

AI-OdooFinder



Tu Asistente Inteligente para Módulos de Odoo

Proyecto Final - AI Expert DevExpert Curso 2025

El Problema 🤔

Los desarrolladores de Odoo pierden horas buscando el módulo correcto:

App Store

Módulos incompatibles con tu versión

Google

Resultados desactualizados y poco confiables

GitHub OCA

Miles de repos sin forma de filtrar por funcionalidad

Resultado: Frustración, pérdida de tiempo y módulos que no funcionan

AI-OdooFinder: Un asistente conversacional con IA que encuentra el módulo exacto que necesitas

Pregunta en lenguaje natural:

"Necesito un módulo para gestionar suscripciones en Odoo 17"

Obtén resultados precisos:

-  Solo módulos compatibles con tu versión
-  Con análisis de dependencias
-  Puntuados por calidad y popularidad

Arquitectura: RAG Híbrido



El Desafío

Un RAG simple (solo vectorial) **NO funciona** para Odoo:

- No puede garantizar compatibilidad de versiones
- Mezclará resultados de v16, v17, v18...
- Perderás la confianza del usuario

Nuestra Solución

Búsqueda en 2 Fases:

- ❑ 1. Filtro Determinista (SQL)

```
version = "17.0" AND "sale" IN depends
```

- ❑ 2. Búsqueda Semántica (Vectorial)

Embeddings del README y description

Stack Tecnológico



Backend & Datos

- **Neon PostgreSQL** (Serverless con pgVector)
- **FastAPI** (Python 3.10+) en Render.com
- **SQLAlchemy** + Alembic para migraciones



IA & Embeddings

- **Qwen3-Embedding-4B** (modelo open source)
- **OpenRouter.ai** para hosting
- **pgVector** para búsqueda vectorial



Interfaz

- **Claude Skill** → Conversacional en claude.ai
- Sin frontend web en MVP

Cómo Funciona



Usuario: "Necesito gestión de inventario en v17"



Claude Skill (Agente)



API Backend (FastAPI)



1 SQL Filter: `version="17.0" AND keywords LIKE "%inventario%"`



2 Vector Search: `similarity(query_embedding, module_embeddings)`



3 Scoring & Ranking: GitHub stars + commits + docs quality



Resultados ordenados → Claude formatea respuesta

Aplicación del Método I.D.E.A.S.

Investigar

- Investigación de arquitecturas RAG híbridas
- Análisis de APIs (GitHub, OpenRouter, Neon)
- Estudio del ecosistema Odoo/OCA

Dialogar

- Prompts para Claude Skill (SKILL.md)
- Context engineering con manifiestos de Odoo
- Chain-of-thought para búsquedas complejas

Aplicación del Método I.D.E.A.S.

Ejecutar

- Desarrollo con **Cursor IDE** y Composer Mode
- Refactoring de código Python con IA
- Generación de tests automáticos

Automatizar

- ETL automático con GitHub Actions
- Scripts para actualización de embeddings
- CI/CD pipeline completo

Aplicación del Método I.D.E.A.S.

 Sintetizar



OpenAI API

Integración para embeddings



Claude API

Interfaz conversacional



Sistema Híbrido

SQL + RAG vectorial

Resultado: Aplicación funcional que demuestra todo lo aprendido

Demo del Flujo



Usuario pregunta:

"Necesito un módulo para gestionar suscripciones en Odoo 17"

Sistema responde:

Sale Subscription (87/100) Recomendado

Repository: OCA/sale-workflow

245 estrellas | Actualizado hace 2 días

Depende de: sale, account

Gestión completa de suscripciones con facturación automática, renovaciones y períodos de prueba.

Compatible con Odoo 17.0

Datos del Proyecto



Fuente de Datos

- **OCA (Odoo Community Association)**
→ ~1000 módulos
- **GitHub API** para obtención automática
- Actualización semanal vía ETL



Base de Datos

- **~1000 módulos** indexados
- **Soporte para versiones:** 14.0, 15.0, 16.0, 17.0, 18.0
- **Embeddings:** 384 dimensiones (Qwen3)



Performance

- Búsqueda < 500ms
- ETL completo: ~15 minutos

Estructura del Proyecto



```
ai-odoofinder/
├── backend/    # FastAPI + servicios
├── scripts/    # ETL y mantenimiento
├── claude-skill/ # SKILL.md para Claude
├── docs/       # Documentación técnica
├── .github/    # CI/CD workflows
└── tests/      # Tests automatizados
```

Archivos Clave

- backend/app/services/search_service.py - Búsqueda híbrida
- scripts/etl_oca_modules.py - Pipeline de datos
- claude-skill/SKILL.md - Instrucciones para Claude

Roadmap de Implementación



✓ Fase 0: Setup (1-2 semanas)

- Configuración de cuentas (Neon, OpenRouter, GitHub)
- Validación técnica de APIs
- Pruebas de concepto con embeddings



✓ Fase 1: MVP (2-3 semanas)

- Backend FastAPI funcional
- ETL para ~200 módulos de OCA
- Claude Skill básica



➡️ Fase 2: Producción (1-2 meses)

- Escalado a 1000+ módulos
- Optimización de rendimiento
- Frontend web opcional (Next.js)

Desafíos Técnicos



1. Parsing de Manifiestos

Problema: __manifest__.py es código Python ejecutable

Solución: ast.literal_eval() para parsing seguro

2. Control de Costos

Problema: Embeddings pueden ser costosos

Solución:

- Caché de embeddings en DB
- Batch processing
- Modelo económico (Qwen3)

3. Compatibilidad de Versiones

Problema: Odoo tiene versionado estricto

Solución: Filtro SQL ANTES de búsqueda vectorial

Resultados y Aprendizajes



Habilidades Aplicadas

- Arquitectura RAG híbrida (SQL + Vectorial)
- Prompt Engineering para Claude Skills
- Context Engineering con datos estructurados
- Desarrollo agéntico con Cursor
- Integración de APIs de IA (OpenAI, Anthropic)
- ETL y procesamiento de datos a escala
- CI/CD y automatización con GitHub Actions

Tecnologías Dominadas

Python, FastAPI, PostgreSQL, pgVector, SQLAlchemy, OpenAI API, Anthropic API, GitHub API, Docker, Render.com

Casos de Uso

Para Desarrolladores Odoo

- Encontrar módulos por funcionalidad
- Verificar compatibilidad de versiones
- Analizar dependencias antes de instalar

Para Empresas

- Acelerar implementaciones de Odoo
- Reducir errores de compatibilidad
- Documentación automática de módulos usados

Para la Comunidad OCA

- Visibilidad de módulos de calidad
- Estadísticas de popularidad
- Identificar gaps en el ecosistema

Próximos Pasos



Mejoras Planificadas

Corto Plazo:

- ✨ Soporte para módulos privados (GitHub Empresas)
- 📊 Sistema de recomendaciones ("usuarios que usaron X también usaron Y")
- 🔔 Alertas de actualizaciones de módulos

Largo Plazo:

- 🌐 Frontend web con Next.js
- 🤖 Análisis de código de módulos con IA
- 📝 Generación automática de documentación

Recursos y Enlaces



Repositorio GitHub

 github.com/santipbarber/ai-odoofinder



Documentación

-  **README.md** - Intro comercial
-  **QUICKSTART.md** - Setup en 10 minutos
-  **TECHNICAL_GUIDE.md** - Guía técnica completa
-  **PROJECT_STRUCTURE.md** - Estructura del código



Demos

-  Video demostración (próximamente)
-  API en vivo: api-odoofinder.render.com

Métricas del Proyecto



2,500

Líneas de código Python

85%+

Cobertura de tests

100%

Type hints (MyPy)

Código

- **~2,500 líneas** de código Python
- **85%+ cobertura** de tests
- **100% type hints** (MyPy)
- **Black + Flake8** para calidad

Documentación

- **5 documentos** técnicos completos
- **20+ ejemplos** de código
- **Diagramas** de arquitectura
- **API reference** completa

Comunidad

- Licencia **MIT** (open source)
- **CONTRIBUTING.md** para colaboradores
- Templates para issues y PRs

Lecciones Aprendidas



✓ Lo que Funcionó Bien

- RAG híbrido > RAG puro para datos estructurados
- Claude Skills son perfectas para MVP sin UI
- Neon PostgreSQL simplifica infraestructura
- Embeddings open source (Qwen3) son viables

✗ Lo que Cambiaría

- Empezar con subset más pequeño (50 módulos)
- Implementar caché de embeddings desde día 1
- Automatizar tests de integración más temprano
- Documentar decisiones de arquitectura en tiempo real

Impacto Esperado

Para la Comunidad Odoo

- **Ahorro de tiempo:** De horas a minutos en búsqueda
- **Menos errores:** Compatibilidad garantizada
- **Mejor adopción:** Descubrimiento de módulos de calidad

Para Desarrolladores

- **Productividad 3x:** Búsquedas más rápidas
- **Confianza:** Resultados verificados
- **Aprendizaje:** Exploración del ecosistema OCA

Métricas de Éxito



 **500+ búsquedas/mes** (objetivo año 1)



 **50+ stars** en GitHub (6 meses)



 **10+ contribuidores** activos

Agradecimientos



Instructores AI Expert

Antonio Leiva y Nino Ruano

Por el método I.D.E.A.S. y las herramientas aprendidas

Comunidad

- **OCA (Odoo Community Association)**
- Por los módulos
- **Anthropic** - Por Claude y las Skills
- **OpenRouter** - Por acceso a modelos open source

Compañeros del Curso

Por el feedback, ideas y soporte durante el desarrollo

Preguntas y Contacto ?

¿Dudas sobre el Proyecto?

-  Slack del curso: @Santiago Pérez Barber
-  Email: santipbr@gmail.com
-  GitHub: github.com/santipbarber

¿Quieres Contribuir?

-  Dale star al repo
-  Revisa **CONTRIBUTING.md**
-  Reporta bugs o sugiere features

Conclusión

AI-OdooFinder Demuestra:

Investigación

Arquitecturas RAG y APIs

Diálogo

Prompt Engineering efectivo

Ejecución

Desarrollo agéntico con Cursor

Automatización

ETL y CI/CD completos

Síntesis

Integración real de IA en apps

Resultado: Un proyecto funcional que resuelve un problema real y aplica todo lo aprendido en AI Expert



¡Gracias!



AI-OdooFinder

Tu asistente inteligente para módulos de Odoo

