

# Diplomado en Inteligencia Artificial Aplicada

Plan de estudios



## Diplomado en Inteligencia Artificial Aplicada

### Plan de Estudios

#### Módulo Propedéutico. Introducción a la programación en Python para IA (12 h)

##### 1. Conceptos Básicos de Programación

Aquí se cubrirán los conceptos fundamentales para comprender la lógica de programación, como variables, tipos de datos y operadores. Los estudiantes aprenderán cómo declarar variables, realizar operaciones básicas y entender la importancia de la sintaxis en Python.

##### 2. Estructuras de Datos y Control de Flujo

Se estudiarán las diferentes estructuras de datos disponibles en Python, como listas, tuplas y diccionarios, así como el control de flujo mediante condicionales (if, elif, else) y bucles (for, while). Esto permitirá a los estudiantes organizar y manipular datos de manera eficiente.

##### 3. Funciones

En esta unidad se abordará el concepto de funciones en Python. Los estudiantes aprenderán a definir funciones, pasar argumentos y devolver valores. También se explorarán temas como el alcance de las variables y la reutilización de código mediante el uso de funciones.

#### Módulo 1. Programación en Python para IA (24 h)

##### 1.1: Introducción a Python y Entorno de Desarrollo

En esta unidad, se dará una visión general de Python, incluyendo su historia, características y ventajas. Los estudiantes aprenderán a instalar Python y configurar entornos virtuales para facilitar el desarrollo de proyectos. Se introducirá el uso de Jupyter Notebooks como una herramienta interactiva para escribir y ejecutar código Python.





## 1.2 Fundamentos de Python para Inteligencia Artificial

Se explorarán los conceptos fundamentales de Python necesarios para la inteligencia artificial, como las estructuras de datos (listas, tuplas, conjuntos y diccionarios), el control de flujo (condicionales, bucles) y el manejo de excepciones. Los estudiantes también aprenderán sobre la definición y uso de funciones y módulos en Python para organizar y reutilizar el código.

## 1.3 Manipulación de Datos con NumPy y Pandas

Esta se centrará en el manejo de datos utilizando las bibliotecas NumPy y Pandas. Se estudiarán los arreglos multidimensionales de NumPy y las operaciones básicas que se pueden realizar con ellos. Además, se explorará la manipulación de datos tabulares utilizando Pandas, incluyendo la limpieza, transformación y visualización de datos.

## 1.4 Desarrollo de Modelos con Scikit-Learn

Se introducirá la biblioteca Scikit-Learn para el desarrollo de modelos de aprendizaje automático en Python. Los estudiantes aprenderán sobre el preprocesamiento de datos, incluyendo el escalado y la codificación de variables categóricas, así como la implementación de algoritmos de aprendizaje supervisado y no supervisado, como la regresión, clasificación, clustering y reducción de dimensionalidad.

## 1.5 Construcción de Redes Neuronales con TensorFlow y Keras

En esta unidad, se explorará el desarrollo de modelos de redes neuronales utilizando las bibliotecas TensorFlow y Keras. Se estudiarán las estructuras de datos y los gráficos computacionales en TensorFlow, así como el diseño y entrenamiento de modelos de redes neuronales utilizando Keras. Los estudiantes también aprenderán sobre la evaluación y ajuste de hiperparámetros de modelos de redes neuronales.

## Módulo 3. Aprendizaje de Máquina (32 h)

### 2.1: Conceptos Básicos de Aprendizaje de Máquina

En esta sección se establecen los pilares fundamentales del Aprendizaje de Máquina, proporcionando una visión amplia que abarca desde la definición del campo hasta sus aplicaciones prácticas en diversos sectores. Además, se introduce a los estudiantes en los conceptos esenciales como datos, rasgos, etiquetas y modelos, sentando las bases para comprender las técnicas de evaluación de modelos.





## 2.2 Minimización Estructural del Riesgo

Esta sección va a profundizar en la teoría del aprendizaje, enfocándose en la importancia de minimizar el riesgo estructural al entrenar modelos de ML. Se exploran conceptos cruciales como el sesgo y la varianza, y se presentan técnicas de regularización como  $L_1$ ,  $L_2$  y Red Elástica, que ayudan a evitar el sobreajuste de los modelos.

## 2.3 Métodos de Optimización para Aprendizaje de Máquina a Gran Escala

Aquí se estudian los métodos de optimización necesarios para el entrenamiento eficiente de modelos de ML a gran escala. Se cubren desde los conceptos básicos de optimización, como el gradiente descendente estocástico.

## 2.4 Aprendizaje Supervisado

Esta sección se va a centrar en el aprendizaje supervisado, donde se presentan técnicas para resolver problemas de regresión y clasificación. Se estudian algoritmos como regresión lineal, regresión logística, árboles de decisión, SVM y redes neuronales, junto con métodos de ensamble como Gradient Boosting y Bagging, que permiten mejorar el rendimiento del modelo.

## 2.5: Aprendizaje No Supervisado

Aquí se exploran los métodos de aprendizaje no supervisado, utilizados cuando no se disponen de etiquetas en los datos de entrenamiento. Se examinan técnicas de aglomeración como K-Medias y DBSCAN, así como métodos de reducción de la dimensionalidad como PCA y t-SNE, que facilitan la comprensión y visualización de conjuntos de datos complejos.

## 2.6: Aprendizaje por Refuerzo

Se introduce el aprendizaje por refuerzo, un enfoque donde un agente aprende a través de la interacción con un entorno. Se exploran conceptos como los procesos de decisión de Markov (MDP) y se presentan algoritmos básicos de aprendizaje por refuerzo.

## Módulo 3. Aprendizaje Profundo (20 h)

### 3.1: Introducción al Aprendizaje Profundo

Esta sección proporciona una introducción a los conceptos básicos de las redes neuronales profundas, así como una visión general de la evolución y las aplicaciones del aprendizaje profundo en diversos campos como la visión por computadora, el procesamiento de lenguaje natural y la bioinformática.



### **3.2: Regularización en Redes Neuronales**

*En esta parte se aborda el problema del sobreajuste en las redes neuronales y se presentan técnicas de regularización, como dropout y regularización L1 y L2, que ayudan a evitar que el modelo se ajuste demasiado a los datos de entrenamiento. También se exploran métodos para prevenir la explosión y el desvanecimiento del gradiente durante el entrenamiento.*

### **3.3: Métodos de Optimización para Redes Neuronales Profundas**

*Aquí se estudian los métodos de optimización utilizados para ajustar los parámetros de las redes neuronales profundas de manera eficiente. Se cubren desde el gradiente descendente y sus variantes hasta técnicas avanzadas como SGD con momentum, RMSProp y Adam, junto con estrategias para la inicialización adecuada de los pesos de la red.*

### **3.4: Redes Neuronales Convolucionales (CNN)**

*Esta sección se enfoca en las redes neuronales convolucionales (CNN), una arquitectura especialmente diseñada para procesar datos de tipo imagen. Se exploran los principios de funcionamiento de las CNN, así como arquitecturas comunes como LeNet, AlexNet, VGG y ResNet, y se presentan aplicaciones prácticas en clasificación de imágenes, detección de objetos y segmentación semántica.*

### **3.5: Redes Neuronales Recurrentes (RNN) y Análisis de Series Temporales**

*Se introduce a las redes neuronales recurrentes (RNN) y se explora su capacidad para modelar dependencias temporales en datos secuenciales. Se analizan aplicaciones en análisis de series temporales, predicción y generación de secuencias, destacando su utilidad en áreas como finanzas, clima y procesamiento de señales.*

### **3.6: Redes Neuronales Recurrentes en Procesamiento de Lenguaje Natural (NLP)**

*En esta última se aborda el uso de las redes neuronales recurrentes en el procesamiento de lenguaje natural (NLP). Se examina el modelado del lenguaje con RNN y se presentan aplicaciones en traducción automática, generación de texto, análisis de sentimientos y reconocimiento de voz, demostrando el potencial de estas técnicas en la comprensión y generación de lenguaje humano.*

## **Módulo 4. Seminario de proyecto terminal (16 h)**

