



Crear Tool MCP



¿Sabías que tus
agentes puedes
unas herramientas
externas de forma
segura y
estandarizada?



Comienza con una función de Python

En este caso comenzamos con una función que obtiene la lista de *issues* de un repositorio

```
# Tienes una función normal
async def list_repository_issues(owner: str, repo_name: str):
    """Lista los issues de un repo de GitHub."""
    # ...lógica de la función...
```



Añade el decorador *@mcp.tool()*

¡Listo! Ya puedes utilizarlo. Tu servidor MCP ya incluirá esta Tool

```
# ¡Conviértela en una Tool para tu IA!
@mcp.tool()
async def list_repository_issues(owner: str, repo_name: str):
    """Lista los issues de un repo de GitHub."""
    # ...lógica de la función...
```




Controla quién
puede y quién no
acceder a tu Tool
MCP



Añade tags a tus Tools

Creamos Tools con diferentes tags

```
# Herramienta pública
@mcp.tool(tags={"public"})
async def list_repository_issues(...):
    # ...devuelve 10 issues

# Herramienta privada para desarrollo
@mcp.tool(tags={"private", "development"})
async def list_more_repository_issues(...):
    # ...devuelve 100 issues
```



Crea MCPs con los
tags que quieras
incluir

Solo se podrán usar las
Tools con el tag *public*

```
# El servidor solo expondrá las tools con el tag "public"
mcp = FastMCP(
    name="GitHubMCP",
    include_tags={"public"}
)
```




¿Tool para desarrollo?

¿Has creado una Tool para imprimir la primera issue y depurar tu Tool?

```
# Una tool que no quieres exponer
@mcp.tool(tags={"public", "first_issue"})
async def first_repository_issue(...):
    # ...
```




Crea MCPs con los tags que quieras excluir

Se excluirán las Tools con el tag *first_issue*

```
# Incluye las públicas, pero excluye las de "first_issue"
mcp = FastMCP(
    name="GitHubMCP",
    include_tags={"public"},
    exclude_tags={"first_issue"}
)
```