



«Talento Tech»

# Iniciación a la Programación con Python

CLASE 8



## Clase N° 8 | Preentrega del Proyecto Final Integrador (PFI)

### Temario:

- Creación de un menú interactivo utilizando bucles while y condicionales if-elif-else
- Implementación de una funcionalidad para agregar productos a una lista
- Visualización de los productos ingresados en el inventario

## Preentrega del Proyecto Final Integrador (PFI)

Hoy llegamos a un punto clave en nuestro curso: la preentrega del Proyecto Final Integrador (PFI). A lo largo de las clases, hemos recorrido juntos un camino de aprendizaje que cubre los fundamentos de Python, desde las variables y operadores hasta las estructuras de control y listas. Ahora es el momento de poner a prueba esos conocimientos y avanzar hacia la construcción de tu PFI.

## Entrega de la Clase 8: Implementación Parcial del Proyecto Final Integrador (PFI)

Hasta este punto del curso, ya hemos cubierto los fundamentos de Python, desde la creación de algoritmos y el manejo de variables, hasta el uso de condicionales, bucles while, y listas. Ahora, es el momento de poner en práctica estos conceptos para avanzar hacia tu Proyecto Final Integrador (PFI). La entrega consiste en construir una versión parcial del sistema de inventario, utilizando los conocimientos adquiridos hasta el momento. Es una gran oportunidad para que pruebes las ideas de tu proyecto y consolides tu aprendizaje.

### Objetivos:

- 1) Crear un menú interactivo utilizando bucles while y condicionales if-elif-else: El menú debe permitir seleccionar entre diferentes opciones relacionadas con la gestión de productos. Entre las opciones deben incluirse: agregar productos al inventario y mostrar los productos registrados.

**Tip:** tu menú puede incluir opciones como "1. Agregar producto", "2. Mostrar productos" y "3. Salir". Usá `int()` para convertir la entrada ingresada en un número.

- 2) Implementar la funcionalidad para agregar productos a una lista: cada producto debe ser almacenado en una lista y tener al menos un nombre y una cantidad asociada.



**Tip:** creá una lista vacía al comienzo del programa donde vayas agregando los productos ingresados y usá `append()` para añadirlos.

- 3) Mostrar los productos ingresados: al seleccionar la opción correspondiente, el sistema debe permitir visualizar los productos almacenados hasta el momento.

**Tip:** si la lista está vacía podés mostrar un mensaje que de aviso en caso de que no haya productos cargados aún. Si hay productos, imprimí cada uno de manera clara.

Esta entrega te va a permitir sentar las bases para el desarrollo del inventario que será parte de tu PFI. Recordá seguir las buenas prácticas de codificación que hemos discutido en clase y utilizar bucles, listas y condicionales de manera eficiente.

## Menú simple con while y if.

Imaginá que tenés un programa que permite interactuar con diferentes opciones. Este menú se basa en un bucle while, que mantiene el programa corriendo hasta que la persona elija la opción de salir. Cada vez que ingresa una opción, el programa responde en consecuencia. Si ingresa una opción incorrecta, también se lo hacemos saber. Veamos cómo funciona el código:

```
opcion = 0

while opcion != 3:
    print("Menú Interactivo")
    print("1. Opción 1")
    print("2. Opción 2")
    print("3. Salir")

    opcion = int(input("Seleccioná una opción: "))

    if opcion == 1:
        print("Seleccionaste la Opción 1")
    elif opcion == 2:
        print("Seleccionaste la Opción 2")
    elif opcion == 3:
        print("Saliendo del menú...")
    else:
        print("Opción inválida, intentá nuevamente.")
```



El programa empieza imprimiendo las opciones disponibles y luego pide que se ingrese una elección. Luego, dependiendo de la opción seleccionada, se ejecuta un bloque de código específico. El bucle sigue corriendo mientras no se elige la opción de salir. Usamos if, elif y else para manejar cada opción del menú.

```
Menú Interactivo
1. Opción 1
2. Opción 2
3. Salir
Seleccioná una opción: 2
Seleccionaste la Opción 2
Menú Interactivo
1. Opción 1
2. Opción 2
3. Salir
Seleccioná una opción: 3
Saliendo del menú...
```

Este tipo de estructura es muy útil para crear menús interactivos en cualquier tipo de aplicación. Pensá en cómo podrías usarlo para gestionar los productos de tu inventario en el Trabajo Final Integrador (TFI).

## Agregar nombres a una lista.

Este es un ejemplo que te muestra cómo agregar nombres a una lista. Lo que hacemos es inicializar una lista vacía. Luego, a través de un bucle while, preguntamos si la persona quiere agregar más nombres (o productos). Si responde que sí, pedimos el nombre (o la descripción del producto) y lo agregamos a la lista.

```
nombres = []

agregar = "sí"

while agregar == "sí":
    nombre = input("Ingresá un nombre: ")
    nombres.append(nombre)

    agregar = input("¿Querés agregar otro nombre? (sí/no): ")

print("Los nombres ingresados son:")
for nombre in nombres:
    print(nombre)
```

La magia ocurre cuando usamos el método `.append()`, que es el que permite ir sumando elementos a la lista uno por uno. El bucle sigue corriendo hasta que la persona ingresa "no". Cuando el bucle termina, mostramos todos los nombres que se han guardado.

```
Ingresá un nombre: Pedro
¿Querés agregar otro nombre? (sí/no): sí
Ingresá un nombre: Ana
¿Querés agregar otro nombre? (sí/no): no
Los nombres ingresados son:
Pedro
Ana
```



Si lo pensás en términos de tu PFI, podrías reemplazar los nombres por productos y la lista de nombres por una lista de inventario. De esta manera, vas sumando productos uno por uno y después podés mostrarlos todos juntos asegurándote de que todo lo ingresado quede registrado.

## Validar entrada de números.

Cuando estás trabajando con un sistema como el de tu inventario, no querés permitir el ingreso de cualquier valor. Por ejemplo, si alguien sugiere una cantidad negativa de productos, eso no tendría sentido para nuestro programa. Por eso este ejemplo es clave: te enseña cómo validar la entrada de quien interactúa con tu aplicación. Veamos cómo:

```
cantidad = -1

while cantidad <= 0:
    cantidad = int(input("Ingresá la cantidad de productos (debe ser mayor a 0): "))
    if cantidad <= 0:
        print("La cantidad no puede ser menor o igual a 0. Intentá de nuevo.")

print("Ingresaste una cantidad válida:", cantidad)
```

Lo que hacemos es pedir una cantidad y si ésta es menor o igual a 0 mostramos un mensaje de error y volvemos a pedir que se ingrese un valor. El bucle sigue corriendo hasta que el valor ingresado es mayor que 0. Sólo entonces el programa continúa.



```
Ingresá la cantidad de productos (debe ser mayor a 0): -3
La cantidad no puede ser menor o igual a 0. Intentá de nuevo.
Ingresá la cantidad de productos (debe ser mayor a 0): 0
La cantidad no puede ser menor o igual a 0. Intentá de nuevo.
Ingresá la cantidad de productos (debe ser mayor a 0): 10
Ingresaste una cantidad válida: 10
```

Es muy importante tener este tipo de control en tu proyecto. Si pensás en tu inventario, no podrías permitir cantidades negativas o precios nulos, porque esos valores no tendrían lógica en un sistema de productos. Este código es una base sólida para asegurarte de que la información que se ingresa en tu sistema siempre sea válida y esté en el formato correcto.

## Recorrer una lista con while.

Recorrer una lista es una de las cosas más comunes que vas a hacer en cualquier programa, no sólo en el PFI. Este ejemplo te muestra cómo hacerlo con un bucle while. Lo que hacemos es definir una lista de productos y luego recorrerla para mostrar cada uno de los elementos.

```
productos = ["Producto A", "Producto B", "Producto C"]
indice = 0

while indice < len(productos):
    print("Producto:", productos[indice])
    indice = indice + 1
```



Usamos un contador (llamado índice) para representar la posición actual en la lista. El bucle sigue corriendo mientras el valor del índice sea menor que la longitud de la lista. En cada iteración, el índice se incrementa para avanzar al siguiente producto y el bucle se detiene cuando hemos recorrido todos los productos.

```
Producto: Producto A
Producto: Producto B
Producto: Producto C
```

En un proyecto como el PFI esta herramienta resulta fundamental. Imaginá que necesitás mostrar todos los productos que tenés almacenados. Usando un bucle como este, podés recorrer tu lista de productos de manera eficiente y presentarlos de forma ordenada. Es una herramienta sencilla, pero increíblemente versátil y que, además, ofrece un gran margen para mejorarla y ampliarla según lo que tu sistema necesite en el futuro.

## Ejercicios Prácticos:

### Registro de ventas por día:

Desarrollá un programa que permita registrar las ventas diarias de un comercio durante cinco días. Al finalizar, el sistema debe mostrar el total de ventas realizadas en cada día y el promedio de ventas.

#### Tips:

- Utilizá un bucle while que permita a la persona usuaria ingresar el monto de las ventas diarias.
- Asegurate de validar que el monto ingresado sea un valor positivo.
- Usá un acumulador para la suma de las ventas.

## Actualización del inventario a partir de un arreglo:

En una tienda es necesario actualizar el inventario cuando se venden productos. A continuación, te proporcionamos un arreglo con una lista de productos, donde cada producto tiene un código, una descripción y una cantidad en stock. Escribí un programa que permita:

- Seleccionar un producto a partir de su código.
- Ingresar la cantidad vendida (que debe ser mayor que cero).
- Actualizar la cantidad en stock de ese producto restando la cantidad vendida.

El arreglo de productos disponibles es el siguiente:

```
productos = [  
    ["P001", "Manzanas", 50],  
    ["P002", "Peras", 40],  
    ["P003", "Bananas", 30],  
    ["P004", "Naranjas", 60]  
]
```

El script que tenés que hacer debe modificar la cantidad en stock de acuerdo a cada venta realizada. Si la cantidad vendida es mayor que la cantidad disponible en stock, el programa debe mostrar un mensaje de error.

**Buenos Aires**  
*aprende* 

Agencia de Habilidades para el Futuro

