

# Guía para Ser Senior en Python

Esta guía cubre los conocimientos y habilidades clave que debes dominar para alcanzar un nivel senior en Python.

## 1. Fundamentos del Lenguaje

- Tipado dinámico y fuerte
- Estructuras de datos integradas: listas, diccionarios, conjuntos, tuplas
- Comprehensions: listas, conjuntos y diccionarios
- Funciones: parámetros, valores por defecto, `*args` y `**kwargs`
- Manejo de excepciones

## 2. Programación Orientada a Objetos

- Clases y objetos
- Encapsulamiento, herencia y polimorfismo
- Decoradores de clases y métodos
- Metaclases

## 3. Programación Funcional

- Funciones de orden superior
- Uso de `map`, `filter` y `reduce`
- Expresiones `lambda`
- Iteradores y generadores

## 4. Módulos y Paquetes

- Creación y organización de módulos
- Importaciones relativas y absolutas
- Gestión de paquetes con `pip` y `venv`

## 5. Gestión de Datos

- Manejo de archivos (lectura/escritura)
- Serialización: JSON, YAML, pickle
- Interacción con bases de datos usando `sqlite3` y ORMs como SQLAlchemy

## 6. Programación Asíncrona

- Uso de `async` y `await`
- Manejo de `asyncio`
- Ejecutores de hilos y procesos

## 7. Buenas Prácticas

- Escribir código limpio y legible (PEP 8)
- Documentación con `docstrings`
- Pruebas unitarias con `unittest`, `pytest`
- Depuración y profiling

## 8. Bibliotecas y Frameworks Clave

- **Web:** Django, Flask, FastAPI
- **Data Science:** NumPy, pandas, Matplotlib, scikit-learn
- **Automatización:** Selenium, requests, BeautifulSoup
- **DevOps:** Paramiko, fabric

## 9. Conocimientos Avanzados

- Multihilos y multiprocesos
- Manejo de memoria y garbage collection
- Internals del intérprete CPython
- Integración con C/C++ usando `ctypes`, `cython`, o `cffi`

## 10. Desarrollo Profesional

- Contribuir a proyectos open source
- Mentoría y liderazgo técnico
- Participación en revisiones de código
- Optimización de procesos de desarrollo