

Curso:

VSCode y GitHub Copilot



Tema 2:

Chat Agent

CONNECT

+ .

¿Qué hace diferente al chat?

La conversación continuada y guardable

¿Qué modos tenemos?

Ask, Edit, Plan... y Agent

Configuración

La carpeta .github

o



CONCEPTS:

Prompts , Instructions, Tools

Prompts: ¿Qué hacer?

- Comandos Slash 🎸
- Oficiales: /clear /doc /explain /fix /new /test
- Biblioteca de peticiones reutilizables:
 - en la carpeta .github/prompts
 - [command].prompt.md
- Enriquecidos con:
 - # : #file #selection
 - @ : @workspace @vscode @terminal @github

Instructions: ¿Cómo hacerlo?

- Generales en fichero
 - `.github/copilot-instructions.md`
 - `AGENTS.md`
- Contextuales:
 - en la carpeta `.github/instructions`
 - `[context].instructions.md`
- Una cabecera en Front-matter que indica:
 - propósito y
 - ficheros a los que aplica.

Tools: ¿Con qué hacerlo?

- Herramientas externas en lenguaje natural
- Ejemplo: API de GitHub
- Cada MCP dispone de un server con +múltiples Tools
- Puede requerir autenticación y configuración extra.
- <https://github.com/mcp>

CONCEPTUAL DEMO

Show me the code;

- ❑ Configure: instructions
 - ❑ Add: IRepository
 - ❑ Git: commit prompt

Context:

- Partiendo de la solución actual con las instrucciones de copilot

Workflow:

1. RF-2: `PUT /rockets/:id` Cambiar capacidad o rango.

Validation:

- ✓ RF-2: `PUT /rockets/:id`
 - ✓ 200 Detalles del cohete cambiados
 - ✓ 404 cohete no encontrado
- ✓ Bonus:
 - ✓ 400 con datos mal formados

CONCRETE PRACTICE

Activity-Bookings

CONCLUSION



- *¿Qué hemos aprendido?*
 - El modo agente será tu pair programmer
 - Configurar con instrucciones
 - Prompts as code
- *¿Qué haremos después?*
 - Usar los agentes especiales...