

Universidad Tecnológica de Panamá
Facultad de Ingeniería en Sistemas Computacionales
Asignatura: Desarrollo Lógico Y Algoritmo
Ejercicio Práctico2

Profesor: Napoleón Ibarra

Estudiante: Ramiro Caballero

Quetzaliris Espinosa 1-762-410

Fecha Inicio: 13/10/2025 --> 4:10 PM

Fecha Entrega: 14/10/2025 -->2:25 PM

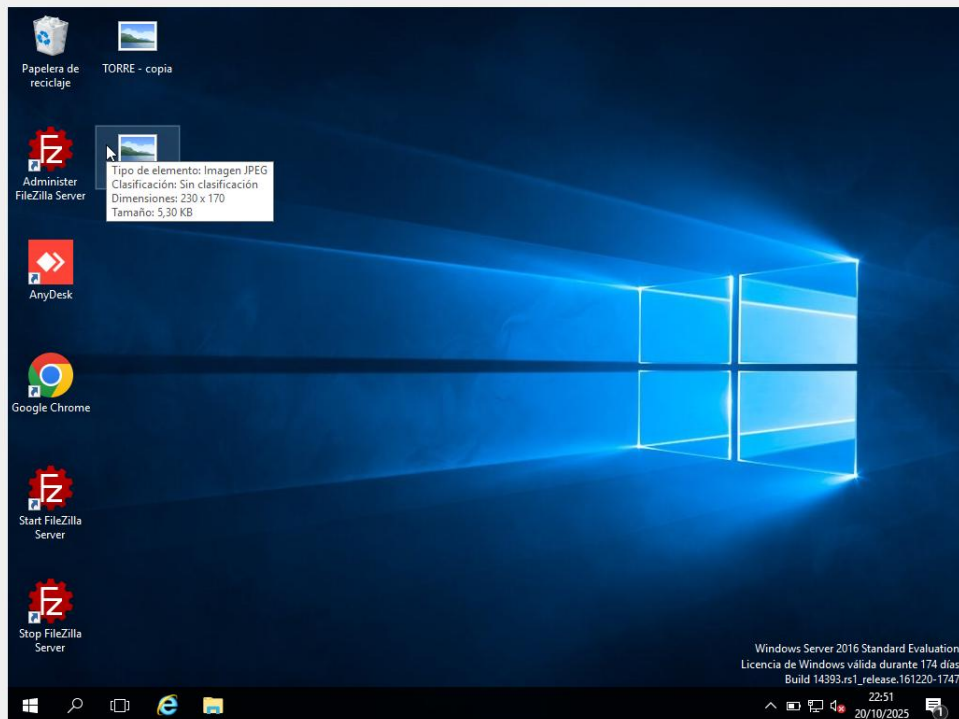
Fecha Cierre (Tope): 20/10/2025 -->4:10 PM

Procedimiento:

- De manera INDIVIDUAL o en grupos de 2 PERSONAS, realizar la asignación.
- Investigue, descargar e instalar en una máquina virtual (Usted elige el IDE) una distribución de Windows Server (2016/2019).
- Se debe entregar al profesor: Documento digital: entrega en la plataforma (TEAM) el y/o los códigos desarrollando los problemas. Sustente su trabajo en el aula de clases.

I Parte. Laboratorio. Valor 20 puntos.

1. Elegir un software para virtualización (Virtual Box / VMware), descargar, instalar y configurarlo.
2. Descargar, instalar la distribución de Windows Server (2016/2019) elegida dentro de la máquina virtual.
3. Realizar el proceso de configuración correspondiente al realizar el proceso de la instalación del SO.
4. Configurar una carpeta para hacer respaldos de códigos dentro del SO, verifique su instalación, configuración.
5. Realizar la configuración del Servidor de Datos (que se accesible por Remote Desktop Connection, ANYDESK, FTP).
6. Hacer pruebas de funcionamiento.



II Parte. Caso de Estudio. *Valor 20 Puntos*

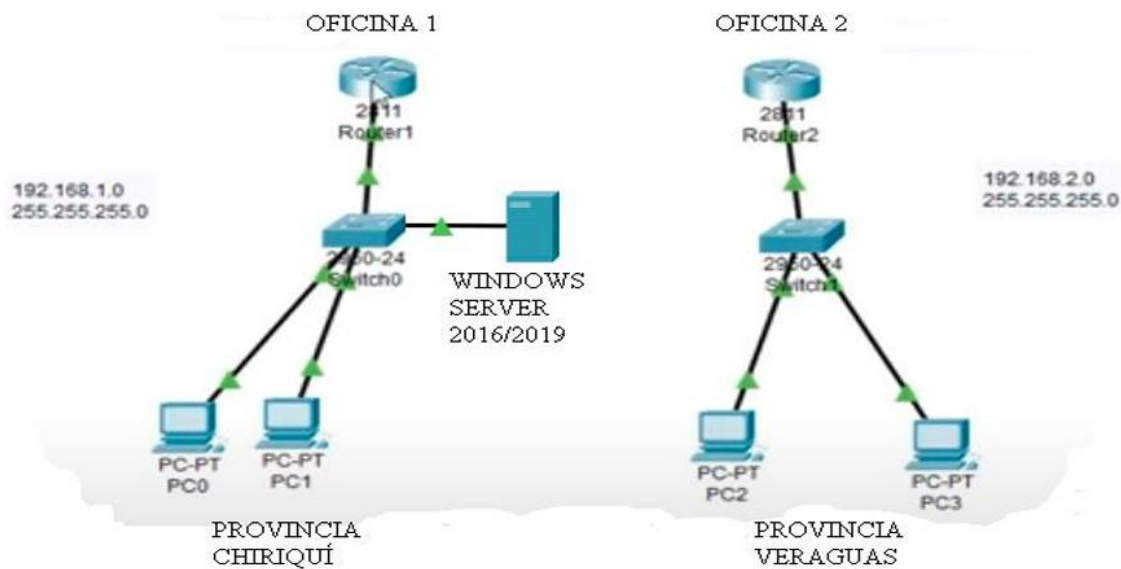
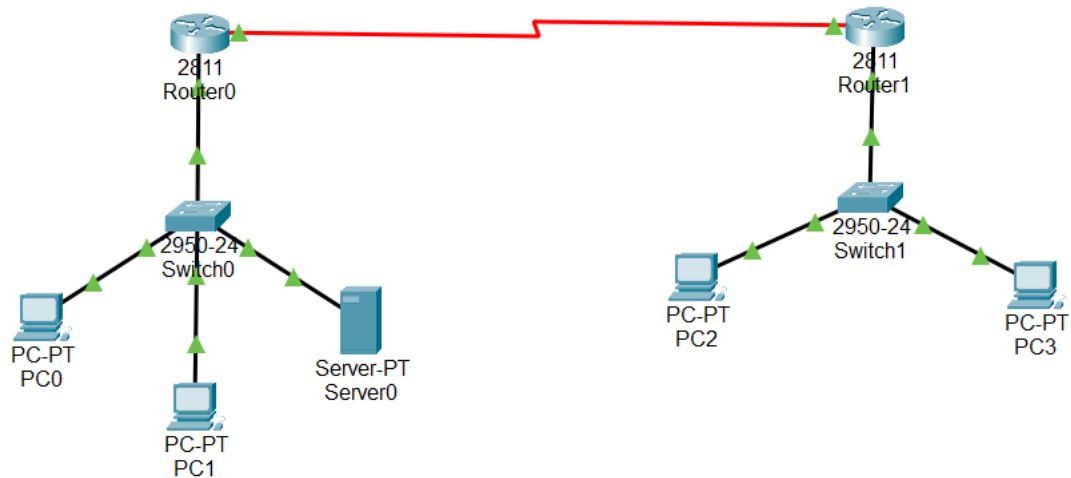


Figura 1. Prototipo de diagrama

Procedimiento:

1. Teniendo en cuenta el siguiente plano de oficina (Figura 1), confeccione el esquema de RED LAN, puesto que en ambas oficinas 1 y 2 se va a utilizar el servidor de datos.



Command Prompt

```

Cisco Packet Tracer SERVER Command Line 1.0
C:\>ping 198.168.20.3

Pinging 198.168.20.3 with 32 bytes of data:

Reply from 192.168.10.1: Destination host unreachable.
Request timed out.
Reply from 192.168.10.1: Destination host unreachable.
Reply from 192.168.10.1: Destination host unreachable.

Ping statistics for 198.168.20.3:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
C:\>
  
```

Command Prompt

```

Cisco Packet Tracer PC Command Line 1.0
C:\>ping 192.168.20.3

Pinging 192.168.20.3 with 32 bytes of data:

Request timed out.
Reply from 192.168.20.3: bytes=32 time=1ms TTL=126
Reply from 192.168.20.3: bytes=32 time=1ms TTL=126
Reply from 192.168.20.3: bytes=32 time=1ms TTL=126

Ping statistics for 192.168.20.3:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 1ms, Average = 1ms
C:\>
  
```

III Parte. Pseudocódigos, diagrama de flujo. Valor 10 Puntos

1. Desarrolle los problemas en pseudocódigo para la parte IV de la asignación.

1. Pseudocódigo.

Inicio

Definir hora_str, horas_str, minutos_str, sufijo Como Cadena

Definir horas, minutos, hora_12 Como

Entero

Escribir 'Ingrese la hora en formato 24 horas por ejemplo 13 45'

Leer hora_str

Si Longitud(hora_str) <> 5 Entonces

Escribir 'Error debe ingresar exactamente 5 caracteres en formato HHMM'

SiNo

Si Subcadena(hora_str,3,3) <> ':' Entonces

Escribir 'Error formato incorrecto debe usar el simbolo de dos puntos entre horas y minutos'

SiNo

horas_str <- Subcadena(hora_str,1,2)

minutos_str <- Subcadena(hora_str,4,5)

horas <- ConvertirANumero(horas_str)

minutos <- ConvertirANumero(minutos_str)

**Si horas<0 O horas>23 O minutos<0 O minutos>59 Entonces
Escribir 'Error hora fuera de rango permitido'**

SiNo

Si horas=0 Entonces

hora_12 <- 12

sufijo <- 'AM'

SiNo

Si horas=12 Entonces

hora_12 <- 12

sufijo <- 'PM'

SiNo

Si horas>12 Entonces

hora_12 <- horas-12

sufijo <- 'PM'

SiNo

hora_12 <- horas

sufijo <- 'AM'

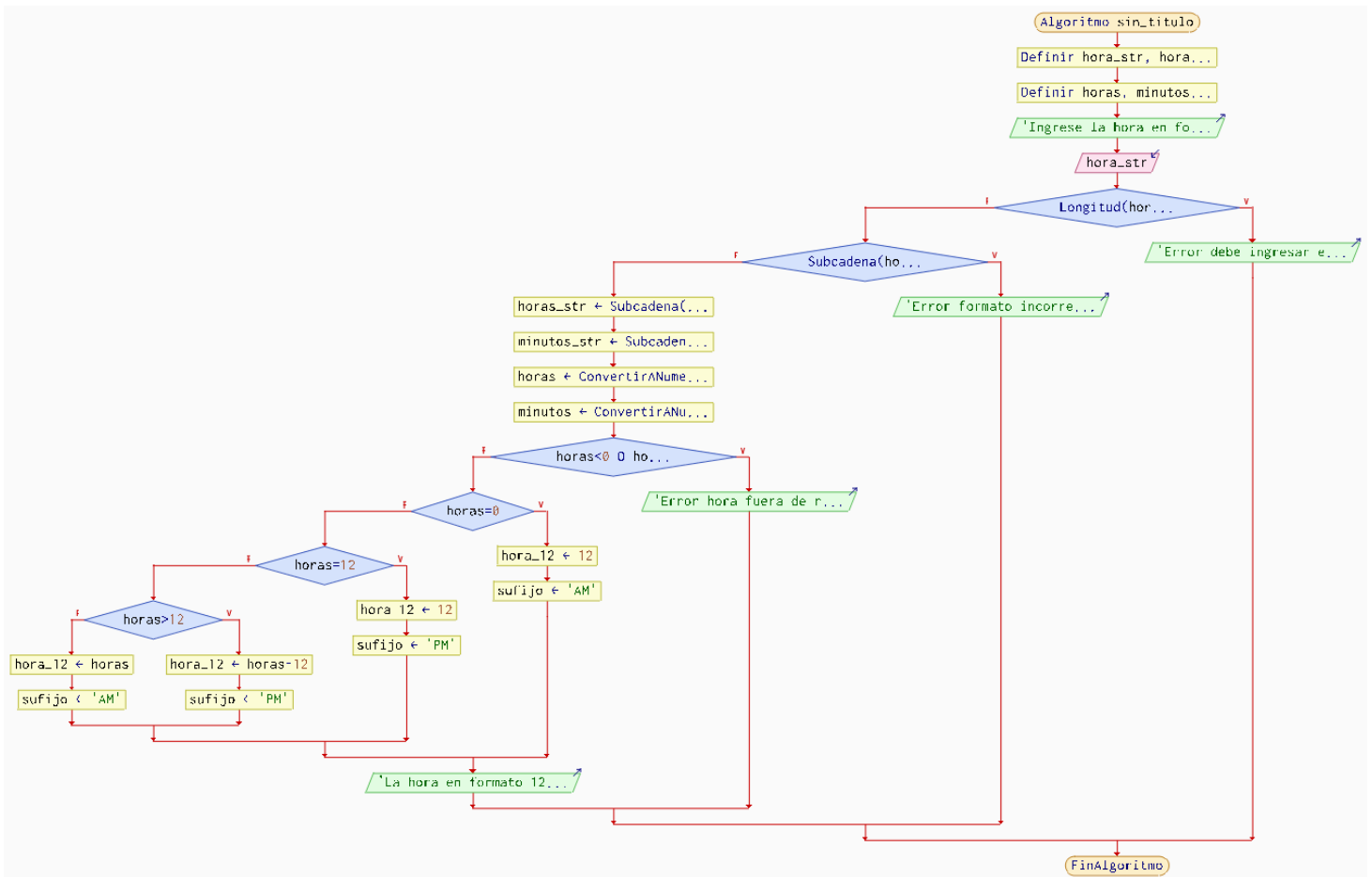
FinSi

Escribir 'La hora en formato 12h es ', hora_12, ' ', minutos, ' ', sufijo

FinSi

Fin

Diagrama de flujo.



2. Pseudocódigo

Inicio

Definir C1, C2, C3 Como Entero

Definir P1, P2, P3, V1, V2, V3, TOTAL Como Real

Escribir 'Ingrese la cantidad vendida del producto Cola (máximo 5000000):'

Leer C1

Escribir 'Ingrese el precio unitario de Cola (en B/.):'

Leer P1

Escribir 'Ingrese la cantidad vendida del producto Naranja (máximo 5000000):'

Leer C2

Escribir 'Ingrese el precio unitario de Naranja (en B/.):'

Leer P2

Escribir 'Ingrese la cantidad vendida del producto Limón (máximo 5000000):'

Leer C3

Escribir 'Ingrese el precio unitario de Limón (en B/.):'

Leer P3

// Validar cantidades

Si (C1<0 O C1>5000000) O (C2<0 O C2>5000000) O (C3<0 O C3>5000000) Entonces

Escribir 'Error: una o más cantidades están fuera del rango permitido'

SiNo

// Calcular totales

V1 <- C1*P1

V2 <- C2*P2

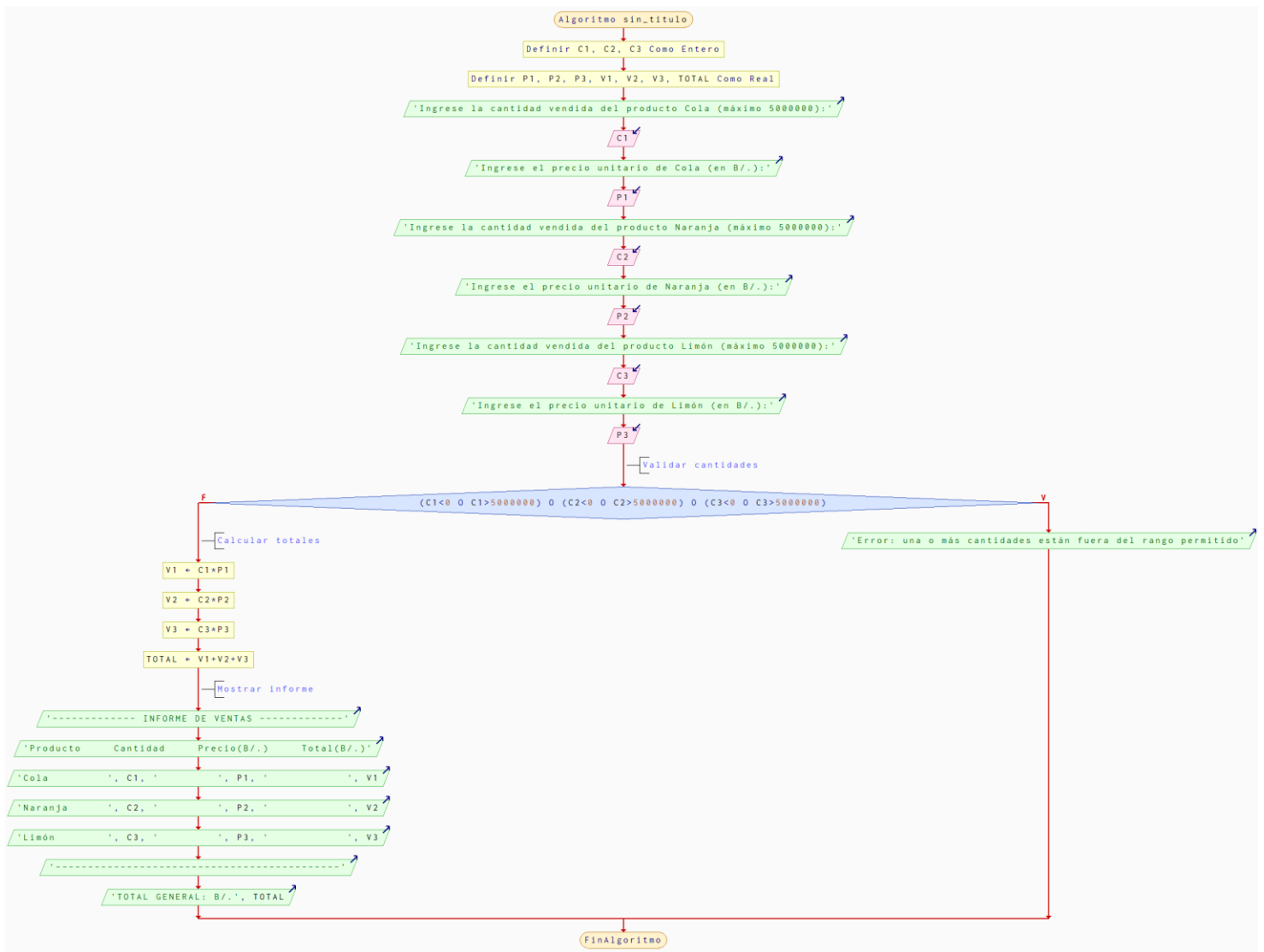
V3 <- C3*P3

```

TOTAL <- V1+V2+V3
// Mostrar informe
Escribir '----- INFORME DE VENTAS -----'
Escribir 'Producto  Cantidad  Precio(B/)  Total(B/.)'
Escribir 'Cola      ', C1, '      ', P1, '      ', V1
Escribir 'Naranja   ', C2, '      ', P2, '      ', V2
Escribir 'Limón     ', C3, '      ', P3, '      ', V3
Escribir '-----'
Escribir 'TOTAL GENERAL: B/.', TOTAL
FinSi
Fin

```

Diagrama de Flujo



IV Parte. Desarrollo de problemas en Python. Valor 20 Puntos

1. Escriba un programa en Python que lea la hora en notación de 24 horas y que imprima en notación de 12; por ejemplo, si la entrada es 13:45, la salida será 1:45 pm. El programa debe solicitar al usuario que introduzca exactamente cinco caracteres para especificar una hora. Valor 10 puntos
import tkinter as tk
from tkinter import messagebox

```
def convertir_hora():
```

```
    hora_24 = entrada_hora.get().strip()
```

```
    # Verificar que tenga 5 caracteres (formato hh:mm)
```

```
    if len(hora_24) != 5 or hora_24[2] != ':':
```

```
        messagebox.showerror("Error", "La hora debe tener el formato hh:mm (5 caracteres).")
```

```
    return
```

```

try:
    horas = int(hora_24[:2])
    minutos = int(hora_24[3:])

    if not (0 <= horas < 24 and 0 <= minutos < 60):
        raise ValueError

    # Convertir a formato 12 horas
    sufijo = "am"
    if horas >= 12:
        sufijo = "pm"
    if horas == 0:
        horas = 12
    elif horas > 12:
        horas -= 12

    # Formatear minutos con dos cifras
    hora_12 = f"{horas}:{minutos:02d}{sufijo}"
    resultado_label.config(text=f"Hora en formato 12: {hora_12}")

except ValueError:
    messagebox.showerror("Error", "Entrada no válida. Usa el formato hh:mm con valores válidos.")

# Crear ventana principal
ventana = tk.Tk()
ventana.title("Conversor de hora 24h a 12h")
ventana.geometry("350x200")

# Etiquetas y campo de entrada
tk.Label(ventana, text="Introduce la hora (hh:mm):").pack(pady=10)
entrada_hora = tk.Entry(ventana, width=10, justify="center")
entrada_hora.pack()

# Botón para convertir
tk.Button(ventana, text="Convertir", command=convertir_hora).pack(pady=10)

# Resultado
resultado_label = tk.Label(ventana, text="", font=("Arial", 12))
resultado_label.pack(pady=10)

ventana.mainloop()

```


2. Una compañía de refrescos comercializa tres productos: de cola, de naranja y de limón. Se dese realizar una aplicación que calcule las ventas realizadas de cada producto. Para ello, se leerá la cantidad vendida (máximo 5000000) y el precio en balboas de cada producto y se mostrará un informe de ventas como el que sigue:

Producto	Ventas	Precio	Total
Cola	1000000	0.17	170000.00
Naranja	350000	0.20	70000.00
Limon	530000	0.19	100700.00
TOTAL			340700.00

NOTA: Para el desarrollo de la parte IV en Python, utilizar la interfaz Tkinter, tome en cuenta después del punto (decimal) 2 cifras significativas.

```
import tkinter as tk
```

```
from tkinter import messagebox
```

```
def calcular_totales():
```

```
    try:
```

```
        # Leer entradas
```

```
        ventasCola = int(entradaColaVentas.get())
```

```
        precioCola = float(entradaColaPrecio.get())
```

```
        ventasNaranja = int(entradaNaranjaVentas.get())
```

```
        precioNaranja = float(entradaNaranjaPrecio.get())
```

```
        ventasLimon = int(entradaLimonVentas.get())
```

```
        precioLimon = float(entradaLimonPrecio.get())
```

```
    # Validar límites
```

```
    if any(v > 5000000 for v in [ventasCola, ventasNaranja, ventasLimon]):
```

```
        messagebox.showerror("Error", "La cantidad vendida no puede superar 5,000,000 unidades.")
```

```
    return
```

```

# Calcular totales

total_cola = ventas_cola * precio_cola

total_naranja = ventas_naranja * precio_naranja

total_limon = ventas_limon * precio_limon

total_general = total_cola + total_naranja + total_limon


# Mostrar resultados con 2 decimales

resultado_text.delete("1.0", tk.END)

resultado_text.insert(tk.END, "Producto\tVentas\tPrecio\tTotal\n")

resultado_text.insert(tk.END, "-----\n")

resultado_text.insert(tk.END, f"Cola\t{ventas_cola}\t{precio_cola:.2f}\t{total_cola:.2f}\n")

resultado_text.insert(tk.END,
f"Naranja\t{ventas_naranja}\t{precio_naranja:.2f}\t{total_naranja:.2f}\n")

resultado_text.insert(tk.END, f"Limon\t{ventas_limon}\t{precio_limon:.2f}\t{total_limon:.2f}\n")

resultado_text.insert(tk.END, "-----\n")

resultado_text.insert(tk.END, f"TOTAL\t\t\t{total_general:.2f}\n")


except ValueError:

    messagebox.showerror("Error", "Por favor, ingrese valores numéricos válidos.")


# Ventana principal

ventana = tk.Tk()

ventana.title("Informe de Ventas de Refrescos")

ventana.geometry("480x420")


# Sección de Cola

tk.Label(ventana, text="Cola").grid(row=0, column=0, padx=10, pady=5)

tk.Label(ventana, text="Ventas:").grid(row=0, column=1)

```

```
entrada_cola_ventas = tk.Entry(ventana, width=10)
entrada_cola_ventas.grid(row=0, column=2)
tk.Label(ventana, text="Precio:").grid(row=0, column=3)
entrada_cola_precio = tk.Entry(ventana, width=10)
entrada_cola_precio.grid(row=0, column=4)
```

Sección de Naranja

```
tk.Label(ventana, text="Naranja").grid(row=1, column=0, padx=10, pady=5)
tk.Label(ventana, text="Ventas:").grid(row=1, column=1)
entrada_naranja_ventas = tk.Entry(ventana, width=10)
entrada_naranja_ventas.grid(row=1, column=2)
tk.Label(ventana, text="Precio:").grid(row=1, column=3)
entrada_naranja_precio = tk.Entry(ventana, width=10)
entrada_naranja_precio.grid(row=1, column=4)
```

Sección de Limón

```
tk.Label(ventana, text="Limón").grid(row=2, column=0, padx=10, pady=5)
tk.Label(ventana, text="Ventas:").grid(row=2, column=1)
entrada_limon_ventas = tk.Entry(ventana, width=10)
entrada_limon_ventas.grid(row=2, column=2)
tk.Label(ventana, text="Precio:").grid(row=2, column=3)
entrada_limon_precio = tk.Entry(ventana, width=10)
entrada_limon_precio.grid(row=2, column=4)
```

Botón para calcular

```
tk.Button(ventana, text="Calcular Totales", command=calcular_totales).grid(row=3, column=0,
columnspan=5, pady=10)
```

Cuadro de resultados

```
resultado_text = tk.Text(ventana, height=10, width=55)
```

```
resultado_text.grid(row=4, column=0, columnspan=5, padx=10, pady=10)
```

```
ventana.mainloop()
```