



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



Profesor: Napoleón Ibarra

Valor: 100 puntos

Estudiante:

Cédula:

**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.

**Criterios de Evaluación:**

<b>Criterios</b>	<b>Puntos (Mínimo=1, Máximo=5)</b>	<b>Porcentaje</b>
<i>Desarrollo</i>	1-5	70 %
<i>Sustentación</i>	1-5	15 %
<i>Puntualidad</i>	1-5	15 %

**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*

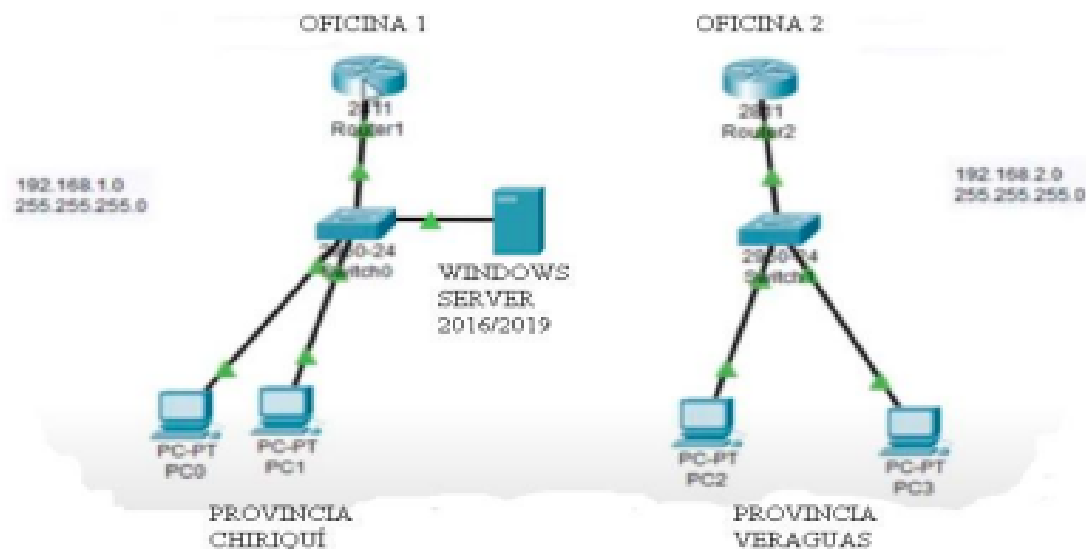
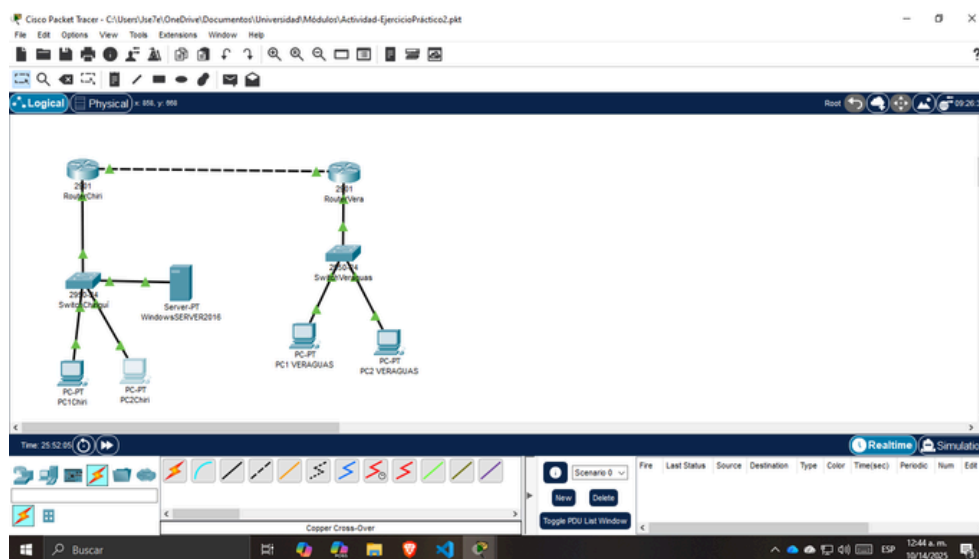


Figura 1. Prototipo de diagrama

### 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.



### Pruebas de escritorio en la sustentación

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

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

Algoritmo ConvertirHora24a12

// Definir variables

Definir hora\_24 Como Cadena

Definir horas, minutos, hora\_12 Como Entero

Definir sufijo Como Cadena

Definir valido Como Logico

valido ← Falso

Repetir

    Escribir "Introduce la hora en formato 24h (HH:MM):"

    Leer hora\_24

    // Validar longitud y ":"

    Si longitud(hora\_24) = 5 Y SubCadena(hora\_24,3,1) = ":" Entonces

        // Convertir horas y minutos

        horas ← ConvertirANumero(SubCadena(hora\_24,1,2))

        minutos ← ConvertirANumero(SubCadena(hora\_24,4,2))

    // Validar rango

    Si horas >= 0 Y horas <= 23 Y minutos >= 0 Y minutos <= 59 Entonces

        valido ← Verdadero

    Sino

        Escribir "Error: hora o minutos fuera de rango"

    FinSi

    Sino

        Escribir "Error: Debes ingresar 5 caracteres en formato HH:MM"

    FinSi

Hasta Que valido = Verdadero

// Conversión a formato 12h

Si horas = 0 Entonces

    hora\_12 ← 12

    sufijo ← "am"

Sino

    Si horas < 12 Entonces

        hora\_12 ← horas

        sufijo ← "am"

    Sino

        Si horas = 12 Entonces

            hora\_12 ← 12

            sufijo ← "pm"

        Sino

            hora\_12 ← horas - 12

            sufijo ← "pm"

        FinSi

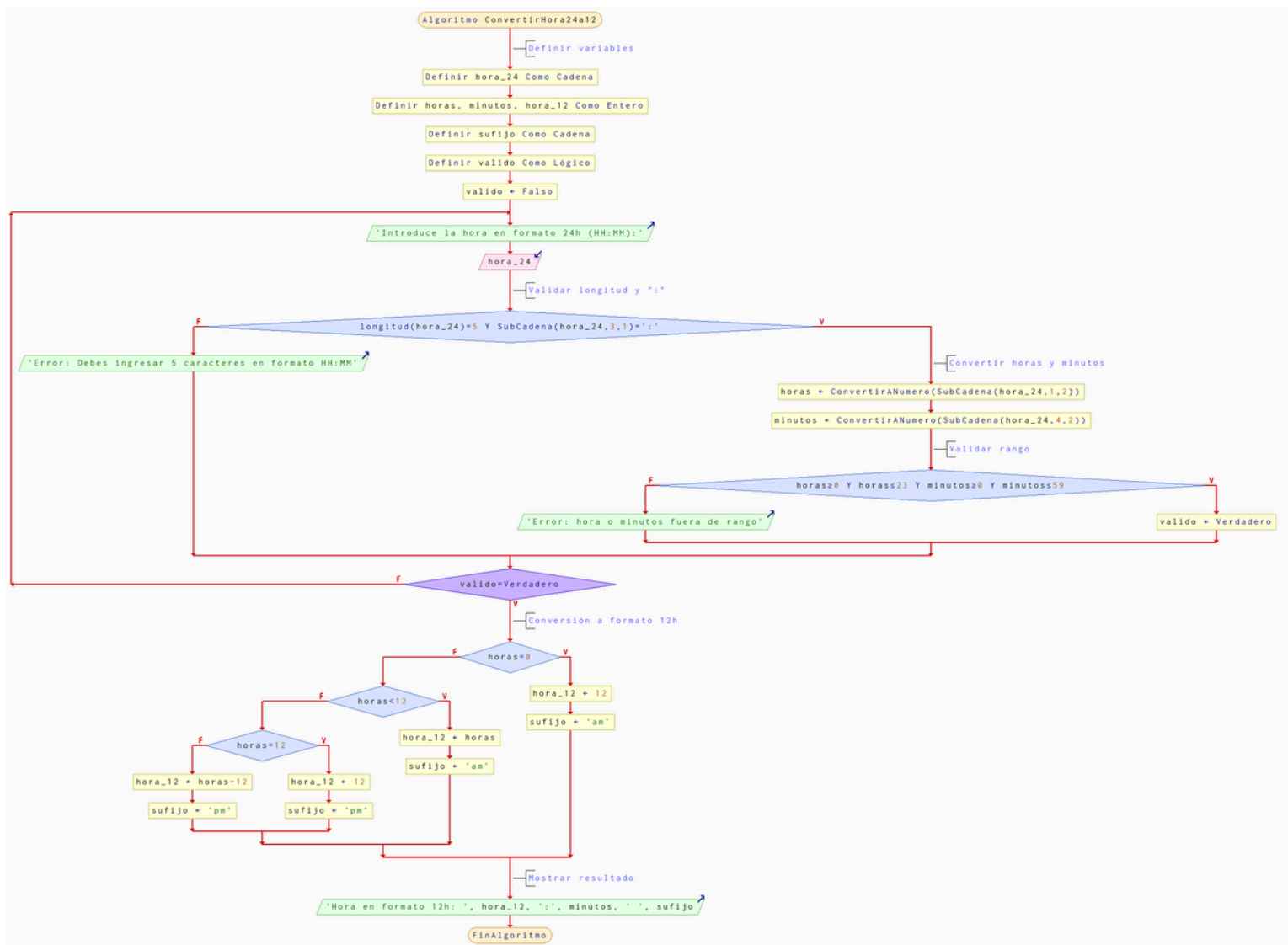
    FinSi

FinSi

// Mostrar resultado

Escribir "Hora en formato 12h: ", hora\_12, ":", minutos, " ", sufijo

FinAlgoritmo



## Algoritmo VentasRefrescos

Definir cantidadCola, cantidadNaranja, cantidadLimon Como Entero

Definir precioCola, precioNaranja, precioLimon Como Real

Definir ventasCola, ventasNaranja, ventasLimon, ventasTotales Como Real

Escribir "Ingrese la cantidad de Cola"

Leer cantidadCola

Escribir "Ingrese el precio de Cola en B/."

Leer precioCola

Escribir "Ingrese la cantidad de Naranja"

Leer cantidadNaranja

Escribir "Ingrese el precio de Naranja en B/."

Leer precioNaranja

Escribir "Ingrese la cantidad de Limón"

Leer cantidadLimon

Escribir "Ingrese el precio de Limón en B/."

Leer precioLimon

$ventasCola \leftarrow cantidadCola * precioCola$

$ventasNaranja \leftarrow cantidadNaranja * precioNaranja$

$ventasLimon \leftarrow cantidadLimon * precioLimon$

$ventasTotales \leftarrow ventasCola + ventasNaranja + ventasLimon$

// Salida de resultados

Escribir "Producto Ventas Precio Total"

Escribir "-----"

Escribir "Cola ", cantidadCola, " ", precioCola, " ", ventasCola

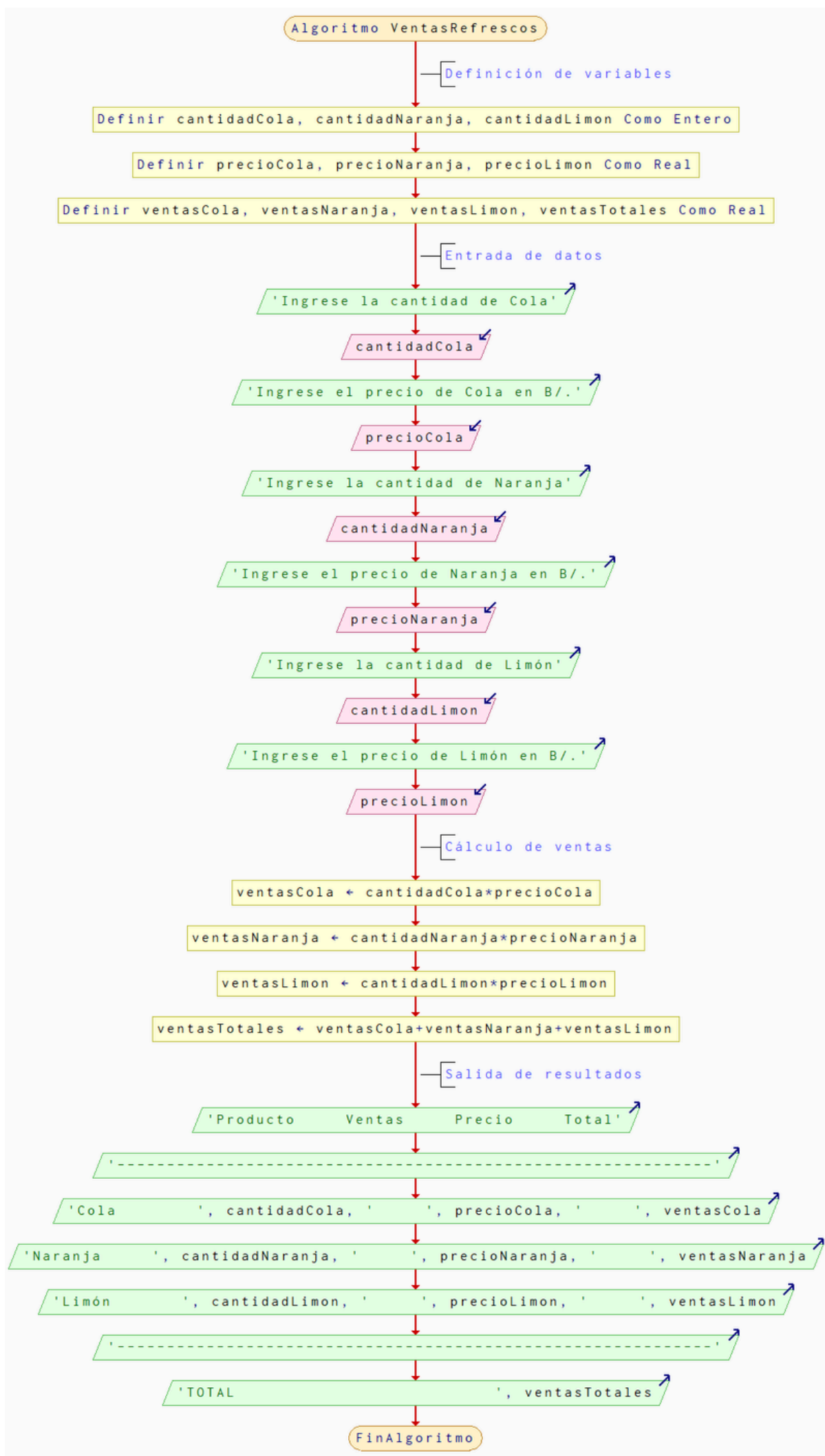
Escribir "Naranja ", cantidadNaranja, " ", precioNaranja, " ", ventasNaranja

Escribir "Limón ", cantidadLimon, " ", precioLimon, " ", ventasLimon

Escribir "-----"

Escribir "TOTAL ", ventasTotales

FinAlgoritmo



#### 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

# Función para convertir hora

def convertir_hora():
    hora_24 = entrada.get()

    # Validamos longitud y formato

    if len(hora_24) != 5 or hora_24[2] != ":":
        messagebox.showerror("Error", "Debes ingresar 5 caracteres en formato HH:MM")
        return

    # Aquí separaremos horas y minutos

    try:
        horas = int(hora_24[:2])
        minutos = int(hora_24[3:])
    except ValueError:
        messagebox.showerror("Error", "Horas y minutos deben ser números")
        return

    # Validamos rango

    if not (0 <= horas <= 23) or not (0 <= minutos <= 59):
        messagebox.showerror("Error", "Hora o minutos fuera de rango")
        return

    # Conversión a 12h

    if horas == 0:
        hora_12 = 12
        sufijo = "am"
    elif horas < 12:
        hora_12 = horas
        sufijo = "am"
    elif horas == 12:
        hora_12 = 12
        sufijo = "pm"
    else:
        hora_12 = horas - 12
        sufijo = "pm"

    # Mostramos resultado en etiqueta

    resultado.set(f"{hora_12}:{minutos:02d} {sufijo}")

# Ventana principal

root = tk.Tk()
root.title("Conversor de formato de horas (De 24 a 12)")
root.geometry("400x250")
root.configure(bg="#692626")

# Etiqueta de la instrucción

tk.Label(root, text="Introduce la hora (HH:MM):", font=("Comic Sans MS", 14, "bold")).pack(pady=10)

# Entrada de teclado

entrada = tk.Entry(root, font=("Comic Sans MS", 12))
entrada.pack(pady=5)

# Botón para ingresar la entrada

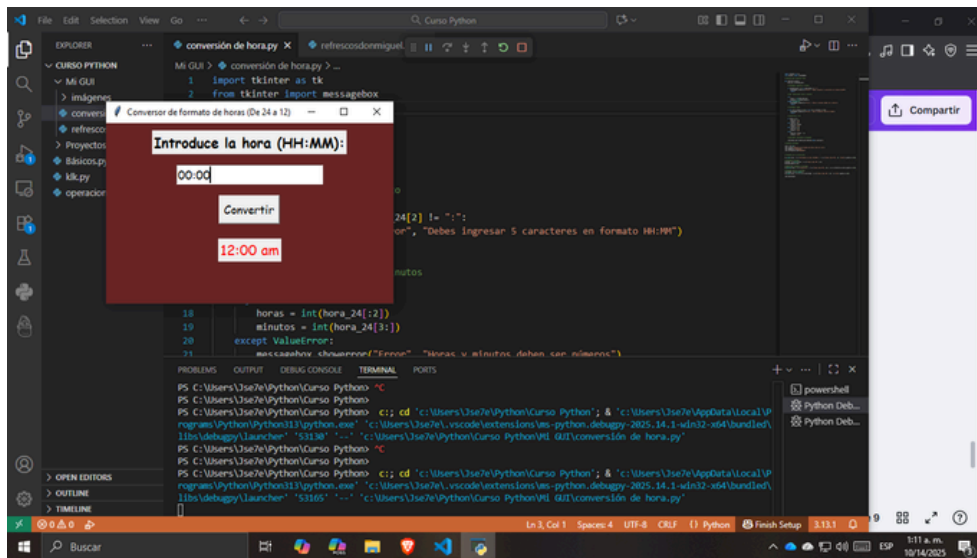
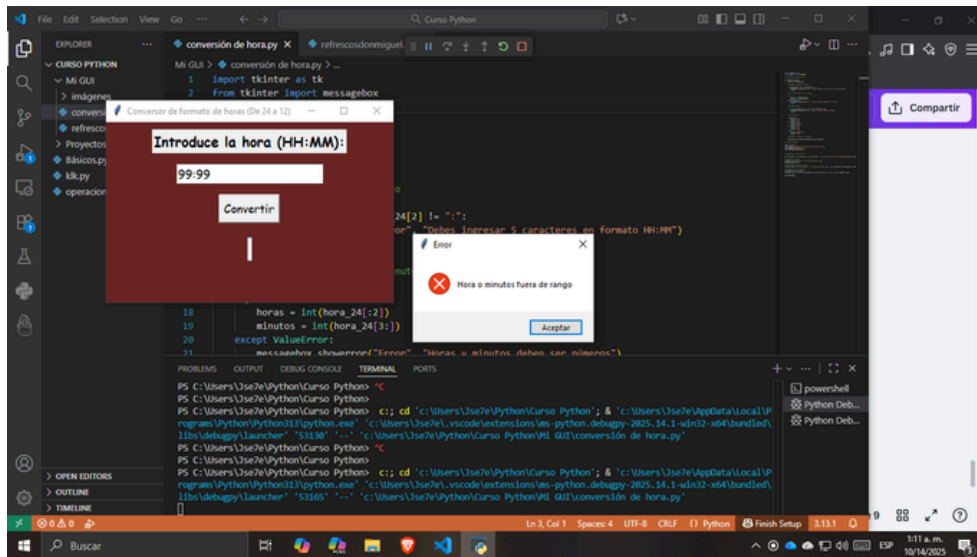
tk.Button(root, text="Convertir", font=("Comic Sans MS", 12), command=convertir_hora).pack(pady=10)

# Etiqueta para el resultado

resultado = tk.StringVar()
tk.Label(root, textvariable=resultado, font=("Comic Sans MS", 14), fg="red").pack(pady=10)

root.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

```
import tkinter as tk
from tkinter import messagebox

# Función para calcular ventas
def calcular_ventas():
    try:
        # Leer cantidades y precios
        cant_cola = int(entrada_cola.get())
        precio_cola = float(precio_cola_entry.get())

        cant_naranja = int(entrada_naranja.get())
        precio_naranja = float(precio_naranja_entry.get())

        cant_limon = int(entrada_limon.get())
        precio_limon = float(precio_limon_entry.get())

        # Comprobando validez de las entradas
        if cant_cola < 0 or cant_cola > 5000000 or cant_naranja < 0 or cant_naranja > 5000000 or cant_limon < 0 or cant_limon > 5000000:
            messagebox.showerror("Error", "Las cantidades deben estar entre 0 y 5,000,000")
            return

        # Validación de precios
        if precio_cola < 0 or precio_naranja < 0 or precio_limon < 0:
            messagebox.showerror("Error", "Los precios deben ser positivos")
            return

        # Calcular totales
        total_cola = cant_cola * precio_cola
        total_naranja = cant_naranja * precio_naranja
        total_limon = cant_limon * precio_limon
        total_general = total_cola + total_naranja + total_limon

        # Mostrar resultados en la etiqueta
        resultado.set(f"""
Informe de Ventas:
Cola: B/. {total_cola:,.2f}
Naranja: B/. {total_naranja:,.2f}
Limón: B/. {total_limon:,.2f}
-----
Total General: B/. {total_general:,.2f}
""")

    except ValueError:
        messagebox.showerror("Error", "Debes ingresar números válidos")

# Ventana principal
root = tk.Tk()
root.title("Calculadora de Ventas de Refrescos")
root.geometry("450x500")
root.configure(bg="#60B2FF")

# Etiquetas y entradas para Cola
tk.Label(root, text="Cola", font=("Arial", 12, "bold"), bg="#60B2FF").pack(pady=5)
entrada_cola = tk.Entry(root, font=("Arial", 12))
entrada_cola.pack(pady=2)
precio_cola_entry = tk.Entry(root, font=("Arial", 12))
precio_cola_entry.pack(pady=2)
precio_cola_entry.insert(0, "Precio B/.") # indicación

# Etiquetas y entradas para Naranja
tk.Label(root, text="Naranja", font=("Arial", 12, "bold"), bg="#60B2FF").pack(pady=5)
entrada_naranja = tk.Entry(root, font=("Arial", 12))
entrada_naranja.pack(pady=2)
precio_naranja_entry = tk.Entry(root, font=("Arial", 12))
precio_naranja_entry.pack(pady=2)
precio_naranja_entry.insert(0, "Precio B/.")

# Etiquetas y entradas para Limón
tk.Label(root, text="Limón", font=("Arial", 12, "bold"), bg="#60B2FF").pack(pady=5)
entrada_limon = tk.Entry(root, font=("Arial", 12))
entrada_limon.pack(pady=2)
precio_limon_entry = tk.Entry(root, font=("Arial", 12))
precio_limon_entry.pack(pady=2)
precio_limon_entry.insert(0, "Precio B/.")

# Botón para realizar la operación de calcular
tk.Button(root, text="Calcular Ventas", font=("Arial", 12, "bold"), fg="white",
          bg="#FF477E", command=calcular_ventas).pack(pady=15)

# Etiqueta para mostrar resultados
resultado = tk.StringVar()
tk.Label(root, textvariable=resultado, font=("Arial", 12), bg="#60B2FF", justify="left").pack(pady=10)

root.mainloop()
```

