

# Pre Entrega de Proyecto

Clase 08 - “Pre entrega de proyecto”



# Repaso



# Listas en Python

Una lista es una estructura que te permite almacenar varios valores dentro de una sola variable. Una lista se crea encerrando los elementos entre corchetes [] y separándolos por comas:

```
vehiculos = ["coche", "tren", "avión", "barco"]
```

Cada elemento tiene una posición específica dentro de la lista, lo que nos permite acceder a ellos cuando lo necesitamos.

```
vehiculos = ["coche", "tren", "avión", "barco"]  
print(vehiculos[0]) # Imprime "coche"  
print(vehiculos[2]) # Imprime "avión"
```



# Recorrer una lista con while

Este ejemplo te muestra cómo recorrer una lista con un bucle while. Lo que hacemos es definir una lista de productos y luego recorrerla para mostrar cada uno de los elementos.

```
productos = ["manzanas", "pan", "leche"]
indice = 0

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

Probá este código.

```
Producto 1 : manzanas
Producto 2 : pan
Producto 3 : leche
```

Verás esto en la terminal.



# ¿Cómo crear un menú?

Podés usar 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.

```
while opcion != 3:  
    print("Menú Interactivo")  
    print("1. Opción 1")  
    print("2. Opción 2")  
    print("3. Salir")  
    opcion = int(input("Seleccioná una opción: "))  
  
# Aquí va el código que gestiona las opciones
```



# Validar datos

Debemos asegurarnos que los datos que ingresa el usuario se ajusten a lo que el script requiere. Recordemos cómo hacerlo:

```
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)
```

# ¡Vamos a la práctica!





# Ejercicios prácticos



Optativos | No entregables

## Registro de ventas por día

Desarrollá un programa que permita registrar las ventas diarias de un comercio durante 5 días. Al finalizar, el sistema debe mostrar el total de ventas realizadas en cada día y el promedio de ventas.

### Tips:

- Usá un bucle while que permita al usuario 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.





# Ejercicios prácticos



Optativos | No entregables

## 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.



# Ejercicios prácticos



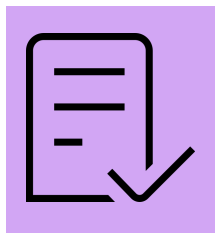
Optativos | No entregables

## Actualización del inventario a partir de un arreglo

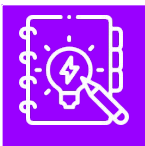
El arreglo de productos disponibles es que ves a continuación.

```
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.



# Pre Entrega de Proyecto



# Revisión de progreso



Obligatorio | Entregable

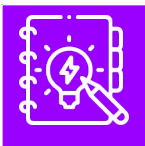
## Revisión de progreso

Estamos acercándonos a un **punto clave en el desarrollo del proyecto**, donde tendrán la oportunidad de compartir los avances realizados hasta el momento actual.

En este punto del proceso es fundamental que **reúnan todo el trabajo que han desarrollado hasta la clase N°8**. Esta es una gran oportunidad para consolidar sus ideas y asegurarse de que están en el camino correcto

## Formato de Entrega:

**Compartir un link al drive** (público) que contenga los archivos y carpetas que conforman tu proyecto. Los links deberán ser entregados en el apartado de “Pre-Entrega de Proyecto” en el Campus Virtual.



# Revisión del progreso



Obligatorio | Entregable

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 tarea de esta entrega consiste en construir una versión parcial del sistema de inventario, utilizando los conocimientos adquiridos.

```
import keras.callbacks as callbacks
from keras import optimizers
from keras.optimizers import Adam

from sklearn.model_selection import GridSearchCV
from keras.wrappers.scikit_learn import KerasClassifier
from keras.callbacks import EarlyStopping
def ANN(optimizer = 'sgd', neurons=16, batch_size=1024, epochs=80, activation='relu', patience=10):
    K.clear_session()
    inputs=Input(shape=(X.shape[1],))
    x=Dense(1000)(inputs)
    x=BatchNormalization()(x)
    x=Activation('relu')(x)
    x=Dropout(0.3)(x)
    x=Dense(256)(inputs)
    x=BatchNormalization()(x)
    x=Activation('relu')(x)
    x=Dropout(0.25)(x)
    x=Dense(2, activation='softmax')(x)
    model=Model(inputs=inputs, outputs=x, name='base_nlp')
    model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
    model.compile(optimizer=Adam(lr = 0.01), loss='categorical_crossentropy', metrics=['accuracy'])
    #
    early_stopping = EarlyStopping(monitor="loss", patience = patience)# early stop patience
    history = model.fit(X, pd.get_dummies(y).values,
                        validation_data=(X_val, pd.get_dummies(y_val).values),
                        batch_size=batch_size,
```



# Revisión del progreso



Obligatorio | Entregable

Esta instancia evaluativa es de carácter obligatorio y es un punto clave dentro de la cursada ya que nos permitirá evaluar tu progreso en el recorrido y asegurar que estás en el camino correcto en la construcción del **Proyecto Final Integrador**.

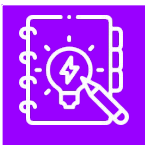
Este espacio de entrega está conformado por:

Cuestionario de  
Autoevaluación



Pre-Entrega del  
Proyecto





# Revisión del progreso



Obligatorio | Entregable

A partir de la **Clase N° 8** tendrás **7 días** de corrido para realizar la autoevaluación y la entrega en el campus virtual



## Cuestionario de Autoevaluación

Te permitirá reflexionar sobre tu propio aprendizaje, progreso y cumplimiento de las consignas o rúbricas previamente establecidas y en caso de ser necesario realizar las modificaciones o ajustes correspondientes antes de realizar la preentrega.

Se encontrará disponible en la Ruta N°2 de Campus Virtual

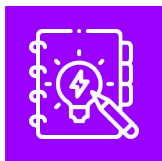


## Preentrega de proyecto:

Se evaluará la aplicación de los conocimientos adquiridos en la realización de un proyecto.

La realización progresiva de los "**Ejercicios prácticos**" y el seguimiento de las rúbricas de la "**Ruta de Avance**" los guiará paso a paso hacia la realización de la "**Preentrega**" y el "**Proyecto Integrador Final**"

Se entregará en la Ruta N°2 de Campus Virtual



# Preentrega de proyecto



Obligatorio | Entregable

**Formato de entrega:** Crear una carpeta en drive (pública) que contenga los archivos y carpetas que conforman tu proyecto. Compartir el link en el apartado de entrega en el Campus Virtual.

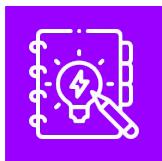
## Requisitos para la entrega:

### 1. Crear un menú interactivo

Crear un menú interactivo utilizando bucles while y condicionales if-elif-else:

- El menú debe permitir al usuario seleccionar entre diferentes opciones relacionadas con la gestión de productos.
- Entre las opciones, deben incluirse: agregar productos al inventario y mostrar los productos registrados.





# Preentrega de proyecto



Obligatorio | Entregable

## 2. Agregar productos al inventario

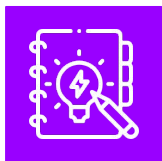
Implementar la funcionalidad para agregar productos a una lista:

- Cada producto debe ser almacenado en una lista, y debe tener al menos un nombre y una cantidad asociada.

## 3. Mostrar el inventario

Mostrar los productos ingresados:

- Al seleccionar la opción correspondiente, el sistema debe permitir visualizar los productos almacenados hasta el momento.



# Preentrega de proyecto



Requisitos para la entrega:

## Recuerda:

Esta instancia de entrega te va a permitir sentar las bases para el desarrollo del inventario que será parte de tu Proyecto Final Integrador.

Recuerda seguir las buenas prácticas de codificación que hemos discutido en clase y utilizar bucles, listas y condicionales de manera eficiente.



Tendrás 7 días de corrido para realizar la entrega



# ¡NUEVO CUESTIONARIO EN CAMPUS!

La resolución del cuestionario es de carácter obligatorio para poder avanzar en la cursada.