuestas	91		15	

Fuente: Encuesta 1 realizada a estudiantes de la carrera de Software
 Autor: Angelo Franco

Figura 9



Fuente: Encuesta 1 realizada a estudiantes de la carrera de Software
 Autor: Angelo Franco

Interpretación:

Los estudiantes que usan IA en sus proyectos a menudo experimentan dificultades para estimar tiempo y recursos, recolectar requisitos y establecer un cronograma el 50-66% de los casos. Los estudiantes que no usan IA tienen problemas más variados, incluyendo definir el alcance del proyecto, recolectar requisitos, establecer y seguir un cronograma (50% de los casos).

Los datos sugieren que la IA puede ayudar a reducir la cantidad de áreas problemáticas.

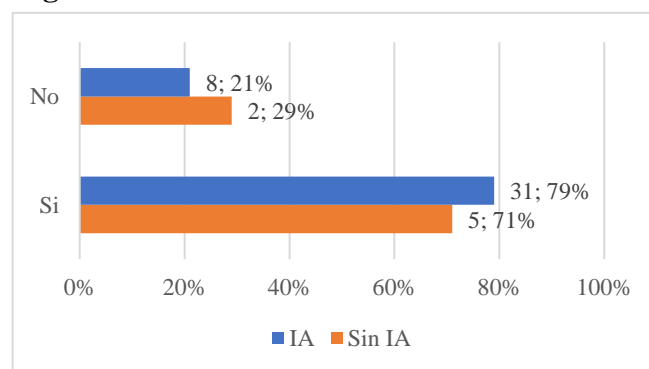
11. ¿Has tenido dificultades en la fase de diseño?

Tabla 13

Opciones	IA		Sin IA	
	Frecuencia	Porcentaje	Frecuencia	Porcentaje
Si	31	79%	5	71%
No	8	21%	2	29%
Total general	39		7	

Fuente: Encuesta 1 realizada a estudiantes de la carrera de Software
Autor: Angelo Franco

Figura 10



Fuente: Encuesta 1 realizada a estudiantes de la carrera de Software
Autor: Angelo Franco

Interpretación:

El 78% de estudiantes manifiestan que sí presentan complicaciones esta fase, incluyendo el 79% que utilizan IA y el 71% que no.

Los resultados muestran que independientemente si utilizan o no IA, enfrentan complicaciones.

12. ¿Alguna de las siguientes frases describe tu problema?

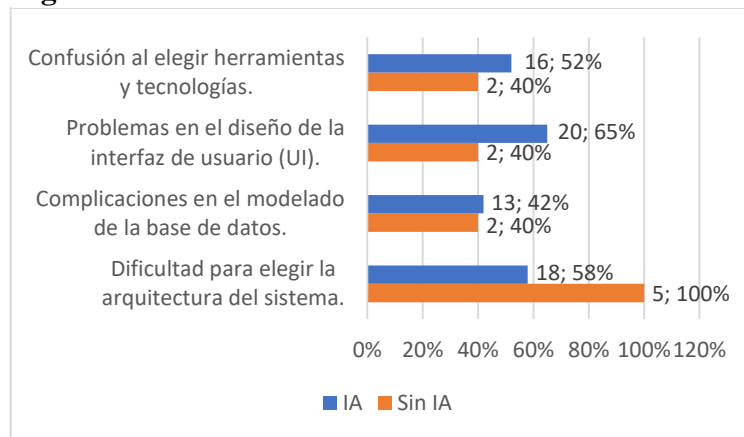
Tabla 14

Opciones	IA		Sin IA	
	Frecuencia	Porcentaje	Frecuencia	Porcentaje
Dificultad para elegir la arquitectura del sistema.	18	58%	5	100%
Complicaciones en el modelado de la base de datos.	13	42%	2	40%
Problemas en el diseño de la interfaz de usuario (UI).	20	65%	2	40%
Confusión al elegir herramientas y tecnologías.	16	52%	2	40%
Total de respuestas	67		11	

Fuente: Encuesta 1 realizada a estudiantes de la carrera de Software

Autor: Angelo Franco

Figura 11



Fuente: Encuesta 1 realizada a estudiantes de la carrera de Software

Autor: Angelo Franco

Interpretación:

Los problemas más frecuentes que presentan los estudiantes que utilizan IA es el diseño de la interfaz de usuario con el 65% y la elección de la arquitectura que tendrá el sistema con el 58%. Mientras el problema más frecuente de los estudiantes que no utilizan IA es la dificultad para elegir la arquitectura del sistema, presente en el 100% de estudiantes.

A partir de los datos se concluye que el desarrollo sin la utilización de IA tiene problemas más focalizados, en contraste al del desarrollo con IA que muestra una dispersión de problemáticas en diferentes actividades.

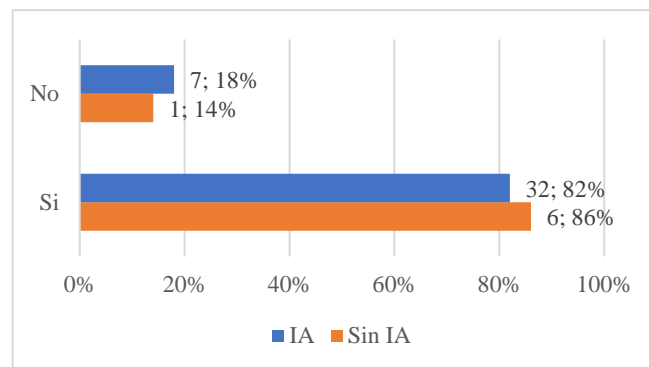
13. ¿Has tenido dificultades en la fase de desarrollo?

Tabla 15

Opciones	IA		Sin IA	
	Frecuencia	Porcentaje	Frecuencia	Porcentaje
Si	32	82%	6	86%
No	7	18%	1	14%
Total general	39		7	

Fuente: Encuesta 1 realizada a estudiantes de la carrera de Software
Autor: Angelo Franco

Figura 12



Fuente: Encuesta 1 realizada a estudiantes de la carrera de Software
Autor: Angelo Franco

Interpretación:

El 83% de estudiantes manifiestan que sí presentan complicaciones en la fase de desarrollo, incluyendo 82% que utilizan IA y 86% que no.

De acuerdo con los resultados se concluye que, independientemente si se utiliza o no IA, se presentan problemas en esta fase. No es algo sorprendente, pues la fase de desarrollo puede ser compleja independientemente de las herramientas que se utilicen.

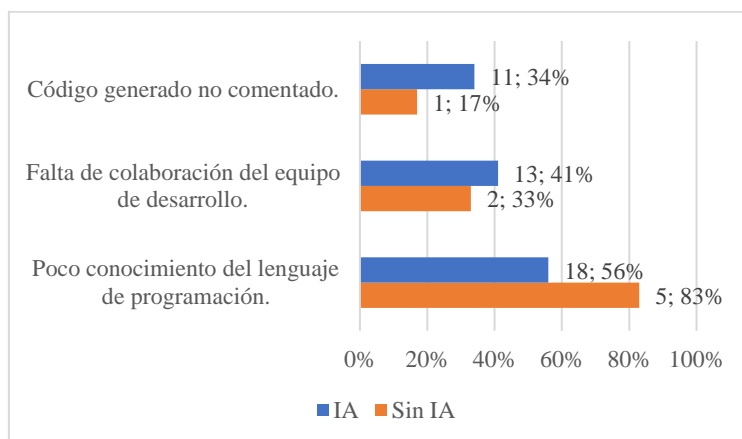
14. ¿Alguna de las siguientes frases describe tu problema?

Tabla 16

Opciones	IA		Sin IA	
	Frecuencia	Porcentaje	Frecuencia	Porcentaje
Poco conocimiento del lenguaje de programación.	18	56%	5	83%
Falta de colaboración del equipo de desarrollo.	13	41%	2	33%
Código generado no comentado.	11	34%	1	17%
Total de respuestas	42		8	

Fuente: Encuesta 1 realizada a estudiantes de la carrera de Software
 Autor: Angelo Franco

Figura 13



Fuente: Encuesta 1 realizada a estudiantes de la carrera de Software
 Autor: Angelo Franco

Interpretación:

Los estudiantes que experimentan problemas en esta fase mencionan que, entre los inconvenientes más comunes, utilizando o no IA, está el poco conocimiento del lenguaje de programación el 56 – 83% respectivamente, la falta de colaboración del equipo el 41 – 33% y el código no es comentado el 34 – 17%.

Los datos muestran que al no utilizar IA el problema es más focalizado en la falta de conocimiento del lenguaje, mientras al utilizar IA los problemas son más dispersos. Esto sugiere que la implementación de la IA puede originar nuevos desafíos. Sin embargo, ayuda ante problemas que pueden surgir a raíz del desconocimiento del lenguaje de programación.

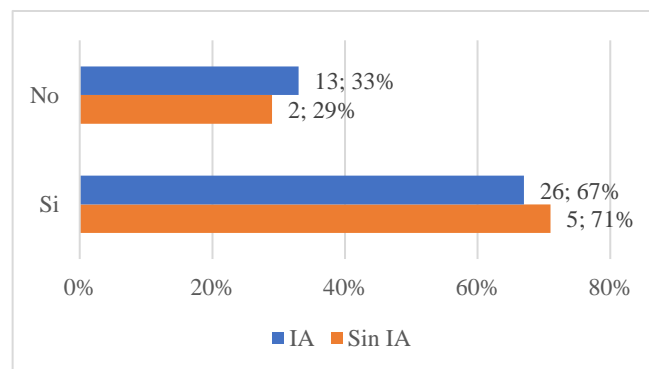
15. ¿Has tenido dificultades en la fase de pruebas?

Tabla 17

Opciones	IA		Sin IA	
	Frecuencia	Porcentaje	Frecuencia	Porcentaje
Si	26	67%	5	71%
No	13	33%	2	29%
Total general	39		7	

Fuente: Encuesta 1 realizada a estudiantes de la carrera de Software
Autor: Angelo Franco

Figura 14



Fuente: Encuesta 1 realizada a estudiantes de la carrera de Software
Autor: Angelo Franco

Interpretación:

El 67% de estudiantes manifiestan que sí presentan complicaciones en la fase de pruebas, incluyendo el 67% que utilizan IA y el 71% que no.

En conclusión, tanto si se utiliza IA o no, surgen dificultades. Esta fase puede resultar compleja pues se prueban todas las partes del sistema y sin una planificación e incorporación de herramientas adecuadas, los problemas aparecen.

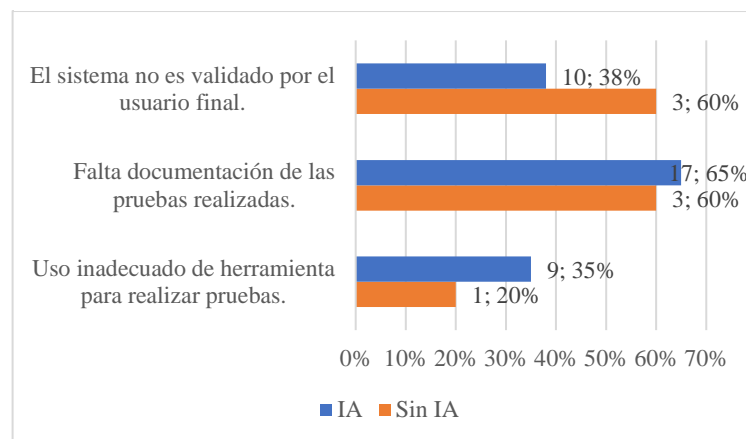
16. ¿Alguna de las siguientes frases describe tu problema?

Tabla 18

Opciones	IA		Sin IA	
	Frecuencia	Porcentaje	Frecuencia	Porcentaje
Uso inadecuado de herramienta para realizar pruebas.	9	35%	1	20%
Falta documentación de las pruebas realizadas.	17	65%	3	60%
El sistema no es validado por el usuario final.	10	38%	3	60%
Total de respuestas	36		7	

Fuente: Encuesta 1 realizada a estudiantes de la carrera de Software
 Autor: Angelo Franco

Figura 15



Fuente: Encuesta 1 realizada a estudiantes de la carrera de Software
 Autor: Angelo Franco

Interpretación:

El 65% de los estudiantes que utilizan IA mencionan que el problema más frecuente es la falta de documentación. El 60% de estudiantes que no utilizan IA indican que los problemas más frecuentes que experimentan es la falta de documentación y la no validación del sistema por parte del usuario final.

Se concluye que la utilización de IA en esta fase presenta menos problemas, siendo en su mayoría la falta de documentación de las pruebas realizadas.

4.1.2 Encuesta para evaluar la utilización del manual y su eficacia.

Encuesta 2

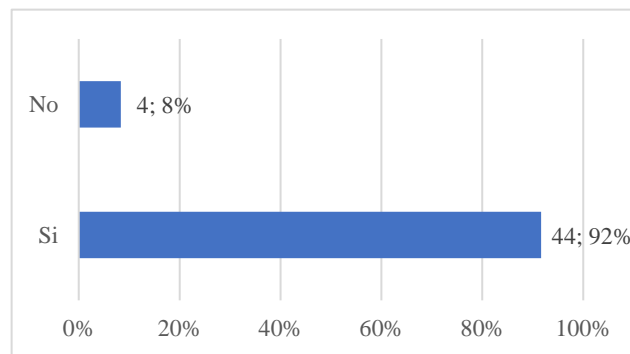
1. En un primer instante, ¿has utilizado inteligencia artificial para desarrollar software?

Tabla 19

Opciones	Frecuencia	Porcentaje
Si	44	92%
No	4	8%
Total general	48	

Fuente: Encuesta 1 realizada a estudiantes de la carrera de Software
Autor: Angelo Franco

Figura 16



Fuente: Encuesta 1 realizada a estudiantes de la carrera de Software
Autor: Angelo Franco

Interpretación:

El 92% de los estudiantes indican que sí utilizan la herramienta, mientras el 8% no. Los datos indican que los estudiantes, en su mayoría, adoptaron a la IA como copiloto en sus proyectos. La gran acogida de esta herramienta se debe a que resulta muy útil para desempeñar tareas.

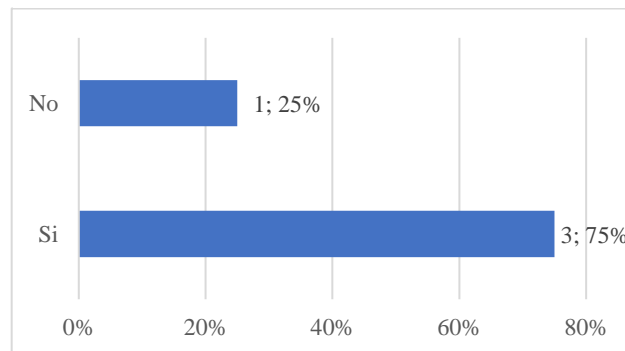
2. ¿Has decidido introducir IA al proceso de desarrollo de tus proyectos?

Tabla 20

Opciones	Frecuencia	Porcentaje
Si	3	75%
No	1	25%
Total general	4	

Fuente: Encuesta 2 realizada a estudiantes de la carrera de Software
Autor: Angelo Franco

Figura 17



Fuente: Encuesta 2 realizada a estudiantes de la carrera de Software
Autor: Angelo Franco

Interpretación:

De los estudiantes que en un primer instante no utilizaba inteligencia artificial como apoyo para desarrollar sus proyectos, el 75% decidió probar la herramienta y el 25% se negó a utilizarla.

De esta forma se muestra que la mayoría acoge de una manera positiva la introducción de nuevas tecnologías.

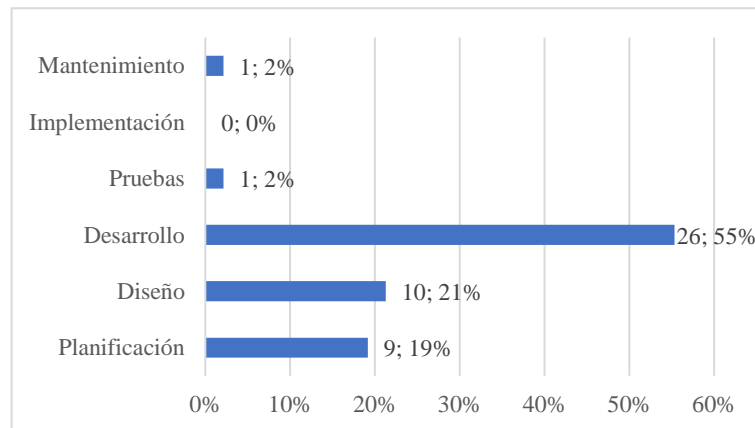
3. Selecciona la fase del proyecto en la que utilizas IA con mayor frecuencia.

Tabla 21

Opciones	Frecuencia	Porcentaje
Planificación	9	19%
Diseño	10	21%
Desarrollo	26	55%
Pruebas	1	2%
Implementación	0	0%
Mantenimiento	1	2%
Total de respuestas	47	

Fuente: Encuesta 2 realizada a estudiantes de la carrera de Software
Autor: Angelo Franco

Figura 18



Fuente: Encuesta 2 realizada a estudiantes de la carrera de Software
Autor: Angelo Franco

Interpretación:

Los estudiantes que decidieron utilizar o ya utilizaban inteligencia artificial para desarrollar sus proyectos, indicaron que las fases más recurrentes donde la utilizan son: desarrollo con el 55%, diseño con el 21%, planificación con el 19% y se manifiesta que ninguno lo utiliza en la implementación.

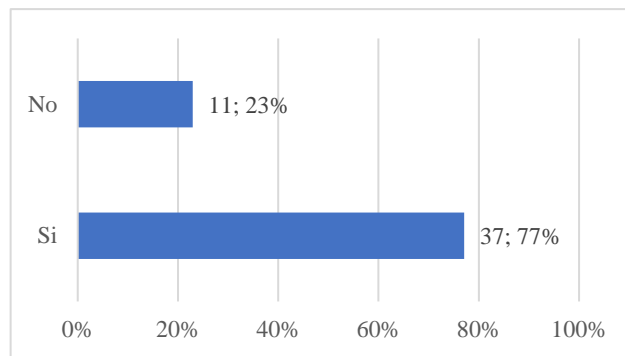
En conclusión, las fases donde más se utiliza IA para realizar tareas son: desarrollo, donde se genera código que sirva como guía o incluso mejorar algoritmos que ya se hayan creado; diseño, donde contribuye a la creación de prototipos en menor tiempo; y planificación, en la cual ayuda a prever problemas potenciales, optimizar recursos y mejorar la toma de decisiones.

4. ¿Aplicas los principios de buenas prácticas al desarrollar tus proyectos?
Tabla 22

Opciones	Frecuencia	Porcentaje
Si	37	77%
No	11	23%
Total general	48	

Fuente: Encuesta 2 realizada a estudiantes de la carrera de Software
Autor: Angelo Franco

Figura 19



Fuente: Encuesta 2 realizada a estudiantes de la carrera de Software
Autor: Angelo Franco

Interpretación:

El 77% de estudiantes indican que sí aplican los principios de buenas prácticas en sus proyectos, el 23% restante no lo hace.

Esto muestra que la mayoría de las estudiantes procuran seguir normas y estándares para asegurar la creación de un software de calidad.

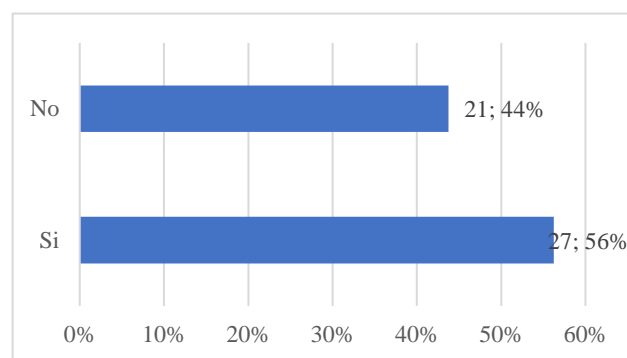
5. ¿Has leído el manual de buenas prácticas para el desarrollo de software apoyado en IA?

Tabla 23

Opciones	Frecuencia	Porcentaje
Si	27	56%
No	21	44%
Total general	48	

Fuente: Encuesta 2 realizada a estudiantes de la carrera de Software
Autor: Angelo Franco

Figura 20



Fuente: Encuesta 2 realizada a estudiantes de la carrera de Software
Autor: Angelo Franco

Interpretación:

El 56% de estudiantes afirman haber leído el manual de buenas prácticas presentado. El 44% restante no lo han leído.

En conclusión, aunque el manual ha sido leído por una mayoría de estudiantes, no ha tenido el recibimiento esperado. Esto sugiere que se debe utilizar otras maneras para promover y presentar el manual, de modo que resulte más atractivo e incite a su revisión.

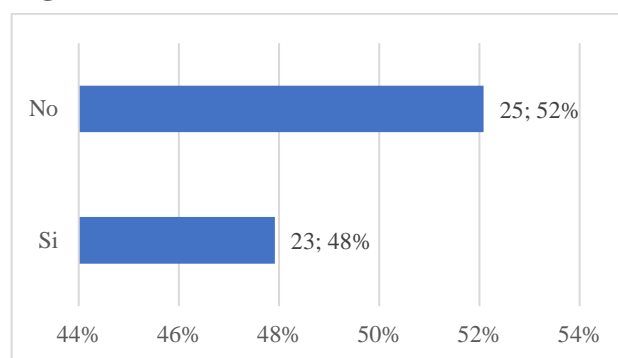
6. ¿Has aplicado las buenas prácticas recomendadas en este manual en tus proyectos?

Tabla 24

Opciones	Frecuencia	Porcentaje
Si	23	48%
No	25	52%
Total general	48	

Fuente: Encuesta 2 realizada a estudiantes de la carrera de Software
Autor: Angelo Franco

Figura 21



Fuente: Encuesta 2 realizada a estudiantes de la carrera de Software
Autor: Angelo Franco

Interpretación:

El 48% de estudiantes manifiestan que han aplicado las prácticas presentes en el manual. El 52%, si bien lo leyeron, no lo han puesto en práctica.

Los datos indican que el manual de buenas prácticas no ha sido aplicado por una gran parte de los estudiantes. Esto da a entender que, aunque las prácticas y propuestas del manual pueden ser valiosas, puede haber desafíos que impiden su adopción y aplicación efectiva.

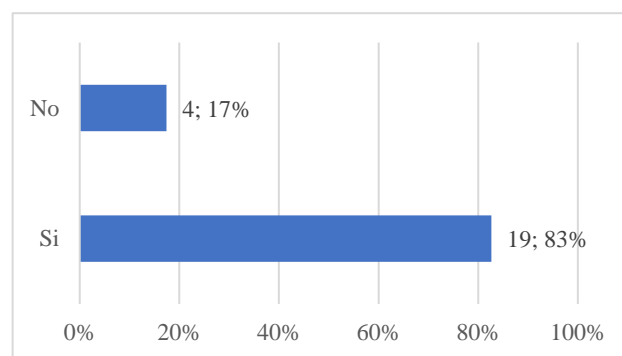
7. ¿Has notado alguna mejora en la eficiencia y calidad de tus proyectos después de aplicar las buenas prácticas del manual?

Tabla 25

Opciones	Frecuencia	Porcentaje
Si	19	83%
No	4	17%
Total general	23	

Fuente: Encuesta 2 realizada a estudiantes de la carrera de Software
Autor: Angelo Franco

Figura 22



Fuente: Encuesta 2 realizada a estudiantes de la carrera de Software
Autor: Angelo Franco

Interpretación:

De los estudiantes que han aplicado las prácticas del manual presentado, el 83% mencionan que han notado una mejora en la eficiencia y calidad de sus proyectos.

El 17% no han notado ninguna mejora.

Los datos muestran que el manual ha sido de utilidad para quienes lo han leído y aplicado. Las prácticas recopiladas en el manual, de ser aplicadas, ayudan al programador a llevar las tareas de una manera más organizada y sencilla.

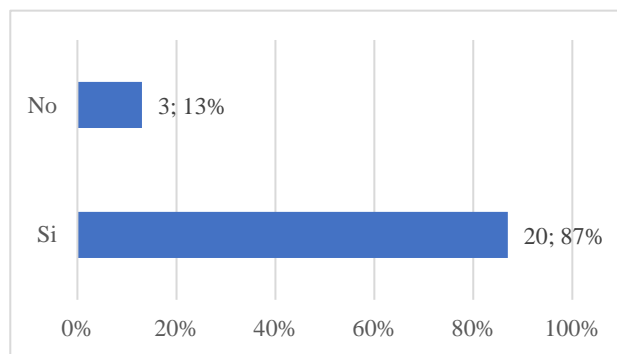
8. ¿Consideras que has comprendido mejor la manera en que la IA puede ayudar en el proceso de desarrollo de software?

Tabla 26

Opciones	Frecuencia	Porcentaje
Si	20	87%
No	3	13%
Total general	23	

Fuente: Encuesta 2 realizada a estudiantes de la carrera de Software
Autor: Angelo Franco

Figura 23



Fuente: Encuesta 2 realizada a estudiantes de la carrera de Software
Autor: Angelo Franco

Interpretación:

El 87% de estudiantes indican que después de aplicar las prácticas presentes en el manual, han entendido cómo la IA puede ayudarlos a desarrollar software. El restante no ha comprendido aún como la herramienta puede ayudarlos.

Se ha logrado que la mayoría de los estudiantes comprendan cómo la IA puede ayudarles en el desarrollo de un proyecto.

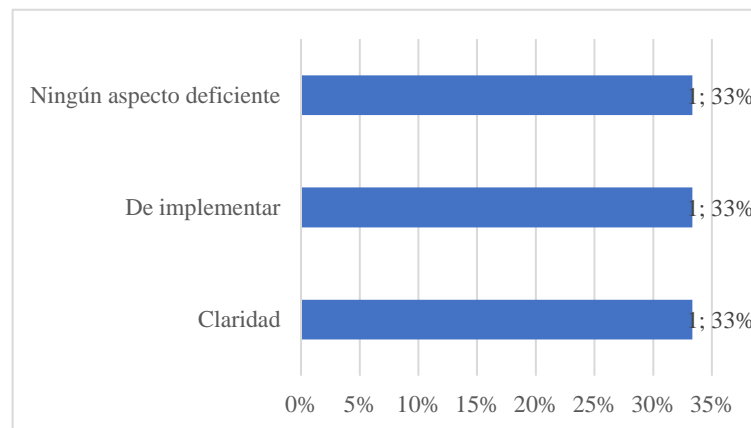
**9. ¿Crees que algún aspecto del manual necesita más claridad o detalle?
(Opcional)**

Tabla 27

Opciones	Frecuencia	Porcentaje
Claridad	1	33%
De implementar	1	33%
Ningún aspecto deficiente	1	33%
Total de respuestas	3	

Fuente: Encuesta 2 realizada a estudiantes de la carrera de Software
Autor: Angelo Franco

Figura 24



Fuente: Encuesta 2 realizada a estudiantes de la carrera de Software
Autor: Angelo Franco

Interpretación:

Algunos estudiantes han dejado su comentario acerca del manual, resaltando aspectos que deberían manejarse de manera más clara o detallada. Las sugerencias se enfocan en que el manual debería ser más claro y fácil de implementar. Mientras uno de los comentarios fue utilizado para declarar que no ha encontrado ningún aspecto deficiente.

Aunque el manual fue redactado de la manera más clara y familiar posible, podría ser mejorado.

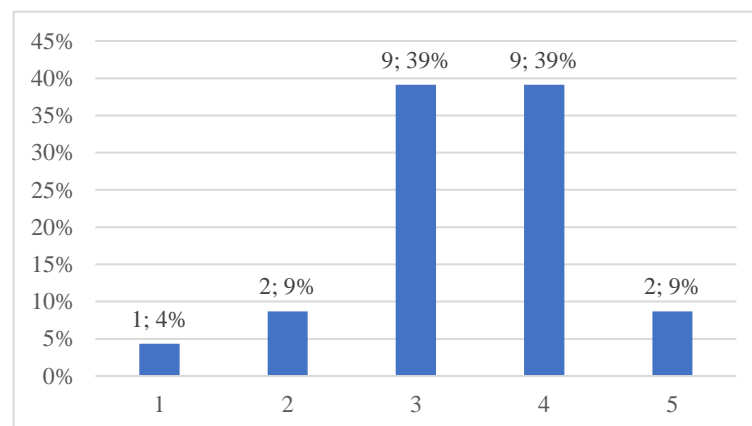
10. Según tu experiencia califica en una escala del 1 al 5 la utilidad del manual.

Tabla 28

Opciones	Frecuencia	Porcentaje
1	1	4%
2	2	9%
3	9	39%
4	9	39%
5	2	9%
Total de respuestas	23	

Fuente: Encuesta 2 realizada a estudiantes de la carrera de Software
Autor: Angelo Franco

Figura 25



Fuente: Encuesta 2 realizada a estudiantes de la carrera de Software
Autor: Angelo Franco

Interpretación:

El 87% de estudiantes puntúa al manual entre bueno y excelente (3-5). El restante lo puntúa entre muy malo y malo (1-2).

La mayoría de estudiantes puntúa el manual con una calificación buena y muy buena, lo que indica que fue de utilidad. Las puntuaciones negativas pueden deberse a que los temas no fueron tratados de la manera esperada, lo que generó rechazo.

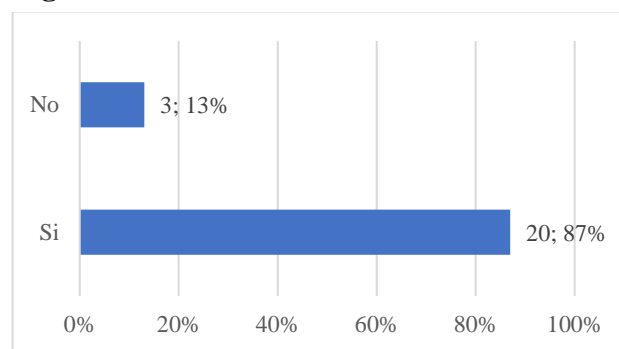
11. ¿Recomendarías este manual a otros desarrolladores que están comenzando a trabajar con IA?

Tabla 29

Opciones	Frecuencia	Porcentaje
Si	20	87%
No	3	13%
Total general	23	

Fuente: Encuesta 2 realizada a estudiantes de la carrera de Software
Autor: Angelo Franco

Figura 26



Fuente: Encuesta 2 realizada a estudiantes de la carrera de Software
Autor: Angelo Franco

Interpretación:

El 87% de estudiantes mencionan que sí recomendarían el manual a quienes están empezando a trabajar con IA, el 17% no.

Gran parte de los estudiantes recomendarían el manual de buenas prácticas, esto se debe a que el manual ha resultado útil y puede ayudar a desarrolladores que empiezan a integrar IA en sus proyectos.

12. ¿Puedes indicar el motivo por el cual decidiste no usar el manual?

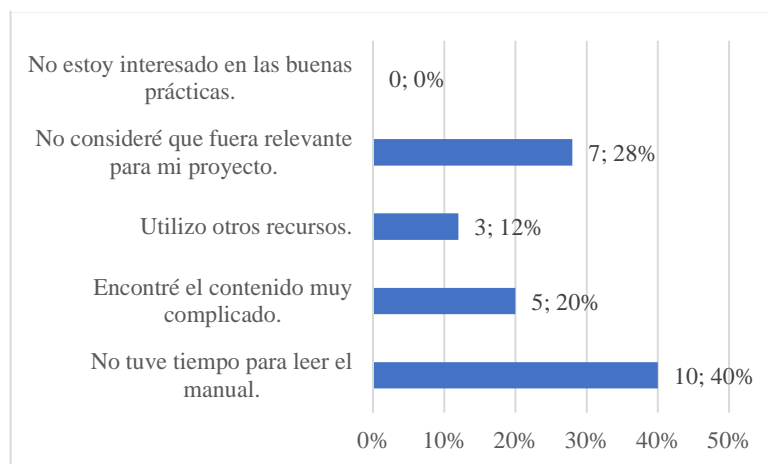
Tabla 30

Opciones	Frecuencia	Porcentaje
No tuve tiempo para leer el manual.	10	40%
Encontré el contenido muy complicado.	5	20%
Utilizo otros recursos.	3	12%
No consideré que fuera relevante para mi proyecto.	7	28%
No estoy interesado en las buenas prácticas.	0	0%
Total de respuestas	25	

Fuente: Encuesta 2 realizada a estudiantes de la carrera de Software

Autor: Angelo Franco

Figura 27



Fuente: Encuesta 2 realizada a estudiantes de la carrera de Software

Autor: Angelo Franco

Interpretación:

Se preguntó a los estudiantes que no utilizaron el manual los motivos por el cual tomaron esa decisión, donde el 40% indica que no tuvieron tiempo para leer el manual, el 28% consideraron que no era relevante para su proyecto, el 20% encontraron el contenido complicado y el 12% utilizan otros recursos.

Los motivos que mencionan los estudiantes evidencian diferentes desafíos y preferencias, que va desde limitaciones de tiempo hasta la utilización de otros recursos. Esto sugiere la necesidad de adaptar y personalizar los recursos de aprendizaje para satisfacer las necesidades y preferencias de los estudiantes.

4.1.3 Discusión de resultados

La primera encuesta reveló una resistencia inicial entre algunos estudiantes de la carrera de software hacia la adopción de la inteligencia artificial en el desarrollo. Se detectaron diversas debilidades que emergen de los desafíos presentes en cada etapa de la creación de un sistema, ya sea en el análisis, planificación, diseño, desarrollo o pruebas. Generalmente, hay un alto porcentaje de estudiantes que presentan dificultades de grado medio.

La segunda encuesta mostró que, a pesar de que la recepción no fue la esperada, aquellos que aplicaron los estándares y técnicas propuestas en el manual observaron una mejora en la eficiencia y calidad de los sistemas que construyeron. Se evidenció que la integración de la IA puede superar dificultades más allá de la generación de código, pudiendo ayudar en cada fase y las tareas que esta comprende. No obstante, para asegurar una adopción exitosa de la IA en el desarrollo de software, es crucial considerar estrategias de implementación efectivas. Al hacerlo, se puede asegurar una recepción más positiva del material de IA entre los estudiantes.

CONCLUSIONES

- A través de técnicas como la encuesta y entrevista se identificaron las debilidades que los estudiantes presentan en las etapas del desarrollo de software, destacando tres aspectos principales:
 1. Un 85% de los estudiantes que utilizan IA presentan dificultades, comparado con el 64% que no utiliza IA.
 2. El grado de dificultad medio es experimentado por el 79% de quienes utilizan IA y el 57% de quienes no la utilizan.
 3. Independientemente si utilizan o no IA, presentan debilidades que en algunos casos son muy dispersas y en otras muy concretas. La tabla adjunta detalla los problemas más comunes:

Tabla 31

Resumen de debilidades identificadas.

Etapa de desarrollo	IA		Sin IA	
	Debilidad	%	Debilidad	%
Planificación	Estimar tiempo y recursos.	66%	Definir el alcance del proyecto.	50%
	Recopilación de requisitos.	56%	Recopilación de requisitos.	50%
	Establecer un cronograma.	50%	Establecer un cronograma.	50%
			Cumplir el cronograma.	50%
Diseño	El diseño de la interfaz de usuario (UI).	65%	Elegir la arquitectura del sistema.	100%
	Elegir la arquitectura del sistema.	58%		
	Elegir herramientas y tecnologías.	52%		
Desarrollo	Poco conocimiento del lenguaje de programación.	56%	Poco conocimiento del lenguaje de programación.	83%

Prueba	Falta documentación de las pruebas realizadas.	65%	Falta documentación de las pruebas realizadas.	60%
			El sistema no es validado por el usuario final.	60%

Nota. En etapas posteriores a las pruebas los estudiantes indican que no utilizan IA, por lo cual no se ven representadas en esta tabla. Sin embargo, el manual ofrece posibles opciones de implementación a estas etapas, implementación y mantenimiento.

- La aplicación de la inteligencia artificial en las diversas fases del desarrollo de software ha potenciado tanto la eficiencia como la calidad del producto final. La IA tiene una relación directa con la calidad del software, ya que tiene la capacidad de procesar y analizar volúmenes masivos de datos, identificando patrones y tendencias que pueden ser aprovechados para mejorar la calidad del software.
- Se ha determinado que la inteligencia artificial contribuye de manera significativa a la creación del software. Esta contribución se manifiesta al ayudar a automatizar tareas repetitivas, generar código, detectar errores, etc. Al permitir un desarrollo rápido y preciso, la IA está revolucionando la industria del software. No obstante, es de vital importancia recordar que la IA es una herramienta que complementa el trabajo de un programador, no lo reemplaza.
- Tras la entrega del manual de buenas prácticas con apoyo en inteligencia artificial, se dio un período de 1 mes para que los estudiantes puedan aplicarlo en sus proyectos. Posteriormente se realizó una encuesta para evaluar la efectividad del manual. Aunque la acogida del manual fue del 48%, el 83% de este grupo notó una mejora en sus proyectos. La comprensión de cómo la IA puede asistir en el desarrollo de software fue clave para estos resultados positivos. Por lo tanto, se puede afirmar que el manual presentado ha sido una solución efectiva. A pesar de ello, la acogida fue menor a la esperada, lo que indica la necesidad de estrategias de implementación más eficaces y un mayor apoyo educativo para un uso efectivo de estas herramientas.

RECOMENDACIONES

- La introducción de tecnologías emergentes en el ámbito de software debe darse de manera gradual para garantizar una transición efectiva. Los recursos utilizados para guiar la implementación deben mantenerse a la par de la evolución de las herramientas, pues de este modo la formación será continua, actualizada y permitirá mantenerse al día ante los cambios rápidos en el panorama tecnológico.
- Es fundamental evaluar continuamente el desempeño de los estudiantes, ya que la adopción de nuevas herramientas generará nuevos desafíos a los que se deberá dar solución.
- Dado que la implementación efectiva de la inteligencia artificial requiere un alto nivel de competencia técnica y una comprensión de los principios de la IA, es importante no depender únicamente de la educación institucional como fuente de conocimiento. La experimentación autónoma y la observación son complementos que enriquecen y expanden el aprendizaje.
- En relación al manual propuesto, es importante recalcar que aquellos que no han utilizado inteligencia artificial en un comienzo, a menudo tienen menos problemas al desarrollar software debido a su amplio conocimiento. Sin embargo, la implementación de la IA puede minimizar aún más estos problemas y acelerar el ritmo con el cual se desarrolla el proyecto.

BIBLIOGRAFÍA

- Acosta Bravo, E. A. (2019). *Documentación y adopción de buenas prácticas para el desarrollo de software basado en los fundamentos DevOps I*. [Documento de trabajo de grado].
https://bibliotecadigital.udea.edu.co/bitstream/10495/16742/1/AcostaEdwin_2019_PracticasDesarrolloSoftware.pdf
- Agudelo, N., Tano, G., & Vargas, C. (2020). *Historia de la Automatización*. Universidad ECCI.
[https://ingenierovizcaino.com/ecci/aut1/corte1/articulos/Historia de la Automatizacion.pdf](https://ingenierovizcaino.com/ecci/aut1/corte1/articulos/Historia%20de%20la%20Automatizacion.pdf)
- Amazon Web Services (AWS). (s.f.) *¿Qué es el ciclo de vida del desarrollo de software (SDLC)?*. <https://aws.amazon.com/es/what-is/sdlc/>
- AXELOS. (2019). *ITIL Foundation*, ITIL 4 Edition. Axelos
- Bodemer, O. (2023). *Revolutionizing Software Development: Harnessing the Power of Artificial Intelligence for Next-Generation Coding Solutions*. TechRxiv. DOI: 10.36227/techrxiv.24592335.v1
- Caballero Fernández, G., García Vázquez, J., & Quintás Corredoira, M. (2007). *La importancia de los stakeholders de la organización: un análisis empírico aplicado a la empleabilidad del alumnado de la universidad española*. *Investigaciones Europeas de Dirección y Economía de la Empresa*, 13(2), 13-32.
- Chávez Arellano, R., Pineda Domínguez, D. & Cuadrado Gallego, J. (2014). *Eficiencia de proyectos de desarrollo de software y modelos de conversión de funcionalidad*. *Investigación administrativa*, 43(113), 45-59.
- Cruz Pérez, M. A., Pozo Vinuesa, M. A., Andino Jaramillo, A. F., & Arias Parra, A. D. (2018). *Las Tecnologías de la Información y la Comunicación (TIC) como forma investigativa interdisciplinaria con un enfoque intercultural para el proceso de formación de los estudiantes*. *E-Ciencias De La Información*, 9(1). <https://doi.org/10.15517/eci.v1i1.33052>
- García Sandoval, M. G., Ariza Torrado, H. D., Pinzón, M. L., & Flórez Fuentes, A. S. (2015). *La implementación de buenas prácticas para el desarrollo de*

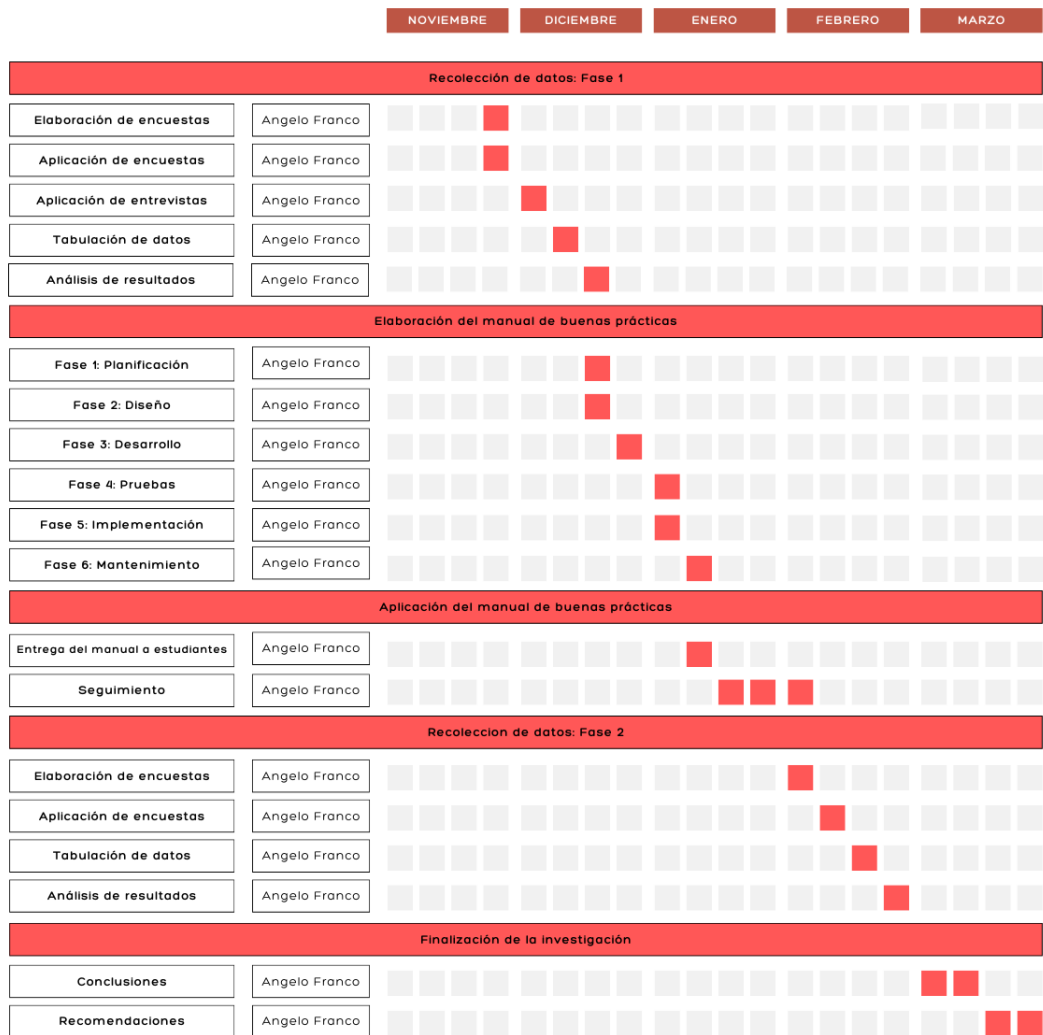
- software colaborativo de aplicativos web*. Mundo FESC, (10). Recuperado de <https://dialnet.unirioja.es/servlet/articulo?codigo=5351802>
- GitLab. (2023). *The State of AI in Software Development*. TSoft Global. <https://www.tsoftglobal.com/wp-content/uploads/2023/11/GitLab-STATE-OF-AI.pdf>
- IBM. (2023). *¿Qué es la Inteligencia artificial (IA)?*. <https://www.ibm.com/es-es/topics/artificial-intelligence>
- ISACA (2019). *COBIT*. <https://www.isaca.org/resources/cobit>
- ISO/IEC/IEEE. (2017). *Systems and software engineering – Vocabulary (ISO/IEC/IEEE 24765:2017)*. International Organization for Standardization.
- Mitra, U. (2014). *Best Practices in Programming*. The University of Texas at Austin. <https://www.cs.utexas.edu/~mitra/csSummer2014/cs312/lectures/bestPractices.html>
- Ocaña, F., Valenzuela, L. & Garro, L. (2019). *Inteligencia artificial y sus implicaciones en la educación superior*. Propósitos y Representaciones, 7(2), 536-568. <https://dx.doi.org/10.20511/pyr2019.v7n2.274>
- Popper, K. (2008). *La lógica de la investigación científica*. Segunda Edición. Tecnos.
- Pressman, R. (2010). *Ingeniería del Software. Un enfoque práctico*. (7^a ed.). España: Ed: McGraw-Hill Interamericana.
- Ruohonen, L.P. (2018). *Inteligencia Artificial: 101 cosas que debes saber hoy sobre nuestro futuro*. Editorial Planeta.
- Sarco Calapiña W. M., Toaquiza Pacheco D. V. (2022). *Aplicación de la Inteligencia Artificial en el Desarrollo de Software*. [Tesis previa a la obtención del Título de Ingeniería en Informática y Sistemas Computacionales]. Universidad Técnica de Cotopaxi. <http://repositorio.utc.edu.ec/handle/27000/9182>

- Soraluz, A., Valles, M., & Lévano, D. (2021). *Desarrollo guiado por comportamiento: buenas prácticas para la calidad de software*. *Ingeniería y Desarrollo*, 39(1), 190-204. <https://doi.org/10.14482/inde.39.1.005.3>
- Stack Overflow. (2023). *Developer sentiment around AI/ML*. Stack Overflow Labs. <https://stackoverflow.co/labs/developer-sentiment-ai-ml/#h2-f23f00df4dec0>
- Vergadia P. (2023). *AI in software development: What you need to know*. *Google Cloud*. <https://cloud.google.com/blog/products/ai-machine-learning/how-ai-impacts-software-development>

Anexo I

Cronograma (Gantt)

Cronograma del proyecto



Anexo II

Presupuesto Ejecutado

Recurso	Descripción	Cantidad	Precio unitario	Subtotal
Laptop	Lenovo IdeaPad 3i core i5 11300H - NVIDIA GeForce 3050	1	1000 \$	1000 \$
Licencia de Office	Licencia de Office 2021 LTSC	1	50.00 \$	50 \$
Arriendo	Departamento con servicios básicos incluido	5 meses	80 \$	400 \$
Impresiones	Certificados, documentos, encuestas	-	0.05 \$	50 \$
			Total	1500 \$

Anexo III

Encuesta 1

Identificación de debilidades en el desarrollo de aplicaciones

En este breve cuestionario, queremos conocer tus experiencias y desafíos al desarrollar aplicaciones. ¡Tus respuestas nos ayudarán a entender mejor cómo podemos apoyarte en tu camino hacia la excelencia en el mundo del software!



1. ¿Utilizas inteligencia artificial?

Si

No

2. ¿Has utilizado la inteligencia artificial para desarrollar software (en cualquiera de sus fases)?

Si

No

3. Indica el motivo por el cual no utiliza inteligencia artificial al desarrollar software.

Desconocimiento de esta posibilidad.

No lo veo necesario.

No se crear prompts para que la IA me entregue lo que quiero.

Otra: _____

4. ¿Estarías dispuesto a utilizar la inteligencia artificial en tus proyectos de desarrollo?

Si

No

5. ¿Has tenido dificultades al desarrollar software con inteligencia artificial?

Si

No

6. ¿Has tenido dificultades al desarrollar software?

Si

No

7. Indica cuál es el grado de la dificultad.

Bajo

Medio

Alto

8. ¿Te ha ocurrido alguno de estos problemas al utilizar la inteligencia artificial?

Es complicado formular el prompt para que la IA me devuelva lo que busco.

El código generado mezcla lenguajes de programación.

El código generado utiliza una versión obsoleta del lenguaje de programación.

Otra: _____

Identificación de las debilidades por fases.

Por favor, comparta sus observaciones y experiencias. Esto nos ayudará a entender mejor las áreas en las que los estudiantes pueden necesitar más apoyo.
¡Gracias por su participación! 🙏

9. ¿Has tenido dificultades en la fase de planificación?

Si

No

10. ¿Alguna de las siguientes frases describe tu problema?

Se dificulta definir el alcance del proyecto.

Dificultades en la recopilación de requisitos.

Complicaciones al estimar tiempo y recursos.

Dificultad para establecer un cronograma.

Dificultad para seguir el cronograma.

Confusión al identificar riesgos del proyecto.

Otra: _____

11. ¿Has tenido dificultades en la fase de diseño?

Si

No

12. ¿Alguna de las siguientes frases describe tu problema?

Dificultad para elegir la arquitectura del sistema.

Complicaciones en el modelado de la base de datos.

Problemas en el diseño de la interfaz de usuario (UI).

Confusión al elegir herramientas y tecnologías.

Otra: _____

13. ¿Has tenido dificultades en la fase de desarrollo?

Si

No

14. ¿Alguna de las siguientes frases describe tu problema?

Poco conocimiento del lenguaje de programación.

Falta de colaboración del equipo de desarrollo.

Código generado no comentado.

Otra: _____

15. ¿Has tenido dificultades en la fase de pruebas?

Si

No

16. ¿Alguna de las siguientes frases describe tu problema?

Uso inadecuado de herramienta para realizar pruebas.

Falta documentación de las pruebas realizadas.

El sistema no es validado por el usuario final.

Otra: _____

17. ¿Has tenido dificultades en la fase de implementación?

Si

No

18. ¿Alguna de las siguientes frases describe tu problema?

Complicaciones al instalar el software.

Dificultades en la migración de las bases de datos.

Otra: _____

19. ¿Has tenido dificultades en la fase de mantenimiento?

Si

No

20. ¿Alguna de las siguientes frases describe tu problema?

Descubrimiento de inconsistencias.

Limitaciones de tiempo o personal para correcciones y mejoras.

Aumento de la complejidad del código tras corregir errores.

Otra: _____

Anexo IV

Encuesta 2

Utilización del Manual y su Eficacia

¡Hola, genios del código! 🚀 ¿Has estado jugando con la IA en tus proyectos de software? ¿Has estado utilizando nuestro manual de buenas prácticas como tu guía de viaje en este emocionante viaje de desarrollo de software? ¡Genial! Queremos saber todo sobre ello.

1. En un primer instante, ¿has utilizado inteligencia artificial para desarrollar software?

Si

No

2. ¿Has decidido introducir IA al proceso de desarrollo de tus proyectos?

Si

No

3. Selecciona la fase del proyecto en la que utilizas IA con mayor frecuencia.

Planificación

Diseño

Desarrollo

Pruebas

Implementación

Mantenimiento

4. ¿Aplicas los principios de buenas prácticas al desarrollar tus proyectos?

Si

No

5. ¿Has leído el manual de buenas prácticas para el desarrollo de software apoyado en IA?

Si

No

6. ¿Has aplicado las buenas prácticas recomendadas en este manual en tus proyectos?

Si

No

7. ¿Has notado alguna mejora en la eficiencia o calidad de tus proyectos después de aplicar las buenas prácticas del manual?

Si

No

8. ¿Consideras que has comprendido mejor la manera en que la IA puede ayudar en el proceso de desarrollo de software?

Si

No

9. ¿Crees que algún aspecto del manual necesita más claridad o detalle? (Opcional)

10. Según tu experiencia califica en una escala del 1 al 5 la utilidad del manual.

Poco útil 1 2 3 4 5 Muy útil

11. ¿Recomendarías este manual a otros desarrolladores que están comenzando a trabajar con IA?

Si

No

12. ¿Puedes indicar el motivo por el cual decidiste no usar el manual?

No tuve tiempo para leer el manual.

Encontré el contenido muy complicado.

Utilizo otros recursos.

No consideré que fuera relevante para mi proyecto.

No estoy interesado en las buenas prácticas.

Otra: _____

Anexo V

Entrevista (guion)

Introducción

- Saludo: “Hola, mi nombre es Angelo y soy estudiante del octavo ciclo de Software. Gracias por tomarte el tiempo para ayudarme”
- Propósito: “Esta conversación tiene como objetivo conocer si nosotros, los estudiantes, nos apoyamos en herramientas de IA para desarrollar software, los desafíos a los que nos enfrentamos y nuestras experiencias, por supuesto.”

Preguntas de la entrevista

- Utilización de herramienta: “¿Utilizas alguna herramienta de IA? Por ejemplo, Chat GPT, Bing Chat, Cloud u otro”

****Preguntas si utilizan herramientas de IA****

- Área focalizada: “¿Para qué la utilizas?”, “¿Dime cuál la fase en que más utilizas la herramienta de IA?”
- Aplicación: “En los proyectos que trabajas ahora ¿los estás realizando con ayuda de la herramienta de IA?”

****Preguntas si no utilizan herramientas de IA****

- Motivo: “¿Por qué no utilizas alguna de estas herramientas?”
- Adopción: “¿Estarías dispuesto a utilizarlas si su adopción resultara en un tiempo de desarrollo más corto y la automatización de tareas tediosas?”
- Área focalizada: “¿Qué herramientas si utilizas? Por ejemplo, algún framework, software externo, tecnología u otra cosa.”

****Preguntas generales****

- Desafíos: “¿Has encontrado alguna complicación al desarrollar aplicaciones de esta manera?”, “¿Cuáles son estos inconvenientes?”
- Experiencia: “Puedes compartir conmigo alguna experiencia”

Preguntas del entrevistado

- “¿Tienes alguna duda acerca del propósito o finalidad de esta entrevista?”

Cierre

- Agradecimiento: “Gracias por tu tiempo. Ha sido muy interesante conocer cómo desarrolla software otra persona.”

Anexo VI

Fotografías

1. Realización de la primera encuesta.



2. Entrega del manual de buenas prácticas.



3. Realización de la segunda encuesta.



4. Reunión con el director y pares académicos.



Anexo VII

Certificado Antiplagio

ING. HENRY ALBÁN EN CALIDAD DE DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

CERTIFICA

Que el trabajo de integración curricular denominado “MANUAL DE BUENAS PRÁCTICAS PARA EL DESARROLLO DE SOFTWARE APOYADO EN INTELIGENCIA ARTIFICIAL PARA ESTUDIANTES DE LA CARRERA DE SOFTWARE DE LA UNIVERSIDAD ESTATAL DE BOLÍVAR, PERÍODO 2023-2024” , presentado por FRANCO ORTEGA ANGELO OCTAVIO estudiante de la carrera de SOFTWARE, pasó el análisis de coincidencia no accidental en la herramienta TURNITIN, reflejando un porcentaje de similitud del 6%, como se puede evidenciar en el documento adjunto.

Guaranda, 04 de abril del 2024

Atentamente,



EDGAR HENRY ALBÁN
YANEZ

Ing. Henry Albán
Director

NOMBRE DEL TRABAJO

Tesis V2.pdf

AUTOR

Angelo Franco

RECUENTO DE PALABRAS

17082 Words

RECUENTO DE CARACTERES

102156 Characters

RECUENTO DE PÁGINAS

113 Pages

TAMAÑO DEL ARCHIVO

6.5MB

FECHA DE ENTREGA

Apr 3, 2024 4:56 PM GMT-5

FECHA DEL INFORME

Apr 3, 2024 4:58 PM GMT-5**● 6% de similitud general**

El total combinado de todas las coincidencias, incluidas las fuentes superpuestas, para cada base de datos.

- 6% Base de datos de Internet
- 1% Base de datos de publicaciones

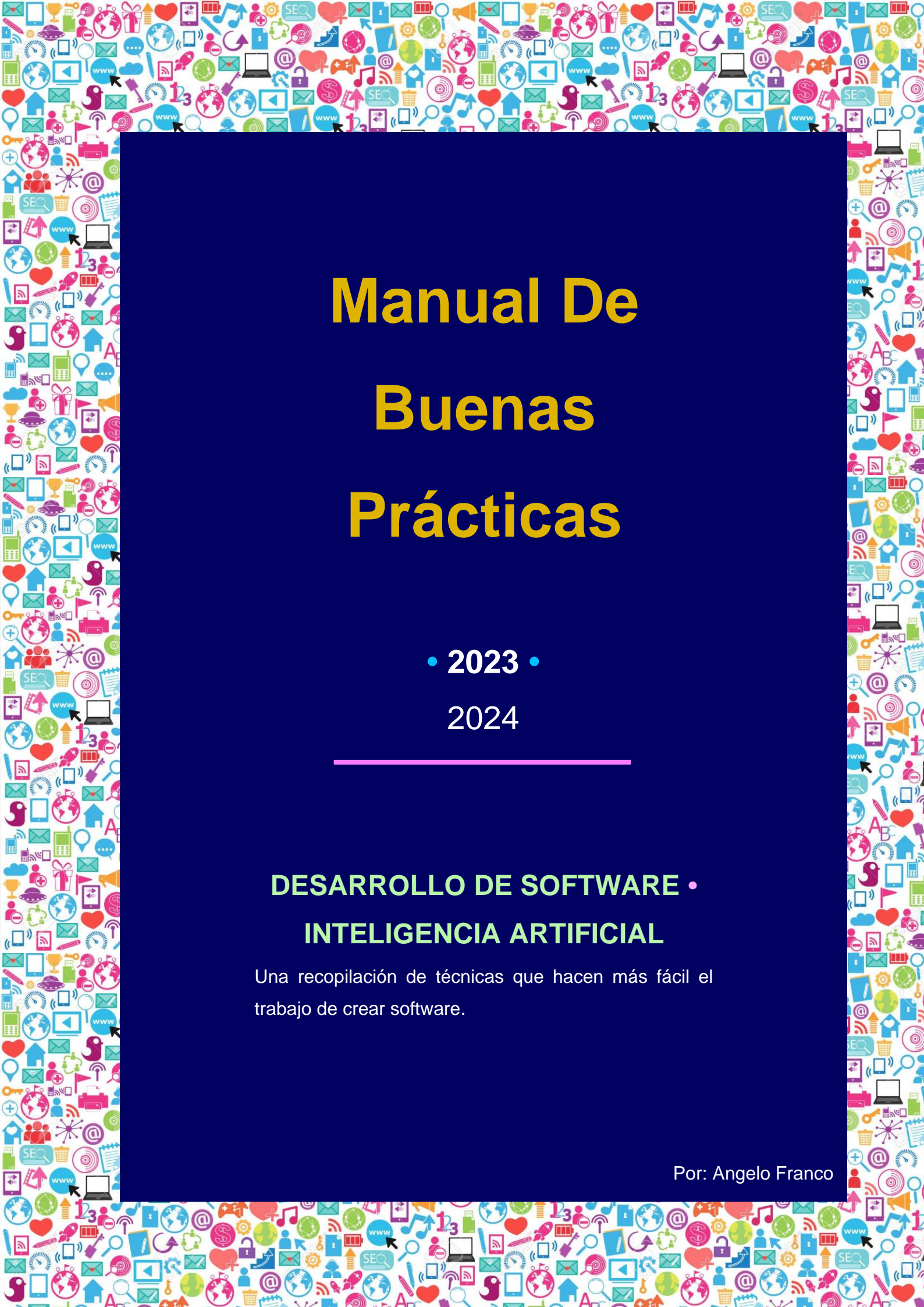
● Excluir del Reporte de Similitud

- Base de datos de Crossref
- Base de datos de contenido publicado de Crossref
- Base de datos de trabajos entregados
- Material bibliográfico
- Material citado
- Material citado
- Coincidencia baja (menos de 8 palabras)
- Fuentes excluidas manualmente



Anexo VIII

Manual



Manual De Buenas Prácticas

• 2023 •
2024

**DESARROLLO DE SOFTWARE •
INTELIGENCIA ARTIFICIAL**

Una recopilación de técnicas que hacen más fácil el
trabajo de crear software.

Por: Angelo Franco

Contenido

Introducción	1
Objetivos	1
Formulación de prompts	2
¿Qué es un prompt?	2
Recomendaciones para crear un prompt de calidad	2
Mejores IA para desarrollo de Software	4
Fases del desarrollo de Software.....	5
Fase de planificación.....	6
Levantamiento de requisitos	6
Documentar los requisitos	6
Identificar los riesgos	7
Estimaciones más acertadas	7
Creación de cronogramas.....	8
Fase de diseño.....	8
Modelo de desarrollo	8
Arquitectura del sistema	10
Diseño de interfaz de usuario	11
Modelado de base de datos.....	12
Tecnologías y herramientas.....	12
Fase de desarrollo	14
Principios para una codificación eficiente y sostenible	14
No todo es programar	15
Comunicación efectiva.....	15
Subdivide el sistema.....	15
Las sugerencias son eso, sugerencias	15
Documentación de código	16
Fase de pruebas	16
Planifica las pruebas.....	16
Pruebas unitarias.....	16
Pruebas de integración.....	17
Validación por parte del usuario.....	17
Fase de implementación	17
Planificación del despliegue.....	17
Despliegue del Software.....	18



Capacitación.....	18
Monitorización	19
Fase de mantenimiento.....	19
Atiende a las sugerencias.....	19
Documenta los cambios.....	19
Pruebas aisladas	20
Pueden ocurrir problemas.....	20
Despliegue inteligente	20
Consideraciones.....	21
Bibliografía.....	21



Introducción

Actualmente la inteligencia artificial (IA) se ha convertido en una herramienta muy utilizada en el ámbito del desarrollo de software. El presente manual de buenas prácticas trata de recopilar técnicas que el programador puede aplicar con la finalidad de facilitar la realización de tareas involucradas en el desarrollo de un sistema.

Se ha prestado especial atención en las debilidades identificadas en estudiantes de la carrera de software de la Universidad Estatal de Bolívar, con el propósito de brindar una solución que ataque directamente a sus inconvenientes. El manual abarca desde la formulación de “prompts”, hasta las fases del desarrollo de software, es decir, la planificación, diseño, desarrollo, pruebas, implementación y mantenimiento.

Objetivos

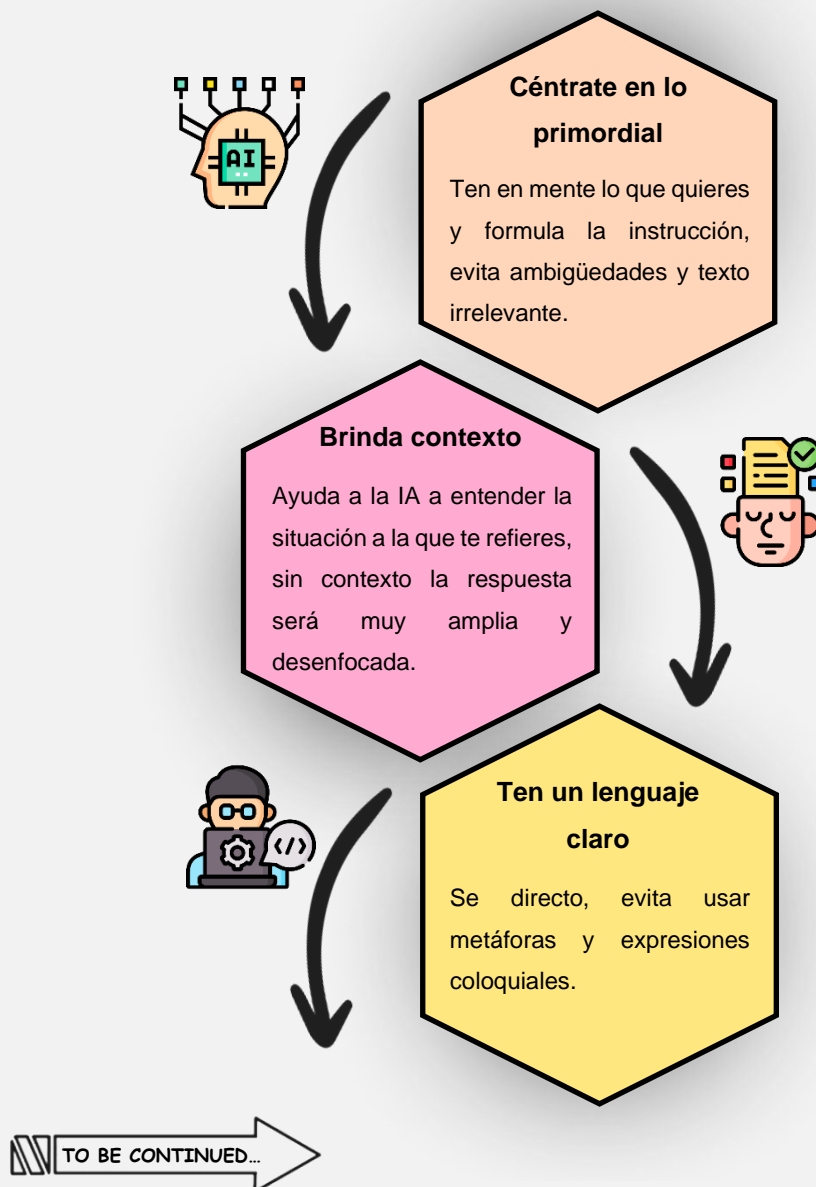
- Recopilar una serie de pautas y recomendaciones que guíen al estudiante en la construcción de un software eficiente y de calidad.
- Fomentar la adopción de buenas prácticas.
- Promover la integración de la inteligencia artificial en el proceso de desarrollo de software.
- Mostrar maneras de automatizar tareas con inteligencia artificial.

Formulación de prompts

¿Qué es un prompt?

Un prompt es una serie de palabras que inicia la creación de contenido mediante inteligencia artificial. Saber crear instrucciones es la clave para obtener el resultado que se espera. Por ello, resulta importante conocer las recomendaciones para la creación de las mismas.

Recomendaciones para crear un prompt de calidad





Establece condiciones
Indica, por ejemplo, el lenguaje, versión, modelo de programación, etc. Esto te dará una respuesta más acertada.



Longitud
Si bien un prompt corto puede ser muy ambiguo, uno largo puede exceder de información y generar respuestas confusas.

Una petición a la vez
"Si una duda tú tienes, separar el problema tu debes." 🙄 Tan solo separa tus preguntas en varios prompts.



Reitera el prompt
Si la respuesta no es lo que esperas, reformula el prompt o brinda más detalles para generar una respuesta más precisa.



Mejores IA para desarrollo de Software

Existen varias herramientas de IA que pueden acompañar al programador en la construcción de un sistema, apoyándolo en cada una de las fases que este supone. Las herramientas utilizadas por excelencia son:

Mejores “copilotos” para desarrollar software

Free

Bing

Asistente de IA enfocado en un entorno más general. Puede responder inquietudes de cualquier índole, generar imágenes, generar código y corregir errores.



Amazon CodeWhisperer

Herramienta que permite a los programadores crear una aplicación de manera rápida y segura. Puede sugerir código, seguimiento de referencias y análisis de seguridad.

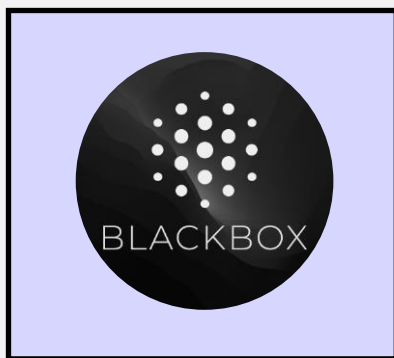
Free

Free

Android Studio Bot

Es un asistente de codificación integrado en Android Studio. Es capaz de generar código, corregir errores y contestar respuestas a preguntas relacionadas con el desarrollo móvil.





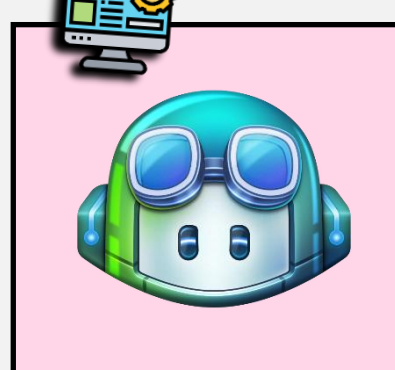
Blackbox Free

Herramienta que realiza un seguimiento al código del programador. Cuenta con un chat para responder inquietudes sobre programación, sugiere código en la marcha y optimiza la gestión de cambios.

\$

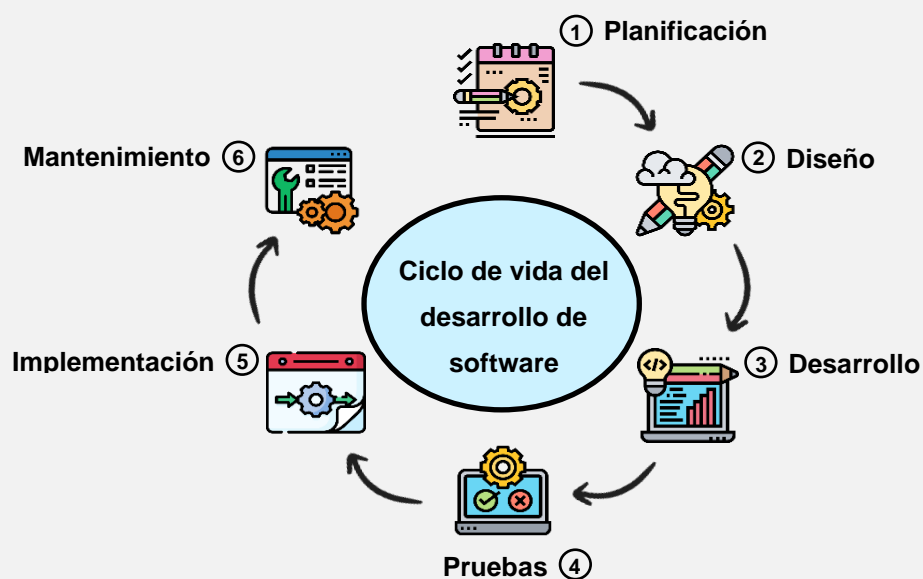
GitHub Copilot

Herramienta destinada específicamente para programadores. Sugiere código y funciones completas, además transforma lenguaje natural en sugerencias de programación en docenas de lenguajes.



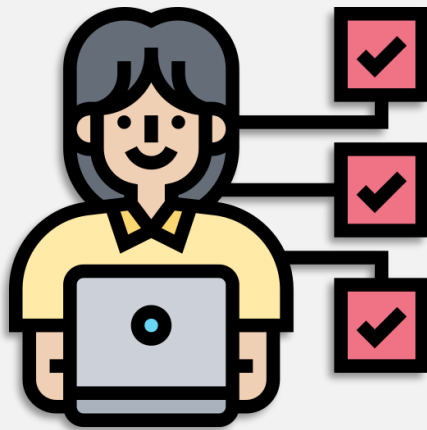
Fases del desarrollo de Software

El programador se enfrenta a diferentes desafíos al construir software y la inteligencia artificial puede ser un apoyo para la realización de estas actividades. A continuación, se presentará una serie de técnicas para reducir los problemas identificados:



Fase de planificación

Es la base sobre el cual se empezará a construir el sistema, por lo mismo debe ser llevado a cabo con rigurosidad. Las maneras de integrar la inteligencia artificial y poder asegurar la correctitud de lo planificado concierne:



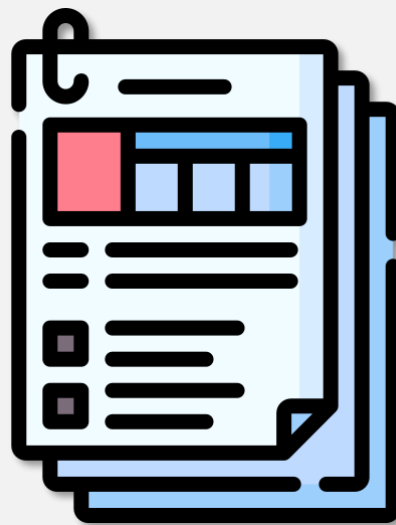
Levantamiento de requisitos

La IA fue entrenada con millones de datos e incluso algunas tienen acceso a todo el contenido de internet. Por lo cual preguntar los requisitos para el sistema que está por desarrollar puede brindar una visión general de las funcionalidades que sistemas similares al tuyo contienen.

NOTA: Esto no reemplazará la colaboración que deberás tener con el cliente. Puede tener más o menos requisitos y es tu obligación atender a ellos.

Documentar los requisitos

Llevar constancia de los requisitos que van surgiendo es primordial, recuerda que deben ser validados por el cliente. Documentar los requisitos debe convertirse en un hábito, pues evitas tener inconvenientes en la etapa final, teniendo evidencia a cerca de los requisitos acordados.



NOTA: Los clientes nunca te dirán sus requerimientos en una sola reunión, debes trabajar estrecha y constantemente con él hasta que se sienta satisfecho.



Identificar los riesgos

La IA te puede proporcionar los riesgos potenciales de proyectos similares al tuyo, incluyendo cuestiones técnicas, de recursos, presupuestos y plazos. Analízalos e identifica los que pueden ocurrir en tu proyecto.

NOTA: Intenta dar soluciones tanto proactivas como reactivas, es esencial tener un plan de contingencia puesto que un riesgo puede ocurrir en cualquier momento.

Estimaciones más acertadas

Estimar costos significa la identificación de diferentes variables y su costo. La inteligencia artificial puede establecer dichas variables de manera más efectiva.



NOTA: El encargado de realizar la estimación tan solo necesitará establecer los costos de cada variable.



Creación de cronogramas

Mediante datos históricos, la IA puede ayudarte en la creación del cronograma de tu proyecto. Brinda el plazo máximo en semanas o meses, las funcionalidades a desarrollar y más contexto para que genere un cronograma, el cual deberás analizar y adaptarlo si fuera necesario.

Fase de diseño

Se toma las especificaciones recopiladas en la fase anterior y revisan diversos temas:

Modelo de desarrollo

Sommerville define al modelo de software como una abstracción de un proceso real, es decir, un modelo representa y describe las etapas, actividades y tareas que se realizan para desarrollar software.

Existen diferentes modelos, entre los más conocidos están:

- De prototipo (A)
- Basado en componentes (NA)
- En espiral(A)
- RAD(A)
- Cascada (NA)

A: Ágil
NA: No ágil

Lo que se debe considerar para la elección de un modelo es:

Requisitos del cliente

¿Son claros y estables, o son cambiantes?

Modelos ágiles se adaptan mejor a entornos flexibles, mientras modelos no ágiles requieren especificaciones sólidas.



Tiempo / presupuesto

¿Cuál es el tiempo disponible para terminar el proyecto?

¿Cuál es el presupuesto?

Algunos modelos son más eficientes que otros en términos de tiempo y recursos.



Naturaleza del proyecto

¿El Proyecto es grande y complejo o pequeño y simple?

Modelos ágiles se desempeñan mejor en proyectos cambiantes, mientras modelos no ágiles funcionan mejor en proyectos más estructurados.



Colaboración del Cliente

¿El cliente desea involucrarse durante todo el proyecto?

Los modelos ágiles promueven la colaboración con el cliente.

Arquitectura del sistema

Es un plan que muestra cómo se organizan las diferentes partes de un programa y como interactúan entre sí, ayudando a los programadores a entender cómo construir el programa y a verificar que las partes encajen de manera correcta. Los factores primordiales que se deben tener en cuenta para elegir la arquitectura son:

Costo

Existen arquitecturas con alto grado de complejidad que requieren mayor infraestructura, lo que se traduce en más inversión.

Necesidad de crecimiento

Si es probable que el software se expanda en el futuro, la arquitectura debe poder adaptarse a ese crecimiento.

Experiencia del equipo


La arquitectura para ser más eficiente debe estar alineada con las habilidades y experiencia del equipo.

Mantenimiento y evolución

La arquitectura debe facilitar el mantenimiento y evolución.

Integración con sistemas

Debe ser compatible con sistemas existentes con los que el software interactuará.



El modelo de desarrollo y la arquitectura del sistema están interrelacionados e influyen mutuamente.

Diseño de interfaz de usuario

Enfoque en el usuario

El interfaz debe estar diseñado para el usuario, así que se debe entender sus necesidades y habilidades.

Respuesta rápida

El tiempo de respuesta del sistema debe ser inmediato, o al menos, debe informar que se está procesando su solicitud.

Familiaridad

La interfaz debe usar términos familiares para el usuario, así mismo los íconos y distribución de elementos.

Accesibilidad

La interfaz debe poderse adaptar a los usuarios, incluyendo a aquellos con discapacidad. Esto implica el tamaño de fuente, subtítulos, audios, etc.

Paleta de colores adecuada

Los colores deben ser agradables y fáciles de ver. El color transmite diferentes sensaciones, por lo cual se debe enlazar la temática con lo que se quiere transmitir.

NOTA: La inteligencia artificial puede ayudar en el diseño de interfaz de usuario, pues con el prompt adecuado puede generar imágenes referenciales muy útiles.

Modelado de base de datos

El modelo de base de datos muestra cómo se organiza los datos y como se relacionan entre sí. Un buen modelado encaminará a un diseño y creación de base de datos más eficiente.

La IA ayuda a modelar una base de datos de manera más precisa, tan solo necesitaremos alimentar de información como que se muestra en la siguiente imagen:

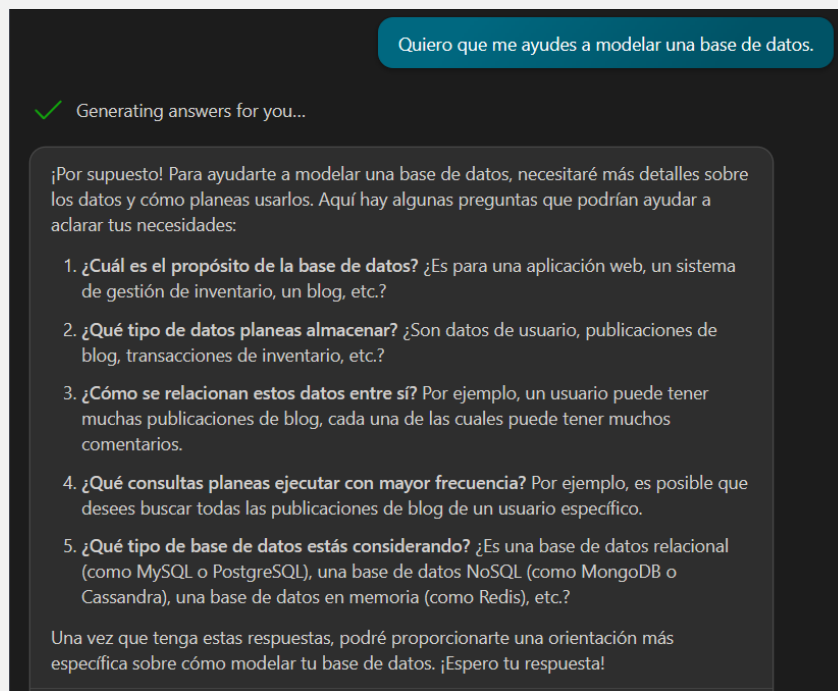
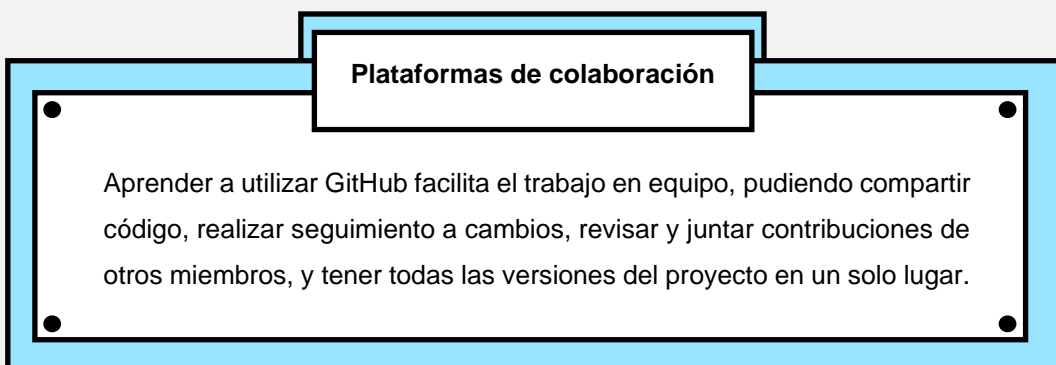


Imagen 1. Información requerida para modelar una BD. Bing.

Tecnologías y herramientas

Se proporciona un conjunto de tecnologías y herramientas que el programador podrá emplear al construir el software:



IDE

Utilizar un entorno de desarrollo integrado que sea adaptable a las necesidades del programador hará que el trabajo de codificación sea una tarea amena. Los IDE más utilizados son VS Code, IntelliJ IDEA, Vim, Android Studio, etc. Cada uno tiene fortalezas, debilidades y su elección dependerá de las necesidades del proyecto.

Frameworks

Brinda al programador una plantilla preestablecida para la creación de software. Utilizar un framework resultará en un menor tiempo de desarrollo y esfuerzo de los programadores.

Para elegir un framework con el cual trabajar se deberá analizar algunas características como el entorno del sistema, lenguaje de programación, tamaño y complejidad del proyecto.

Framework Scrum

Framework utilizado para gestionar proyectos.

Spring

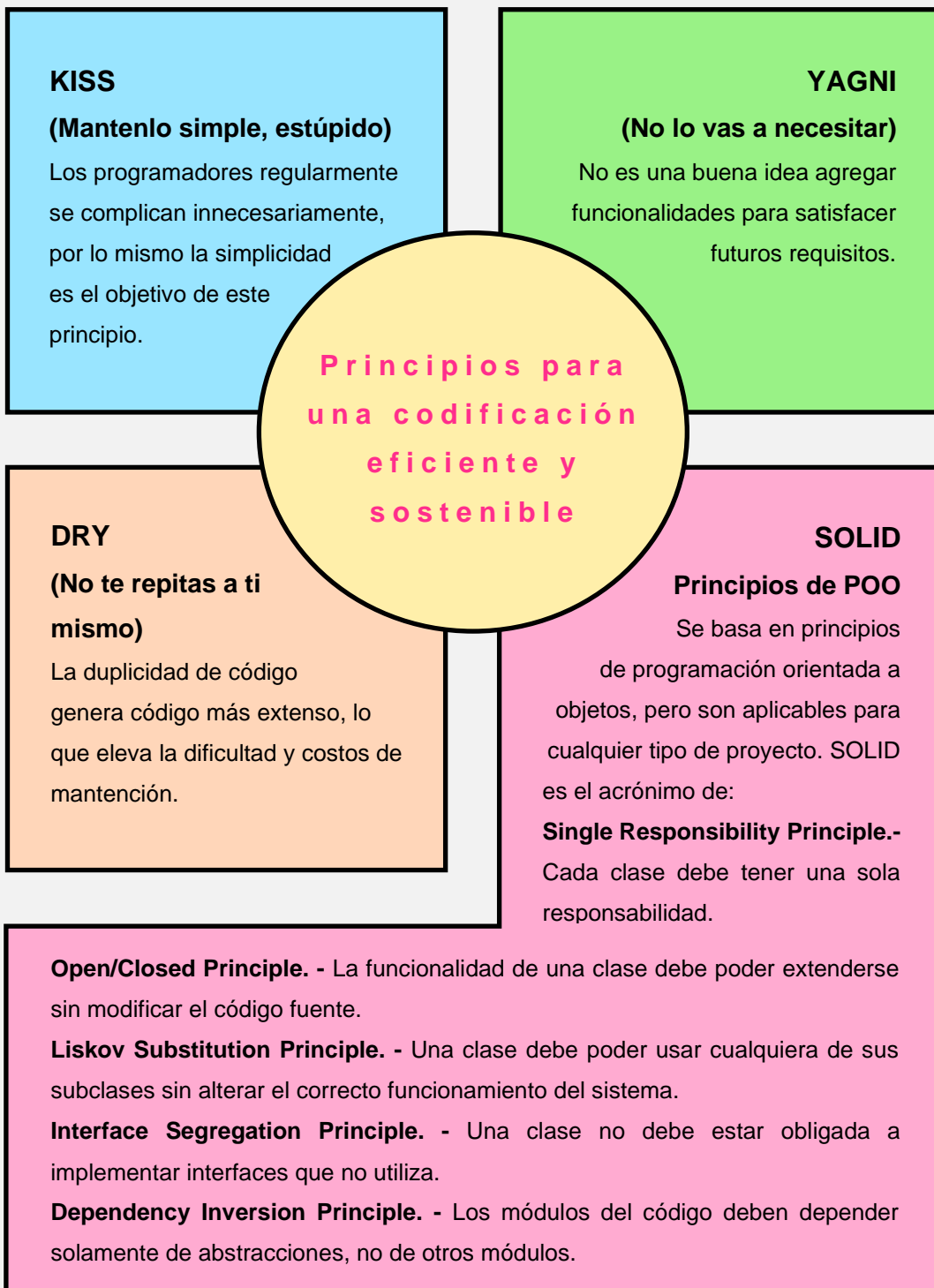
Framework de Java para desarrollar aplicaciones empresariales.

React

Biblioteca de JavaScript para crear interfaces de usuario.

Fase de desarrollo

En esta fase se empieza a crear el código fuente del programa. Los desarrolladores escriben el código necesario para implementar las funcionalidades establecidas en fases anteriores, considerando y aplicando:



No todo es programar

Uno de los errores más comunes al empezar con el desarrollo de un sistema es empezar directamente con la programación, sin antes tener una base sólida sobre la cual ir construyendo el sistema. Dichas bases se van estableciendo en las fases anteriores a esta.



Comunicación efectiva



Si se está trabajando en conjunto es esencial tener una buena comunicación, informando sobre avances y tareas que se están realizando. Puede ocurrir que ambos estén desarrollando la misma funcionalidad o que parte de un código sea reutilizable en otro.

Subdivide el sistema

Un sistema siempre podrá dividirse en varios componentes o módulos. Esto facilitará el desarrollo y mantenimiento, además si se trabaja en equipo se podrá dividir las tareas entre el equipo y el proyecto se concretará en menos tiempo.



Las sugerencias son eso, sugerencias

Al generar código con ayuda de la IA muchas veces se tendrá que modificar para adaptarlo a lo requerido. No tomes ese código como la verdad absoluta, aplica tus conocimientos y realiza pruebas correspondientes para validarlo.



Documentación de código

Un código bien documentado explica de manera clara y concisa la función y lógica detrás de un fragmento de código en específico. La documentación es importante para que otros desarrolladores e incluso el mismo autor comprendan el código y facilite el mantenimiento futuro.



Fase de pruebas

En esta fase se pondrá a prueba el sistema para verificar y validar su correcto funcionamiento. Las siguientes actividades ayudarán al programador a realizar esta labor con menor probabilidad de fallo:

Planifica las pruebas

Las pruebas deben ser planificadas en paralelo al desarrollo. Esto permite que al terminar una parte del software se identifique y solucione rápidamente los errores.

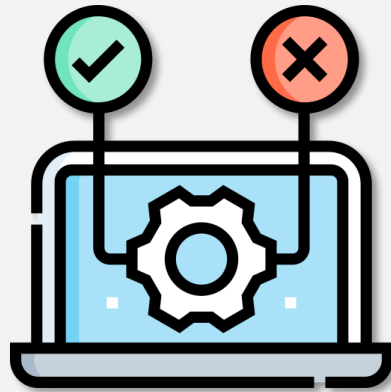


Pruebas unitarias

Prueba con la cual se valida la funcionalidad de un componente en concreto. Estas pueden ser automatizadas utilizando herramientas o bibliotecas que facilitan la creación, ejecución y análisis de pruebas unitarias como MochaJS, módulo unittest de python, Jasmine, etc.

Pruebas de integración

Si el sistema ha sido desarrollado mediante componentes o modularmente, se debe comprobar que interactúe correctamente con los demás. La IA puede aportar generando y ejecutando casos de prueba, utilizando herramientas de automatización de pruebas como Testim.io, AppliTools, entre otros.



Validación por parte del usuario

El sistema debe ser presentado al usuario para que pueda interactuar con él y verificar si cumple con sus necesidades y expectativas. Esto brinda la oportunidad de corregir cualquier problema antes de que el sistema sea implementado.

Fase de implementación

En esta fase el software desarrollado será alojado en un entorno de producción y será accesible para los usuarios. Las buenas prácticas sugieren que se debe:

Planificación del despliegue

Para garantizar una transición más llevadera del desarrollo al entorno de producción se debe evaluar lo que requerirá el despliegue, identificar las dependencias, asegurarse de tener los recursos necesarios (servidores, personal, ancho de banda) y detallar cada paso que se llevará a cabo.



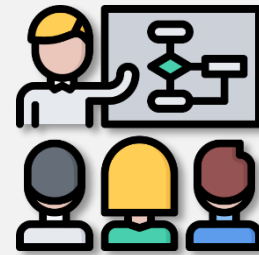
Despliegue del Software



Debe realizarse de manera controlada y ordenada, apegándose a la planificación inicial para evitar contratiempos. Puede utilizarse herramientas como GitLab CI/CD para automatizar el despliegue, prueba y monitorización.

Capacitación

Enseñar al usuario como utilizar el sistema permite que todas las funcionalidades implementadas sean conocidas y usadas. La capacitación se puede realizar desde diversos enfoques:



Manual de Usuario

Guía que detalla cómo se utiliza el sistema paso a paso. Incluye descripciones claras de las funciones y soluciones a problemas que el usuario puede encontrarse.

Sesiones de capacitación

Reunión en la que alguien que conoce muy bien el sistema enseña a los demás a cómo usarlo. Debe incluir demostraciones prácticas y un espacio para resolver dudas e inquietudes.

Webinars grabados

Son videos instructivos a los que el usuario puede acceder en cualquier momento. Este webinar debe incluir una revisión general del sistema e instrucciones acerca de funcionalidades en específico.

Monitorización



Tras desplegar el sistema se debe llevar una revisión exhaustiva de los recursos que se está utilizando, los tiempos de respuesta, errores, etc. Para esta tarea podremos implementar herramientas de monitorización como Prometheus, Grafana y ELK Stack.


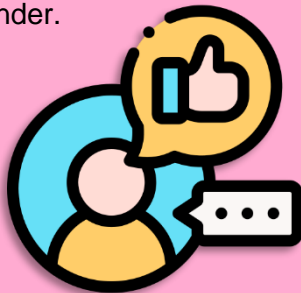
Fase de mantenimiento

En esta fase se pulirá el sistema dando solución a errores o bugs, o realizando mejora el rendimiento, actualizaciones de seguridad y eliminación o expansión de funcionalidades.

Algunas consideraciones que se debe tener en esta fase son:

Atiende a las sugerencias

Los usuarios del software muchas veces sugerirán funcionalidades, mejoras de interfaz, o informarán sobre errores. Los desarrolladores deberán evaluar estas sugerencias y determinar cuál atender.



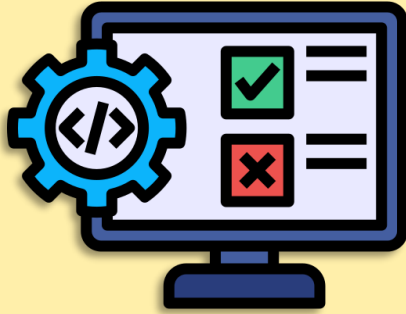
El mantenimiento implica una iteración constante de todas las fases.



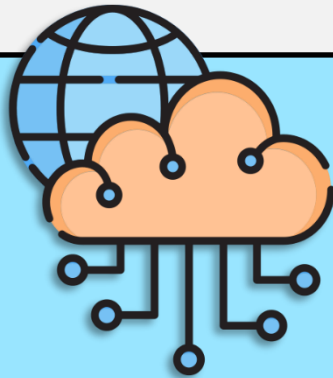
Documenta los cambios

Los cambios realizados durante esta fase deben ser documentados. Se debe incluir la descripción, razón e impacto del cambio.

Pruebas aisladas



Antes de lanzar un cambio en el sistema se debe realizar pruebas en un entorno separado al de producción para verificar que el cambio no cause problemas inesperados.



Despliegue inteligente

El despliegue de las actualizaciones debe realizarse en un momento en el que el software no se utilice mucho. Esto para interrumpir al menor número de usuarios posible.

Pueden ocurrir problemas

Aunque todo se planifique y se tomen precauciones, pueden ocurrir problemas después de la implementación. Si esto ocurre se deberá recurrir a la versión anterior al software mientras se resuelve el problema.



Consideraciones

Las buenas prácticas recopiladas en este manual han sido tomadas de una serie de recursos disponibles en la web, tesis y fragmentos de normas y estándares de la Organización Internacional de Normalización (ISO) y el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE).

Se ha tratado de plasmar la información más relevante de dichos recursos, pero se recomienda consultar las versiones completas de estas normas para una mayor comprensión de las mejores prácticas en el desarrollo de software.

Bibliografía

Delgado Olivera, L. de la C., & Díaz Alonso, L. M. (2021). Modelos de desarrollo de Software. Revista cubana de ciencias informáticas, 15(1), 37-51. <https://www.redalyc.org/journal/3783/378366538003/html/>

Domínguez S. (2023). Domina la fórmula para crear prompts efectivos en ChatGPT. OpenWebinars. <https://openwebinars.net/blog/domina-la-formula-para-crear-prompts-efectivos-en-chatgpt/#elementos-clave-de-un-prompt-efectivo>

González Jaimes, Elvira Ivone, López Chau, Asdrúbal, Trujillo Mora, Valentín, & Rojas Hernández, Rafael. (2018). Estrategia didáctica de enseñanza y aprendizaje para programadores de software. RIDE. Revista Iberoamericana para la Investigación y el Desarrollo Educativo, 9(17), 688-712. <https://doi.org/10.23913/ride.v9i17.402>

ISO/IEC. (2016). ISO/IEC TR 29110-1:2016 Systems and software engineering – Lifecycle profiles for Very Small Entities (VSEs) – Part 1: Overview. <https://www.iso.org/standard/62711.html>

ISO/IEC/IEEE. (2017). ISO/IEC/IEEE 12207:2017 Systems and software engineering – Software life cycle processes [Estándar]. <https://www.iso.org/standard/63712.html>

ISO/IEC/IEEE. (2020). ISO/IEC/IEEE 29119-1:2020 Software and systems engineering – Software testing – Part 1: Concepts and definitions. <https://www.iso.org/standard/81291.html>

Morejòn Labrada, S. (2020). Principios del proceso de diseño de interfaz de usuario. *Revista Cubana De Transformación Digital*, 1(3), 143–155. <https://rctd.uic.cu/rctd/article/view/96>

Ruiz Mendoza, K. K. (2023). El uso de ChatGPT 4.0 para la elaboración de exámenes: crear el prompt adecuado. *LATAM Revista Latinoamericana De Ciencias Sociales Y Humanidades*, 4(2), 6142–6157. <https://doi.org/10.56712/latam.v4i2.1040>

Xirgo, L. R., & González, A. J. V. (2019). Aplicación del principio KISS a la enseñanza de los fundamentos de programación. *Actas de las XXV Jornadas sobre Enseñanza Universitaria de la Informática*, 4, 55-62.