

Curso: Introducción al Machine Learning

Manual de prácticas y teoría para el curso de ML. De cero a redes neuronales.

- [Capítulo 0: Información General del Curso](#)
 - [Temario y Calendario](#)
 - [Criterios de Evaluación](#)
 - [Recursos necesarios](#)
- [Capítulo 1: Fundamentos y Primeros Pasos](#)
 - [Práctica 1: La Magia sin Código \(Teachable Machine\).](#)
 - [Práctica 2: Nuestro Laboratorio \(Intro a Google Colab & Python\).](#)
 - [Práctica 3: El Petróleo del Siglo XXI \(Carga de datos con Pandas\).](#)
- [Capítulo 2: Manipulación y Visualización](#)
 - [Práctica 4: Datos que Cuentan Historias \(Gráficas\).](#)
 - [Práctica 5: Detectives de Datos \(Limpieza de datos\).](#)
- [Capítulo 3: Aprendizaje Supervisado \(Predicciones\)](#)
 - [Práctica 6: Adivinando el Futuro \(Regresión Lineal\).](#)
 - [Práctica 7: ¿Clase A o Clase B? \(KNN - Vecinos cercanos\).](#)
 - [Práctica 8: El Árbol de las Decisiones \(Decision Trees\).](#)
 - [Práctica 9: ¿Qué tan bueno es mi modelo? \(Evaluación y Métricas\).](#)
- [Capítulo 4: Aprendizaje No Supervisado \(Patrones\)](#)
 - [Práctica 10: Agrupando sin Etiquetas \(Clustering K-Means\).](#)
- [Capítulo 5: Redes Neuronales \(Deep Learning\)](#)
 - [Práctica 11: La Neurona Artificial \(Lógica del Perceptrón\).](#)

- [Práctica 12: Cerebros Digitales \(Redes Densas y MNIST\).](#)
- [Práctica 13: Visión por Computadora \(Clasificación de ropa/imágenes\).](#)

Capítulo 0: Información General del Curso

Temario y Calendario

Bienvenido a la ruta de aprendizaje de **Introducción al Machine Learning**. Durante las próximas **16 semanas**, pasaremos de no saber nada sobre Inteligencia Artificial a construir nuestras propias Redes Neuronales capaces de "ver".

Este curso es **práctico**. La teoría es importante, pero aquí aprenderemos haciendo.

?? Mapa de Ruta Semanal

A continuación se detalla qué práctica realizaremos cada semana. Las semanas marcadas con **PA** corresponden a evaluaciones o periodos vacacionales.

Semana	Módulo	Actividad / Práctica	Tema Principal
01	Módulo 1	PA Práctica 1: La Magia sin Código	Intro a IA & Teachable Machine
02	Módulo 1	PA Práctica 2: Nuestro Laboratorio	Google Colab & Python Básico
03	Módulo 1	PA Práctica 3: El Petróleo del Siglo XXI	Carga de datos con Pandas
04	Módulo 2	PA Práctica 4: Datos que Cuentan Historias	Visualización de Datos
05	Módulo 2	PA Práctica 5: Detectives de Datos	Limpieza y Preprocesamiento
06	PA PAUSA	PA PRIMERA EVALUACIÓN PARCIAL	<i>Revisión de Módulos 1 y 2</i>
07	Módulo 3	PA Práctica 6: Adivinando el Futuro	Regresión Lineal
08	Módulo 3	PA Práctica 7: ¿Clase A o Clase B?	K-Vecinos Cercanos (KNN)
09	Módulo 3	PA Práctica 8: El Árbol de las Decisiones	Árboles de Decisión
10	Módulo 3	PA Práctica 9: ¿Qué tan bueno es mi modelo?	Métricas de Evaluación
11	PA PAUSA	PA PUENTE / SEGUNDA EVALUACIÓN	<i>Descanso o Repaso</i>

Semana	Módulo	Actividad / Práctica	Tema Principal
12	Módulo 4	📄 Práctica 10: Agrupando sin Etiquetas	Clustering (K-Means)
13	Módulo 5	📄 Práctica 11: La Neurona Artificial	Lógica del Perceptrón
14	Módulo 5	📄 Práctica 12: Cerebros Digitales	Redes Densas (MNIST)
15	Módulo 5	📄 Práctica 13: Visión por Computadora	Redes Convolucionales (CNN)
16	📄 CIERRE	📄 PROYECTO FINAL / EXPO	<i>Presentación de Resultados</i>

? Metodología de Trabajo

- 1. **Lectura:** Cada semana, lee la introducción teórica en la hoja correspondiente aquí en BookStack.
- 2. **Práctica:** Sigue los pasos para programar tu modelo en **Google Colab**.
- 3. **Reto:** Al final de cada práctica, encontrarás un pequeño desafío que deberás resolver solo.
- 4. **Entrega:** Sube tu notebook `.ipynb` o el enlace compartido a la plataforma de tareas.

“ **Nota:** Este calendario está sujeto a ajustes dependiendo del ritmo del grupo y las fechas oficiales de la institución.

Criterios de Evaluación

?? Criterios de Evaluación

Para aprobar este curso no necesitas memorizar fórmulas complejas, pero sí necesitas **constancia** . La programación y el Machine Learning son habilidades que se desarrollan practicando semana tras semana.

A continuación se detalla cómo se compondrá tu calificación final.

? Desglose de Calificación

Concepto	Peso	Descripción
Portafolio de Prácticas	60%	La suma de las 13 prácticas realizadas en clase y tarea.
Evaluaciones Parciales	20%	Dos revisiones de conocimientos (Semana 6 y Semana 11).
Proyecto Final	20%	Aplicación de lo aprendido en un problema real (Semana 16).
Total	100%	

? Detalles de los Entregables

1. Portafolio de Prácticas (60%)

Cada semana realizaremos una práctica guiada. Para que tu práctica cuente con el puntaje completo, debe cumplir con los siguientes requisitos en tu **Google Colab**:

- **Código Funcional:** El notebook debe correr de principio a fin sin errores.
- **Comentarios:** Debes explicar con tus palabras qué hace el código en las secciones clave.
- **Reto Resuelto:** Cada práctica tiene un pequeño "Reto Final" que debes resolver por tu cuenta.

- **Organización:** El notebook debe tener título, tu nombre y secciones ordenadas.

“ ⚠ **Política de entregas tardías:** Las prácticas se entregan semanalmente.

- Entrega a tiempo: 100% del valor.
- 1 día de retraso: 80% del valor.
- 2 días de retraso: 50% del valor.
- Más de 2 días: No se acepta (salvo justificante médico).

2. Evaluaciones Parciales (20%)

Realizaremos dos cortes para asegurar que los conceptos teóricos están claros.

- **Parcial 1 (Semana 6):** Enfoque en manejo de datos (Pandas) y visualización.
- **Parcial 2 (Semana 11):** Enfoque en modelos supervisados y evaluación.
- *Formato:* Pueden ser cuestionarios rápidos o ejercicios breves de código en clase.

3. Proyecto Final (20%)

En la última semana, presentarás un modelo de Machine Learning creado por ti.

- Elegirás un dataset de tu interés (deportes, música, salud, videojuegos, etc.).
- Entrenarás un modelo para resolver un problema.
- Expondrás los resultados ante el grupo.

? Uso de Inteligencia Artificial (ChatGPT, Gemini, etc.)

Estás en un curso de Inteligencia Artificial, por lo que usar estas herramientas no está prohibido, pero **su uso debe ser inteligente:**

- ☐ **PERMITIDO:** Usar IA para explicarte un error de código, para sugerirte cómo mejorar una gráfica o para entender un concepto matemático.
- ☐ **PROHIBIDO:** Copiar y pegar la solución completa del "Reto Final" sin entenderla.

Regla de Oro: Si te pregunto "¿Qué hace esta línea de código?" y no sabes explicarlo, la práctica se anula, aunque el código funcione perfectamente.

? Puntos Extra

Podrás obtener puntos extra sobre la calificación parcial si:

- Ayudas a compañeros a "debuggear" (solucionar errores) en sus códigos (sin pasarles la respuesta).
- Encuentras formas más eficientes de resolver los retos planteados.
- Participas activamente en las discusiones de clase.

Recursos necesarios

?? Recursos y Herramientas Necesarias

Para este curso, **no necesitas instalar nada complejo** en tu computadora. Hemos diseñado las prácticas para trabajar 100% en la nube. Esto significa que podrás avanzar en tus proyectos desde la escuela, tu casa o cualquier lugar con internet.

A continuación, la lista de lo que necesitas tener listo antes de empezar.

? 1. Hardware (Tu equipo)

- **Computadora:** Laptop o escritorio (Windows, Mac o Linux/Chromebook).
 - *Nota:* Aunque se puede acceder desde tablets, **no se recomienda** para escribir código cómodamente.
 - **Conexión a Internet:** Estable, ya que trabajaremos en línea todo el tiempo.
-

? 2. Cuentas Requeridas

? Una Cuenta de Google

Necesitarás una cuenta de Gmail (personal o institucional, si la escuela lo permite). Esta es la llave maestra para todas nuestras herramientas.

- **¿Por qué?** Usaremos el ecosistema de Google para programar y guardar archivos.

? Kaggle (Opcional por ahora)

[Kaggle.com](https://www.kaggle.com) es la comunidad de Ciencia de Datos más grande del mundo.

- **¿Por qué?** De aquí descargaremos muchos de los "datasets" (conjuntos de datos) para las prácticas. Te avisaremos cuándo registrarte.
-

?? 3. El Laboratorio: Google Colab

Nuestro entorno de desarrollo principal será **Google Colaboratory** (o "Colab").

- **¿Qué es?** Es como un documento de Word, pero que te permite escribir y ejecutar código de Python.
- **Ventaja:** Te presta una computadora virtual potente (GPU) de Google gratis.
- **Acceso:** colab.research.google.com

“📁 **Tarea Previa:** Entra a tu Google Drive, crea una carpeta llamada `Curso_ML` y dentro crea otra llamada `Datasets`. Ahí guardaremos todo para tener orden.

? 4. Sitios Web Clave

Guarda estos enlaces en tus favoritos, los usaremos recurrentemente:

1. **Teachable Machine:** teachablemachine.withgoogle.com (Para la Práctica 1).
 2. **Scikit-Learn Docs:** scikit-learn.org (La "biblia" de las herramientas que usaremos).
 3. **Pandas Cheat Sheet:** [Guía rápida de Pandas](#) (Tu "acordeón" autorizado para manipular datos).
-

? 5. Bibliografía de Apoyo (Opcional)

Si eres curioso y quieres profundizar más allá de las clases, estos recursos son excelentes:

- **Videos:** Canal de YouTube "Dot CSV" (Explica IA en español de forma genial).
 - **Libro:** *Python Data Science Handbook* (Disponible gratis en línea en inglés).
-

¡Eso es todo! Si tienes tu cuenta de Google y acceso a internet, estás listo para la **Práctica 1**.

Capítulo 1: Fundamentos y Primeros Pasos

Práctica 1: La Magia sin Código (Teachable Machine).

Antes de escribir una sola línea de código, necesitamos entender cómo "piensa" una Inteligencia Artificial.

Hoy no vamos a programar. Hoy vamos a **entrenar**. Usaremos una herramienta de Google llamada *Teachable Machine* para crear un modelo capaz de distinguir entre dos cosas usando la cámara de tu computadora.

? Objetivo de la Práctica

- Entender qué es un **Dataset** (Conjunto de datos).
 - Comprender el proceso de **Entrenamiento** (Training).
 - Ver en tiempo real cómo la máquina **Clasifica** (Predice) nuevos datos.
-

? Conceptos Clave (Lo que debes saber)

Para que una IA aprenda, generalmente seguimos 3 pasos:

1. **Recopilar Datos (Input):** Le mostramos muchos ejemplos (fotos, audios, texto).
2. **Entrenar (Train):** La computadora busca patrones matemáticos en esos ejemplos.
3. **Probar (Test):** Le mostramos algo nuevo y vemos si adivina qué es.

“ **Analogía:** Es como enseñar a un niño a diferenciar entre una manzana y una naranja. Primero le muestras muchas manzanas y le dices "esto es manzana". Luego haces lo mismo con las naranjas. Al final, le das una fruta nueva y le preguntas "¿Qué es esto?".

?? Instrucciones Paso a Paso

Paso 1: Entrando al laboratorio

1. Abre el navegador y ve a: teachablemachine.withgoogle.com
2. Haz clic en el botón azul **"Get Started"**.
3. Selecciona **"Image Project"** (Proyecto de Imágenes) y luego **"Standard Image Model"**.

Paso 2: Definiendo las "Clases"

Verás dos columnas. Estas son las categorías que queremos que la IA aprenda.

1. En **Class 1**, cambia el nombre y ponle: 
2. En **Class 2**, cambia el nombre y ponle: 

Paso 3: Recolectando Datos (El Dataset)

¡Hora de las fotos!

1. En la columna , haz clic en **Webcam**.
2. Ponte frente a la cámara, levanta el pulgar y mantén presionado el botón **Hold to Record** unos segundos.
 - *Tip:* Mueve tu mano un poco (acércala, aléjala, muévela a los lados) para que la IA la reconozca en varias posiciones. Intenta tomar unas **50-100 fotos**.
3. Repite lo mismo en la columna , pero ahora con el pulgar hacia abajo.

Paso 4: Entrenando al Modelo

1. En la columna del centro, haz clic en el botón **"Train Model"**.
2. **¡NO cierres la pestaña!** Espera a que termine. Tu navegador está haciendo cálculos matemáticos complejos en este momento.

Paso 5: Probando tu Creación

A la derecha verás el cuadro de **Preview**.

1. Activa la cámara si está apagada.
2. Haz gestos frente a la cámara.
3. Observa las barras de abajo: ¿Cambian cuando subes o bajas el dedo? Si la barra de "Mano Arriba" se llena al 100% cuando haces el gesto, ¡felicidades! Has creado tu primer modelo de IA.

? Experimentación (Rompe el modelo)

Las IAs no son perfectas. Intenta engañarla:

- ¿Qué pasa si muestras solo la palma de la mano (sin pulgar)?
- ¿Qué pasa si te sales de la cámara?
- ¿Qué pasa si otra persona se pone frente a la cámara?

Si la IA se confunde, es porque **sus datos de entrenamiento no incluían esos ejemplos**.

? Reto Final (Entregable)

Para completar esta práctica, debes hacer lo siguiente:

1. Crea un modelo **nuevo** que distinga entre **dos objetos escolares** (ej. "Pluma" vs "Cuaderno" o "Celular" vs "Calculadora").
2. Asegúrate de que funcione bien.
3. Toma una **Captura de Pantalla** donde se vea el modelo funcionando (detectando uno de los objetos con una certeza alta).
4. Responde la siguiente pregunta en tu entrega:

“Cuando mostraste un objeto que NO era ni la opción A ni la opción B (ejemplo: tu cara o el fondo vacío), ¿Qué decisión tomó la IA y por qué crees que pasó eso?”

Sube la captura y tu respuesta a la plataforma de tareas.

Práctica 2: Nuestro Laboratorio (Intro a Google Colab & Python).

Antes de que una computadora pueda aprender, debemos poder darle instrucciones. Esto lo hacemos usando un lenguaje de programación. Para Machine Learning, el lenguaje más popular es **Python**.

Hoy vamos a preparar nuestro laboratorio digital: **Google Colab**.

? Objetivo de la Práctica

- Familiarizarse con la interfaz de **Google Colab** (celdas de código y texto).
 - Repasar o aprender los conceptos clave de **Python** que usaremos: variables, listas y funciones `print()`.
 - Entender cómo ejecutar y guardar un *Notebook*.
-

? Conceptos Clave (Lo que debes saber)

Notebooks (Cuadernos)

Un *notebook* (como Colab) es un documento interactivo que mezcla **texto explicativo** (como este) y **código ejecutable**. Es la herramienta estándar en ciencia de datos.

Variables

Son como "cajas" para guardar información. Tienen un nombre y contienen un valor.

- **Ejemplo:** `edad = 17`
- El nombre de la caja es `edad`, el valor dentro es `17`.

Listas

Son variables que guardan una **colección ordenada** de datos. Se usan corchetes `[]`.

- **Ejemplo:** `frutas = ["manzana", "platano", "pera"]`
-

?? Instrucciones Paso a Paso (¡Manos a la Obra!)

Paso 1: Abrir Colab y Nombrar el Archivo

1. Ve a colab.research.google.com.
2. Haz clic en **"Archivo"** y luego **"Cuaderno Nuevo"**.
3. En la parte superior, cambia el nombre del archivo a: `Práctica 2 - Mi Nombre`.

Paso 2: Usando Celdas de Texto (Markdown)

1. En la primera celda, haz clic en `+ Texto`.
2. Escribe el título de la práctica (usa el símbolo `#` para hacerlo grande, como en BookStack).
3. Esta es la forma de documentar tu código. **¡Una práctica sin documentación no es una buena práctica!**

Paso 3: Declarando Variables

Ahora vamos a la primera celda de `+ Código`.

1. Escribe y ejecuta (presiona el botón de **Play** o `Shift + Enter`):

```
# 1. Declara tu nombre y tu edad en variables
mi_nombre = "Tu Nombre"
mi_edad = 17

# 2. Usa la función print() para mostrar el valor
print("Hola, soy", mi_nombre, "y tengo", mi_edad, "años.")
```

Práctica 3: El Petróleo del Siglo XXI (Carga de datos con Pandas).

Dicen que "los datos son el nuevo petróleo". Pero el petróleo crudo no sirve de mucho si no se refina para convertirlo en gasolina.

En el mundo de Python, nuestra "refinería" se llama **Pandas**. Es una librería increíblemente potente que nos permite tomar archivos gigantes de Excel o CSV y manipularlos en milisegundos. Hoy dejaremos de ver datos como simples números y empezaremos a ver información.

? Objetivo de la Práctica

- Entender qué es un **DataFrame**.
 - Aprender a cargar datos externos desde internet (`.read_csv()`).
 - Aprender a explorar datos desconocidos (`.head()`, `.tail()`, `.describe()`).
 - Realizar ordenamientos básicos para responder preguntas.
-

? Conceptos Clave

? Pandas

No es el oso 🐼. Es la librería de **Panel Data**. Es la herramienta estándar en la industria para análisis de datos.

? DataFrame (df)

Es el objeto principal de Pandas. Imagínalo simplemente como una **Tabla de Excel** **superpoderosa** que vive en la memoria de tu código. Tiene filas (índices) y columnas (variables).

? CSV (Comma Separated Values)

Es el formato de archivo más común para datos. Es texto plano donde cada columna está separada por una coma. Es ligero y cualquier programa lo puede leer.

?? Instrucciones Paso a Paso

Para esta práctica, usaremos una base de datos con las estadísticas de combate de los **Pokémon**.

Paso 1: Importar la librería

En tu primera celda de código, necesitamos llamar a nuestra herramienta.

```
import pandas as pd

# El "as pd" significa que de ahora en adelante,
# en lugar de escribir "pandas", solo escribiremos "pd" para ahorrar tiempo.
```

Paso 2: Cargar los datos (El oleoducto)

Vamos a traer los datos directamente desde un repositorio en Internet. Copia esta URL con cuidado.

```
# Guardamos la dirección del archivo en una variable
url =
"[https://raw.githubusercontent.com/KeithGalli/pandas/master/pokemon_data.csv](https://raw.githubusercontent.com/KeithGalli/pandas/master/pokemon_data.csv)"

# Leemos el archivo CSV y lo convertimos en un DataFrame llamado "df"
df = pd.read_csv(url)

print(";Datos cargados correctamente!")
```

Paso 3: Exploración Inicial (Mirando los datos)

Nunca confíes en datos que no has visto. Vamos a ver las primeras filas.

```
# .head(n) muestra las primeras n filas
df.head(5)
```

Observa las columnas: Name, Type 1, HP (Vida), Attack, Defense, etc.

También podemos ver el final de la tabla:

```
# .tail(n) muestra las últimas n filas
df.tail(3)
```

Paso 4: Estadísticas Rápidas

¿Cuál es el promedio de ataque de todos los Pokémon? ¿Cuál es la defensa máxima? Pandas te lo dice en una línea.

```
# .describe() nos da un resumen estadístico automático
df.describe()
```

Busca la fila "mean" (promedio), "min" (mínimo) y "max" (máximo).

Paso 5: Ordenando Datos (Respondiendo preguntas)

Vamos a responder: ¿Cuáles son los Pokémon más débiles? Para eso, ordenamos la tabla por "Total" (la suma de sus poderes) de menor a mayor.

```
# sort_values ordena los valores.
# ascending=True significa "ascendente" (de menor a mayor)
df.sort_values("Total", ascending=True).head(5)
```

Ahora al revés: ¿Cuáles son los más fuertes?

```
# ascending=False significa "descendente" (de mayor a menor)
df.sort_values("Total", ascending=False).head(5)
```

? Reto Final (Entregable)

En tu Notebook, escribe el código necesario para responder las siguientes preguntas utilizando lo que aprendiste hoy.

- Muestra los 5 Pokémon con mayor Defensa (Defense).
- Muestra los 5 Pokémon con mayor Velocidad (Speed).
- Pregunta de reflexión (escríbela en una celda de texto): Si tuvieras que elegir un Pokémon basándote solo en estos datos para ganar una carrera, ¿cuál elegirías y por qué?
- Sube el enlace de tu Colab a la plataforma.

Capítulo 2: Manipulación y Visualización

Práctica 4: Datos que Cuentan Historias (Gráficas).

Los seres humanos somos visuales. Si te muestro una tabla con 800 filas de números, te dolerá la cabeza. Pero si te muestro un gráfico colorido, tu cerebro detectará patrones en segundos.

Hoy convertiremos los datos aburridos en historias visuales. Usaremos dos librerías artísticas: **Matplotlib** (la base) y **Seaborn** (el maquillaje para que se vea genial).

? Objetivo de la Práctica

- Entender la importancia de la visualización de datos.
 - Crear **Histogramas** para ver distribuciones (¿Qué tan frecuentes son los datos?).
 - Crear **Diagramas de Dispersión** (Scatter Plots) para ver relaciones entre dos variables.
 - Interpretar gráficos para sacar conclusiones.
-

? Conceptos Clave

? Matplotlib vs Seaborn

- **Matplotlib:** Es el abuelo de las gráficas en Python. Es muy poderoso pero a veces requiere mucho código para que se vea bonito.
- **Seaborn:** Es una librería construida sobre Matplotlib. Hace gráficos estadísticos increíbles con una sola línea de código. ¡Usaremos ambos!

? Tipos de Gráficos Básicos

1. **Histograma:** Barras que nos dicen "cuántos hay de cada cosa".
 2. **Scatter Plot:** Puntos en un plano X-Y. Sirve para ver si una variable afecta a la otra (ej. ¿Más altura = Más peso?).
-

?? Instrucciones Paso a Paso

Nota: Como es una nueva sesión de Colab, necesitamos cargar los datos de nuevo.

Paso 1: Importar Librerías y Datos

En tu primera celda, preparamos todo. Fíjate que ahora importamos dos cosas nuevas.

```
import pandas as pd
import matplotlib.pyplot as plt # La librería gráfica base
import seaborn as sns # La librería gráfica bonita

# Cargamos el dataset de Pokémon nuevamente
url =
"[https://raw.githubusercontent.com/KeithGalli/pandas/master/pokemon_data.csv](https://raw.githubusercontent.com/KeithGalli/pandas/master/pokemon_data.csv)"
df = pd.read_csv(url)

print("Librerías gráficas y datos listos.")
```

Paso 2: El Histograma (Distribución del Ataque)

Queremos saber: ¿La mayoría de los Pokémon son débiles, fuertes o promedios? Para esto, analizaremos la columna Attack.

Copia y ejecuta:

```
# Configuramos el tamaño de la figura (Ancho, Alto)
plt.figure(figsize=(10, 6))

# Creamos el histograma
sns.histplot(df['Attack'], bins=20, color='skyblue')

# Ponemos títulos (¡Gráfico sin título no sirve!)
plt.title('Distribución del Ataque de los Pokémon')
plt.xlabel('Puntos de Ataque')
plt.ylabel('Cantidad de Pokémon')
plt.show() # Mostramos el dibujo
```

Analiza el resultado: Verás una "montaña". La parte más alta te dice dónde se encuentra la mayoría de los Pokémon (probablemente entre 50 y 80 de ataque).

Paso 3: Diagrama de Dispersión (Ataque vs. Defensa)

Ahora una pregunta más compleja: ¿Los Pokémon que pegan fuerte también aguantan mucho? (¿Hay relación entre Ataque y Defensa?).

Usaremos un Scatter Plot (nube de puntos).

```
plt.figure(figsize=(10, 6))

# x = Eje horizontal, y = Eje vertical
sns.scatterplot(data=df, x='Attack', y='Defense')

plt.title('Relación Ataque vs Defensa')
plt.show()
```

Analiza: Verás una nube dispersa. Si los puntos formaran una línea diagonal hacia arriba, significaría que "más ataque = más defensa". Aquí parece que no hay una regla clara, están por todos lados.

Paso 4: Agregando una tercera dimensión (Color)

El gráfico anterior es un poco aburrido. Vamos a ver si el Tipo de Pokémon (Fuego, Agua, Roca, etc.) influye. Usaremos el argumento hue (matiz/color) para colorear los puntos según su Type 1.

```
plt.figure(figsize=(12, 8)) # La hacemos más grande

# Agregamos hue='Type 1' para colorear por tipo
sns.scatterplot(data=df, x='Attack', y='Defense', hue='Type 1')

plt.title('Ataque vs Defensa por Tipo')
plt.show()
```

Analiza: ¡Ahora es arte moderno!

- Busca los puntos de color Roca (Rock) o Acero (Steel). ¿Notas que suelen estar más arriba en el eje Y (Defensa)?
- Busca los puntos Normal o Bicho (Bug). ¿Dónde suelen estar?

? Reto Final (Entregable)

Ahora te toca a ti ser el analista de datos.

- Crea un nuevo Scatter Plot que compare la Velocidad (Speed) en el eje X contra la Vida (HP) en el eje Y.
- Usa hue para colorearlos según si son Legendarios o no (la columna se llama Legendary).
- Pregunta de reflexión: Observando tu gráfico, ¿Los Pokémon legendarios (puntos de un color distinto) tienden a ser más rápidos y tener más vida que los normales? ¿O están mezclados?

Sube tu Notebook con los gráficos generados.

Práctica 5: Detectives de Datos (Limpieza de datos).

Capítulo 3: Aprendizaje Supervisado (Predicciones)

Práctica 6: Adivinando el Futuro (Regresión Lineal).

Práctica 7: ¿Clase A o Clase B? (KNN - Vecinos cercanos).

Práctica 8: El Árbol de las Decisiones (Decision Trees).

Práctica 9: ¿Qué tan bueno es mi modelo? (Evaluación y Métricas).

Capítulo 4: Aprendizaje No Supervisado (Patrones)

Práctica 10: Agrupando sin Etiquetas (Clustering K-Means).

Capítulo 5: Redes Neuronales (Deep Learning)

Práctica 11: La Neurona Artificial (Lógica del Perceptrón).

Práctica 12: Cerebros Digitales (Redes Densas y MNIST).

Práctica 13: Visión por Computadora (Clasificación de ropa/imágenes).