

# Copilot para Desarrolladores Web (.Net y Node)

Argenis Gonzalez

Sesión 1 - 11 Noviembre 2025

1. ¿Qué es un MCP Server?
2. Top 5 MCP Servers para Desarrollo
3. Chat vs Autocompletado
4. Casos de Uso Fullstack
5. Copilot vs ChatGPT
6. Limitaciones y Riesgos
7. ¿Qué es un Promt?
8. Tipos de Promts
9. Promt Genérico vs Efectivo
10. ¿Qué es un MCP Server?

# ¿Qué es GitHub Copilot?

-  **Asistente AI integrado en tu editor**
-  **Entrenado con billones de líneas de código público**
-  **Sugerencias en tiempo real** mientras escribes

## Tecnología detrás

- Modelos LLM (Large Language Models)
- Basado en OpenAI Codex
- Aprende de tu contexto actual



**No reemplaza al desarrollador, lo potencia**

# Integración en Editores

## Dónde funciona

- -  **Visual Studio Code** (Principal)
  -  **Visual Studio**
  -  **JetBrains IDEs** (IntelliJ, WebStorm)
  -  **Neovim** (para valientes)

## Instalación sencilla

- 1. Extensión oficial
  2. Login con GitHub
  3. ¡Listo para usar!

# Chat vs Autocompletado



## Autocompletado Inteligente

- Sugerencias **mientras escribes**
- Aparece en **gris** (texto fantasma)
- **Tab** = Aceptar | **Esc** = Rechazar



## Copilot Chat

- Conversación interactiva
- Preguntas y explicaciones
- Refactorización guiada
- Comandos con / (ej: /explain, /fix)

Usa ambos según necesites

# Casos de Uso Fullstack

## Backend

-  Controladores API REST
-  Middlewares de autenticación
-  Queries a base de datos
-  Validaciones y DTOs

## Frontend

-  Componentes React/Angular
-  Estilos CSS/SCSS
-  Llamadas HTTP
-  Formularios y validación

Testing y documentación automática

# Copilot vs ChatGPT

Aspecto	GitHub Copilot	ChatGPT
Contexto	Tu proyecto completo	Solo lo que escribes
Integración	Dentro del editor	Navegador separado
Velocidad	Instantáneo (autocompletar)	Respuesta por mensaje
Uso ideal	Escribir código	Aprender conceptos

👉 Copilot = Productividad | ChatGPT = Aprendizaje

# Limitaciones y Riesgos

## Ten en cuenta

- ❖ **NO ES PERFECTO** - Revisa siempre el código
- **Seguridad** - Puede sugerir código vulnerable
- **Licencias** - Código público puede tener copyright
- **Dependencia** - No pierdas habilidad de programar

## En proyectos grandes

- Contexto limitado (~8000 tokens)
- No entiende arquitectura completa
- Puede generar inconsistencias



**TÚ eres el responsable del código final**

# ¿Qué es un Prompt?

- Un prompt es la instrucción o pregunta que le das a una IA (como Copilot, ChatGPT, Claude, etc.) para obtener una respuesta.
- - 1. **\*\*CLARIDAD\*\*** - Sé específico
  - 2. **\*\*CONTEXTO\*\*** - Tipos y reglas
  - 3. **\*\*EJEMPLOS\*\*** - Muestra patrones

# Tipos de Prompts:

- σ 1. Descripción

**Define qué necesitas de forma clara**

- 2. Contexto

**Proporciona información de fondo relevante**

- 3. Tarea

**Especifica la acción que quieras que realice**

- 4. Ejemplo\*\*

**Muestra un modelo de lo que esperas obtener**

# Prompt Genérico vs. Efectivo

- 
- **✗ Prompt Genérico**  
" Genera un controller en C# "  
**\*\*Problema:\*\* Simple, sin contexto, código genérico y poco útil**
-

# Prompt Genérico vs. Efectivo



## Prompt Efectivo

"Genera un API Controller en C# .NET 8 para gestionar productos de un e-commerce.

○ Incluye: endpoints CRUD (GET, POST, PUT, DELETE), validación de datos con DataAnnotations, manejo de respuestas HTTP y documentación con comentarios XML. Usa inyección de dependencias para el repositorio."

Incluye:

- Descripción: API Controller para gestión de productos
- Contexto: E-commerce, .NET 8, buenas prácticas
- - Tarea: Crear CRUD con validaciones y documentación
- Ejemplo de formato: Especifica patrón (inyección de dependencias, DataAnnotations)

\*\*Resultado:\*\* Código específico, estructurado y listo para usar

# Prompt Genérico vs. Efectivo

## Model Context Protocol (2025)

-  \*\*Estándar abierto\*\* para conectar IA + herramientas
-  \*\*Arquitectura cliente-servidor\*\* (JSON-RPC)
-  \*\*Seguro y stateful\*\*

### Cómo funciona:

\*\*LLM\*\* → \*\*MCP\*\* → \*\*Herramienta externa\*\*

- Local (STDIO) o Remoto (HTTP)
- Descubre capacidades y gestiona permisos

### ○ Beneficios:

- Workflows agénticos (Copilot + DBs/repos)
- Soluciona problema NxM (muchos modelos × herramientas)

# ¿QUÉ ES UN MCP SERVER?

## Model Context Protocol (2025)

-  \*\*Estándar abierto\*\* para conectar IA + herramientas
-  \*\*Arquitectura cliente-servidor\*\* (JSON-RPC)
-  \*\*Seguro y stateful\*\*

### Cómo funciona:

\*\*LLM\*\* → \*\*MCP\*\* → \*\*Herramienta externa\*\*

- Local (STDIO) o Remoto (HTTP)
- Descubre capacidades y gestiona permisos

### ○ Beneficios:

- Workflows agénticos (Copilot + DBs/repos)
- Soluciona problema NxM (muchos modelos × herramientas)

# TOP 5 MCP SERVERS

## 📁 #1 File System

- Acceso al sistema de archivos local
- Lee/escribe/crea directorios
- **Uso:** Generar y guardar código directamente

## 🧠 #2 Sequential Thinking

- Razonamiento paso a paso estructurado
- Mejora decisiones complejas del LLM
- **Uso:** Debugging y arquitectura de código

## 📚 #3 Context7

- Gestión avanzada de contexto
- Memoria extendida para conversaciones
- **Uso:** Mantener coherencia en proyectos grandes

## 🔍 #4 Tavily

- Búsqueda web especializada para IA
- Acceso a información actualizada
- **Uso:** Consultar docs, APIs o tecnologías recientes

## 🗄️ #5 Base de Datos (PostgreSQL/MongoDB)

- Inspecciona esquemas y ejecuta queries
- Operaciones read-only seguras
- **Uso:** Validaciones y análisis de datos en desarrollo

