



# Roadmap to AI Engineer

Cómo comenzar y avanzar en el mundo de la ingeniería de inteligencia artificial.

Por Juan Duran

# ¿Qué hace un AI Engineer?

Un **AI Engineer** **diseña, desarrolla e implementa** sistemas y modelos de **inteligencia artificial** que permiten a las máquinas “aprender” de los datos y tomar decisiones.

Este perfil combina conocimientos en **programación, matemáticas, estadística y machine learning**, y suele trabajar en la creación de modelos predictivos, asistentes inteligentes, sistemas de recomendación, visión por computadora, procesamiento de lenguaje natural (NLP) y más.


El rol de AI Engineer es cada vez más demandado y está presente en muchos y diferentes sectores . Entre sus tareas típicas encontramos:


- Recolectar, limpiar y preparar **datos**
- Entrenar y validar **modelos de ML/DL**
- **Optimizar** modelos para **producción**
- **Desplegar** modelos con **APIs** o en la **nube**
- **Colaborar** con Data Scientists y DevOps





# Habilidades para ser AI Engineer

Convertirte en AI Engineer implica adquirir una serie de habilidades técnicas y conceptuales:


 **Programación:** Python es el lenguaje más usado en IA. Otros como R, Java o C++ también se usan en entornos específicos.

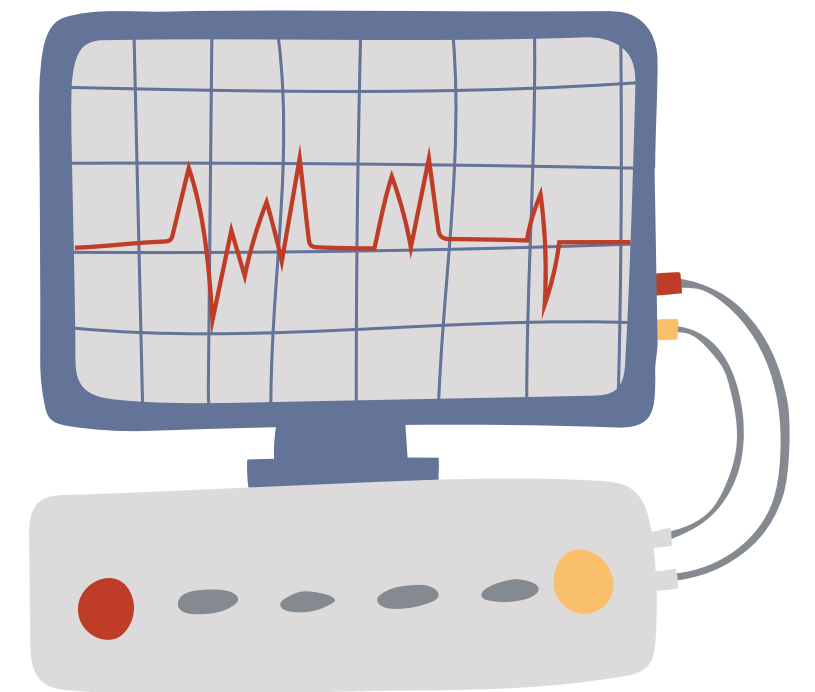
 **Matemáticas y estadística:** Fundamentos de álgebra lineal, cálculo, probabilidad y estadística son imprescindibles para entender cómo funcionan los modelos.

 **Machine Learning y Deep Learning:** Comprender los algoritmos, sus casos de uso, ventajas, limitaciones y cómo aplicarlos.

 **Ingeniería de datos:** Saber trabajar con grandes volúmenes de datos, manejar bases de datos y diseñar pipelines de datos robustos.

 **Cloud & DevOps:** Familiaridad con plataformas como AWS, Azure o GCP, y habilidades básicas de CI/CD para desplegar modelos.

 **Curiosidad y aprendizaje constante:** La IA cambia rápido. Estar actualizado es parte del trabajo.



# Fundamentos de programación

Si estás empezando, **Python** es el mejor punto de partida: sencillo, legible y con una gran comunidad en el mundo de la **IA**.

## Comienza por los fundamentos:

- Variables, estructuras de control y funciones
- Clases y manejo de errores

## Luego avanza hacia librerías clave:

- NumPy y Pandas para manipulación de datos
- Matplotlib y Seaborn para visualización
- Scikit-learn para aplicar algoritmos de machine learning

Dominar la programación no solo te permite construir modelos, sino entender el comportamiento de los datos y crear soluciones eficientes.

**Es el primer gran paso en tu camino hacia la ingeniería de IA.**



# Matemáticas para IA

Para ser AI Engineer no hace falta dominar fórmulas complejas, pero sí entender los conceptos básicos que sustentan el aprendizaje automático.

## Álgebra lineal

- Vectores, matrices, multiplicación matricial y transformaciones
- Es la base del manejo de datos y del funcionamiento interno de muchos algoritmos, especialmente en deep learning.

## Probabilidad y estadística

- Conceptos como media, varianza, desviación estándar y distribuciones
- Probabilidad condicional, Bayes y análisis exploratorio de datos
- Te ayuda a entender el muestreo y la calidad de los datos.

## Cálculo

- Derivadas, gradientes y optimización
- Fundamental para entender cómo se ajustan los pesos en una red neuronal durante el entrenamiento.



# Fundamentos de Machine Learning

Aquí empieza lo emocionante. Debes aprender a construir modelos que puedan aprender de los datos:

## Conceptos clave:

- Overfitting vs underfitting
- Validación cruzada
- Métricas de evaluación (accuracy, precision, recall, F1-score)

## Algoritmos más usados:

- Regresión lineal y logística
- Árboles de decisión y random forest
- KNN, SVM, clustering
- Modelos de boosting (XGBoost, LightGBM)

✓ Usa datasets reales (como los de Kaggle o Scikit-learn) y practica entrenando modelos desde cero.



# Deep Learning y redes neuronales

El siguiente nivel es el aprendizaje profundo (**Deep Learning**), una subrama del ML enfocada en redes neuronales.

## 🧠 Fundamentos que debes conocer:

- Neuronas artificiales y capas
- Funciones de activación
- Arquitecturas básicas (MLP, CNN, RNN)
- Backpropagation y optimización

## 📦 Librerías populares:

- TensorFlow y Keras
- PyTorch (más flexible, usada en investigación)

🌐 Con deep learning puedes abordar tareas más complejas como visión por computador, reconocimiento de voz o NLP.





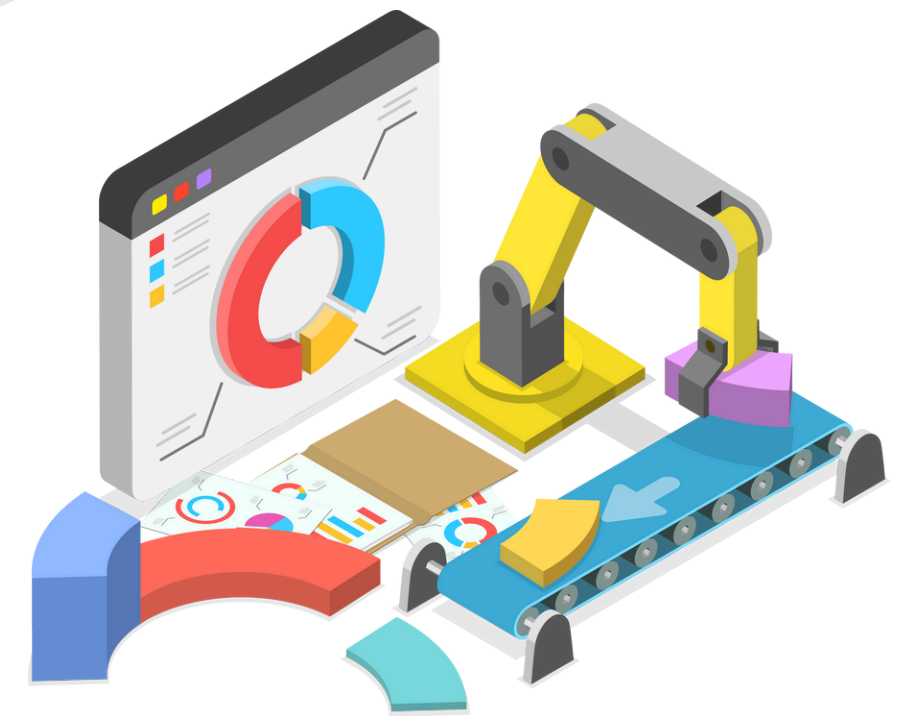
# Despliegue y MLOps

Un modelo que funciona en tu Jupyter Notebook no sirve de mucho si no se puede usar en el mundo real.

## 🎯 Aprende a:

- Guardar modelos (joblib, pickle, ONNX)
- Crear APIs con Flask o FastAPI para servirlos
- Usar herramientas de contenedores como Docker
- Automatizar procesos con CI/CD
- Desplegar en la nube (AWS, GCP, Azure)

💡 Aquí nace el concepto de **MLOps**, que une el mundo del desarrollo de modelos con la infraestructura y el mantenimiento.





# Herramientas y recursos recomendados

Aquí tienes una lista básica de herramientas que usarás en tu camino como AI Engineer:

**Lenguajes y librerías:** Python, Scikit-learn, TensorFlow, PyTorch, NumPy, Pandas, OpenCV, NLTK

**Datos y almacenamiento:** SQL, MongoDB, Hadoop, Spark

**Entornos y despliegue:** Docker, Kubernetes, Flask, FastAPI

**Cloud:** AWS (SageMaker), Google Cloud (Vertex AI), Azure ML

## Recursos:

- Coursera (Machine Learning – Andrew Ng)
- fast.ai (Deep Learning)
- Kaggle (competencias y datasets)
- GitHub (repositorios de proyectos)



# Conclusiones y próximos pasos

Convertirse en AI Engineer no es un camino rápido, pero sí muy enriquecedor y con muchísimas salidas profesionales.

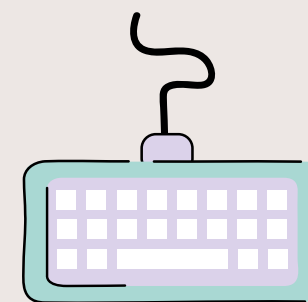
Es importante avanzar paso a paso, practicando y aplicando lo aprendido.

Aquí algunos consejos clave:

- ✓ Aprende programación sólida (Python)
- ✓ Refuerza tus bases matemáticas
- ✓ Crea proyectos propios (¡enséñalos en GitHub!)
- ✓ Participa en comunidades (Kaggle, Discord, LinkedIn)
- ✓ No busques saberlo todo antes de empezar
- ✓ Aprende de los errores: fallar también es parte del camino

**Empieza hoy. El mejor aprendizaje en IA viene de construir, equivocarse, entender y volver a intentarlo.**





# Gracias



Por Juan Duran

“Coding, Gaming and Leveling Up”