Script completo con interfaz Tkinter integrada para que el usuario pueda seleccionar:

* Producto
* Estación
* Fecha inicio
* Fecha final

Y luego se genere el PDF con los totales por día y el total acumulado.

SCRIPT: Reporte PDF con Tkinter

**import** mysql.connector

**from** reportlab.lib.pagesizes **import** letter

**from** reportlab.pdfgen **import** canvas

**from** reportlab.lib.units **import** inch

**import** tkinter **as** tk

**from** tkinter **import** ttk, messagebox

**from** tkcalendar **import** DateEntry

**import** **datetime**

*# Configuración de conexión*

DB\_CONFIG = {

'host': 'localhost',

'user': 'Usuario',

'password': 'Clave',

'database': 'gasolinas1',

'charset': 'utf8mb4',

'collation': 'utf8mb4\_general\_ci'

}

*# ===================== FUNCIONES BASE DE DATOS =====================*

def obtener\_productos():

try:

conn = mysql.connector.connect(\*\*DB\_CONFIG)

**cursor** = conn.**cursor**()

**cursor**.**execute**("SELECT nombre FROM productos ORDER BY nombre")

productos = [fila[0] **for** fila **in** **cursor**.fetchall()]

**cursor**.**close**()

conn.**close**()

**return** productos

**except** Exception **as** e:

print(f"Error obteniendo productos: {e}")

**return** []

def obtener\_estaciones():

try:

conn = mysql.connector.connect(\*\*DB\_CONFIG)

**cursor** = conn.**cursor**()

**cursor**.**execute**("SELECT id, nombre FROM estacion ORDER BY nombre")

estaciones = **cursor**.fetchall()

**cursor**.**close**()

conn.**close**()

**return** estaciones

**except** Exception **as** e:

print(f"Error obteniendo estaciones: {e}")

**return** []

*# ===================== REPORTE PDF =====================*

def generar\_reporte\_pdf(producto, estacion\_id, estacion\_nombre, fecha\_inicio, fecha\_fin):

try:

conn = mysql.connector.connect(\*\*DB\_CONFIG)

**cursor** = conn.**cursor**()

**cursor**.**execute**("""

**SELECT** i.fecha, i.cantidad, i.precio\_venta, i.venta\_soles

**FROM** inventario i

**JOIN** productos p **ON** i.productoid = p.id

**WHERE** p.nombre = %s **AND** i.estacionid = %s **AND** i.fecha **BETWEEN** %s **AND** %s

**ORDER** **BY** i.fecha

""", (producto, estacion\_id, fecha\_inicio, fecha\_fin))

datos = **cursor**.fetchall()

**cursor**.**close**()

conn.**close**()

**if** **not** datos:

messagebox.showwarning("Sin datos", "No se encontraron registros para ese rango.")

**return**

*# Crear PDF*

nombre\_pdf = f"reporte\_{producto}\_{estacion\_nombre.replace(' ', '\_')}.pdf"

c = canvas.Canvas(nombre\_pdf, pagesize=letter)

width, height = letter

c.setFont("Helvetica-Bold", 16)

c.drawString(1\*inch, height - 1\*inch, f"Estación: {estacion\_nombre}")

c.setFont("Helvetica", 12)

c.drawString(1\*inch, height - 1.4\*inch, f"Producto: {producto}")

c.drawString(1\*inch, height - 1.7\*inch, f"Desde: {fecha\_inicio} Hasta: {fecha\_fin}")

c.setFont("Helvetica-Bold", 11)

c.drawString(1\*inch, height - 2.1\*inch, "Fecha")

c.drawString(2.5\*inch, height - 2.1\*inch, "Cantidad (gal)")

c.drawString(4.2\*inch, height - 2.1\*inch, "Precio Venta")

c.drawString(5.8\*inch, height - 2.1\*inch, "Venta S/")

**y** = height - 2.4\*inch

total\_venta = 0

c.setFont("Helvetica", 10)

**for** fecha, cantidad, precio, venta **in** datos:

**if** **y** < 1\*inch:

c.showPage()

**y** = height - 1\*inch

c.setFont("Helvetica-Bold", 16)

c.drawString(1\*inch, height - 1\*inch, f"Estación: {estacion\_nombre}")

c.setFont("Helvetica-Bold", 11)

c.drawString(1\*inch, height - 1.5\*inch, "Fecha")

c.drawString(2.5\*inch, height - 1.5\*inch, "Cantidad (gal)")

c.drawString(4.2\*inch, height - 1.5\*inch, "Precio Venta")

c.drawString(5.8\*inch, height - 1.5\*inch, "Venta S/")

**y** = height - 1.8\*inch

c.setFont("Helvetica", 10)

c.drawString(1\*inch, **y**, fecha.strftime("%Y-%m-%d"))

c.drawRightString(3.9\*inch, **y**, f"{cantidad:.2f}")

c.drawRightString(5.4\*inch, **y**, f"{precio:.2f}")

c.drawRightString(7.4\*inch, **y**, f"{venta:.2f}")

total\_venta += venta

**y** -= 0.25\*inch

c.setFont("Helvetica-Bold", 11)

**y** -= 0.2\*inch

c.drawString(1\*inch, **y**, "Total Venta S/:")

c.drawRightString(7.4\*inch, **y**, f"{total\_venta:.2f}")

c.save()

messagebox.showinfo("Éxito", f"PDF generado: {nombre\_pdf}")

**except** Exception **as** e:

messagebox.showerror("Error", f"No se pudo generar el PDF: {e}")

*# ===================== INTERFAZ =====================*

def crear\_interfaz():

root = tk.Tk()

root.title("Generar Reporte PDF")

root.**geometry**("450x300")

*# === Productos ===*

tk.Label(root, **text**="Producto:").pack(pady=5)

producto\_cb = ttk.Combobox(root, **values**=obtener\_productos(), state="readonly")

producto\_cb.pack()

*# === Estaciones ===*

tk.Label(root, **text**="Estación:").pack(pady=5)

estaciones = obtener\_estaciones()

estacion\_dict = {nombre: id\_ **for** id\_, nombre **in** estaciones}

estacion\_cb = ttk.Combobox(root, **values**=**list**(estacion\_dict.**keys**()), state="readonly")

estacion\_cb.pack()

*# === Fechas ===*

tk.Label(root, **text**="Fecha Inicio:").pack(pady=5)

fecha\_inicio = DateEntry(root, width=15, date\_pattern='yyyy-mm-dd')

fecha\_inicio.set\_date(**datetime**.**date**.today())

fecha\_inicio.pack()

tk.Label(root, **text**="Fecha Fin:").pack(pady=5)

fecha\_fin = DateEntry(root, width=15, date\_pattern='yyyy-mm-dd')

fecha\_fin.set\_date(**datetime**.**date**.today())

fecha\_fin.pack()

*# === Botón generar ===*

def generar():

producto = producto\_cb.**get**()

estacion\_nombre = estacion\_cb.**get**()

estacion\_id = estacion\_dict.**get**(estacion\_nombre)

f\_ini = fecha\_inicio.get\_date().strftime("%Y-%m-%d")

f\_fin = fecha\_fin.get\_date().strftime("%Y-%m-%d")

**if** **not** producto **or** **not** estacion\_id:

messagebox.showwarning("Campos vacíos", "Selecciona todos los campos.")

**return**

generar\_reporte\_pdf(producto, estacion\_id, estacion\_nombre, f\_ini, f\_fin)

tk.Button(root, **text**="Generar PDF", command=generar, bg="#4CAF50", fg="white").pack(pady=15)

root.mainloop()

*# === Ejecutar*

**if** \_\_name\_\_ == "\_\_main\_\_":

crear\_interfaz()

Asegúrate de tener estas librerías instaladas:

* Ejecuta en un CMD

pip install mysql-connector-python reportlab tk tkcalendar

PASOS FINALES PARA EJECUTAR EL SCRIPT

* Abre una terminal (CMD o PowerShell) en la carpeta donde guardaste el script .py
* Ejecuta el script con el siguiente comando

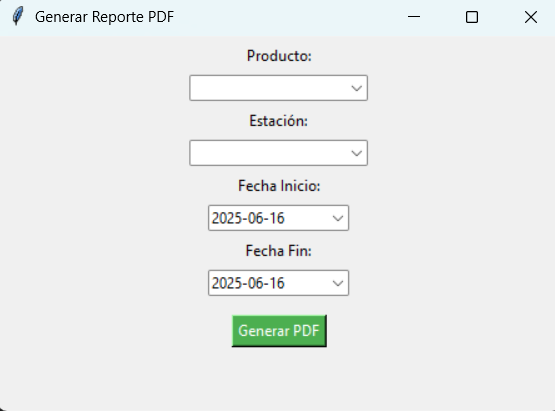
python nombre\_del\_script.py

* Por ejemplo, si lo guardaste como reporte\_pdfl\_Tkinter.script.py, ejecuta:

python reporte\_pdfl\_Tkinter.script.py

DEBERÍAS VER:

* Se abrirá una ventana con esta interfaz:
* Un **menú desplegable** para seleccionar el **producto**
* Otro para seleccionar la **estación (empresa)**
* Dos calendarios para escoger **fecha de inicio y fin**.
* Un botón **“Generar PDF”**.



¿QUÉ SUCEDE AL PRESIONAR “Generar PDF”?

* El sistema consulta la base de datos con los filtros.
* Si hay datos, genera un archivo PDF llamado: reporte\_<producto>\_<nombre\_estacion>.pdf
* El archivo quedará guardado en la misma carpeta donde ejecutaste el script.
* Si no hay datos, te mostrará un mensaje.

VERIFICA:

* Que los datos existen en la BD para las fechas/producto seleccionados.
* Que la conexión a MySQL esté funcionando.

