



# Temario del curso: Domina GitHub Copilot

Este curso está orientado a que el alumnado pase de usar GitHub Copilot como autocompletado a trabajar con flujos **agentic** completos en GitHub, con foco en calidad, gobernanza y DevOps.

Es un temario **muy ambicioso**, diseñado para exprimir al máximo GitHub Copilot en escenarios reales de desarrollo. Para facilitar el aprendizaje, las sesiones se grabarán y se compartirán con el alumnado para que puedan revisarlas tantas veces como necesiten.

Además, durante el curso y unos días de gracia posteriores, el alumnado dispondrá de un canal de Discord para resolver dudas y afianzar los contenidos vistos en clase.

## ■ Alcance y expectativas del curso

Este curso está orientado a uso profesional de GitHub Copilot en flujo real de equipo.

### Sí cubrimos:

- Uso práctico de GitHub Copilot en editor, chat, CLI y flujos con agentes.
- Personalización (MCP, skills, prompts, agentes) aplicada a proyectos reales.
- Calidad y gobernanza en pull requests y en el ciclo DevOps.

### No cubrimos:

- Fundamentos de programación, Git o DevOps desde cero.
- Ejercicios guiados en directo: las sesiones priorizan la explicación y las demos en vivo realizadas por la profesora. Se propondrán ejercicios sugeridos para practicar offline.

## ■ Estructura del curso

- Clase 1 · Fundamentos
- Clase 2 · Avanzado y personalización
- Clase 3 · Agentic DevOps

## ■ Prerrequisitos recomendados

- Cuenta de GitHub con acceso activo a GitHub Copilot (Individual, Pro, Business o Enterprise).
- Conocimientos básicos de Git/GitHub (commits, ramas, pull requests).
- VS Code instalado y actualizado (o editor compatible con GitHub Copilot).
- Terminal básica (navegación y ejecución de comandos).
- Permisos para instalar extensiones y herramientas de línea de comandos en tu equipo.

## ■ Clase 1 · Fundamentos de GitHub Copilot

### Objetivo



Entender qué es GitHub Copilot hoy, sus modos de uso y cómo integrarlo en el día a día de desarrollo.

## Contenidos

- Qué es GitHub Copilot y evolución desde autocompletado.
- Diferencia entre **asistente** y **agente**.
- Formas de interacción: editor, chat, inline, terminal y CLI.
- GitHub Copilot Chat: modos, modelos disponibles, tools y consumo (AI Credits / premium requests).
- Funcionalidades clave:
  - Code completion
  - Next Edit Suggestions
  - Slash commands
  - Computer vision
  - Integrated browser
  - Plan mode
- Presentación de la aplicación de ejemplo del curso (React + .NET), extrapolable a cualquier lenguaje de programación moderno.
- Mapa visual del IDE: identificación rápida de las superficies de GitHub Copilot (iconos, accesos y puntos de entrada más usados).
- Introducción a Cloud Agents y GitHub Copilot CLI.
- Ejercicios sugeridos a desarrollar offline.

## Resultado esperado

El alumnado será capaz de usar GitHub Copilot de forma productiva en tareas diarias de implementación y revisión.

## ■ Clase 2 · Avanzado: personalización y extensibilidad

### Objetivo

Personalizar GitHub Copilot para adaptarlo al contexto del proyecto y del equipo.

### Contenidos

- Personalización de GitHub Copilot.
- **MCP Servers:**
  - Qué son y por qué usarlos
  - Instalación y gestión
  - Servidor MCP básico



- **Skills** y su aplicación práctica.
- **Prompts** reutilizables y generación de prompts desde conversaciones.
- **AGENTS.md** y **nested AGENTS.md**.
- **Custom Agents** y demo de subagentes.
- **Agent Hooks**.
- Chat Customizations, Chat Debug y evaluaciones.
- Ejercicios sugeridos a desarrollar offline.

## Resultado esperado

El alumnado será capaz de definir un entorno de GitHub Copilot ajustado al dominio, convenciones y procesos de su equipo.

## ■ Clase 3 - Agentic DevOps con GitHub

### Objetivo

Aplicar GitHub Copilot y agentes dentro de un flujo DevOps gobernado, trazable y auditable.

### Contenidos

- GitHub Agentic AI Developer: marco de trabajo y ciclo de vida.
- Cultura DevOps aplicada a agentes.
- GitHub como plano de control para trabajo agentic.
- Fases del flujo con agentes:
  - Planificación
  - Desarrollo
  - CI/CD
  - Mantenimiento
- Control de calidad y gobernanza en GitHub:
  - Pull Requests y reviews
  - Status checks
  - CODEOWNERS
  - Rulesets
  - Environments
- Riesgos, antipatrones y mitigaciones.
- Trazabilidad y observabilidad de contribuciones generadas por agentes.
- Rastro mínimo de auditoría y modelo de contribución.
- Rúbrica práctica para evaluar PRs de agentes (qué revisar y cómo se ve una buena contribución).
- GitHub Agentic Workflows:



- Estructura y frontmatter YAML (triggers, permisos, tools, safe outputs)
- Mantenimiento de documentación
- Ejemplos de integraciones con el ecosistema
- Ejercicios sugeridos a desarrollar offline.

## Resultado esperado

El alumnado será capaz de incorporar IA agentic en su SDLC con controles reales de calidad, seguridad y cumplimiento.

### ■ Qué se lleva el alumnado al finalizar

- Criterio para elegir cuándo usar asistente vs agente.
- Comprensión práctica de cómo usar GitHub Copilot Chat, CLI y Cloud Agents a partir de casos reales y demos en vivo realizadas por la profesora.
- Conocimiento de las opciones de personalización de GitHub Copilot (MCP, Skills, prompts y agentes) y cuándo aporta valor cada una.
- Marco de referencia para incorporar Agentic DevOps con control del proceso, calidad y gobernanza.

### ■ Enfoque metodológico

- Primera parte de cada sesión orientada a fundamentos y marco conceptual.
- Segunda parte centrada en demos en vivo, donde la profesora mostrará la aplicación práctica de lo explicado.
- Énfasis en expectativas realistas: productividad con gobernanza, no automatización sin control.

### ■ Nota final

El temario podrá ajustarse durante el curso para incorporar novedades, anuncios o cambios relevantes que aparezcan mientras se desarrolla la formación.