

# Universidad Tecnológica de Panamá

Asignatura: Desarrollo Lógico y Algoritmo

## Ejercicio Práctico 2

Profesor: Napoleón Ibarra

Valor: 100 puntos

Estudiante: Eliana Martínez

Cédula: 4-834-1382

Detalle	Fecha y hora
<b>Fecha Inicio:</b>	<b>13/10/2025 --&gt; 16:10</b>
<b>Fecha de entrega:</b>	<b>14/10/2025 --&gt; 14:25</b>
<b>Fecha Cierre (Tope):</b>	<b>20/10/2025 --&gt; 16:10</b>

### Procedimiento

- ⊕ De manera INDIVIDUAL o en grupos de 2 PERSONAS, realizar la asignación.
- ⊕ Investigue, descargue e instale 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 desarrollados los problemas. Sustente su trabajo en el aula de clases.

### Criterios de evaluación

Criterios	Puntos (Mínimo=1, Máximo=5)	Porcentaje
Desarrollo	1-5	70%
Sustentación	1-5	15%
Puntualidad	1-5	15%

### I Parte. Laboratorio. Valor 20 puntos

- ⊕ Elegir un software para virtualización (Virtual Box / VMWare), descargarlo, instalarlo y configurarlo.
- ⊕ Descargar, instalar la distribución de Windows Server (2016/2019) elegida dentro de la máquina virtual.
- ⊕ Realice el proceso de configuración correspondiente al realizar el proceso de instalación del SO.
- ⊕ Configure una carpeta para hacer respaldos de códigos dentro del SO, verifique su instalación, configuración.

- Realice la configuración del Servidor de Datos (que sea accesible mediante Conexión a Escritorio Remoto, ANYDESK, FTP).
- Hacer pruebas de funcionamiento.

**Oracle VirtualBox Administrador**

Nueva Open Configuración Descartar Mostrar

server 2016 Corriendo

Detalles

General

Nombre: server  
Sistema operativo: Windows Server 2016 (64-bit)

Sistema

Pantalla

Memoria de video: 128 MB  
Controlador gráfico: VBoxSVGA  
Servidor de escritorio remoto: Inhabilitado  
Grabación: Inhabilitado

Almacenamiento

Controlador: SATA  
Puerto SATA 0: server.vdi (Normal, 50.00 GB)  
Puerto SATA 1: [Unidad óptica] VBoxGuestAdditions.iso (50.67 MB)  
Puerto SATA 2: [Unidad óptica] Unattended-4a2e3daa-241f-47dc-bb10-49a261f825c4-aux-iso.viso (0 B)  
Controlador Floppy  
Dispositivo de disquete 0: Unattended-4a2e3daa-241f-47dc-bb10-49a261f825c4-aux-floppy.img (1.41 MB)

Audio

Controlador de anfitrión: Predeterminado  
Controlador: Audio Intel HD

Red

Adaptador 1: Intel PRO/1000 MT Desktop (NAT)

USB

Controlador USB: xHCI  
Filters de dispositivos: 0 (0 activo)

Carpetas compartidas

Ninguno

Descripción

Ninguna

**Previsualización**

**server - Administrador de archivos**

Administrador de archivos

Sistema de archivos anfitrión: C: > Users > Usuario

Nombre	Tamaño	Cambiar hora	Propietario	Permisos
..	0 B	01.09.2025 1...	rwx...	
zin...	0 B	20.10.2025 1...	rwx...	
Vir...	0 B	02.01.2025 2...	r-xr...	
Vi...	0 B	19.11.2024 0...	rwx...	
Se...	0 B	19.11.2024 0...	r-xr...	
Se...	0 B	19.11.2024 0...	r-xr...	
Sa...	0 B	19.11.2024 0...	r-xr...	
Re...	0 B	19.11.2024 0...	rwx...	

Sistema de archivos invitado:

Nombre	Tamaño	Cambiar hora	Propietario	Permisos
A:	0 B			
C:	0 B			
D:	0 B			
E:	0 B			

ELI ..... Cerrar sesión

Preferencias Operaciones Log

Listar carpetas arriba  Tamaños legibles por humanos  
 Preguntar antes de eliminar  Mostrar objetos ocultos  
 Interactive Column Widths

Cerrar Ayuda

**Linux**

Windows\_Server\_2016\_Datacenter\_EVAL.... 10/14/2025 6:40 p. m. Archivo de image... 6,808,810 KB

python-3.14.0-amd64.exe 10/14/2025 11:49 a. m. Aplicación 29,200 KB

python-3.14.0-aarch64-linux-android.tar.... 10/14/2025 11:21 a. m. Archivo WinRAR 20,418 KB

VSCodeUserSetup-x64-1.105.0.exe 10/14/2025 11:06 a. m. Aplicación 112,776 KB

Python 3.11 Installer.exe 10/13/2025 10:18 p. m. Aplicación 1,080 KB

Microsoft.Management.Deployment.win... 10/13/2025 10:18 p. m. Archivo WINMD 31 KB

Visual Studio Code Installer (2).exe 10/13/2025 10:17 p. m. Aplicación 1,080 KB

Visual Studio Code Installer (1).exe 10/13/2025 10:10 p. m. Aplicación 1,080 KB

451 elementos

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

Figura 1. Prototipo de diagrama

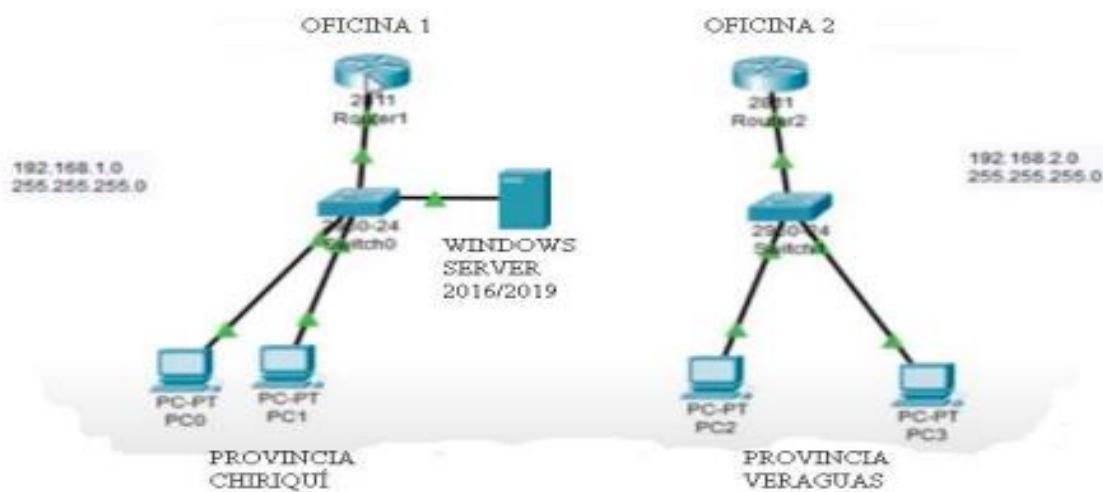
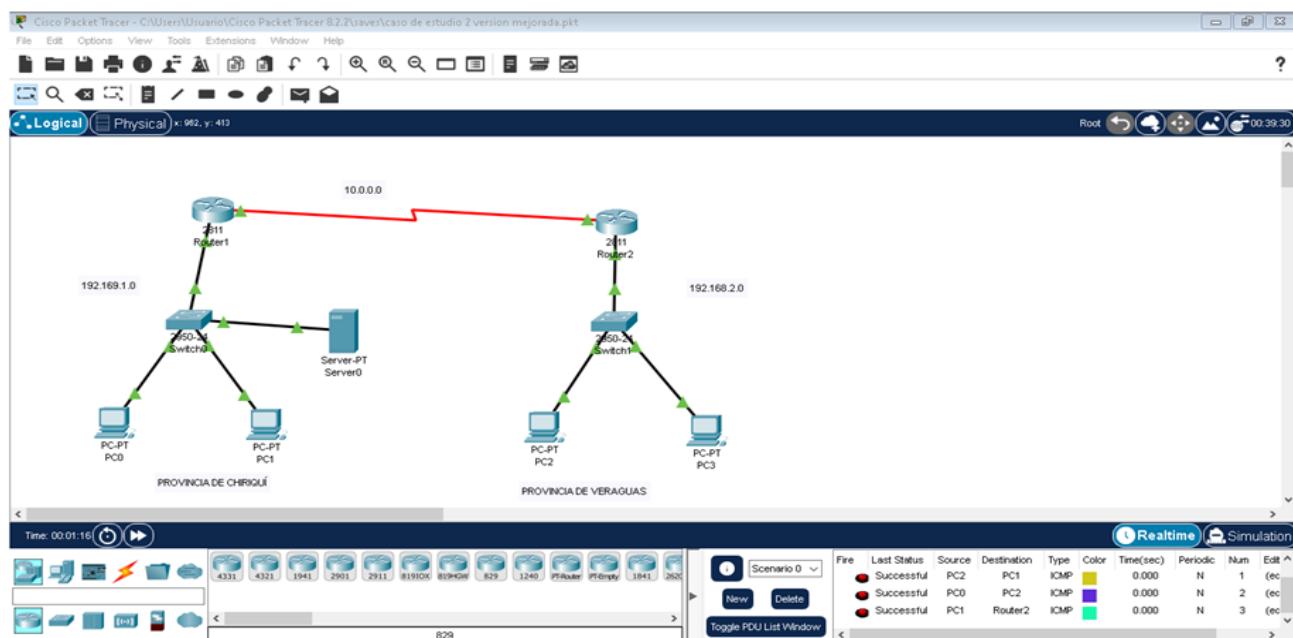


Figura 1. Prototipo de diagrama

### 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.



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

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

### Problema 1: Conversión de hora de 24h a 12h

Pseudocódigo	Diagrama de flujo
<p><b>INICIO</b></p> <p><b>MOSTRAR</b> "Ingrese una hora en formato 24h (HH:MM):"</p> <p><b>LEER</b> hora_24</p> <p><b>SI</b> longitud de hora_24 ≠ 5 O el carácter 3 no es ':'</p> <p><b>ENTONCES</b></p> <p><b>MOSTRAR</b> "Error: formato incorrecto"</p> <p><b>TERMINAR</b></p> <p><b>FIN SI</b></p> <p><b>DIVIDIR</b> hora_24 en horas y minutos usando ':' como separador</p> <p><b>CONVERTIR</b> horas y minutos a números enteros</p> <p><b>SI</b> horas &lt; 0 O horas &gt; 23 O minutos &lt; 0 O minutos &gt; 59 <b>ENTONCES</b></p> <p><b>MOSTRAR</b> "Error: hora no válida"</p> <p><b>TERMINAR</b></p> <p><b>FIN SI</b></p> <p><b>DEFINIR</b> sufijo ← "am"</p> <p><b>SI</b> horas == 0 <b>ENTONCES</b>     horas ← 12</p> <p><b>SINO SI</b> horas == 12 <b>ENTONCES</b>     sufijo ← "pm"</p> <p><b>SINO SI</b> horas &gt; 12 <b>ENTONCES</b>     horas ← horas - 12     sufijo ← "pm"</p> <p><b>FIN SI</b></p> <p><b>MOSTRAR</b> "Hora en formato 12h: " horas ":" minutos " " sufijo</p> <p><b>FIN</b></p>	<pre> graph TD     Start([Algoritmo ConvertirHora]) --&gt; Definir[Definir hora_24, horas...]     Definir --&gt; Definir2[Definir horas, minutos...]     Definir2 --&gt; Leer[Leer hora_24]     Leer --&gt; Longitud{Longitud(hora_24)}     Longitud --&gt; ErrorFormato['Error: formato incorrecto!']     Longitud --&gt; Separar[Separar horas y minutos]     Separar --&gt; Subcadena1[Subcadena...]     Subcadena1 --&gt; Subcadena2[Subcadena...]     Subcadena2 --&gt; Numeros[Convertir a números]     Numeros --&gt; Horas[horas + ConvertirANumer...]     Numeros --&gt; Minutos[minutos + ConvertirANu...]     Horas --&gt; Validacion{Validar rango de valores}     Minutos --&gt; Validacion     Validacion --&gt; ErrorHoraNoValida['Error: hora no válida!']     Validacion --&gt; Sufijo["sufijo ← 'am'"]     Sufijo --&gt; Conversion12h["Conversión a formato 12h"]     Conversion12h --&gt; Horas12{horas&lt;0 o horas&gt;23}     Horas12 --&gt; ErrorHoraNoValida     Horas12 --&gt; Horas0{horas=0}     Horas0 --&gt; Horas12h[horas + 12]     Horas12h --&gt; SufijoPm["sufijo ← 'pm'"]     SufijoPm --&gt; Mostrar[Mostrar resultado form...]     Mostrar --&gt; Hora12h["Hora en formato 12h: ..."]     Hora12h --&gt; FinAlgoritmo([FinAlgoritmo])     </pre>

## Problema 2: Cálculo de ventas de refrescos.

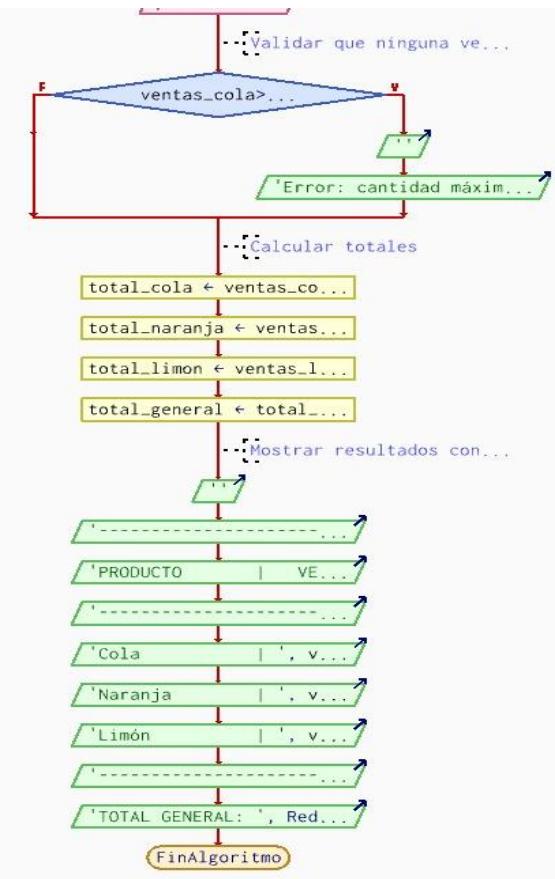
Pseudocódigo	
<p><b>INICIO</b></p> <p><b>MOSTRAR "Ingrese los datos de ventas (máx. 5,000,000):"</b></p> <p><b>MOSTRAR "Cola - cantidad vendida:"</b> LEER ventas_colas</p> <p><b>MOSTRAR "Cola - precio (Balboas):"</b> LEER precio_colas</p> <p><b>MOSTRAR "Naranja - cantidad vendida:"</b> LEER ventas_naranja</p> <p><b>MOSTRAR "Naranja - precio (Balboas):"</b> LEER precio_naranja</p> <p><b>MOSTRAR "Limón - cantidad vendida:"</b> LEER ventas_limon</p> <p><b>MOSTRAR "Limón - precio (Balboas):"</b> LEER precio_limon</p> <p><b>SI</b> alguna de las ventas &gt; 5000000 <b>ENTONCES</b></p> <p style="padding-left: 20px;"><b>MOSTRAR "Error: cantidad máxima excedida"</b></p> <p style="padding-left: 20px;"><b>TERMINAR</b></p> <p><b>FIN SI</b></p> <pre> total_colas ← ventas_colas * precio_colas total_naranja ← ventas_naranja * precio_naranja total_limon ← ventas_limon * precio_limon total_general ← total_colas + total_naranja + total_limon  <b>MOSTRAR "Producto Ventas Precio Total"</b> <b>MOSTRAR "-----"</b> <b>MOSTRAR "Cola      ", ventas_colas, " ",</b> <b>precio_colas, " ", total_colas</b> <b>MOSTRAR "Naranja  ", ventas_naranja, " ",</b> <b>precio_naranja, " ", total_naranja</b> </pre>	<pre> graph TD     Start([Algoritmo Registro_Ventas]) --&gt; DefCola[Definir ventas_colas, v...]     DefCola --&gt; DefPrecio[Definir precio_colas, p...]     DefPrecio --&gt; DefTotal[Definir total_colas, to...]     DefTotal --&gt; Ingresar[Ingrese los datos de ...]     Ingresar --&gt; Cola[Cola - cantidad vendida...]     Cola -- ventas_colas --&gt; Naranja[Naranja - cantidad ve...]     Naranja -- ventas_naranja --&gt; Limon[Limón - cantidad vend...]     Limon -- ventas_limon --&gt; Validar{Validar que ninguna ve...}     Validar --&gt; Decision{ventas_colas&gt;...}     Decision -- S --&gt; Error[Error: cantidad máxi...]     Decision -- NO --&gt; Total[Total de ventas]     Total --&gt; Fin([Fin]) </pre>

**MOSTRAR "Limón ", ventas\_limon, " ",  
precio\_limon, " ", total\_limon**

**MOSTRAR "-----"**

**MOSTRAR "TOTAL: ", total\_general**

**FIN**



**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, la salida será. El programa debe solicitar al usuario que introduzca exactamente cinco caracteres para especificar una hora. valor 10 punto

```
 1  main.py > ...
 2  import tkinter as tk
 3  from tkinter import messagebox
 4
 5  def convertir_hora():
 6      hora_24 = entrada.get()
 7      if len(hora_24) != 5 or hora_24[2] != ':':
 8          messagebox.showerror("Error", "Debe ingresar la hora en formato HH:MM")
 9          return
10
11     try:
12         horas, minutos = map(int, hora_24.split(':'))
13         if not (0 <= horas < 24 and 0 <= minutos < 60):
14             raise ValueError
15         messagebox.showerror("Error", "Hora no válida.")
16         return
17
18     sufijo = "am"
19     if horas == 0:
20         horas = 12
21     elif horas == 12:
22         sufijo = "pm"
23     elif horas > 12:
24         horas -= 12
25         sufijo = "pm"
26
27     resultado = f"{horas}:{minutos:02d} {sufijo}"
28     etiqueta_resultado.config(text=f"Hora en formato 12h: {resultado}")
29
30 # Interfaz gráfica
31 ventana = tk.Tk()
32 ventana.title("Conversión de Hora 24h a 12h")
33
34 tk.Label(ventana, text="Ingrese hora en formato 24h (HH:MM)").pack(pady=5)
35 entrada = tk.Entry(ventana)
36 entrada.pack(pady=5)
37
38 tk.Button(ventana, text="Convertir", command=convertir_hora).pack(pady=5)
39 etiqueta_resultado = tk.Label(ventana, text="")
40 etiqueta_resultado.pack(pady=5)
41
42 ventana.mainloop()
43
```

2. Una compañía de refrescos comercializa tres productos: de cola, de naranja y de limón. Se desea realizar una aplicación que calcule las ventas realizadas de cada producto. Para ello, se

leerá la cantidad vendida (máximo) y el precio en Balboas de cada producto y se mostrará un informe de ventas como el que sigue:

```
main.py problema2_refrescos.py > calcular

# Interfaz gráfica
ventana = tk.Tk()
ventana.title("Cálculo de Ventas de Refrescos")

# Cola
tk.Label(ventana, text="Ventas (máx. 5,000,000)").grid(row=0, column=1)
tk.Label(ventana, text="Precio (Balboas)").grid(row=0, column=2)

# Naranja
tk.Label(ventana, text="Naranja").grid(row=2, column=0)
entry_naranja = tk.Entry(ventana)
entry_naranja.grid(row=2, column=1)
entry_precio_naranja = tk.Entry(ventana)
entry_precio_naranja.grid(row=2, column=2)

# Limón
tk.Label(ventana, text="Limón").grid(row=3, column=0)
entry_limon = tk.Entry(ventana)
entry_limon.grid(row=3, column=1)
entry_precio_limon = tk.Entry(ventana)
entry_precio_limon.grid(row=3, column=2)

tk.Button(ventana, text="Calcular", command=calcular).grid(row=4, column=0, columnspan=3)

text_resultado = tk.Text(ventana, height=10, width=50)
text_resultado.grid(row=5, column=0, columnspan=3)

ventana.mainloop()
```

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