

# Roadmap: Backend + IA + APIs con Python

## Fase 1: Fundamentos de Backend en Python

- Repaso de Python avanzado: async/await, excepciones, typing, pytest.
- FastAPI core: rutas, validación de datos con pydantic, manejo de errores, documentación automática.
- Proyecto práctico #1: API de gestión de tareas con CRUD.

## Fase 2: Bases de Datos y Persistencia

- ORM: SQLAlchemy o Tortoise ORM.
- Bases de datos: PostgreSQL o MySQL (producción), SQLite (pruebas).
- Autenticación y seguridad: JWT, OAuth2 con FastAPI.
- Proyecto práctico #2: API de usuarios + autenticación con JWT + PostgreSQL.

## Fase 3: APIs y Microservicios

- REST APIs con endpoints versionados.
- GraphQL (opcional) con Strawberry o Ariadne.
- Background tasks con FastAPI, Celery.
- Deploy con Docker en Railway, Render, AWS, GCP, Azure.
- Proyecto práctico #3: API de notificaciones con microservicios.

## Fase 4: Machine Learning y APIs de IA

- ML con Python: scikit-learn, pandas, numpy.
- Entrenar modelos simples y guardarlos con joblib o pickle.
- Integrar modelos con FastAPI para exponer predicciones vía JSON.
- Proyecto práctico #4: API que prediga precios de casas con ML.

## Fase 5: LLMs y NLP

- Fundamentos NLP: nltk, spacy, transformers (Hugging Face).
- LLMs locales: GPT-2, LLaMA-2, Mistral con transformers.
- LLMs en la nube: OpenAI, Anthropic, Cohere.
- Frameworks: LangChain, LangGraph.
- Proyecto práctico #5: API de chatbot con FastAPI + LLM.

## Fase 6: Infraestructura y Producción

- CI/CD con GitHub Actions.
- Monitorización: logs con structlog, métricas con Prometheus + Grafana.
- Optimización: cache con Redis, rate limiting.
- Escalabilidad: Kubernetes, serverless con AWS Lambda o Google Cloud Functions.
- Proyecto práctico #6: Plataforma fullstack con FastAPI + ML/LLM, frontend React, deploy con Docker.

## ■ Orden sugerido de aprendizaje

- FastAPI (Fase 1 y 2).
- Microservicios y deploy (Fase 3).
- ML clásico + API (Fase 4).
- LLMs + LangChain (Fase 5).
- Infraestructura avanzada (Fase 6).