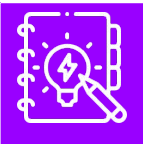


Proyecto Final

Clase 16 - “ Rúbricas PFI”



Proyecto Integrador

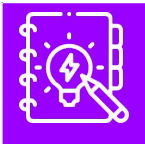


Obligatorio | Entregable

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



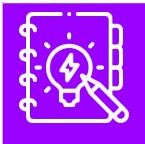
Obligatorio | Entregable

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.



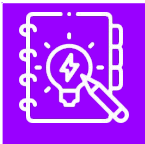
Proyecto Integrador



Obligatorio | Entregable

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).



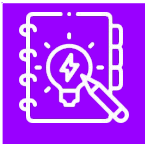
Proyecto Integrador



Obligatorio | Entregable

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

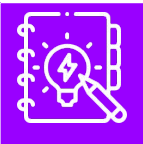
Interfaz de usuario

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.

Menú Principal

1. Agregar producto
2. Mostrar productos
3. Actualizar cantidad de producto
4. Eliminar producto
5. Buscar producto
6. Reporte de bajo stock
7. Salir

Seleccione una opción:



Proyecto Integrador

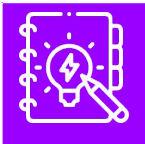


Obligatorio | Entregable

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.

```
22 #-----
23 def producto existe(id_producto):
24     """Función para verificar si un producto existe en la base de datos"""
25     cursor.execute("SELECT * FROM productos WHERE id = ?", (id_producto,))
26     return cursor.fetchone() is not None
27
28 #-----
29
30 def agregar_producto():
31     """Función para agregar un nuevo producto al inventario"""
32     nombre = input("Ingrese el nombre del producto: ")
33     descripcion = input("Ingrese una breve descripción del producto: ")
34     cantidad = int(input("Ingrese la cantidad disponible: "))
35     precio = float(input("Ingrese el precio del producto: "))
36     categoria = input("Ingrese la categoría del producto: ")
37
38     cursor.execute('INSERT INTO productos (nombre, descripcion, cantidad, precio, cat
39                   VALUES (?, ?, ?, ?, ?)', (nombre, descripcion, cantidad, precio
40     conn.commit()
41     print(Fore.GREEN + "Producto agregado exitosamente.")
42
43 #-----
44
45 def mostrar_productos():
46     """Función para mostrar todos los productos en el inventario en forma de tabla"""
47     cursor.execute("SELECT * FROM productos")
48     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.