

a. Basado en los tres casos de uso (baja, media y alta complejidad), ¿qué elementos técnicos específicos determinan que la IA sea un aliado real o un obstáculo en el desarrollo? y ¿Por qué cree que en tareas de alta complejidad el ahorro de tiempo suele ser menor? generalmente es un aliado para temas que son repetitivos o que no requieren de un contexto grande para poder entregar una respuesta altamente efectiva, cuando son de construcción de herramientas, o que no requiere de conocimiento muy avanzado del producto ya construido. Suele volverse obstaculo cuando no tienen un entrenamiento adecuado y carece del contexto necesario para ejecutar la tarea. En tareas de alta complejidad la complejidad no esta en la generación del código por parte de la IA, sino en todo el contexto que se debe armar, dependencias, arquitectura, etc. Casos de Uso y Reglas de Negocio claramente definidas. En terminos tecnicos la complejidad dejo de ser obstaculo y factor de poco rendimiento con IA, el tema pasa mas por definiciones, reglas y alcances claros, con estos elementos la eficiencia aumenta de lo contrario el rendimiento seria bajo por los reprocesos asi sea con IA. Los elementos técnicos que determinan que la IA sea un aliado real o un obstáculo son principalmente: la claridad de las reglas de negocio, el nivel de reutilización de componentes, el grado de acoplamiento entre módulos, la cantidad de dependencias existentes y la criticidad del proceso. La IA funciona mejor cuando las funcionalidades son específicas, repetitivas y tienen entradas y salidas bien definidas, como la generación de PDFs o formularios. En cambio, se vuelve menos efectiva cuando existen múltiples dependencias entre módulos, reglas implícitas no documentadas o procesos críticos que afectan otros componentes del sistema, como pagos, carteras o liquidaciones. En tareas de alta complejidad el ahorro de tiempo suele ser menor porque el mayor esfuerzo no está en escribir código, sino en comprender el negocio, analizar dependencias, validar escenarios y ejecutar pruebas de regresión. Aunque la IA puede acelerar la generación de código, aún se requiere intervención humana para garantizar que los cambios no produzcan errores o impactos no previstos en procesos relacionados. Por ello, mientras en tareas simples la IA reduce significativamente el tiempo de implementación, en tareas complejas el beneficio disminuye debido a la necesidad de análisis y validación adicional.	b. Si usted estima un tiempo de supervisión para las sugerencias de la IA, ¿Cómo decide cuándo confiar plenamente en el código generado y qué criterios utiliza para detectar posibles alucinaciones o errores lógicos? considero que nunca se debería confiar a ciegas, para eso se deben generar los mecanismos que se encargen de inspeccionar el código, las pruebas unitarias, pruebas de integración, funcionales, etc, es lo que debe garantizar que lo generado cumpla con lo solicitado. El código generado yo no es problema siempre y cuando los modelos se maduren con estándares y políticas de desarrollo, patrones y skill adecuados. Si esto se hace bien, confío por encima del 90%. Las alucinaciones o errores se miden con base en los resultados esperados. La decisión de confiar plenamente en el código generado por la IA depende del nivel de impacto y criticidad de la funcionalidad. En tareas simples y de bajo riesgo, donde las reglas son claras y el alcance es limitado, el nivel de confianza puede ser mayor. Sin embargo, en funcionalidades que afectan procesos transaccionales o múltiples módulos, la supervisión debe ser más estricta. Para detectar posibles alucinaciones o errores lógicos se utilizan criterios como: Verificar que el código cumpla exactamente las reglas de negocio definidas. Revisar que las consultas, variables, tablas o servicios utilizados realmente existan en el sistema. Validar que no se hayan asumido escenarios o comportamientos no especificados en los requisitos. Analizar impactos sobre otros módulos o procesos dependientes. Ejecutar pruebas unitarias, funcionales y de regresión. Revisar manejo de excepciones, condiciones límite y casos no contemplados. Comparar el resultado generado con patrones o implementaciones existentes en el sistema. En la práctica, la confianza total rara vez es automática, aumenta a medida que el código supera revisiones técnicas y pruebas que confirman que su comportamiento coincide con las necesidades reales del requerimiento.	c. ¿Cómo influye la arquitectura actual empleada en su empresa en la velocidad con la que puede integrar código generado por IA? ¿Considera que las herramientas de IA se alinean bien con los estándares de calidad de la empresa? influye mucho la arquitectura, al menos en la parte del backend donde esta la logica de negocio, dado que se requiere demasiado contexto, y cualquier error en no detectado puede afectar en procesos de mision critica. Creo que todavia nos falta mucho para alinear los estandares de calidad con la IA, porque no estan diseñados y construidas con las reglas precisas y los controles adecuados, pero ya se esta en camino a organizarlos para que puedan alinearse. Una buena arquitectura soportada en Patrones, Skill fuertes, políticas y estándares hacen que los desarrollos sean mas rapidos. Las herramientas de IA se maduran con los elementos mencionados anteriormente, por lo tanto se alinean. La arquitectura actual influye directamente en la velocidad de integración del código generado por IA. Si la arquitectura es modular, con capas bien definidas, servicios desacoplados y reglas de negocio centralizadas, la integración suele ser más rápida porque la IA puede generar componentes aislados con menor riesgo de afectar otras partes del sistema. En cambio, cuando existen arquitecturas heredadas con lógica distribuida en formularios, procedimientos almacenados, triggers o componentes altamente dependientes entre sí, la integración se vuelve más lenta porque requiere mayor análisis, adaptación y validación. Las herramientas de IA se alinea parcialmente con los estándares de calidad de la empresa. Son eficientes para acelerar tareas repetitivas, generar estructuras base y apoyar la documentación o pruebas; sin embargo, el código generado no siempre cumple automáticamente con estándares internos relacionados con arquitectura, nomenclatura, seguridad, manejo de errores, rendimiento o reglas específicas del negocio. Por ello, el uso de IA debe complementarse con revisiones técnicas, pruebas y procesos de aseguramiento de calidad antes de incorporarse a ambientes productivos.	a. Más allá de dar instrucciones básicas, ¿cómo ha perfeccionado su técnica de ingeniería de prompts (uso de contexto, delimitadores y ejemplos) para obtener resultados de mayor calidad? estructurando mejor los prompts de tal manera que se pueda dar un contexto completo y se alinee a la IA para que de respuestas efectivas. Definiendo los roles, objetivos, restricciones, etc, utilizando archivos de contextos como los markdowns, los skills, etc, todos los componentes que soporten un buen prompt, porque no es solo el prompt. Claridad, precision y alcance acotado y por partes, siempre exigiendo planes de ejecucion. La técnica de ingeniería de prompts se a venido perfeccionando dando contexto detallado, delimitando claramente el problema y utilizando ejemplos concretos. El contexto permite que la IA comprenda el dominio del negocio, la arquitectura existente, las tecnologías involucradas y las restricciones del sistema. Los delimitadores ayudan a separar requisitos, reglas de negocio, entradas y resultados esperados para reducir ambigüedades. Los ejemplos permiten orientar la respuesta hacia un formato o comportamiento específico. Adicionalmente, se mejora la calidad definiendo explícitamente el rol que debe asumir la IA (por ejemplo, arquitecto de software, desarrollador o analista funcional), estableciendo criterios de salida esperados y agregando restricciones como estándares de codificación, patrones arquitectónicos o condiciones de validación. También se utiliza refinamiento iterativo, donde las respuestas iniciales se ajustan progresivamente hasta alcanzar un nivel adecuado de precisión y calidad.	b. Con el uso de asistentes, el trabajo manual disminuye. ¿Qué habilidades de diseño modular y pensamiento estratégico considera que son ahora más importantes que el simple conocimiento de la sintaxis del lenguaje? ya los lenguajes y sintaxis toman una relevancia muy baja, ahora lo importante es la ingeniería de prompts y todo el contexto alrededor de los prompt, de tal forma que con esta informacion cualquier la pueda generar código sin importar el lenguaje y su sintaxis Frente al diseño modular, la desagregación o segmentación del tema a desarrollar o problema a solucionar, y en el pensamiento estrategico, la precisión en la definición de alcances y objetivos buscados frente a temas de velocidad, seguridad, concurrencia, etc. Las habilidades que adquieren mayor relevancia es la capacidad de diseñar arquitecturas modulares, comprender los procesos del negocio y analizar dependencias entre cada componente. Más que conocer la sintaxis específica de un lenguaje, se vuelve más importante definir correctamente cómo se estructuran los servicios, cómo interactúan los módulos y cómo mantener bajo acoplamiento y alta cohesión en el sistema. Por ello, el pensamiento estratégico es para identificar riesgos, priorizar funcionalidades, anticipar impactos de cambios y tomar decisiones de diseño orientadas a escalabilidad, mantenibilidad y rendimiento. Adicionalmente, habilidades como el análisis de reglas de negocio, la capacidad de validar resultados, el diseño de pruebas y el criterio para evaluar la calidad del código generado son fundamentales, ya que permiten asegurar que las soluciones sean técnicamente correctas y alineadas con las necesidades reales del negocio.	c. ¿Qué procesos sigue para asegurar que el código sugerido por la IA no introduzca vulnerabilidades de seguridad (CWE) o problemas de propiedad intelectual en los repositorios de la empresa? para esto se deben aplicar herramientas de auditoria e inspeccion de código, como el sonar, y aplicar desarrollo seguro (OWASP), sobre estos temas aun nos falta trabajar mucho para minimizar código con vulnerabilidades Definir herramientas de control y supervisión (github), validar el nivel de revisión de acuerdo al criticidad de componentes, hacer trazas, realizar formacion continua al equipo, entre otras. Para asegurar que el código sugerido por la IA no introduzca vulnerabilidades de seguridad (CWE) o problemas de propiedad intelectual, se siguen procesos de revisión técnica y validación antes de integrar cualquier cambio al repositorio. Primero, se realiza análisis del código para identificar patrones inseguros como inyección SQL, exposición de credenciales, validación insuficiente de entradas, manejo inadecuado de autenticación o permisos y otros riesgos asociados a vulnerabilidades conocidas. Adicionalmente, se utilizan herramientas de análisis estático y revisión automatizada para detectar posibles problemas de seguridad y dependencias vulnerables. El código también pasa por revisiones entre pares y pruebas funcionales para validar su comportamiento dentro del contexto del sistema. Respecto a propiedad intelectual, se evita incorporar código generado de forma automática sin revisión previa, especialmente si incluye fragmentos extensos o implementaciones específicas que puedan provenir de terceros. También se verifica el origen y licenciamiento de librerías o componentes externos utilizados, asegurando cumplimiento con las políticas y estándares internos de la empresa. De esta manera, la IA se utiliza como apoyo al desarrollo, pero la responsabilidad final sobre seguridad y cumplimiento permanece dentro del proceso de validación del equipo técnico.	a. Si la curva de aprendizaje se percibe como empinada, ¿qué tipo de formación específica o acompañamiento técnico cree que facilitaría que el equipo adoptara la IA con mayor confianza? primero la ingeniería de prompts, luego la automatización de pruebas y configuración de inspeccion de código, y detección de vulnerabilidades 1. Marco Conceptual. 2. Ingeniería de Prompts 3. Acompañamiento tecnico y funcional. la adopción de IA se facilitaría mediante formación práctica y acompañamiento enfocado en casos reales del entorno de trabajo. Más que capacitaciones generales, resultan más útiles entrenamientos orientados a cómo integrar la IA en actividades cotidianas del equipo, como generación de código, análisis de requisitos, documentación, pruebas y revisión técnica. También sería valioso incluir formación específica en ingeniería de prompts, buenas prácticas para validar resultados, identificación de errores o alucinaciones, aspectos de seguridad y uso responsable de la información. Adicionalmente, el acompañamiento técnico mediante sesiones guiadas, mentorías, ejemplos aplicados a proyectos reales y definición de estándares internos de uso ayudaría a generar confianza y reducir la incertidumbre sobre cuándo y cómo utilizar la IA de manera efectiva.	b. En cuanto al riesgo de estandarización del código (sego). ¿De qué manera utiliza la IA para potenciar su creatividad en lugar de simplemente repetir patrones preexistentes? lo primero es apoyarse en la la para ayudarte a hacer lluvias de ideas, retando a la misma ia, y poniendola a prueba, para que sea una aliada en el proceso creativo Aplicar siempre ideas de innovación, dejando de lado la forma tradicional de hacer y contruir código, siempre trabajando con la necesidad del usuario final o consumidor del producto. Para evitar el riesgo de estandarización del código, la IA se utiliza como una herramienta de exploración y generación de alternativas, no como un mecanismo para aceptar automáticamente la primera solución propuesta. Se aprovecha para comparar diferentes enfoques arquitectónicos, evaluar patrones de diseño, generar múltiples opciones de implementación y analizar ventajas y desventajas de cada una. La creatividad se potencia utilizando la IA para cuestionar soluciones tradicionales, proponer escenarios alternativos y descubrir enfoques que quizás no se habían considerado inicialmente. Sin embargo, la decisión final se basa en el contexto específico del negocio, las restricciones técnicas y la experiencia del equipo. De esta manera, la IA complementa el pensamiento estratégico y la capacidad de diseño, en lugar de limitarse a replicar patrones preexistentes.	c. ¿En qué medida la preocupación por la privacidad de los datos de la empresa limita su uso diario de herramientas basadas en la nube como ChatGPT o Claude? de momento no ha sido un impedimento, igual lo que se busca es mejorar la productividad, manteniendo altos estandares de calidad, por el momento no se ha tenido restricciones sobre el tema Esto es un seego aplicado por las políticas de Gobernanza de los datos, sin embargo siempre debe prevalecer la política de privacidad o confidencialidad de los datos, tratando de garantizar que no se vulneren los derechos de personas e instituciones. La preocupación por la privacidad de los datos puede limitar significativamente el uso diario de herramientas basadas en la nube, especialmente cuando se trabaja con información sensible, código propietario, datos de clientes o reglas de negocio estratégicas. En estos casos, se evita compartir información confidencial, credenciales, datos personales, estructuras internas o detalles críticos de la arquitectura empresarial. Para reducir riesgos, el uso de estas herramientas suele apoyarse en prácticas como anonimizar información, utilizar datos simulados, compartir únicamente fragmentos necesarios y seguir las políticas internas de seguridad y gobierno de datos de la empresa. La adopción también depende del nivel de confianza en los controles de privacidad ofrecidos por la herramienta y de la existencia de versiones empresariales que incorporen mayores garantías de seguridad y cumplimiento normativo.
---	---	---	--	--	---	--	---	--