

Universidad Tecnológica de Panamá
Facultad de Ingeniería en Sistemas Computacionales
Asignatura: Desarrollo Lógico Y Algoritmo
Ejercicio Práctico 2

Profesor: Napoleón Ibarra

Valor: 100 puntos

Estudiantes: Carlos Jiménez y Arvin Guerra

Cédulas: 4-845-1014 y 4-836-808

I PARTE. Caso de Estudio. Valor 70 puntos

Una empresa familiar (CHINOS CAFE) le contrata a Usted (es) por servicios profesionales para que realicen el siguiente proyecto. Esta empresa desea crear, configurar, gestionar un Sistema de Información (PYTHON y MYSQL Server Máquina Virtual) para gestionar sus datos. El negocio cuenta con 2 establecimientos (1 prestablecido, otro por inaugurar). Dentro de sus recursos informáticos tenemos 1 Impresora, 1 Servidor, 3 PC, 3 Laptop, 1 Punto de Acceso para los Clientes. 1 Router Inalámbrico, 1 Tableta. Estos equipos se encuentran en la oficina que está en operación. Se debe replicar los mismos equipos en la otra oficina.

1. Configuración del Servidor Local (MySQL Win11):

```
C:\Program Files\MySQL\MySQL Server 8.0\bin\mysql.exe Ver 8.0.44 for Win64 on x86_64 (MySQL Community Server - GPL)

Connection id:          24
Current database:       chinos_cafe
Current user:           root@localhost
SSL:                  Cipher in use is TLS_AES_256_GCM_SHA384
Using delimiter:        ;
Server version:         8.0.44 MySQL Community Server - GPL
Protocol version:       10
Connection:             localhost via TCP/IP
Server characterset:   utf8mb4
Db      characterset:  utf8mb4
Client characterset:   cp850
Conn. characterset:    cp850
TCP port:              3306
Binary data as:         Hexadecimal
Uptime:                8 days 18 hours 4 min 23 sec

Threads: 2  Questions: 132  Slow queries: 0  Opens: 216  Flush tables: 3  Open tables: 135  Queries per second avg: 0.000

mysql> use chinos_cafe;
Database changed
mysql> show tables;
+-----+
| Tables_in_chinos_cafe |
+-----+
| contactos            |
| detalle_ventas        |
| productos             |
| proveedores           |
| sucursales            |
| ventas                |
+-----+
6 rows in set (0.01 sec)

mysql> describe productos;
+-----+
| Field      | Type     | Null | Key | Default | Extra
+-----+
| id_producto | int      | NO   | PRI  | NULL    | auto_increment
| nombre       | varchar(100)| YES  |      | NULL    |
| precio       | decimal(10,2) | YES  |      | NULL    |
| stock        | int      | YES  |      | NULL    |
| id_proveedor | int      | YES  | MUL  | NULL    |
| id_sucursal  | int      | YES  | MUL  | NULL    |
+-----+
6 rows in set (0.00 sec)

mysql> describe contactos;
+-----+
| Field      | Type     | Null | Key | Default | Extra
+-----+
| id_contacto | int      | NO   | PRI  | NULL    | auto_increment
| nombre       | varchar(100)| YES  |      | NULL    |
| email        | varchar(100)| YES  |      | NULL    |
| mensaje      | text      | YES  |      | NULL    |
| fecha_envio  | datetime | YES  |      | CURRENT_TIMESTAMP | DEFAULT_GENERATED
+-----+
5 rows in set (0.00 sec)

mysql> describe proveedores;
+-----+
| Field      | Type     | Null | Key | Default | Extra
+-----+
| id_proveedor | int      | NO   | PRI  | NULL    | auto_increment
| nombre       | varchar(100)| YES  |      | NULL    |
| telefono    | varchar(20) | YES  |      | NULL    |
| email        | varchar(100)| YES  |      | NULL    |
+-----+
4 rows in set (0.01 sec)

mysql> describe detalle_ventas;
+-----+
| Field      | Type     | Null | Key | Default | Extra
+-----+
| id_detalle | int      | NO   | PRI  | NULL    | auto_increment
| id_venta   | int      | YES  | MUL  | NULL    |
| id_producto | int      | YES  | MUL  | NULL    |
| cantidad    | int      | YES  |      | NULL    |
| subtotal    | decimal(10,2)| YES  |      | NULL    |
+-----+
5 rows in set (0.00 sec)

mysql> describe sucursales;
+-----+
| Field      | Type     | Null | Key | Default | Extra
+-----+
| id_sucursal | int      | NO   | PRI  | NULL    | auto_increment
| nombre       | varchar(50) | YES  |      | NULL    |
| direccion   | varchar(100)| YES  |      | NULL    |
+-----+
3 rows in set (0.00 sec)

mysql> describe ventas;
+-----+
| Field      | Type     | Null | Key | Default | Extra
+-----+
| id_venta   | int      | NO   | PRI  | NULL    | auto_increment
| fecha       | datetime | YES  |      | CURRENT_TIMESTAMP | DEFAULT_GENERATED
| total       | decimal(10,2) | YES  |      | NULL    |
| id_sucursal | int      | YES  | MUL  | NULL    |
+-----+
4 rows in set (0.00 sec)
```

2. Configuración del Servidor Virtual (MariaDB UbuntuServer/Linux):

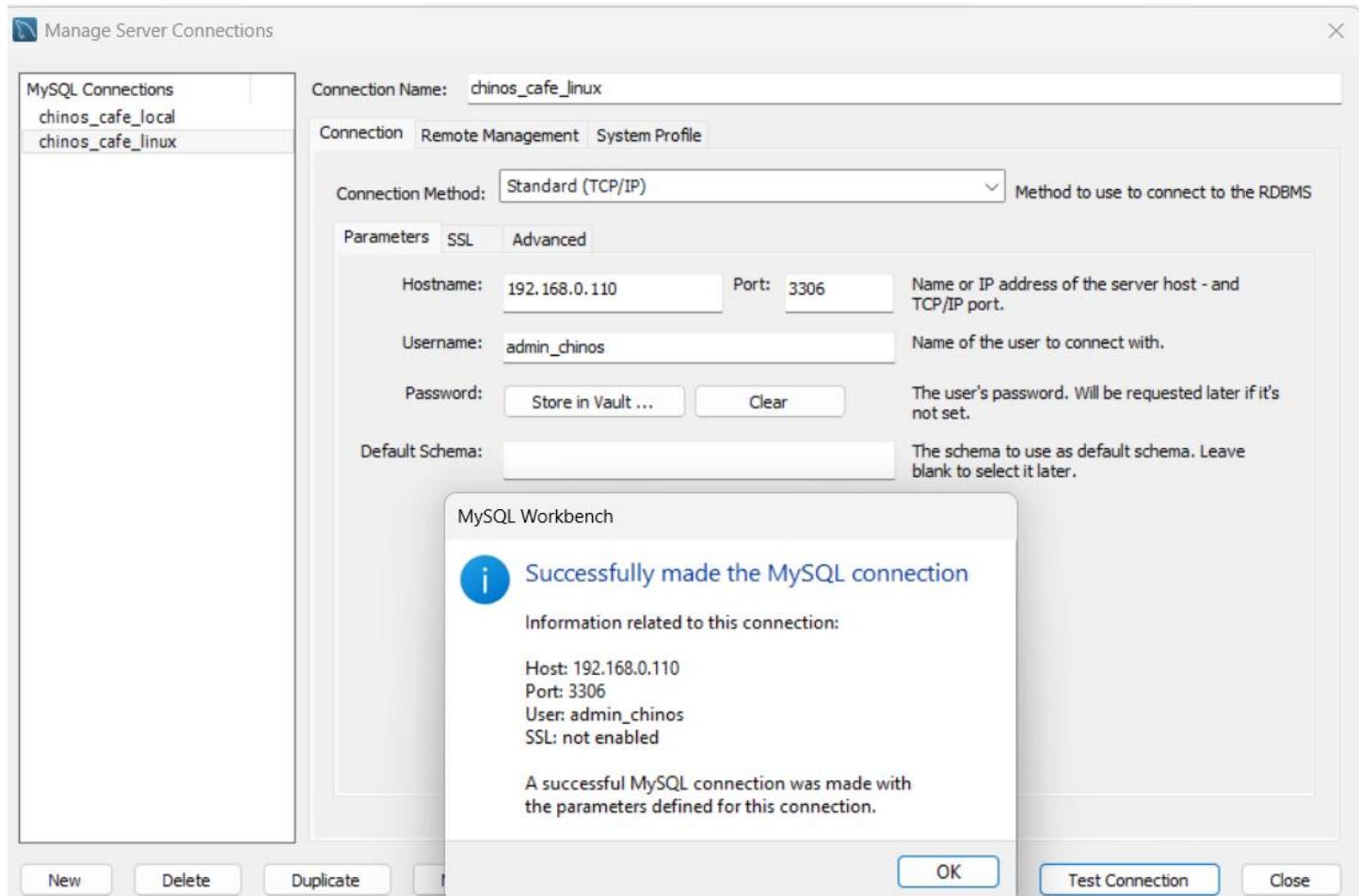
```
carlos@localhost:~$ sudo systemctl status mariadb
● mariadb.service - MariaDB 10.11 database server
  Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; preset: disabled)
  Active: active (running) since Wed 2025-11-12 18:07:20 EST; 8min ago
  Invocation: 3062366b117649cd80bc2b5781dcc3b5
    Docs: man:mariadb(8)
          https://mariadb.com/kb/en/library/systemd/
  Main PID: 4131 (mariadb)
    Status: "Taking your SQL requests now..."
      Tasks: 9 (limit: 35077)
     Memory: 204.4M (peak: 231.1M)
        CPU: 1.548s
      CGroup: /system.slice/mariadb.service
              └─4131 /usr/libexec/mariadb --basedir=/usr
```

3. Configuración del Firewall Virtual (Habilitar Puerto 3306):

```
carlos@localhost:~$ sudo systemctl status firewalld
● firewalld.service - firewalld - dynamic firewall daemon
  Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; preset: enabled)
  Active: active (running) since Wed 2025-11-12 17:51:19 EST; 49min ago
  Invocation: de9a9945065045fdb674610c8e06c05d
    Docs: man:firewalld(1)
  Main PID: 895 (firewalld)
    Tasks: 2 (limit: 35077)
   Memory: 47.2M (peak: 69.7M)
      CPU: 1.336s
    CGroup: /system.slice/firewalld.service
            └─895 /usr/bin/python3 -sP /usr/sbin/firewalld --nofork --nopid

nov 12 17:51:16 localhost systemd[1]: Starting firewalld.service - firewalld - dynamic firewall daemon...
nov 12 17:51:19 localhost systemd[1]: Started firewalld.service - firewalld - dynamic firewall daemon.
carlos@localhost:~$ sudo firewall-cmd --add-port=3306/tcp --permanent
success
carlos@localhost:~$ sudo firewall-cmd --reload
success
carlos@localhost:~$ sudo firewall-cmd --list-ports
3306/tcp
```

4. Conexión SQL de Windows 11 (MySQL-Server) a UbuntuServer (MariaDB):



Con la configuración del servidor ya terminada, se procede a realizar el sistema POS en PYTHON con su respectiva conexión en la BD ubicado de manera local.

OBSERVACIÓN: La conexión SQL Windows-Linux se realiza mediante la librería de Python "mySqlConnector", por lo que el código es idéntico, excepto en los valores de "host" y "user", ya que estos dependen del SO en el que se ejecuten.

¿Qué requiere la organización? I PARTE. INTERFAZ. Valor 50 Puntos.

- 1. ¿Qué arquitectura SGBD, aparte de servidor de archivo, se recomendaría? ¿Por qué? ¿Es necesario que el SGBD sea multihilo? Sí o no y por qué.**

Se recomienda una arquitectura cliente servidor con MySQL Server en una máquina virtual dedicada. Esta arquitectura organiza las transacciones y mantiene la información centralizada. El servidor de archivos separado guarda documentos, comprobantes y respaldos para evitar carga sobre el motor de base de datos. El SGBD necesita multihilo porque varias estaciones trabajan al mismo tiempo. El procesamiento en paralelo mantiene tiempos de respuesta estables durante ventas, inventario y consultas administrativas.

- 2. Adecuación y configuración del diagrama entidad relación, registros, tablas y base de datos.**

Identificación y gestión del servidor MySQL Server, servidor de archivos y servidor web.

El modelo entidad relación incluye tablas para productos, categorías, inventario, usuarios, roles, clientes, ventas, detalle de venta, proveedores, compras, detalle de compra, facturas, sucursales y dispositivos. Cada tabla usa claves primarias y claves foráneas que garantizan integridad. Se indexan campos como sku, id_producto, id_sucursal y fecha de venta para acelerar consultas. El servidor MySQL gestiona toda la información estructurada del negocio. El servidor web ejecuta la aplicación escrita en Python y entrega la interfaz del POS y el sitio. El servidor de archivos conserva reportes, comprobantes y copias de seguridad. Cada servidor se configura con firewall, usuarios con permisos limitados y respaldos automáticos.

- 3. Gestione un Sistema POS (Punto de Venta) en Python. Contemple procesos necesarios cotidianos como sección ventas, proveedores, inventario y factura.**

El POS procesa las ventas mediante búsqueda de productos, cálculo de totales, impuestos e impresión del comprobante. El módulo de inventario registra movimientos por ventas, compras y ajustes, además de alertas por niveles bajos de existencias. El módulo de proveedores administra información de contacto y compras. Las compras generan entrada a inventario y registro de facturas. El módulo de facturación crea comprobantes fiscales y los almacena en el servidor de archivos. El sistema mantiene roles, permisos y auditoría de actividades.

- 4. Gestione el prototipo de sitio web. Incluya adecuación en plantilla, configuración, diseño de interfaz y un formulario que se guarde en la base de datos para contactos posteriores.**

El sitio web presenta información del negocio, menú, sucursales y una sección de contacto. El formulario incluye nombre, correo, asunto y mensaje. El servidor web recibe los datos, los valida y los guarda en una tabla de contactos junto con la fecha de envío. El diseño usa una plantilla responsive y una interfaz clara para facilitar la navegación.

- 5. Conexión entre el prototipo desarrollado y los servidores. Es decir, intercambio de datos o registros y acceso de administración.**

La comunicación se realiza mediante una API REST en el servidor web. El prototipo usa solicitudes HTTPS para enviar y recibir datos. Las credenciales se manejan mediante variables de entorno. El control de acceso limita funciones administrativas a usuarios autorizados. Las estaciones de cada sucursal se conectan al servidor web para registrar ventas y consultar inventario. Si la red falla de forma temporal, las operaciones se guardan localmente y se sincronizan cuando regresa la conexión. El sistema mantiene registros de actividad, monitoreo y respaldos para asegurar continuidad.

6. Diagrama de Flujo y Pseudocódigo.



Algoritmo POS_Simple

```
// Arreglos para productos (máximo 10)
Dimension nombres[10], precios[10], stocks[10]
// Carrito (máximo 5 artículos)
Dimension carrito_nombres[5], carrito_cantidades[5], carrito_subtotales[5]

// Inicializar algunos productos de ejemplo
nombres[0] <- "Café"
precios[0] <- 2.5
stocks[0] <- 10

nombres[1] <- "Pastel"
precios[1] <- 3.0
stocks[1] <- 5

cant_productos <- 2
cant_carrito <- 0

Repetir
    Escribir "___ CHINOS CAFE ___"
    Escribir "1. Ver productos"
    Escribir "2. Agregar al carrito"
    Escribir "3. Ver carrito y pagar"
    Escribir "4. Salir"
    Leer opcion

    Segun opcion Hacer
        1:
            Escribir "-- Productos --"
            Para i <- 0 Hasta cant_productos - 1 Hacer
                Escribir i + 1, ". ", nombres[i], " - $", precios[i], " (Stock: ", stocks[i], ")"
            FinPara

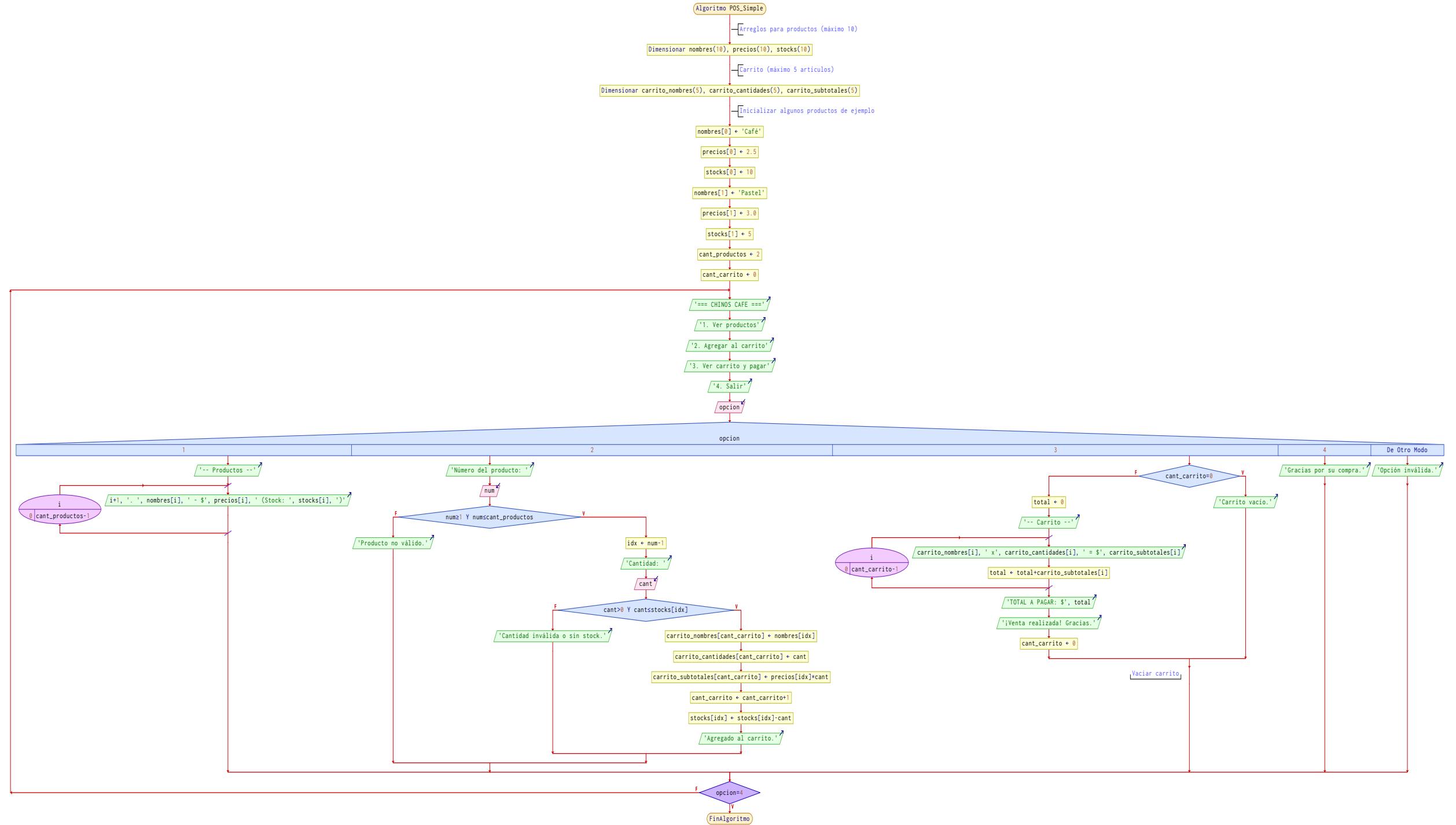
        2:
            Escribir "Número del producto: "
            Leer num
            Si num >= 1 Y num <= cant_productos Entonces
                idx <- num - 1
                Escribir "Cantidad: "
                Leer cant
                Si cant > 0 Y cant <= stocks[idx] Entonces
                    carrito_nombres[cant_carrito] <- nombres[idx]
                    carrito_cantidades[cant_carrito] <- cant
                    carrito_subtotales[cant_carrito] <- precios[idx] * cant
                    cant_carrito <- cant_carrito + 1
                    stocks[idx] <- stocks[idx] - cant
                    Escribir "Agregado al carrito."
                Sino
                    Escribir "Cantidad inválida o sin stock."
                FinSi
            Sino
                Escribir "Producto no válido."
            FinSi

        3:
            Si cant_carrito = 0 Entonces
                Escribir "Carrito vacío."
            Sino
                total <- 0
                Escribir "-- Carrito --"
                Para i <- 0 Hasta cant_carrito - 1 Hacer
                    Escribir carrito_nombres[i], " x", carrito_cantidades[i], " = $",
                carrito_subtotales[i]
                    total <- total + carrito_subtotales[i]
                FinPara
                Escribir "TOTAL A PAGAR: $", total
                Escribir "¡Venta realizada! Gracias."
                cant_carrito <- 0 // Vaciar carrito
            FinSi

        4:
            Escribir "Gracias por su compra."

    DeOtroModo:
        Escribir "Opción inválida."
    FinSegun
Hasta Que opcion = 4

FinAlgoritmo
```

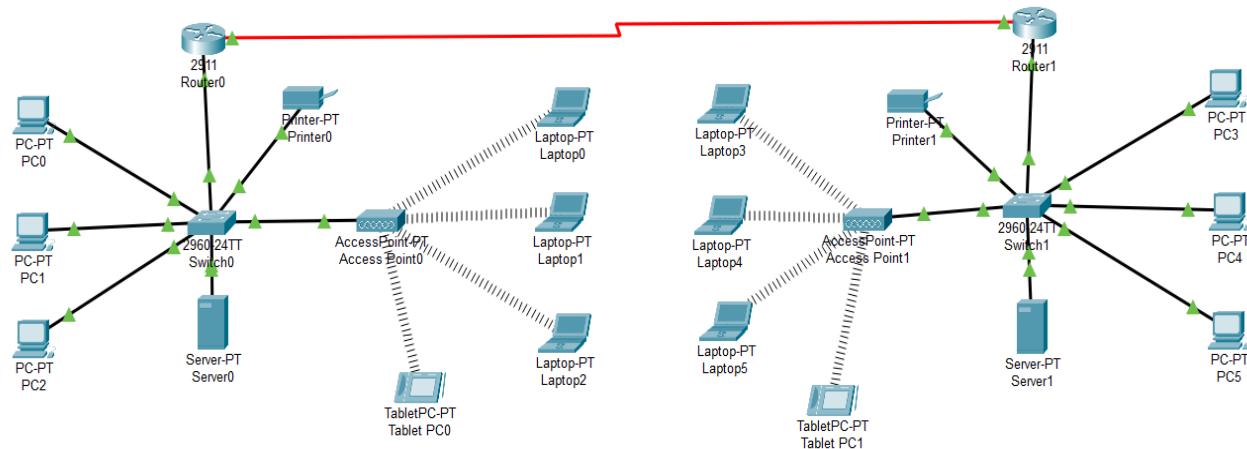


II PARTE. RED LAN. Valor 15 Puntos.

1) Confeccione, simule el diagrama de red para cada sucursal.

Conecte las 2 sucursales (Vía CACSA y por inaugurar (San Mateo), Sugerencia: utilice VLAN.

2) Pruebas de funcionamiento del proyecto.



III PARTE. DESARROLLO. Valor 15 Puntos.

1. Gestione, configure un Firewall (cortafuego) para el Servidor Web Apache. Usted debe elegir la aplicación y configurarlo para que permita mitigar ataques como: DDOS, otros.

```
● firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; preset: enabled)
   Active: active (running) since Sun 2025-11-16 19:11:12 EST; 1h 16min ago
     Invocation: dec3ced1eb6347b297d732e7e0f8bd01
       Docs: man:firewalld(1)
     Process: 911 ExecStartPost=/usr/bin/firewall-cmd --state (code=exited, status=0/SUCCESS)
    Main PID: 896 (firewalld)
      Tasks: 2 (limit: 21599)
     Memory: 50.8M (peak: 71.5M)
        CPU: 6.414s
       CGroup: /system.slice/firewalld.service
               └─896 /usr/bin/python3 -sP /usr/sbin/firewalld --nofork --nopid

nov 16 19:10:48 localhost systemd[1]: Starting firewalld.service - firewalld - dynamic firewall daemon...
nov 16 19:11:12 localhost systemd[1]: Started firewalld.service - firewalld - dynamic firewall daemon.
carlos@localhost:~/Descargas$ sudo firewall-cmd --permanent --add-service=http
success
carlos@localhost:~/Descargas$ sudo firewall-cmd --permanent --add-service=ssh
Warning: ALREADY_ENABLED: ssh
success
carlos@localhost:~/Descargas$ sudo firewall-cmd --list-all
public (default, active)
  target: default
  ingress-priority: 0
  egress-priority: 0
  icmp-block-inversion: no
  interfaces: enp0s3
  sources:
    services: cockpit dhcpcv6-client ssh
    ports: 3306/tcp
    protocols:
    forward: yes
    masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

```
carlos@localhost:~/Descargas$ sudo firewall-cmd --zone=web-secure --list-all
web-secure (active)
  target: %%REJECT%%
  ingress-priority: 0
  egress-priority: 0
  icmp-block-inversion: no
  interfaces: eth0
  sources:
    services: http
  ports:
  protocols:
  forward: no
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
    rule family="ipv4" source address="127.0.0.0/8" reject
    rule protocol value="icmp" reject
    rule family="ipv4" source address="0.0.0.0/8" reject
```

```
carlos@localhost:~/Descargas$ sudo iptables -L -n -v
Chain INPUT (policy ACCEPT 660 packets, 167K bytes)
pkts bytes target     prot opt in     out     source          destination
  0    0 ACCEPT      all  --  lo      *       0.0.0.0/0        0.0.0.0/0
 32  3356 ACCEPT     all  --  *       *       0.0.0.0/0        0.0.0.0/0        state RELATED,ESTABLISHED
  0    0 ACCEPT      tcp   --  *       *       0.0.0.0/0        0.0.0.0/0        tcp dpt:22
  0    0 ACCEPT      tcp   --  *       *       0.0.0.0/0        0.0.0.0/0        tcp dpt:80
  0    0 DROP        icmp  --  *       *       0.0.0.0/0        0.0.0.0/0
551  141K REJECT     all  --  *       *       0.0.0.0/0        0.0.0.0/0        reject-with icmp-host-prohibited
  0    0 REJECT     all  --  *       *       0.0.0.0/0        0.0.0.0/0        reject-with icmp-host-prohibited

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target     prot opt in     out     source          destination

Chain OUTPUT (policy ACCEPT 69 packets, 5578 bytes)
pkts bytes target     prot opt in     out     source          destination
```

Configuraciones del servidor apache:

```
carlos@localhost:~$ sudo systemctl status httpd
● httpd.service - The Apache HTTP Server
  Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
  Active: inactive (dead)
    Docs: man:httpd.service(8)
carlos@localhost:~$ sudo systemctl start httpd
carlos@localhost:~$ sudo systemctl enable httpd
Created symlink '/etc/systemd/system/multi-user.target.wants/httpd.service' → '/usr/lib/systemd/system/httpd.service'.
carlos@localhost:~$ sudo systemctl status httpd
● httpd.service - The Apache HTTP Server
  Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
  Active: active (running) since Mon 2025-11-17 09:38:55 EST; 12s ago
  Invocation: ba97a233349f457a8b687d703e2aba9b
    Docs: man:httpd.service(8)
  Main PID: 6699 (httpd)
    Status: "Total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; Bytes served/sec: 0 B/sec"
    Tasks: 177 (limit: 21599)
   Memory: 13.4M (peak: 13.7M)
      CPU: 115ms
  CGroup: /system.slice/httpd.service
          ├─6699 /usr/sbin/httpd -DFOREGROUND
          ├─6700 /usr/sbin/httpd -DFOREGROUND
          ├─6701 /usr/sbin/httpd -DFOREGROUND
          ├─6702 /usr/sbin/httpd -DFOREGROUND
          └─6703 /usr/sbin/httpd -DFOREGROUND

nov 17 09:38:54 localhost.localdomain systemd[1]: Starting httpd.service - The Apache HTTP Server...
nov 17 09:38:55 localhost.localdomain (httpd)[6699]: httpd.service: Referenced but unset environment variable evaluates
nov 17 09:38:55 localhost.localdomain httpd[6699]: AH00558: httpd: Could not reliably determine the server's fully qual
nov 17 09:38:55 localhost.localdomain systemd[1]: Started httpd.service - The Apache HTTP Server.
nov 17 09:38:55 localhost.localdomain httpd[6699]: Server configured, listening on: port 80
[lines 1-22/22 (END)]
carlos@localhost:~$ sudo netstat -tulnp | grep :80
tcp6      0      0 :::::80                           ::::*                  LISTEN      6699/httpd
```

The screenshot shows a browser window for `127.0.0.1:5000` displaying the CHINOS CAFE website. The page features a header with "CHINOS CAFE", a menu with "Inicio" and "Contacto", and a main section with the heading "¡Bienvenido a CHINOS CAFE!" and the text "Disfruta de los mejores cafés y postres en Panamá.", followed by a blue button labeled "Contáctanos". To the right of the browser is a Visual Studio Code debugger interface. The code editor shows a file named `web.py` with the following content:

```
# ===== EJECUCIÓN (solo para desarrollo local) =====
if __name__ == '__main__':
    app.run(debug=True, host='0.0.0.0', port=5000)
```

The debugger sidebar includes sections for VARIABLES, WATCH, CALL STACK, and BREAKPOINTS. The CALL STACK shows the application is running on multiple threads. The BREAKPOINTS section has a checked checkbox for "Uncaught Exceptions". The TERMINAL tab displays log messages from the application's console.

The screenshot shows a browser window for `127.0.0.1:5000/contacto` displaying the CHINOS CAFE contact form. The form has fields for "Nombre completo" (with a placeholder "Introduce tu nombre completo..."), "Email" (with a placeholder "Introduce tu correo electrónico..."), and "Mensaje" (with a large text area). Below the form is a green "Enviar Mensaje" button. To the right of the browser is a Visual Studio Code debugger interface, identical to the one in the first screenshot, showing the same code in `web.py` and the same debugger state.

Configuraciones adicionales para el proceso de replicar:

```
C:\Windows\System32>netsh advfirewall firewall add rule name="MySQL" dir=in action=allow protocol=TCP localport=3306  
Aceptar  
MariaDB [(none)]> change master to  
  -> master_host='192.168.0.163',  
  -> master_user='admin_chinos',  
  -> master_password='1234',  
  -> master_log_file='CARLOS-GX-bin.000004',  
  -> master_log_pos=26297,  
  -> master_port=3306;  
Query OK, 0 rows affected, 1 warning (0.094 sec)  
  
MariaDB [(none)]> show variables like "replicate_do_db";  
+-----+-----+  
| Variable_name | Value |  
+-----+-----+  
| replicate_do_db | chinos_cafe |  
+-----+-----+  
1 row in set (0.003 sec)  
  
MariaDB [(none)]>   
  
MariaDB [(none)]> select user, Host from mysql.user;  
+-----+-----+  
| User      | Host    |  
+-----+-----+  
| admin_chinos | %      |  
| mariadb.sys  | localhost |  
| mysql       | localhost |  
| root        | localhost |  
+-----+-----+  
4 rows in set (0.032 sec)  
  
MariaDB [(none)]> grant replication slave, replication client on *.* to 'admin_chinos'@'%';  
Query OK, 0 rows affected (0.021 sec)  
  
MariaDB [(none)]> show grants for 'admin_chinos'@'%';  
+-----+  
| Grants for admin_chinos@% |  
+-----+  
| GRANT REPLICATION SLAVE, BINLOG MONITOR ON *.* TO 'admin_chinos'@`%` IDENTIFIED BY PASSWORD '*6BB4837EB74329105EE4568DDA7DC67ED2CA2AD9' |  
| GRANT ALL PRIVILEGES ON `chinos_cafe`.* TO 'admin_chinos'@`%` |  
+-----+  
2 rows in set (0.000 sec)
```