

## GUIÓN PRESENTACION

### **Ipek Ozkaya**

Ipek Ozkaya es investigadora principal y directora técnica del grupo Engineering Intelligent Software Systems en el Software Engineering Institute (SEI) de la Universidad Carnegie Mellon.

Ipek trabaja en desarrollar técnicas que permitan mejorar la eficiencia en el desarrollo de software y la evolución de sistemas, enfocándose en prácticas de arquitectura de software, economía del software y desarrollo ágil.

Recientemente, su trabajo se ha centrado en la gestión de la deuda técnica en sistemas complejos y al desarrollo de sistemas habilitados por inteligencia artificial.

### **Introducción a la Arquitectura del Software**

Ipek comienza dando una definición académica de la Arquitectura del Software: “La arquitectura del software de un sistema es el conjunto de estructuras necesarias para razonar sobre el sistema que comprende los elementos de software, las relaciones entre ellos y las propiedades de ambos.”

Esto realmente significa que la arquitectura de un sistema está compuesta por estructuras que representan las distintas vistas de un sistema, como pueden ser el comportamiento de dicho sistema en tiempo de ejecución o la vista de despliegue.

### **Riesgos y oportunidades del uso de IA en el entorno profesional**

Tal y como expresa Ipek Ozkaya en el podcast, nos enfrentamos tanto a una oportunidad como a un riesgo. Una oportunidad radica en que, al tener estas herramientas de IA generativa a nuestro alcance, es más fácil incorporar todo el conocimiento que poseemos, ya que los modelos de lenguaje pueden procesar esta información. Gracias al modo interactivo de las herramientas de chat, se permite la transferencia de conocimiento podría mejorarse significativamente, permitiendo a los arquitectos acceder a patrones de diseño y mejores prácticas de forma más fluida. Además, la automatización de tareas rutinarias libera a los arquitectos para que se centren en el pensamiento estratégico. La IA puede encargarse de tareas como la generación de documentación y resúmenes de código, permitiéndonos invertir nuestro tiempo en la toma de decisiones de alto nivel y la resolución de problemas complejos.

El riesgo potencial radica en que, cuando se hace uso de estas herramientas para tareas como la implementación, el alcance de las interacciones con ellas es limitado y puede llevar a la implementación inconsistente por la falta de contexto. Aunque actualmente dichas herramientas están mejorando, la pregunta que plantea Ipek Ozkaya es: "¿Estoy incorporando la pregunta de implementación que planteo de la misma manera con el resto del sistema? ¿Se hace de forma consistente? ¿Se tienen en cuenta algunas de las preocupaciones que podrían surgir de escenarios arquitectónicos significativos, etc.?" Vemos que muchas herramientas disponibles son de gran ayuda para diferentes roles y partes, pero requieren una implementación cuidadosa para evitar inconsistencias.

### **Casos de uso de IA en arquitectura del software**

Como señala Ipek Ozkaya, estamos en las primeras etapas de la investigación de estas herramientas para tareas más complejas. Algunos de los casos de uso pueden ser: la documentación de la arquitectura, las tareas y el resumen de código para comprender el contexto, y preguntar todo tipo de conocimiento arquitectónico en términos de, bien, ¿cómo implemento este marco en particular? ¿Qué significa usar un patrón arquitectónico que uso que aparece antes de que aparezca para este sistema en particular? ¿Qué otros sistemas similares puedo aprender de? Además, la IA puede ayudar en la generación de modelos 3D y simulaciones de arquitectura, optimizar la distribución de espacios y la planificación de proyectos, y mejorar la creación de propuestas de diseño y renders.

### **Cómo ayuda la IA en las "trade-off decisions"**

Las IA pueden ayudar en las "trade-off decisions" en tareas como el análisis de datos con el fin de identificar patrones arquitectónicos. Las herramientas de IA pueden generar documentación sobre las decisiones arquitectónicas tomadas, facilitando la trazabilidad y justificación de los "trade-offs". La IA generativa puede predecir cómo afectará una elección arquitectónica en el futuro. Además, la IA puede ayudar a la simulación de diferentes escenarios, lo que permite a los arquitectos evaluar los posibles resultados de sus decisiones, y analizar el impacto de las decisiones arquitectónicas en factores como el costo, el rendimiento y la sostenibilidad.

### **Uso de IA en proyectos "legacy"**

Tal y como menciona en el podcast Ipek Ozkaya, el uso de las IA puede ser muy útil para tareas como la creación de documentación o para la lectura y contexto de un código que nos facilite la comprensión de este. Esto puede resultar de mucha utilidad para proyectos "legacy" debido a la común falta de documentación o la implementación con código difícil de mantener o antiguo. El uso de la IA facilitaría mucho la transformación de un proyecto "legacy", ayudando en tareas de refactorización de código, documentación y pruebas, y actualización de dependencias. También puede ayudar en

la migración de sistemas "legacy" a plataformas más modernas y en la extracción de datos para su integración con nuevas aplicaciones y servicios.

### **¿Qué modelo es mejor?**

Ipek comenta que la elección del modelo depende de varios factores. Actualmente, existen muchísimos modelos disponibles, cada uno con características distintas. Algunos son más rápidos, mientras que otros destacan por su precisión. La elección adecuada depende del objetivo que se persiga, el tipo de resultado esperado y los datos con los que el modelo haya sido entrenado.

Por ejemplo, si solicitamos información sobre un tema de última generación y el modelo no ha sido entrenado con esos datos, su velocidad no importará, ya que no podrá proporcionar una respuesta precisa e incluso podría generar información errónea.

Otro aspecto clave es la capacidad del modelo para procesar y comprender correctamente la información que se le proporciona en los prompts.

Para un arquitecto de software, lo más importante es que el modelo sea consistente, es decir, que mantenga una estrategia uniforme para resolver problemas y responda de manera coherente en situaciones similares.

### **Importancia de los prompts**

La generación efectiva de prompts es crucial en herramientas basadas en inteligencia artificial generativa (Gen AI) aplicadas a contextos complejos como la arquitectura de software, ya que permiten expresar claramente restricciones, patrones arquitectónicos específicos y requisitos de calidad. Un prompt detallado, que incluya vocabulario arquitectónico, facilita cerrar la brecha entre decisiones abstractas del arquitecto y la implementación concreta en código, asegurando que la solución generada responda fielmente a las intenciones iniciales.

Además, los prompts bien contruidos tienen un impacto directo en la transferencia y democratización del conocimiento arquitectónico. Al explicitar patrones, decisiones y supuestos que normalmente podrían quedar implícitos, ayudan a reducir la dependencia de arquitectos aislados y facilitan que desarrolladores con menos experiencia puedan implementar soluciones arquitectónicamente consistentes mediante interacciones guiadas con herramientas como ChatGPT o Copilot.

Finalmente, debido a la naturaleza probabilística de las herramientas de Gen AI, la calidad del prompt determina en gran medida la precisión y consistencia del resultado generado. Un prompt cuidadosamente diseñado no solo mejora la calidad del software

resultante, sino que también permite que expertos validen rápidamente las soluciones propuestas, identificando posibles errores o desviaciones, y asegurando así un desarrollo más ágil y confiable en entornos complejos.

### **Evolución de las IAs**

Cada semana, las IAs evolucionan y presentan nuevos avances. En poco tiempo, han mejorado en múltiples aspectos, como la integración de documentos, el uso de varios modelos dentro de una misma IA y la capacidad de procesar imágenes, videos e incluso audios.

Otro aspecto en el que continuarán mejorando es su capacidad para recordar información, es decir, la cantidad de mensajes previos que pueden retener, el contexto de la conversación e incluso datos de chats anteriores.

Además, las herramientas especializadas en mantener el contexto también están avanzando. Aunque una IA no pueda recordar todo por sí sola, existen soluciones que permiten conservar el hilo de la información a lo largo del tiempo.

Por otro lado, el *prompt engineering* sigue perfeccionándose, lo que significa que las IAs están mejorando en la construcción de *prompts* optimizados para generar resultados más precisos y alineados con las expectativas del usuario.

Las aplicaciones basadas en IA también evolucionan, permitiendo usos cada vez más complejos y específicos, lo que representa una gran oportunidad para los ingenieros de software. Hay un enorme entusiasmo y un potencial increíble en estas herramientas. Sin embargo, a medida que las IAs se vuelven más avanzadas y capaces, surge una reflexión inevitable sobre el futuro del rol de los ingenieros de software en este nuevo panorama tecnológico.

### **Futuro de las IAs**

El avance en el uso de prompts en inteligencia artificial está transformando el desarrollo y la arquitectura de software. Actualmente, existen patrones generales para la construcción de prompts, aplicables a diversas áreas, no solo a la ingeniería de software. Sin embargo, el siguiente paso será el desarrollo de patrones más específicos para tareas concretas, como la documentación de código o la corrección de errores. En este contexto, surge la posibilidad de que los documentos de arquitectura de software evolucionen hacia documentos de prompts, ya que estos funcionan como una forma estructurada de expresar requisitos y necesidades del sistema.

Por otro lado, aunque la IA es útil para tomar decisiones en cuestiones como la selección de algoritmos o el diseño de bases de datos, su naturaleza probabilística implica que las

respuestas pueden variar y, en algunos casos, ser incorrectas. Por ello, es fundamental validar los resultados obtenidos. A pesar de esta limitación, los desarrolladores con experiencia en un área específica pueden aprovechar mejor la IA si saben construir prompts efectivos. Esta evolución ha acelerado la utilidad de herramientas como code completion, lo que a su vez abre la posibilidad de explorar tareas aún más complejas en el futuro.

### **Problemas de seguridad en las IAs**

El uso de herramientas de IA generativa en el diseño y la arquitectura de sistemas plantea serias preocupaciones de seguridad, como la filtración de datos y la desinformación. Aunque los desarrolladores de estos modelos pueden revelar cómo se manejan los datos, el riesgo aumenta cuando se interactúa con modelos públicos, ya que la información ingresada puede terminar en el dominio público. Un ejemplo notable fue el caso de Samsung, donde empleados introdujeron información propietaria en una IA, exponiéndola sin darse cuenta.

Más allá de la seguridad y la fuga de datos, el riesgo también abarca el uso indebido de información confidencial. Por esta razón, es responsabilidad de los usuarios ser cautelosos al compartir datos sensibles con estas herramientas. Aunque los desarrolladores de IA están mejorando los mecanismos de seguridad, aún pueden producirse errores, lo que refuerza la necesidad de validar la información generada.

### **Validación de una arquitectura**

En los test de código hay métodos como las pruebas unitarias que nos permiten comprobar su funcionamiento. En cuanto a la arquitectura no existe una forma cuantitativa de verificar que la arquitectura sea correcta. Normalmente la validación de una arquitectura se sostiene en la opinión de un experto en la materia, por lo que una IA generativa no puede ayudar en esta situación.

### **¿Las IAs reducen la creatividad?**

Si la IA es consistente en sus respuestas, efectivamente reduce la creatividad, al dar la misma solución al mismo problema siempre. De todas formas, no es un problema ya que la propia existencia del conocimiento arquitectónico y de los patrones de diseño reduce significativamente la creatividad. Lo importante es más que la solución obtenida satisfaga los requisitos que se presenta de forma eficiente y no tanto que lo haga de forma creativa.

### **¿Son las IAs capaces de producir una buena arquitectura?**

Hoy en día las IAs pueden generar arquitecturas de software, pero siguen presentando dudas en cuanto a su calidad y fiabilidad.

Si bien es cierto que pueden asistir en el proceso de diseño, proponiendo diseños basados en patrones conocidos, no garantizan una solución óptima sin la intervención humana. Factores como la escalabilidad, la mantenibilidad y la alineación con los requisitos del negocio requieren la figura del arquitecto.

Por lo tanto, la inteligencia artificial es una herramienta muy útil, pero no puede reemplazar el papel del arquitecto del software.