

Práctica: Desarrollo de un Agente Basado en LLM con FastMCP y la API de OpenAI

1. Descripción general

En esta práctica, se deberá diseñar y desarrollar una aplicación que implemente un **agente conversacional basado en un LLM** utilizando:

- **API de OpenAI** (Modelos con soporte para herramientas / MCP)
- **FastMCP**, para definir y alojar herramientas personalizadas en el propio proyecto.

El propósito del agente será elegido libremente por el grupo de estudiantes. Ejemplos orientativos:

- Recepción de hotel (gestión de reservas, información al cliente).
- Servicio de atención al estudiante.
- Gestor de instalaciones deportivas (reservas, horarios, información).
- Asistente para un comercio electrónico.
- Asistente de soporte técnico.
- Agente para un museo, centro cultural o biblioteca.

El objetivo principal consiste en que el agente **razone, llame y combine herramientas** (MCP) definidas por el grupo y, si los desean, **utilice además MCP públicos**, como Google Calendar u otros disponibles.

2. Requisitos funcionales

La aplicación debe incluir:

Un agente conversacional basado en un modelo de OpenAI

- Uso de un modelo compatible con herramientas / MCP (como los modelos recientes de la API).
- Gestión de diálogo básica.
- Capacidad para invocar herramientas cuando el modelo lo considere necesario.

Desarrollo de al menos un MCP propio mediante FastMCP

El MCP creado podrá realizar, por ejemplo:

- Acceso a datos locales (ficheros, inventario, información de instalaciones...).
- Operaciones CRUD sobre una pequeña base de datos o JSON local.
- Consultas a recursos públicos.
- Gestión de agenda o reservas.
- Cálculos, conversión de datos o funciones utilitarias.

Integración opcional de MCP públicos

Se valorará:

- Uso de **Google Calendar MCP** u otras herramientas públicas existentes.
- Combinación fluida entre la herramienta propia y las herramientas externas.

Aplicación final ejecutable

La aplicación puede ser un servicio tipo **CLI** interactivo, una pequeña aplicación web (p. ej., con Flask o FastAPI) o un bot de mensajería (Telegram).

Debe permitir probar el agente y observar:

- Peticiones al modelo.
- Llamadas a herramientas.
- Flujo conversacional coherente.

3. Documentación recomendada

Se recomienda apoyarse en la siguiente documentación:

- **OpenAI – Herramientas y MCP**
<https://platform.openai.com/docs/guides/tools>
- **FastMCP – Introducción y guía de desarrollo**
<https://gofastmcp.com/getting-started/welcome>