

El uso adecuado de la AI en el SDLC: desarrollo & reingeniería

Modificación de 10 programas SQLRPGLE completada en 7 horas con AI-First development

Un proyecto IBM i con estimación de 10 días fue completado en 7 horas al adoptar el paradigma AI-First Development, cumpliendo un compromiso crítico con tiempo de sobra.

Industria: **Tecnología / Desarrollo de software**

Resumen

Un pase de AS400 había sido devuelto por el Jefe de Ingeniero de Software (JIS) en 2 ocasiones por malas prácticas de desarrollo, estándares inadecuados y problemas de arquitectura. El retraso acumulado era de 2 meses. La jefatura de desarrollo se comprometió a entregar en un máximo de 5 días. La estimación tradicional para modificar los 10 programas SQLRPGLE era de al menos 10 días. Aplicando AI-First Development, el equipo completó todos los cambios en 7 horas.

Principales desafíos

- Pase devuelto 2 veces por el JIS por malas prácticas y estándares inadecuados.
- Retraso acumulado de 2 meses en el desarrollo.
- Compromiso de entrega en máximo 5 días con una estimación real de 10 días.
- 10 programas SQLRPGLE requiriendo modificación y corrección.
- Observaciones de la primera devolución no habían sido atendidas correctamente en la segunda.

Enfoque

Se adoptó el paradigma AI-First Development, donde la inteligencia artificial participó activamente en cada etapa — no como asistente reactivo, sino como eje central del proceso:

Análisis AI-First de los 10 programas para comprender contexto y problemas	Generación colaborativa de código con AI
Validación automática de sintaxis con AI	Refactorización inteligente para corregir arquitectura y estándares

Resultados

- 10 programas modificados, corregidos y entregados en tiempo récord.
- Tiempo reducido de 10 días estimados a 7 horas reales (93% de reducción).
- Compromiso con el cliente cumplido dentro del plazo de 5 días.

Tecnologías utilizadas

IBM i / AS400

SQLRPGLE

GitHub Copilot (AI-First)



Conclusión

El paradigma AI-First transforma la productividad cuando la AI participa activamente en todo el ciclo de desarrollo. La diferencia no está en tener acceso a las herramientas — el equipo ya tenía GitHub Copilot— sino en usarlas como eje central de cada etapa, no como apoyo ocasional.