

Guía de uso de Copilot en VS Code

1. Herramientas ia que utilizaremos:

- Copilot en visual studio code

2. Qué es:

- Copilot Chat permite usar lenguaje natural para obtener asistencia inteligente mientras programas: preguntar por partes del código, generar nuevos fragmentos, entender lógica, depurar o incluso planificar cambios.

3. Cómo acceder

- Hay tres formas de uso:
 - **Chat view** (panel lateral) para conversaciones completas y tareas complejas.
 - **Inline chat** (directamente en archivo/editor) con atajo **Ctrl+I** (Linux/Win) o **⌘I** (macOS).
 - **Quick chat overlay** con **Ctrl+Shift+Alt+L** (Windows/Linux) o **⇧⌥⌘L** (macOS) para preguntas rápidas sin cambiar de vista.
- Nosotros principalmente utilizaremos el **chat view** y la herramienta **inline chat**

4. Modos incorporados (built-in chat modes)

- **Agent**: ideal para tareas complejas que incluyan múltiples archivos, comandos de terminal o automatización.
- **Plan**: (en vista previa) útil para estructurar cambios grandes como un nuevo módulo o refactorizado completo.

- **Ask:** para preguntar sobre fragmentos de código, ideas, explicaciones; adecuado para temas concretos, pequeños cambios.
- **Edit:** para generar automáticamente ediciones en varios archivos del proyecto cuando ya sabes qué deseas cambiar.

5. Mejores prácticas para prompts

- Añade **contexto** usando referencias como `#file`, `#codebase`, `#terminalSelection`.
- Usa comandos especiales precedidos por `/`, por ejemplo `/new`, `/explain`.
- Si utilizas herramientas adicionales en el chat, referencia su nombre con `#`. Por ejemplo, `#fetch` para recuperar contenido web.

6. Beneficios para programar eficientemente

- Puedes generar código funcional rápidamente en varios archivos.
- Obtener explicaciones de lógica compleja o legacy code sin tener que investigar extensamente.
- Acelerar tareas repetitivas como refactorización o pruebas unitarias.
- Mantener flujo de trabajo ágil con menos cambios de contexto (no tener que abandonar editor para buscar documentación).

7. Consideraciones/Limitaciones

- Revisa siempre el código generado: puede necesitar ajustes o no cumplir al 100 % con tus requisitos.

- Añade contexto relevante: sin suficiente contexto, las sugerencias pueden no encajar perfectamente.
- Asegúrate de que los cambios automáticos no rompan tu arquitectura o convenciones de equipo.
- Mantén buen control de versiones: cuando la IA realiza ediciones en varios archivos, es útil ver los cambios antes de confirmar.

8. Referencias

Documentación de Visual Studio sobre copilot:

https://code.visualstudio.com/docs/copilot/chat/copilot-chat#_builtin-chat-modes