

Domina GitHub Copilot y da el salto al desarrollo con agentes

Clase 1: Básicos

Introducción a GitHub Copilot

- Qué es y qué no es (expectativas reales)
- Cómo funciona por debajo (contexto, embeddings, prompt implícito)
- Licencias y planes (Individual, Business, Enterprise)

Primeros pasos

- Activación y configuración inicial
- IDEs soportados (VS Code, JetBrains, Visual Studio, Neovim...)
- Extensiones y requisitos

Code completions: donde nació todo

- Cómo leer y aceptar sugerencias
- Sugerecias inline vs. panel de sugerencias
- Cómo mejorar las sugerencias con contexto:
 - Comentarios
 - Nombres de variables
 - Archivos abiertos
- Ghost text y atajos de teclado clave

Chat

- Modos:
 - Inline chat
 - Panel lateral
 - Quick chat
- Slash commands:
 - `/explain`
 - `/fix`
 - `/tests`
 - `/doc`
- Variables de contexto:
 - `#file`
 - `#selection`
 - `#codebase`
- Modelos disponibles y cuándo cambiar de modelo

Modo agente

- Qué cambia respecto al chat normal
- Herramientas que puede usar:
 - Terminal
 - Filesystem
 - Extensiones MCP
- Cuándo usarlo y cuándo no

Buenas prácticas desde el principio

- Cómo escribir buenos prompts
 - Revisar siempre el output
 - El dev sigue siendo responsable del código
-

Clase 2: Avanzado

Modo agente: arquitectura y extensibilidad

- Cómo funciona el agente
- **AGENTS.md**: definir comportamiento del agente a nivel de repo
- Skills:
 - Qué son
 - Cómo se definen
 - Cuándo usarlas
- Agent Hooks:
 - Reaccionar a eventos del agente (PostToolUse, etc.)
- Plugins:
 - Extender las capacidades del agente

MCP Servers

- Qué es MCP y el rol que juega en el ecosistema de GitHub Copilot
- MCP servers built-in vs. custom
- Cómo conectar y configurar un MCP server
- Demo: Copilot usando un MCP server real

Un MCP Server custom

GitHub Copilot CLI (public preview)

- Qué es: agente autónomo en la terminal
- Instalación
- Qué puede hacer:
 - **plan**
 - **autopilot**
 - **fleet**

GitHub Copilot SDK

- Qué es: SDKs para integrar el agente de Copilot CLI en tus propias apps y servicios
 - Arquitectura:
 - Tu app → SDK client → JSON-RPC → Copilot CLI (server mode)
 - SDKs disponibles
 - Casos de uso:
 - Experiencias personalizadas
 - Automatizaciones
 - Herramientas internas
 - Demo: agente propio construido sobre el SDK
-

Clase 3: Agentic DevOps

Qué es Agentic DevOps

- Del autocompletado al agente: la evolución del rol del dev
- El ciclo completo:
 - Planificación → Desarrollo → CI/CD → Mantenimiento

Fase de planificación con Copilot

- Descomposición de requisitos en issues con Copilot
- Generación de architecture decision records (ADRs)
- Wireframes y specs técnicas asistidas

Fase de Desarrollo: Copilot Cloud Agent

- Asignar tareas a Copilot desde GitHub Issues
- Flujo completo:
 - Issue → Rama → PR → Review
- Qué delegar y qué no (todavía)

Fase de CI/CD con GitHub Actions

- Generación y mantenimiento de workflows con Copilot
- Autofix de errores en pipelines
- Revisión automática de PRs

GitHub Agentic Workflows

- Qué son y cómo se componen
- Casos de uso:
 - Dependabot + Copilot
 - Security alerts → fix automático
 - Issue triage
- Orquestación de múltiples agentes en un mismo repo

Fase de mantenimiento

- Análisis de incidencias y debugging asistido

- Refactoring continuo guiado por Copilot
- Documentación viva:
 - Mantener docs en sync con el código

Reflexión final

- El dev como orquestador: qué habilidades cobran más valor

Q&A