



INSTITUTO TECNOLÓGICO HOWARD W. HUNTER
5TO. BACHILLER EN CIENCIAS Y LETRAS
CON ORIENTACIÓN EN COMPUTACIÓN

CURSO: Laboratorio 2

DOCENTE: Jhonathan Cocon

TEMA:
Sistema De Inventario

CHAN POL, JUNIOR WILFREDO ANTONIO
AREVALO CASTAÑEDA. MILTHON ISAI

SANTA ELENA, FLORES, PETÉN 13 DE MAYO DE 2025

INDICE

| | |
|---|------------|
| INTRODUCCION | II |
| CONCEPTOS | III |
| Sistema para un Negocio | III |
| SQL Server..... | III |
| Python..... | III |
| Código 1. Gestión de pedidos en Python con interfaz Tkinter y conexión a SQL Server | VI |

INDICE DE IMAGENES

Ilustración 1 Inicio IV

Ilustración 2 Botones..... IV

Ilustración 3 Consulta de Datos V

Ilustración 4 Eliminación V

INTRODUCCION

En la actualidad, las organizaciones dependen cada vez más de soluciones tecnológicas para optimizar sus procesos internos y mejorar la toma de decisiones. Un sistema de negocio eficiente permite automatizar tareas, gestionar datos de manera efectiva y generar reportes útiles para la administración. Este proyecto presenta el desarrollo de un sistema básico de gestión para una empresa ficticia, utilizando SQL Server como sistema de gestión de bases de datos (SGBD) y Python como lenguaje de programación principal para la lógica de negocio e interacción con la base de datos. Se incluye una interfaz gráfica amigable para el usuario y se documenta su funcionamiento mediante capturas y notas explicativas.

Es muy importante destacar que, tal como mencionas, los principios de diseño y la lógica implementada en este sistema con SQLite pueden ser perfectamente adaptados a motores de bases de datos más robustos como SQL Server. Esto le otorga al sistema una gran escalabilidad, permitiendo que crezca y se adapte a las necesidades de empresas de mayor tamaño sin tener que empezar desde cero.

CONCEPTOS

Sistema para un Negocio

Un sistema para un negocio es una aplicación informática diseñada para automatizar procesos administrativos y operativos dentro de una empresa. Estos sistemas permiten mejorar la eficiencia, reducir errores y tomar decisiones basadas en datos actualizados.

SQL Server

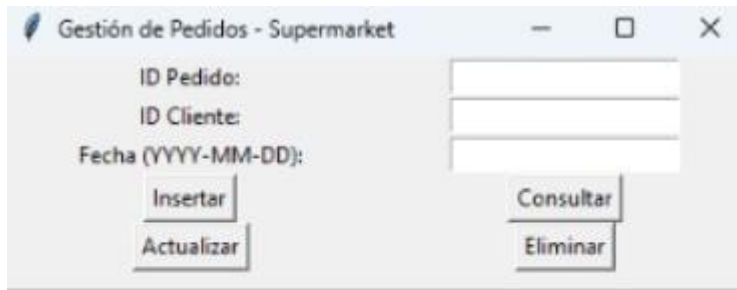
SQL Server es un sistema de gestión de bases de datos relacional (RDBMS) desarrollado por Microsoft. Se utiliza ampliamente en entornos empresariales para almacenar y administrar grandes volúmenes de información mediante el lenguaje SQL.

Python

Python es un lenguaje de programación de alto nivel, ampliamente utilizado por su sintaxis clara y versatilidad. Junto con bibliotecas como Tkinter, permite el desarrollo de aplicaciones con interfaz gráfica de usuario (GUI), conectadas a bases de datos como SQLite o SQL Server.

CAPTURAS

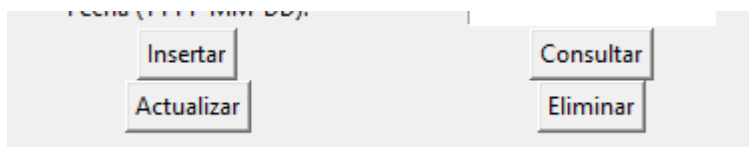
Ilustración 1 Inicio



The screenshot shows a window titled "Gestión de Pedidos - Supermarket". It contains three input fields: "ID Pedido:", "ID Cliente:", and "Fecha (YYYY-MM-DD):". Below these fields are four buttons: "Insertar", "Actualizar", "Consultar", and "Eliminar". The "Insertar" and "Actualizar" buttons are on the left, while "Consultar" and "Eliminar" are on the right.

NOTA: En esta imagen se muestra el inicio de el sistema de inventarios donde nos muestra los diferentes botones para acceder a funciones y el cual ingresar el ID de pedido como el del cliente.

Ilustración 2 Botones



This is a close-up of the buttons from the previous screenshot. It shows the "Insertar" and "Actualizar" buttons on the left, and the "Consultar" and "Eliminar" buttons on the right.

NOTA: En dicha imagen se muestran los diferentes botones de la aplicación

Ilustración 3 Consulta de Datos

The screenshot shows a window titled "Gestión de Pedidos - Supermarket". It contains three input fields: "ID Pedido:" with the value "2", "ID Cliente:" with the value "1", and "Fecha (YYYY-MM-DD):" with the value "2000/12/23". Below these fields are four buttons: "Insertar", "Actualizar", "Consultar", and "Eliminar". The "Consultar" button is highlighted. Below the buttons, the following data is displayed:

| | | |
|-----|----|---------------|
| (1, | 1, | '2000-12-23') |
| (3, | 1, | '2000-12-23') |
| (4, | 1, | '2000-12-23') |

NOTA: En esta imagen se muestra los Pedidos y Productos que se encuentran en la base de datos permitiendo visualizar cada uno

Ilustración 4 Eliminación

The screenshot shows the same window as before, but with different input values: "ID Pedido:" is "coca", "ID Cliente:" is "manuel", and "Fecha (YYYY-MM-DD):" is "2000/12/23". The "Eliminar" button is highlighted. Below the buttons, the data area is empty.

NOTA: En la imagen se muestra la eliminación de los datos de la base de datos.

Código 1. Gestión de pedidos en Python con interfaz Tkinter y conexión a SQL Server

```

import sys
import tkinter
from tkinter import messagebox

# Conexión a SQL Server usando Autenticación de Windows
def get_connection():
    server = 'LAPTOP-M6RT9JJF'
    database = 'Supermarket'
    conn_str = f'DRIVER={{SQL Server}};SERVER={server};DATABASE={database};Trusted_Connection=yes;'
    return pyodbc.connect(conn_str)

# Función para insertar un pedido
def insertar():
    ID_Cliente = entry_cliente.get()
    Fecha = entry_fecha.get()
    try:
        with get_connection() as conn:
            cursor = conn.cursor()
            cursor.execute("INSERT INTO Pedidos (ID_Cliente, Fecha) VALUES (?, ?)", (ID_Cliente, Fecha))
            conn.commit()
            output.insert(tk.END, "Pedido insertado correctamente.\n")
    except Exception as e:
        messagebox.showerror("Error", str(e))

# Consultar pedidos
def consultar():
    try:
        with get_connection() as conn:
            cursor = conn.cursor()
            cursor.execute("SELECT * FROM Pedidos")
            rows = cursor.fetchall()
            output.delete(1.0, tk.END)
            for row in rows:
                output.insert(tk.END, f"{row}\n")
    except Exception as e:
        messagebox.showerror("Error", str(e))

# Actualizar pedido
def actualizar():
    ID_Pedido = entry_id.get()
    ID_Cliente = entry_cliente.get()
    Fecha = entry_fecha.get()
    try:
        with get_connection() as conn:
            cursor = conn.cursor()
            cursor.execute("UPDATE Pedidos SET ID_Cliente = ?, Fecha = ? WHERE ID_Pedido = ?", (ID_Cliente, Fecha, ID_Pedido))
            conn.commit()
            output.insert(tk.END, "Pedido actualizado.\n")
    except Exception as e:
        messagebox.showerror("Error", str(e))

```



```

        except Exception as e:
            messagebox.showerror("Error", str(e))

# Eliminar pedido
def eliminar():
    ID_Pedido = entry_id.get()
    try:
        with get_connection() as conn:
            cursor = conn.cursor()
            cursor.execute("DELETE FROM Pedidos WHERE ID_Pedido = ?",
                (ID_Pedido,))
            conn.commit()
            output.insert(tk.END, "Pedido eliminado.\n")
    except Exception as e:
        messagebox.showerror("Error", str(e))

# Interfaz gráfica
root = tk.Tk()
root.title("Gestión de Pedidos - Supermarket")

tk.Label(root, text="ID Pedido:").grid(row=0, column=0)
entry_id = tk.Entry(root)
entry_id.grid(row=0, column=1)

tk.Label(root, text="ID Cliente:").grid(row=1, column=0)
entry_cliente = tk.Entry(root)
entry_cliente.grid(row=1, column=1)

tk.Label(root, text="Fecha (YYYY-MM-DD):").grid(row=2, column=0)
entry_fecha = tk.Entry(root)
entry_fecha.grid(row=2, column=1)

tk.Button(root, text="Insertar", command=insertar).grid(row=3, column=0)
tk.Button(root, text="Consultar", command=consultar).grid(row=3, column=1)
tk.Button(root, text="Actualizar", command=actualizar).grid(row=4, column=0)
tk.Button(root, text="Eliminar", command=eliminar).grid(row=4, column=1)

output = tk.Text(root, height=10, width=50)
output.grid(row=5, column=0, columnspan=2, pady=10)

root.mainloop()

```

Nota. Este código fue desarrollado por el autor para un sistema de gestión de pedidos de supermercado, utilizando Python, Tkinter y SQL Server como base de datos. La interfaz permite insertar, consultar, actualizar y eliminar registros de pedidos.