

 **MCP + Python = ¡Magia!**

**Cómo hacer que tu IA hable con tu ERP sin
morir en el intento**

¿Quién soy?

- Programo en python desde el 2001
- Un desarrollador que se cansó de hacer copy-paste entre [ChatGPT] y Otras Apps
- En 30 años aprendí que cada vez que automatizo algo, creo otra actividad manual

El Problema

Lo que veo a cada rato:

1. Copiar pregunta de ChatGPT
2. Buscar en su app favorita manualmente
3. Copiar datos
4. Pegar en ChatGPT/Gemini/Claude.....
5. Repetir hasta el cansancio... 😴

¿No sería mejor si la IA pudiera hablar directamente con las Apps?




Una Solución: MCP (Por Ahora)

Model Context Protocol

- Es como WhatsApp pero para que las IAs hablen con tus aplicaciones
- Creado por Anthropic (los de Claude)
- Open source y gratis (¡como la pizza en los meetups antiguos!)

¿Qué vamos a construir hoy?

Un servidor MCP super simple que:

-  Se conecta a Odoo 18 SaaS (Es lo free más rápido que encontré)
-  Permite a Claude leer y escribir datos
-  No requiere PhD en ingeniería

La Arquitectura



(No te asustes, es simple), es básicamente un traductor

Componentes

```
odoo-mcp-simple/  
├── server.py      # El cerebro 🧠  
├── odoo_client.py # El mensajero 📬  
├── config.json    # Los secretos 🔒  
└── requirements.txt # Las dependencias 📦
```



El Cliente Odoo

```
class OdooClient:
    def __init__(self, url, db, username, password):
        self.url = url
        self.db = db
        # XML-RPC es retro pero funciona 🕒
        self.common = xmlrpc.client.ServerProxy(f'{url}/xmlrpc/2/common')
        self.uid = self.common.authenticate(db, username, password, {})

    def search(self, model, domain):
        # Buscar cosas en Odoo
        return "🔍 Encontré algo!"
```


Las Herramientas MCP

```
@mcp.tool()
async def search_records(model: str, filters: list):
    """Busca registros como un detective 🕵️"""
    return odoo_client.search(model, filters)

@mcp.tool()
async def create_record(model: str, data: dict):
    """Crea registros como un mago ✨"""
    return odoo_client.create(model, data)
```

⚙️ Configuración MCP en (Cualquier cosa)

```
{  
  "mcpServers": {  
    "odoo-mcp": {  
      "command": "python",  
      "args": ["/ruta/a/tu/proyecto/server.py"],  
      "env": {}  
    }  
  }  
}
```

Ejemplo completo de configuración

```
{
  "mcpServers": {
    "odoo-simple": {
      "command": "python3",
      "args": ["/home/usuario/odoo-mcp-simple/server.py"],
      "env": {
        "PYTHONPATH": "/home/usuario/odoo-mcp-simple"
      }
    }
  }
}
```

3. Reiniciar Claude y ¡listo! 🎉



¡DEMO TIME!

(La parte donde todo puede salir mal)

Live Session

Vamos a:

1. Conectarnos a `uoft-demo.odoo.com`
2. Crear clientes para el CRM
3. Interactuar con algunos objetos
4. Ver cómo Claude maneja todo solito (Siempre Falla)

¿Qué podría salir mal? 😅

Caso de Uso Real #1

"Claude, ¿cuántos clientes tenemos en Chile?"

```
# Antes: 10 minutos navegando en Odoo  
# Ahora:  
await search_records('res.partner',  
                     [ ('country_id.name', '=', 'Chile') ] )  
# ¡2 segundos! 🚀
```



Caso de Uso Real #2

"Claude, crea una orden de venta para el cliente ACME"

Claude automáticamente:

1. Busca el cliente
2. Obtiene sus datos
3. Crea la orden
4. Te muestra el resultado

Todo sin tocar Odoo 🎉



Errores Comunes

Error #1: Credenciales incorrectas

"🙄 Authentication failed" # → *Revisa tu password*

Error #2: Modelo no existe

"🧑 Model 'res.partner' not found" # → *Es 'partner'!*

Error #3: Sin permisos

"🚫 Access denied" # → *Habla con tu admin*





Mejoras Posibles

- **LittleLLM:** Para crear una KB con lo que el sistema aprende!
- **Seguridad:** Tokens JWT en vez de passwords
- **Performance:** Cache de consultas frecuentes
- **Analytics:** Tracking de uso de la API
- **Más IA:** Que Claude sugiera optimizaciones



Osea...





- **Versión 1.0**
 - CRUD básico 
- **Versión 2.0**
 - Workflows automáticos
 - Reportes inteligentes
- **Versión 3.0** (Cuando consiga café)
 - IA que programa a Odoo sola
 - Skynet? 



Ventajas de MCP

- ✓ **Simple:** No necesitas ser un genio
- ✓ **Rápido:** Minutos para implementar
- ✓ **Flexible:** Funciona con cualquier Odoo
- ✓ **Gratis:** Como el aire (pero mejor)

Consideraciones





-  **Seguridad:** No hardcodees passwords
-  **Rate Limits:** Odoo SaaS tiene límites
-  **Testing:** Prueba en sandbox primero
-  **Documentación:** Sí, hay que escribirla



Lo que intente explicar hoy

1. MCP conecta IAs con aplicaciones
2. XML-RPC sigue vivo (como los zombies)
3. 100 líneas de código pueden hacer magia
4. Claude + Odoo = Productividad ∞




Recursos

-  [MCP Docs](#)
-  [Odoo API](#)
-  [Código de hoy](#)
-  [Awesome MCP Servers](#)

¿Preguntas?

Tengo respuestas (tal vez correctas)

Contacto:

-  @fsmw14
-  fsanmartinw@uft.edu
-  Discord: fsmw



¡Gracias!

Ahora ve y automatiza todo

**PD: Si esto te gustó, ahora a pensar cual es la próxima pega manual
que vamos a generar**



Bonus: Código Rápido

```
# El servidor MCP más simple del mundo
import mcp
from xmlrpc.client import ServerProxy

@mcp.tool()
async def do_magic(query: str):
    """Hace magia con .... ✨"""
    # Tu código aquí
    return ¡Magia completada! 🎩

# ¡Eso es todo amigos!
```