

Inteligencia Artificial Generativa Para la Ciencia de Datos



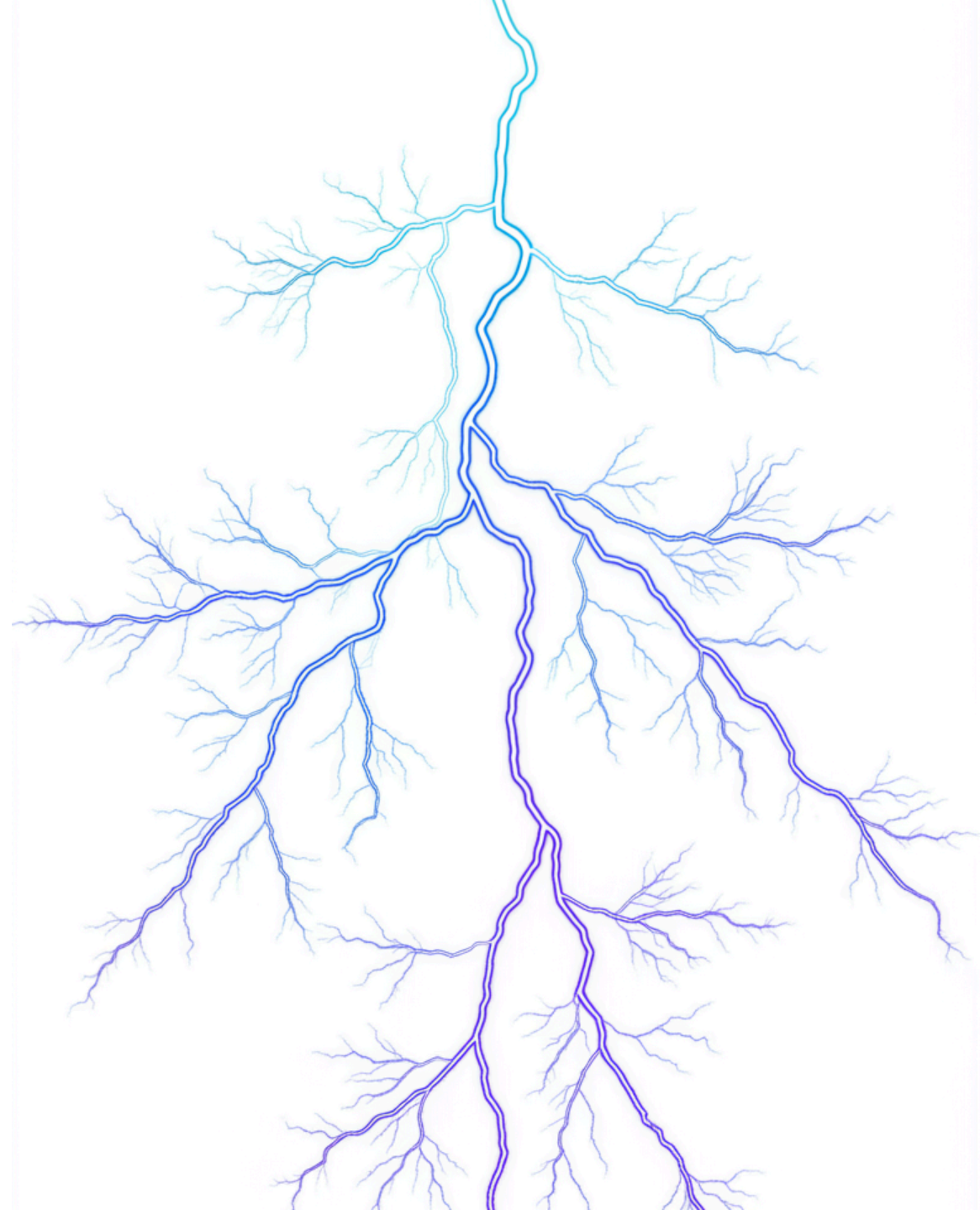
Profesor

Juan Camilo Vega Barbosa

Consultor IA - Ingeniero IA/ML

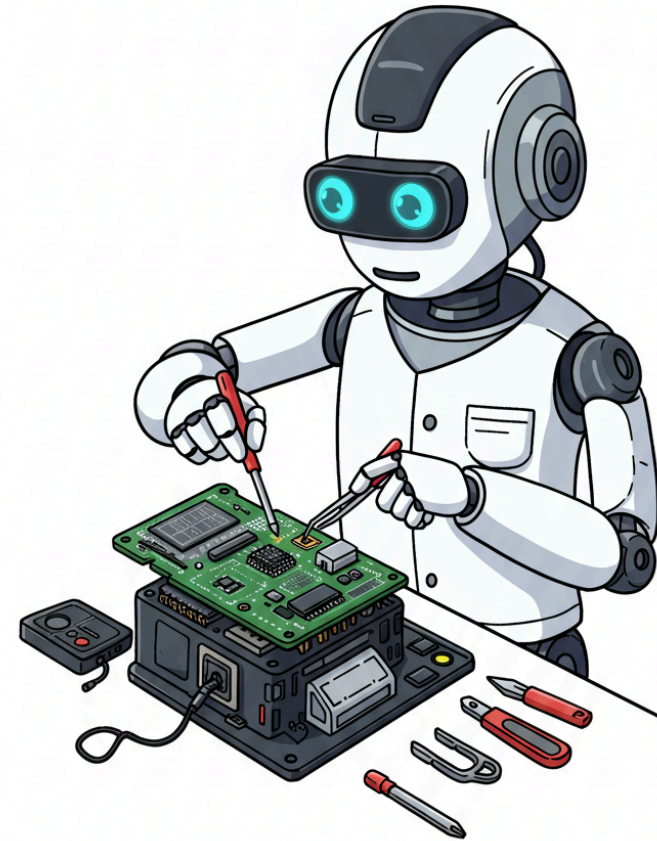


LinkedIn



Model Context Protocol (MCP)

El estándar abierto para conectar
IA con el mundo real





¿Qué es MCP?

El **Model Context Protocol (MCP)** es un estándar abierto desarrollado por Anthropic que permite a las aplicaciones de IA conectarse de manera segura y estandarizada con fuentes de datos externas, APIs y herramientas.



Piénsalo como: El puerto USB-C de las aplicaciones de IA. Así como USB-C estandariza las conexiones entre dispositivos, MCP estandariza la conexión entre modelos de IA y sistemas externos.



Objetivo principal: Eliminar las integraciones fragmentadas y reemplazarlas con un protocolo universal.



Arquitectura de MCP

MCP utiliza una **arquitectura cliente-servidor** inspirada en el Language Server Protocol (LSP):



Host: Aplicación principal (Claude Desktop, Cursor, etc.)

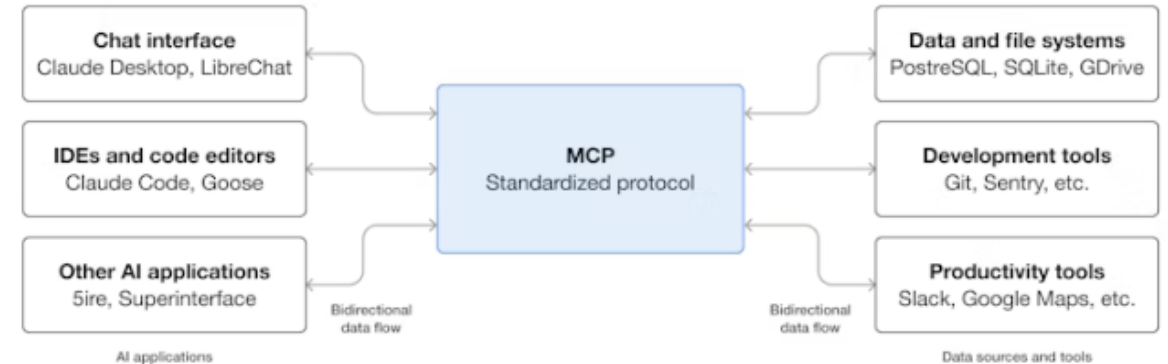


Client: Interfaz que vive dentro del Host



Server: Conecta con herramientas y datos específicos

Fuente: [Documentación oficial MCP](#)





¿Cómo Funciona MCP?

1. 🍌 **Inicialización:** El cliente MCP se conecta con los servidores MCP
2. 🔍 **Descubrimiento:** El cliente consulta qué capacidades ofrece cada servidor
3. 💬 **Solicitud del Usuario:** El usuario envía un prompt al modelo de IA
4. 🧠 **Decisión de la IA:** El modelo decide qué herramientas necesita usar
5. ⚡ **Ejecución:** El cliente MCP ejecuta las herramientas en el servidor correspondiente
6. ✉️ **Respuesta:** Los resultados se integran en el contexto de la IA

¿Por Qué Usar MCP?

 **Interoperabilidad:** Un solo protocolo para múltiples modelos y sistemas

 **Facilidad de Integración:** API clara y extensible para desarrolladores

 **Soporte Multiplataforma:** Compatible con múltiples lenguajes

 **Seguridad:** Autenticación y autorización robustas

 **Empresas que ya usan MCP:** Block (Square), Apollo, Zed, Replit, Codeium, Sourcegraph

Instalación: Paso a Paso

1 Instalar Node.js

Descargar desde nodejs.org o usar un gestor de paquetes:

macOS con Homebrew:

```
brew install node
```

Windows con Chocolatey:

```
choco install nodejs
```

Ubuntu/Debian:

```
sudo apt install nodejs npm
```

Instalación: Claude Desktop

2 Descargar Claude Desktop

Descarga **Claude Desktop** desde:

 [Sitio oficial de Claude](#)

 Disponible para:

- Windows
- macOS
- Linux

Una vez instalado, **inicia Claude Desktop** y ciérralo para que cree los archivos de configuración necesarios.

 **Importante:** Claude Desktop debe ejecutarse al menos una vez para crear la estructura de configuración antes de continuar.

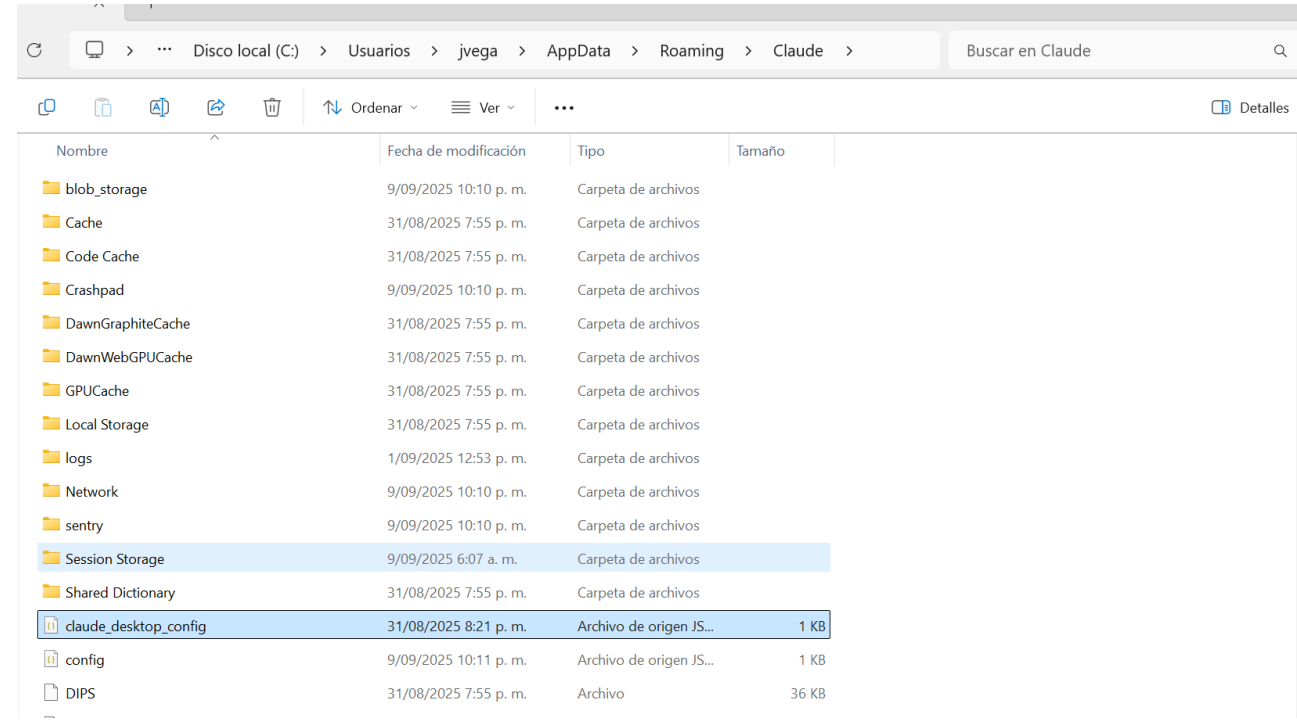
⚙ Configuración de MCP

3 Abrir Configuración de Desarrollador

🔧 En Claude Desktop:

1. Ve a configuración ⚙
2. Busca la sección "Developer"
3. Haz clic en "Edit Config"
4. Se abrirá el archivo

`claude_desktop_config.json`






Configurar MCP Servers

4 Ejemplo de Configuración

```
{
  "mcpServers": {
    "postgres": {
      "command": "npx",
      "args": [
        "-y",
        "@modelcontextprotocol/server-postgres",
        "postgresql://localhost/mydb"
      ]
    }
  }
}
```

 **Tip:** Usa rutas absolutas para evitar problemas de configuración. Puedes obtener la ruta actual con `pwd` (Linux/macOS) o `cd` (Windows).

Instalación Adicional de MCP

Instalación desde Terminal

```
# Instalar servidor de GitHub
npm install -g @modelcontextprotocol/server-github

# Instalar servidor de PostgreSQL
npm install -g @modelcontextprotocol/server-postgres

# Instalar servidor de Git
npm install -g @modelcontextprotocol/server-git

# Verificar instalación
npx @modelcontextprotocol/server-github --help
```

 **Recomendación:** Instala los servidores MCP globalmente para facilitar su configuración y reutilización en múltiples proyectos.

Dónde Encontrar MCP Servers

Repositorio Oficial:

- [Servidores oficiales MCP](#)

Marketplace MCP:

- [MCP.so](#) (16,509+ servidores)

Documentación:

- [Sitio oficial MCP](#)

Directorios Especializados:

- [Directorio de servidores MCP](#)
- [Buscador de servidores MCP](#)



Seguridad y Buenas Prácticas



Autenticación y Autorización:

- MCP implementa OAuth 2.1 con PKCE obligatorio
- Validación de tokens específicos por audiencia
- Rotación automática de tokens de refresh




Riesgos de Seguridad:

- **Confused Deputy:** Servidores MCP pueden actuar fuera del contexto del usuario
- **Token Leakage:** Tokens robados pueden ser reutilizados
- **Data Mining:** Concentración de acceso a múltiples servicios

Recomendaciones de Seguridad

Mejores Prácticas:

- Usa solo servidores MCP de fuentes confiables
- Implementa principio de menor privilegio
- Valida y sanitiza todas las entradas
- Implementa rate limiting
- Audita y monitorea todas las conexiones

 **Nunca hagas:** - Usar tokens con audiencia incorrecta - Pasar tokens entre servicios - Instalar MCP servers no verificados - Exponer credenciales en logs

Ejemplo: MCP GitHub Server

Permite interactuar con repositorios de GitHub, crear issues, revisar pull requests y gestionar el código a través de MCP.

Configuración:

```
{
  "mcpServers": {
    "github": {
      "command": "npx",
      "args": [
        "-y",
        "@modelcontextprotocol/server-github"
      ],
      "env": {
        "GITHUB_PERSONAL_ACCESS_TOKEN": "your-token-here"
      }
    }
  }
}
```

Ejemplo: MCP Playwright Server

Proporciona automatización de navegadores web para testing end-to-end y web scraping usando Playwright.

```
{
  "mcpServers": {
    "playwright": {
      "command": "npx",
      "args": [
        "@playwright/mcp@latest"
      ]
    }
  }
}
```

 Fuente: [Microsoft Playwright MCP](#)

 **Capacidades:** Navegación web, captura de pantallas, testing automatizado, extracción de datos.

Ejemplo: MCP Google Cloud Server

Permite interactuar con servicios de Google Cloud Platform a través de MCP.

```
{
  "mcpServers": {
    "gcloud": {
      "command": "npx",
      "args": [
        "-y",
        "@google-cloud/storage-mcp"
      ]
    }
  }
}
```

 Fuente: [Google Cloud MCP](#)


 **Capacidades:** Gestión de recursos GCP, BigQuery, Cloud Storage, Compute Engine.

 **Nota:** Reemplaza `storage-mcp` por el paquete específico que necesites (ej: `bigquery-mcp`).

Ejemplo: MCP PostgreSQL Server





Permite ejecutar consultas SQL y explorar esquemas de bases de datos PostgreSQL.

```
{
  "mcpServers": {
    "postgres": {
      "command": "npx",
      "args": [
        "-y",
        "@modelcontextprotocol/server-postgres",
        "postgresql://user:pass@localhost/db"
      ]
    }
  }
}
```

 **Capacidades:** Consultas SQL, exploración de esquemas, análisis de datos, optimización de queries.

Reiniciar y Probar

5 Verificación Final

1.  **Guardar Configuración:** Guarda el archivo `claude_desktop_config.json`
2.  **Reiniciar Claude Desktop:** Cierra y vuelve a abrir la aplicación
3.  **Verificar Conexión:** Busca el ícono de herramientas en Claude Desktop
4.  **Probar Funcionalidad:** Haz una consulta que requiera el MCP configurado







Próximos Pasos en Clase



Laboratorio Práctico:

En la siguiente sesión implementaremos y probaremos:

-  **GitHub MCP** - Gestión de repositorios y código
-  **Playwright MCP** - Automatización de navegadores
-  **Google Cloud MCP** - Integración con servicios GCP
-  **PostgreSQL MCP** - Consultas y análisis de datos

Veremos configuración avanzada, manejo de errores, y mejores prácticas de seguridad en entornos reales.

Recursos Adicionales

Documentación Oficial:

- [Documentación completa MCP](#)
- [Repositorio principal en GitHub](#)
- [Especificación técnica del protocolo](#)

Herramientas y SDKs:

- [SDK de Python para MCP](#)
- [SDK de TypeScript para MCP](#)
- [Implementación en Go](#)

Recursos Adicionales

Tutoriales y Ejemplos:

- [Guías paso a paso oficiales](#)
- [Ejemplos de servidores MCP](#)
- [Templates para crear servidores](#)

Ecosistema MCP:

- [Directorio de 1,000+ servidores](#)
- [Soporte oficial de Anthropic](#)
- [Claude Desktop con MCP](#)