Buenos Aires aprende
Agencia de Habilidades para el Futuro

Talento Tech

Ruta de Avance

Clase 04 - “Primeros pasos”

Buenos Aires Ciudad

Ruta de Avance

Optativos | No entregables

Nos ponemos en marcha

A lo largo del curso debemos desarrollar un Proyecto Integrador, de forma individual.

Este proyecto consiste en una aplicación con Python que permita gestionar el inventario de una pequeña tienda o comercio. La aplicación debe ser capaz de **registrar**, **actualizar**, **eliminar** y **mostrar** productos en el inventario. Además, debe incluir funcionalidades para realizar búsquedas y generar reportes de stock.

Objetivos de esta etapa

- Revisar la definición inicial de los requisitos del Proyecto Final Integrador (PFI).
- Construir la estructura básica del programa utilizando los conceptos de entrada, procesamiento y salida de datos.

Ruta de Avance

Optativos | No entregables

Entrada, Procesamiento y Salida:

- Desarrollar un pequeño script que permita al usuario ingresar datos (por ejemplo, el nombre y la cantidad de un producto) y luego mostrar esos datos de manera simple.
- Utilizar funciones input() para la entrada y print() para la salida, siguiendo el modelo de Entrada-Proceso-Salida discutido en clases anteriores.

Esqueleto del Proyecto:

- Crear un esqueleto de su programa, donde definen las funciones principales que necesitarán (sin implementarlas por completo).
- Enfocarse en cómo organizar el código para futuras expansiones, utilizando variables y estructuras básicas

Buenos Aires aprende
Agencia de Habilidades para el Futuro

Talento Tech

Pre - Entrega de Proyecto

Clase 08 - “Pre - Entrega del Proyecto”

Buenos Aires Ciudad

Revisión de 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 kcallbacks
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
    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')
    model.compile(optimizer=Adam(lr = 0.01),loss='categorical_crossentropy',metrics=['a
#
early_stopping = EarlyStopping(monitor="loss", patience = patience)# early stop patie
history = model.fit(X, pd.get_dummies(y).values,
                    batch_size,
```

Buenos Aires aprende
Agencia de Habilidades para el Futuro

Buenos Aires Ciudad

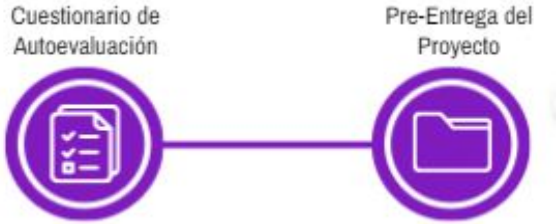
Talento Tech

Revisión de 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:



Revisión de 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 Pre-Entrega.

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

Pre-Entrega 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 "**Pre - Entrega**" y el "**Proyecto Integrador Final**"

Se entregará en la Ruta N°2 de Campus Virtual

Pre Entrega 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.

Pre Entrega 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.

Pre Entrega de Proyecto

Requisitos para la entrega:

Recuerda:

Esta pre-entrega e 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

Buenos Aires aprende
Agencia de Habilidades para el Futuro

Talento Tech

Ruta de Avance

Clase 12 - “ Funciones”

Buenos Aires Ciudad

Ruta de Avance

Optativos | No entregables

Objetivos de esta etapa

1. Trabajaremos en la creación de **funciones modulares** que se integrarán en el proyecto.
2. Nos aseguraremos que las funciones implementadas hasta ahora sean **reutilizables** y estén **organizadas** de manera lógica.
3. Prepararemos la estructura del código para la futura integración con bases de datos.

Ruta de Avance

Optativos | No entregables

Desarrollo de funciones modulares

1. Crearemos funciones específicas como **agregar_producto**, **mostrar_productos**, **actualizar_producto**, y **eliminar_producto** utilizando los conocimientos adquiridos hasta la **Clase 11**.
2. Cada función debe ser independiente y modular, facilitando la integración con otras partes del código.

Práctica y Feedback

1. Implementamos y probamos las funciones.
2. Analizamos maneras de mejorar la modularidad y la reutilización del código.
3. Prepararse para la integración de bases de datos SQLite en la Clase 16, donde se completará la estructura del Proyecto Final Integrador.

Buenos Aires aprende
Agencia de Habilidades para el Futuro

Talento Tech

Proyecto Final

Clase 15 - “Rúbricas PFI”

Buenos Aires Ciudad

Proyecto Integrador

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.

Proyecto Final Integrador:

1. Deberán desarrollar una aplicación en Python que permita gestionar el inventario de una pequeña tienda.
2. La aplicación debe ser capaz de registrar, actualizar, eliminar y mostrar productos en el inventario.
3. Además, debe incluir funcionalidades para realizar búsquedas y generar reportes de stock.

Proyecto Integrador

Obligatorio | Entregable

Requerimientos

1. Crear una base de datos SQLite para almacenar los datos de los productos (nombre, descripción, cantidad, precio, categoría).
2. Implementar una interfaz de usuario básica para interactuar con la base de datos desde la terminal (línea de comandos).
3. Incluir funcionalidades de registro, actualización, eliminación y visualización de productos.
4. Generar reportes de productos con bajo stock.

Objetivos de aprendizaje

1. Implementar estructuras de control y funciones en Python.
2. Desarrollar habilidades de manipulación de archivos y manejo de datos.
3. Aplicar conocimientos de bases de datos SQLite.

Proyecto Integrador

Obligatorio | Entregable

Funcionalidades de la aplicación

- Registro de productos: La aplicación debe permitir al usuario agregar nuevos productos al inventario, solicitando los siguientes datos: nombre, descripción, cantidad, precio y categoría.
- Visualización de productos: La aplicación debe mostrar todos los productos registrados en el inventario, incluyendo su ID, nombre, descripción, cantidad, precio y categoría.
- Actualización de productos: La aplicación debe permitir al usuario actualizar la cantidad disponible de un producto específico utilizando su ID.
- Eliminación de productos: La aplicación debe permitir al usuario eliminar un producto del inventario utilizando su ID.
- Búsqueda de productos: La aplicación debe ofrecer una funcionalidad para buscar productos por su ID, mostrando los resultados que coincidan con los criterios de búsqueda. De manera opcional, se puede implementar la búsqueda por los campos nombre o categoría.
- Reporte de Bajo Stock: La aplicación debe generar un reporte de productos que tengan una cantidad igual o inferior a un límite especificado por el usuario.

Proyecto Integrador

Obligatorio | Entregable

Base de datos

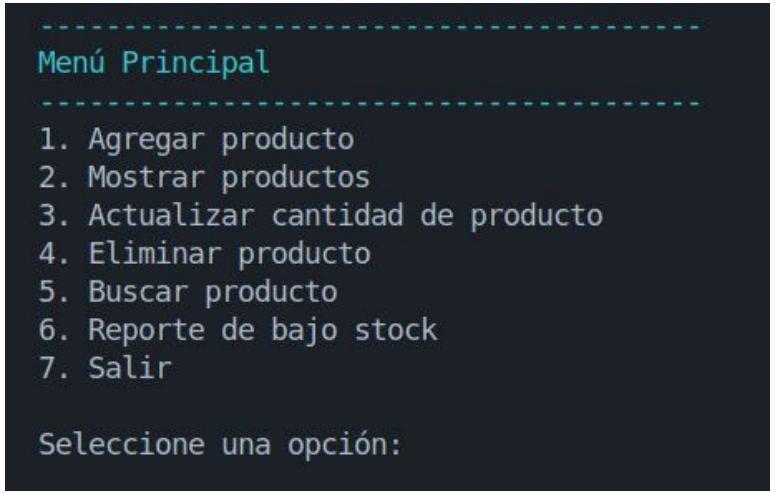
- Crear una base de datos SQLite llamada 'inventario.db' para almacenar los datos de los productos.
- La tabla 'productos' debe contener las siguientes columnas:
- 'id': Identificador único del producto (clave primaria, autoincremental).
- 'nombre': Nombre del producto (texto, no nulo).
- 'descripcion': Breve descripción del producto (texto).'cantidad': Cantidad disponible del producto (entero, no nulo).
- 'precio': Precio del producto (real, no nulo).
- 'categoria': Categoría a la que pertenece el producto (texto).

Proyecto Integrador

Obligatorio | Entregable

Interfaz de usuario

1. Implementar una interfaz de usuario básica para interactuar con la base de datos a través de la línea de comandos (terminal). La interfaz debe incluir un menú principal con las opciones necesarias para acceder a cada funcionalidad descrita anteriormente.
2. Opcional: Utilizar la librería 'colorama' para mejorar la legibilidad y experiencia de usuario en la terminal, añadiendo colores a los mensajes y opciones.

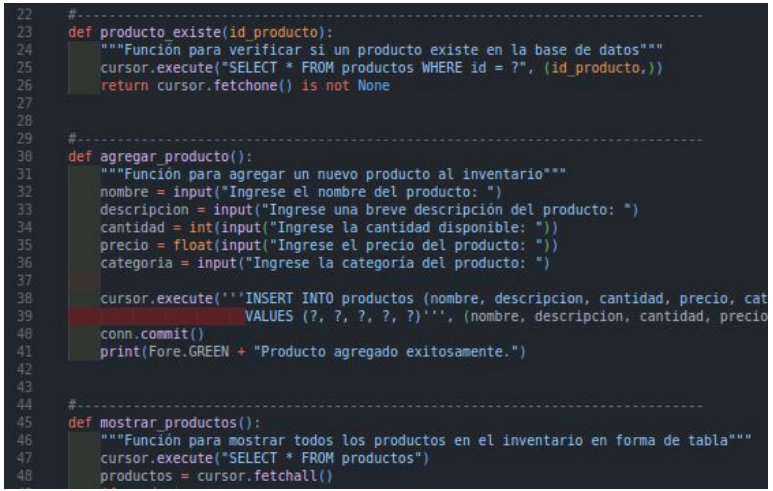


Proyecto Integrador

Obligatorio | Entregable

Requisitos técnicos

1. El código debe estar bien estructurado, utilizando funciones para modularizar la lógica de la aplicación.
2. Los comentarios deben estar presentes en el código, explicando las partes clave del mismo.



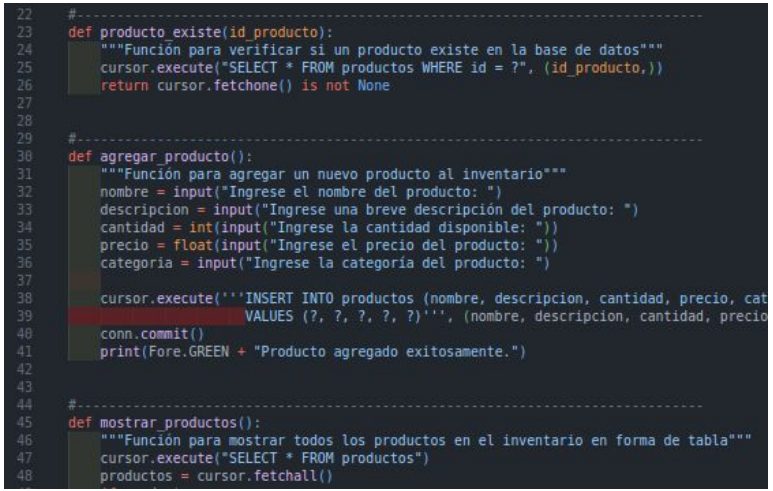
```python
#------------------------------------------------------------------------------
def producto_existe(id_producto):
    """Función para verificar si un producto existe en la base de datos"""
    cursor.execute("SELECT * FROM productos WHERE id = ?", (id_producto,))
    return cursor.fetchone() is not None


#------------------------------------------------------------------------------
def agregar_producto():
    """Función para agregar un nuevo producto al inventario"""
    nombre = input("Ingrese el nombre del producto: ")
    descripcion = input("Ingrese una breve descripción del producto: ")
    cantidad = int(input("Ingrese la cantidad disponible: "))
    precio = float(input("Ingrese el precio del producto: "))
    categoria = input("Ingrese la categoría del producto: ")

    cursor.execute('''INSERT INTO productos (nombre, descripcion, cantidad, precio, cat
                      VALUES (?, ?, ?, ?, ?)''', (nombre, descripcion, cantidad, precio
    conn.commit()
    print(Fore.GREEN + "Producto agregado exitosamente.")


#------------------------------------------------------------------------------
def mostrar_productos():
    """Función para mostrar todos los productos en el inventario en forma de tabla"""
    cursor.execute("SELECT * FROM productos")
    productos = cursor.fetchall()
```

Proyecto Integrador

Obligatorio | Entregable

Requisitos técnicos

1. El código debe estar bien estructurado, utilizando funciones para modularizar la lógica de la aplicación.
2. Los comentarios deben estar presentes en el código, explicando las partes clave del mismo.



```python
#------------------------------------------------------------------------------
def producto_existe(id_producto):
    """Función para verificar si un producto existe en la base de datos"""
    cursor.execute("SELECT * FROM productos WHERE id = ?", (id_producto,))
    return cursor.fetchone() is not None


#------------------------------------------------------------------------------
def agregar_producto():
    """Función para agregar un nuevo producto al inventario"""
    nombre = input("Ingrese el nombre del producto: ")
    descripcion = input("Ingrese una breve descripción del producto: ")
    cantidad = int(input("Ingrese la cantidad disponible: "))
    precio = float(input("Ingrese el precio del producto: "))
    categoria = input("Ingrese la categoría del producto: ")

    cursor.execute('''INSERT INTO productos (nombre, descripcion, cantidad, precio, cat
                      VALUES (?, ?, ?, ?, ?)''', (nombre, descripcion, cantidad, precio
    conn.commit()
    print(Fore.GREEN + "Producto agregado exitosamente.")


#------------------------------------------------------------------------------
def mostrar_productos():
    """Función para mostrar todos los productos en el inventario en forma de tabla"""
    cursor.execute("SELECT * FROM productos")
    productos = cursor.fetchall()
```

Proyecto Integrador

Obligatorio | Entregable

Entrega

1. El proyecto final debe ser entregado en el campus virtual mediante un LINK. Los archivos que conforman el proyecto deberán estar alojados en una carpeta de Google Drive (público) y debe incluir:
2. El script en Python ('.py') con el código fuente de la aplicación.
3. La base de datos SQLite ('inventario.db'), si es que se ha generado con datos de prueba.
4. Un archivo 'README.txt' explicando cómo ejecutar la aplicación y las funcionalidades implementadas.