

guayerd

Fundamentos IA

Introducción IA y datos

Clase 5

En colaboración con
IBM SkillsBuild





¡Bienvenidos!

¿Nos presentamos?

- ¿Qué recuerdan de la clase anterior?
- ¿Qué esperan aprender?
- ¿Tienen alguna pregunta?

Contenidos

Por temas

05

- Copilot Chat y prompts
- Demo asincrónica

06

- Limpieza y transformación

07

- Estadística aplicada

08

- Visualización

Objetivos de la clase



- Copilot Chat
- Prompting y escritura efectiva

Introducción a la IA y los datos

Copilot Chat y prompts

Plataforma Skill Build: IA generativa



Plan de formación

Inteligencia artificial en la práctica

40 minutos  29.752 ★★★★★ 891



eLearning

cree su primer chatbot

1 hora  En curso

¿Qué es Copilot Chat?

Herramienta de IA integrada en entornos de desarrollo.

- Asiste en redacción, análisis y programación
- Interactúa mediante lenguaje natural
- Disponible en VS Code, GitHub y otros entornos

Acceso en VS Code

- **Activación:** Panel lateral o `ctrl+shift+l`
- **Requisitos**
 - Extensión instalada de Copilot
 - Sesión iniciada con cuenta de GitHub



Qué es GitHub ?

GitHub es una de las principales plataformas para **crear proyectos de manera colaborativa**

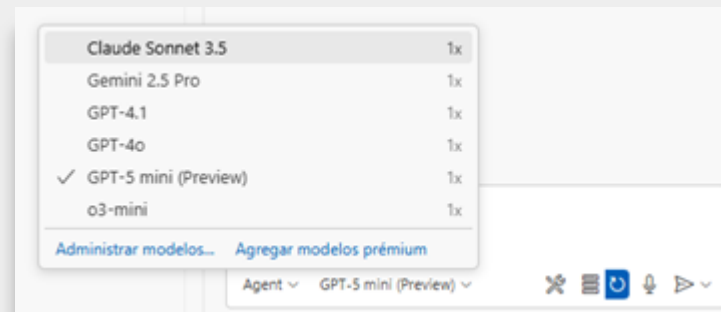
Para qué sirve?

Aloja cientos de miles de repositorios, la mayoría de los cuales son proyectos de código abierto lo que le permite a los usuarios entrar, dejar sus comentarios, e incluso proponer cambios directamente sobre el código de otras personas para contribuir a su mejora.

[GitHub Copilot](#) *es una herramienta de inteligencia artificial que asiste en la escritura de código y donde la seguridad se mantiene como un aspecto crítico. GitHub Copilot ayuda a escribir código más seguro y eficiente, pero siempre es importante revisar y validar el código generado.*

Selector de modelos

Elegir el motor de IA (GPT-5 mini, Gemini 2.5 Pro, Claude Sonnet 3.5) según velocidad o profundidad.



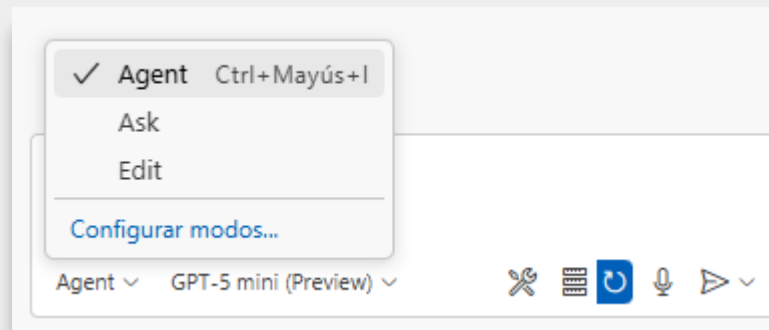
Motor de IA

Formas de utilizarlo

- **Autocompletado inteligente**
 - Sugiere código al escribir
 - Uso: **Tab** para aceptar, **Esc** para rechazar
 - Alternativas: **Alt +]** y **Alt + [**
- **Inline chat**
 - Conversación en el editor
 - Activar: **Ctrl + I**
 - Ejecutar: **Ctrl + Enter**
- **Chat completo**
 - Modo Agent, Ask y Edit
 - Activar: **Ctrl + Alt + I**
- **Acciones rápidas**
 - Icono de bombilla cuando detecta mejoras

Modos de conversación

- **Agent:** Tareas complejas y de seguimiento
- **Ask:** Consultas rápidas (modo más común)
- **Edit:** Modificaciones específicas



Se cambian desde el desplegable del chat

Comandos

Acciones rápidas sobre el código (/)

- `/explain`: explica código seleccionado
- `/fix`: sugiere correcciones para errores
- `/test`: genera casos de prueba
- `/doc`: crea documentación
- `/optimize`: mejora rendimiento del código

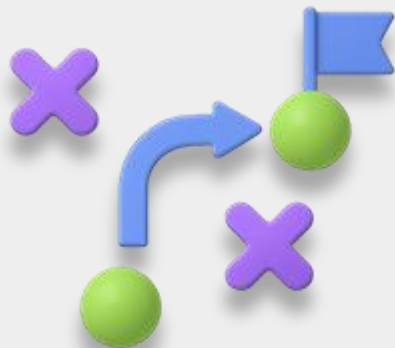
Referencias (#)

- Permiten mencionar issues, pull request o archivos
- Ejemplo: #42 para referirse a un issue

Contexto de trabajo (@)

- `@workspace`: usar todo el proyecto
- `@file`: solo el archivo actual
- `@selection`: código seleccionado
- `@terminal`: interactuar con la terminal
- `@vscode`: ayuda sobre VS Code

Contexto



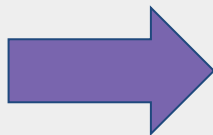
El contexto **es la información que Copilot toma como referencia** para generar respuestas.

- Puede ajustarse con comandos @
- Cuanto más preciso sea el contexto, más relevantes y útiles serán las respuestas

Identifica el comando



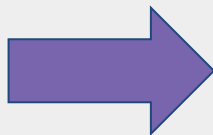
- Entender código complejo
- Solicitar ayuda sobre VS Code
- Identificar y corregir un bucle infinito
- Generar casos de prueba
- Buscar en todo el proyecto
- Crear documentación



Identifica el comando



- Entender código complejo
- Solicitar ayuda sobre VS Code
- Identificar y corregir un bucle infinito
- Generar casos de prueba
- Buscar en todo el proyecto
- Crear documentación



- /explain
- @vscode
- /fix
- /test
- @workspace
- /doc

Prompting

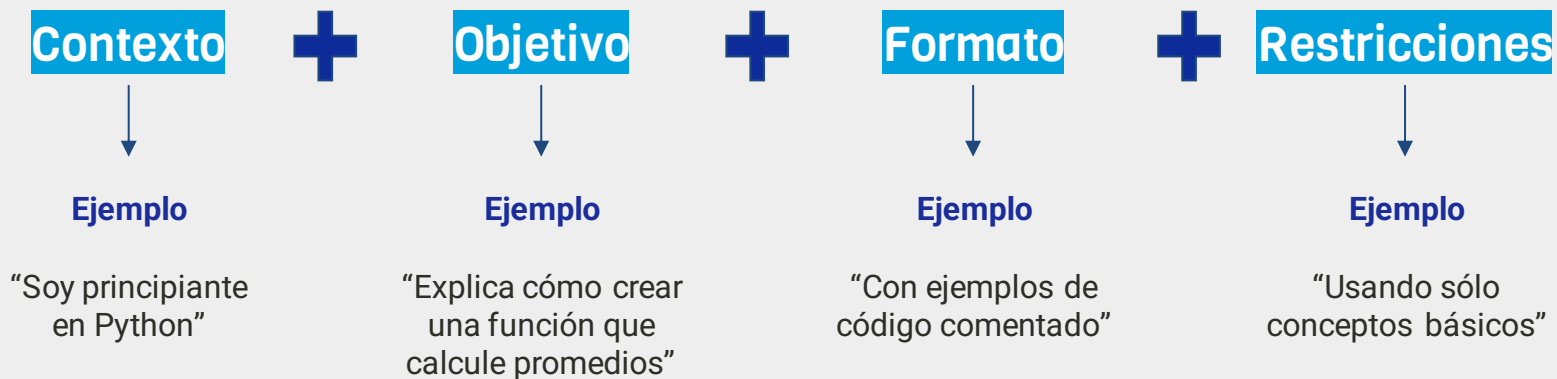
Es el proceso de **redactar instrucciones** claras para obtener respuestas útiles de una IA.

- Precisión en el lenguaje = mejores resultados
- Evitar ambigüedades y redundancias
- Usar formato estructurado cuando sea posible
- Incluir contexto relevante

Se recomienda incluir en el proyecto un archivo [Markdown \(.md\)](#) con [lineamientos](#) claros sobre cómo debe asistir Copilot

Prompts

Estructura



Prompts

Recomendaciones

Reglas generales

- Ser específico
- Proporcionar contexto
- Definir resultado esperado
- Incluir ejemplos

Mejores prácticas

- **Iterativos:** refinar con nuevas preguntas
- **Chaining:** usar la salida de uno como entrada del siguiente
- **Role prompting:** asignar un rol
- **Constraint prompting:** definir límites

Prompts

Tipos

- **Consulta:** “¿Qué es?”, “¿Cómo funciona?”
- **Creación:** “Crea”, “desarrolla”
- **Análisis:** “Analiza este código”, “identifica errores”
- **Optimización:** “Mejora este código”, “sugiere alternativas”
- **Documentación:** “Documenta esta función”, “crea un README”

Tips

Tips para un buen prompting

1. Ser **específico**: indicar qué se quiere y para qué.
2. Incluir **contexto**: lenguaje de programación, público objetivo, estilo
3. **Dar ejemplos** para guiar la respuesta.
4. Hacer preguntas **cortas y claras**.

+info

Prompts

Recomendaciones

Mejores prácticas

- **Iterativos:** refinar con nuevas preguntas
- **Chaining:** usar la salida de uno como entrada del siguiente
- **Role prompting:** asignar un rol
- **Constraint prompting:** definir límites



El dominio del prompting determina la efectividad de tu
colaboración con IA.

Buenas practicas/ Consejos

- La calidad de sus respuestas depende de la calidad de tus preguntas
(*Copilot no te va a leer la mente*)
- Revisar siempre el código (*tu eres el piloto*)
- Prestar especial atención a los bugs y a los errores de lógica
- No quieras correr antes de caminar, comienza con tareas sencillas
- No dependas demasiado del copiloto

¿Cómo pueden mejorarse estos prompts?



- “Haz una función que calcule algo matemático”
- “Este código tiene errores, ayúdame”
- “Explica machine learning con ejemplos y fácil”

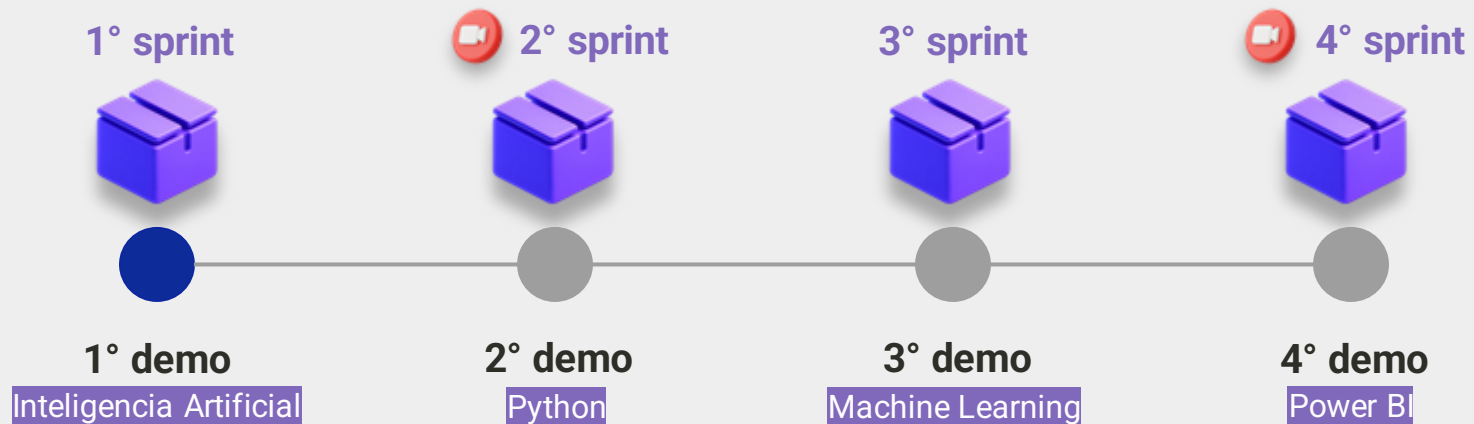
Proyecto

Tienda Aurelion

- **Documentación:** notebook Markdown
- **Desarrollo técnico:** programa Python
- **Visualización de datos:** dashboard en Power BI
- **Presentación oral:** problema, solución y hallazgos



Sprints Project



Características

1º demo: asincrónica

- **Tema:** Inteligencia Artificial
- **Medio:** Carpeta personal en Drive
- **Fecha límite:** 06/10
- **Devolución:** En salitas por equipo

Contenido

1º demo: asincrónica

Documentación (.md)

- Tema, problema y solución
- Dataset de referencia: fuente, definición, estructura, tipos y escala
- Información, pasos, pseudocódigo y diagrama del programa
- Sugerencias y mejoras aplicadas con Copilot

Programa (.py)

- Debe permitir obtener información del proyecto

Instrucciones (.md)

- Instrucciones para Copilot



Llamando a mi Copiloto

Trabajo en equipo



1. **Configurar Copilot Chat** en VS Code
2. **Crear carpeta del proyecto** en VS Code
3. Crear archivo **Instrucciones.md** con lineamientos
4. **Migrar programa .py** a la carpeta del proyecto
5. Generar **Documentación.md** con ayuda de Copilot
6. **Aplicar mejoras sugeridas por Copilot** en código y documentación



Retro

¿Cómo nos vamos?

- ¿Qué fue lo más útil de la clase?
- ¿Qué parte te costó más?
- ¿Qué te gustaría repasar o reforzar?