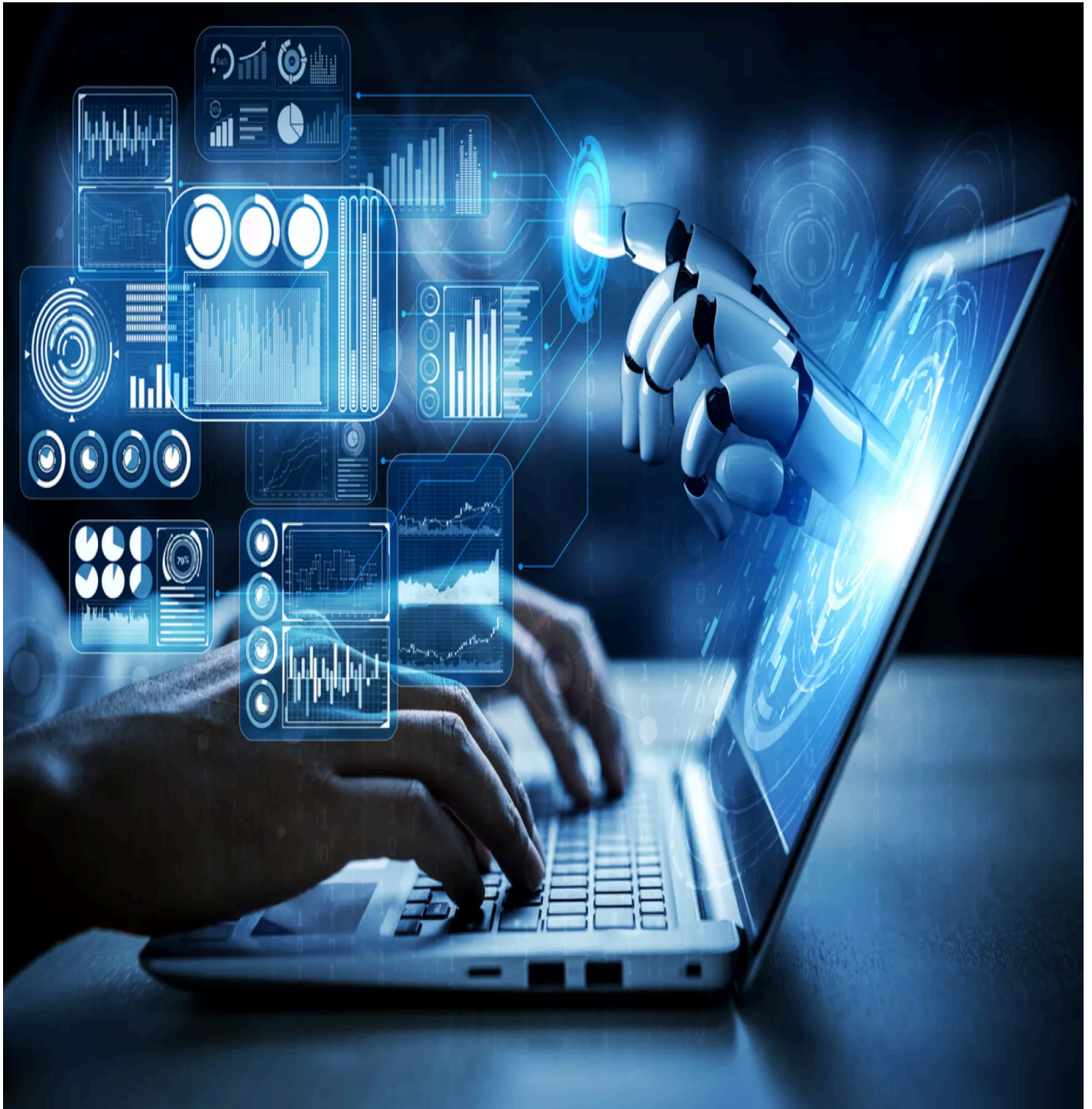


# Programación de Inteligencia Artificial



Nombre: Victoria Jiménez Martín

Módulo: Programación de Inteligencia Artificial

Curso: Especialización de Inteligencia Artificial y Big Data

# Índice

<b>Apartado 1: Programa tareas sencillas con Python en Colab</b>	<b>3</b>
• En la primera celda de código, escribe y ejecuta esta orden: "import this"	3
• En una siguiente celda, escribe el código necesario en Python para crear una lista de 10 nombres de ciudades y luego haz que se muestre utilizando print().	3
• En una nueva celda de código, crea una nueva lista con 10 nombres de artistas o cantantes, y utiliza print() para mostrarla.	4
• En una nueva celda de código, crea un diccionario en el que la clave sea el nombre del artista y el valor sea el nombre de una ciudad.	4
• En una nueva celda de código, escribe el código necesario para que, al ejecutarlo, se muestre en pantalla "Estos son los conciertos para esta semana: " + relación de artistas y ciudades	4
• En una nueva celda de código, escribe el código necesario para cambiar a de ciudad a todos los artistas para un nuevo concierto de su gira. Vuelve a programar también que se muestre la relación de artistas y ciudades.	5
<b>Apartado 2: Utiliza el lenguaje de marcación para dar contexto y documentar el notebook.</b>	<b>6</b>
• Incluye celdas de texto en tu cuaderno, entre las celdas de código, incluyendo, como mínimo, las siguientes:	6
○ Título y subtítulo.	6
○ Texto en negrita y en itálica.	6
○ Un enlace a la página web de Python.	6
○ Una imagen.	6

## Apartado 1: Programa tareas sencillas con Python en Colab

- En la primera celda de código, escribe y ejecuta esta orden: "import this"

```
import this
```

The Zen of Python, by Tim Peters

```
Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!
```

Apartado 1: Programa tareas sencillas con Python en Colab

- En una siguiente celda, escribe el código necesario en Python para crear una lista de 10 nombres de ciudades y luego haz que se muestre utilizando print().

```
Apartado 1: Programa tareas sencillas con Python en Colab
```

```
[ ] ciudades = ["Madrid", "Barcelona", "Valencia", "Sevilla", "Zaragoza", "Málaga", "Murcia", "Palma", "Bilbao", "Alicante"]
print(ciudades)
```

```
['Madrid', 'Barcelona', 'Valencia', 'Sevilla', 'Zaragoza', 'Málaga', 'Murcia', 'Palma', 'Bilbao', 'Alicante']
```

- En una nueva celda de código, crea una nueva lista con 10 nombres de artistas o cantantes, y utiliza print() para mostrarla.

```
[ ] artistas = ["Rosalía", "Bad Bunny", "Dua Lipa", "Drake", "Adele", "Taylor Swift", "Ed Sheeran", "Billie Eilish", "Bruno Mars", "Beyoncé"]
print(artistas)

['Rosalía', 'Bad Bunny', 'Dua Lipa', 'Drake', 'Adele', 'Taylor Swift', 'Ed Sheeran', 'Billie Eilish', 'Bruno Mars', 'Beyoncé']
```

- En una nueva celda de código, crea un diccionario en el que la clave sea el nombre del artista y el valor sea el nombre de una ciudad.

```
▶ conciertos = {
    "Rosalía": "Madrid",
    "Bad Bunny": "Barcelona",
    "Dua Lipa": "Valencia",
    "Drake": "Sevilla",
    "Adele": "Zaragoza",
    "Taylor Swift": "Málaga",
    "Ed Sheeran": "Murcia",
    "Billie Eilish": "Palma",
    "Bruno Mars": "Bilbao",
    "Beyoncé": "Alicante"
}
```

- En una nueva celda de código, escribe el código necesario para que, al ejecutarlo, se muestre en pantalla "Estos son los conciertos para esta semana: " + relación de artistas y ciudades

```
▶ mensaje = "Estos son los conciertos para esta semana: \n" + "\n".join([f"{artista} en {ciudad}" for artista, ciudad in conciertos.items()])
print(mensaje)

Estos son los conciertos para esta semana:
Rosalía en Madrid
Bad Bunny en Barcelona
Dua Lipa en Valencia
Drake en Sevilla
Adele en Zaragoza
Taylor Swift en Málaga
Ed Sheeran en Murcia
Billie Eilish en Palma
Bruno Mars en Bilbao
Beyoncé en Alicante
```

- En una nueva celda de código, escribe el código necesario para cambiar a de ciudad a todos los artistas para un nuevo concierto de su gira. Vuelve a programar también que se muestre la relación de artistas y ciudades.

```
ciudades_rotadas = ciudades[-1:] + ciudades[:-1]
conciertos_actualizados = {artista: ciudad for artista, ciudad in zip(artistas, ciudades_rotadas)}

mensaje_actualizado = "Estos son los conciertos para la próxima semana: \n" + "\n".join([f"{artista} en {ciudad}" for artista, ciudad in conciertos_actualizados.items()])
print(mensaje_actualizado)
```

Estos son los conciertos para la próxima semana:

Rosalía en Alicante  
Bad Bunny en Madrid  
Dua Lipa en Barcelona  
Drake en Valencia  
Adele en Sevilla  
Taylor Swift en Zaragoza  
Ed Sheeran en Málaga  
Billie Eilish en Murcia  
Bruno Mars en Palma  
Beyoncé en Bilbao

## Apartado 2: Utiliza el lenguaje de marcación para dar contexto y documentar el notebook.

- Incluye celdas de texto en tu cuaderno, entre las celdas de código, incluyendo, como mínimo, las siguientes:
  - Título y subtítulo.
  - Texto en negrita y en itálica.
  - Un enlace a la página web de Python.
  - Una imagen.

✓ Título Aquí

Subtítulo Aquí

**\*\* Este es el texto en negrita\*\***

*Este es el texto en itálica*

Enlace a la página de python

[Visita la página oficial de Python](#)

Esta es la imagen



**Enlace al entorno colaborativo:**

<https://colab.research.google.com/drive/1IPeZMJmTxkQK-pVnHK4mtaeFsq1UMnF?usp=sharing>