

Profesor: Napoleón Ibarra	Valor: 100 puntos
Estudiante: Isaac Moreno, Rick Jiménez, Dianys Pitty 4-835-1777	Cédula: 4-836-1992, 4-904-2017
<b>Fecha Inicio: 11/11/2025. Hora: 3:20 PM Fecha Entrega: 18/11/2025. Hora 4:10 PM</b>	

- De manera individual o en grupo de 2 personas confeccione la asignación. Utilice la herramienta Internet como apoyo para realizar su proceso de investigación.
- Una vez culminada la actividad, subirla a la plataforma Moodle (PDF) y presentar su debida sustentación. **Criterios de Evaluación:**

**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 preestablecido, otro por inaugurarse). 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.

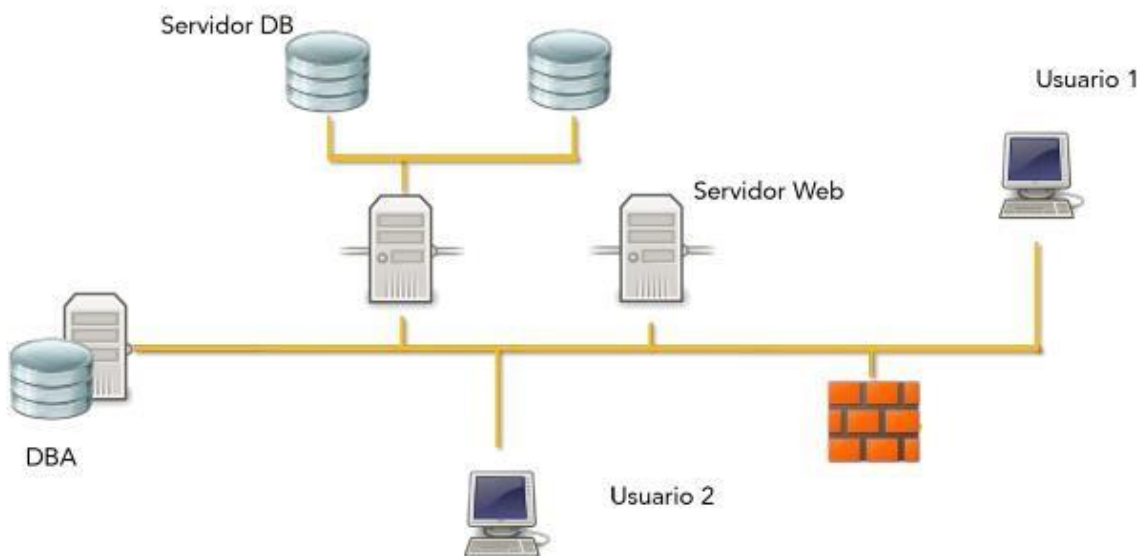


Figura 1. Esquema propuesto

¿Qué requiere la organización?

I PARTE. INTERFAZ. Valor 50 Puntos.

1. ¿Qué arquitectura SGBD, aparte de servidor de archivo recomendaría? ¿Por qué? ¿Es necesario que el SGBD sea multihilo? Si/No. ¿Por qué? Sustente sus respuestas.

**Arquitectura Recomendada: Cliente-Servidor (Clásica)**

La arquitectura recomendada es **Cliente-Servidor (Clásica)**.

**¿Por qué Cliente-Servidor?**

- 1. **Seguridad y Centralización:** El SGBD reside completamente en el Servidor DB (192.168.56.10). Los clientes (POS en Windows, Servidor Web) solo envían comandos SQL. Esto evita que los clientes accedan a los archivos de datos directamente, lo cual es un riesgo grave en la arquitectura de Servidor de Archivo.
- 2. **Integridad de Datos:** El Servidor MySQL gestiona la **conurrencia** (varios usuarios a la vez) y asegura la **integridad transaccional** (ACID), fundamental para el control de stock en un POS.
- 3. **Rendimiento:** Las consultas y el procesamiento de datos se realizan en el servidor (más potente), enviando solo los resultados al cliente, optimizando el tráfico de red.

**¿Es Necesario que el SGBD sea Multihilo? Sí**

**Sustento:**

Un SGBD **multihilo** es esencial en este entorno (CHINOS CAFE) porque permite que el Servidor MySQL:

- Atienda simultáneamente la conexión del **POS Python** (realizando una venta) y la conexión del **Servidor Web** (registrando un nuevo contacto).
- Cada solicitud de conexión (o cada usuario) es manejada por un hilo separado, previniendo bloqueos y caídas de servicio, garantizando la **disponibilidad continua** del sistema de ventas.

2. Adecuación, configuración de diagrama entidad-relación, registros, tablas, base de datos (DB). Identifique, gestione el Servidor MYSQL Server, Servidor de Archivos, Servidor Web.

**Estructura de la Base de Datos (chinos\_cafe)**

La DB está diseñada con tres tablas principales para una facturación atómica:

Tabla	Propósito	Claves
productos	Inventario y control de stock.	id_producto (PK)

Tabla	Propósito	Claves
ventas	Cabecera de la factura (Fecha y Total).	id_venta (PK)
detalle_venta	Líneas de la factura (qué producto, cuánta cantidad).	id_detalle (PK), id_venta (FK), id_producto (FK)

Comando de Creación (Master):  
SQL

```
CREATE DATABASE chinos_cafe;
USE chinos_cafe;
-- Tabla productos
CREATE TABLE productos (
    id_producto INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    precio DECIMAL(10, 2) NOT NULL,
    stock INT NOT NULL DEFAULT 0
);
-- Tabla ventas (Cabecera)
CREATE TABLE ventas (
    id_venta INT AUTO_INCREMENT PRIMARY KEY,
    fecha_venta DATETIME NOT NULL,
    total DECIMAL(10, 2) NOT NULL
);
-- Tabla detalle_venta (Líneas de la factura)
CREATE TABLE detalle_venta (
    id_detalle INT AUTO_INCREMENT PRIMARY KEY,
    id_venta INT NOT NULL,
    id_producto INT NOT NULL,
    cantidad INT NOT NULL,
    precio_unitario DECIMAL(10, 2) NOT NULL,
    FOREIGN KEY (id_venta) REFERENCES ventas(id_venta),
    FOREIGN KEY (id_producto) REFERENCES productos(id_producto)
);
```

Identificación y Gestión de Servidores

Servidor	Rol/Ubicación	IP Clave	Gestión
Servidor MYSQL Server (Master)	Almacenamiento primario y Escrituras	192.168.56.10	Habilitación de Log Binario y server-id = 1
Servidor MYSQL Server (Slave)	Réplica (Backup y Alta Disponibilidad)	192.168.56.20	Configurado para recibir logs del Master (SHOW REPLICA STATUS)
Sistema POS (Host)	Ejecución de la aplicación Python	192.168.56.1	Conexión remota al Master (192.168.56.10)
Servidor Web	Alojamiento del Sitio Web/Formulario	<i>Puede ser el Slave o una tercera MV</i>	Instalación de Apache/Nginx y PHP/Python.

3. Gestione un Sistema POS (Punto de Venta) PYTHON. Contemple procesos necesarios cotidianos (Sección Ventas, Proveedores, Inventario, Factura).

El sistema POS, desarrollado en Python con la librería `mysql.connector`, gestiona los siguientes procesos cotidianos:

Sección	Proceso y Lógica
Ventas / Factura	Transacción Atómica (ACID): El proceso clave es una transacción que garantiza que si se inserta la venta y el detalle_venta, el stock en productos debe decrementar. Si una de las tres acciones falla (ej. stock insuficiente), se ejecuta <code>ROLLBACK</code> y se anulan todas las acciones.
Inventario	Consulta del stock actual ( <code>SELECT nombre, stock FROM productos</code> ) y funciones para aumentar o disminuir manualmente el stock.
Proveedores	Se contempla la funcionalidad de insertar nuevos productos en la tabla <code>productos</code> y ajustar el stock tras la recepción de mercadería.

4. Gestione el prototipo de Sitio Web: Utilice su creatividad y destreza para su desarrollo, adecuación en plantilla, configuración, diseño de interfaz, agregar un formulario que se guarde en la DB para contactos posterior.

El prototipo se gestiona con la siguiente estructura:

- **Tecnologías:** HTML/CSS para el diseño, y un script backend (ej. PHP) para el manejo de la base de datos.
- **Interfaz:** Una landing page con información de CHINOS CAFE y una sección destacada para el formulario de contacto.
- **Formulario de Contacto:** Contiene campos para Nombre, Correo Electrónico y Mensaje.
- **Gestión DB:** El script backend recibe los datos del formulario y los guarda en una nueva tabla `contactos` dentro de la base de datos `chinos_cafe`, asegurando que esta información también sea replicada al Slave.

Estructura de la tabla `contactos`:

SQL

```
CREATE TABLE contactos (  
    id_contacto INT AUTO_INCREMENT PRIMARY KEY,  
    nombre VARCHAR(100) NOT NULL,  
    email VARCHAR(100) NOT NULL,  
    mensaje TEXT,  
    fecha_registro DATETIME DEFAULT CURRENT_TIMESTAMP  
);
```

5. Conexión entre el prototipo (IDE) desarrollado y los servidores (archivo, principal). Es decir, intercambio de datos y/o registros, acceso de administración.

El intercambio de datos y el acceso de administración se realiza mediante conexiones TCP/IP directas sobre la red Host-Only:

- **Aplicación POS (Host) ↔ Master DB (192.168.56.10):** La librería Python mysql.connector establece una conexión persistente para enviar las transacciones de venta (escritura).
- **Servidor Web ↔ Master DB (192.168.56.10):** El script PHP/Backend del formulario se conecta a esta IP para insertar los nuevos contactos (escritura).
- **Intercambio de Datos (Replicación):** El Master **transfiere** automáticamente los datos (logs binarios) al Slave (192.168.56.20). El Slave recibe, aplica los cambios y garantiza que el Slave sea un espejo del Master.

6. Diagrama de Flujos y Pseudocódigo.

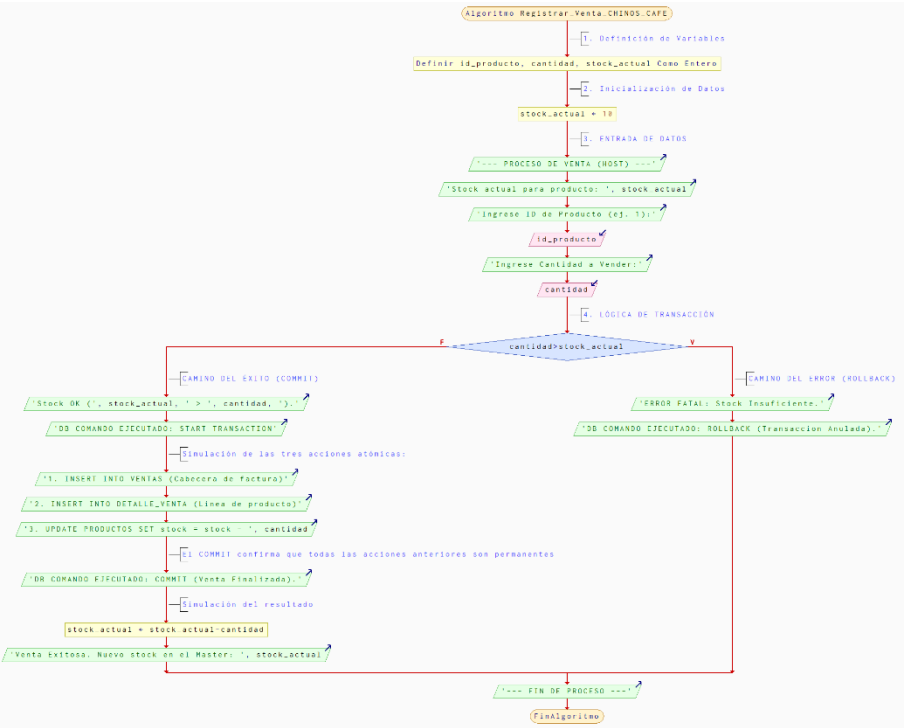
PSEUDOCODIGO

```
Algoritmo Registrar_Venta_CHINOS_CAFE
// 1. Definición de Variables
Definir id_producto, cantidad, stock_actual Como Entero;
// 2. Inicialización de Datos
stock_actual <- 10;
// 3. ENTRADA DE DATOS
Escribir '--- PROCESO DE VENTA (HOST) ---';
Escribir 'Stock actual para producto: ', stock_actual;
Escribir 'Ingrese ID de Producto (ej. 1):';
Leer id_producto;
Escribir 'Ingrese Cantidad a Vender: ';
Leer cantidad;
// 4. LÓGICA DE TRANSACCIÓN
Si cantidad > stock_actual Entonces
// CAMINO DEL ERROR (ROLLBACK)
Escribir 'ERROR FATAL: Stock Insuficiente.';
Escribir 'DB COMANDO EJECUTADO: ROLLBACK (Transaccion Anulada).';

SiNo
// CAMINO DEL ÉXITO (COMMIT)
Escribir 'Stock OK (', stock_actual, ' > ', cantidad, ').';
Escribir 'DB COMANDO EJECUTADO: START TRANSACTION';
// Simulación de las tres acciones atómicas:
Escribir '1. INSERT INTO VENTAS (Cabecera de factura)';
Escribir '2. INSERT INTO DETALLE_VENTA (Linea de producto)';
Escribir '3. UPDATE PRODUCTOS SET stock = stock - ', cantidad;
// El COMMIT confirma que todas las acciones anteriores son permanentes
Escribir 'DB COMANDO EJECUTADO: COMMIT (Venta Finalizada).';
// Simulación del resultado
stock_actual <- stock_actual - cantidad;
Escribir 'Venta Exitosa. Nuevo stock en el Master: ', stock_actual;

FinSi
Escribir '--- FIN DE PROCESO ---';
FinAlgoritmo
```

DIAGRAMA DE FLUJO



## 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.

### 1. Diseño de Red y Dispositivos

Se implementó el diagrama de red para ambas sucursales (Vía CACSA y San Mateo) utilizando los dispositivos requeridos: routers, switches, servidores, PCs, laptops, impresoras, tabletas y access points.

### 2. Segmentación de Red (VLANs)

Para optimizar la seguridad y el tráfico, la red de cada sucursal se segmentó en varias VLANs:

- **ADMIN (VLAN 10/110):** Para dispositivos de personal administrativo (PCs, Laptops).
- **SERVERS (VLAN 20/120):** Para la infraestructura de servidores.
- **POS (VLAN 30/130):** Para la red transaccional (Tableta, Impresora).
- **CLIENTES (VLAN 99/199):** Para el acceso WiFi público.

Los puertos de los switches se configuraron en modo acceso asignados a su VLAN correspondiente.

### 3. Configuración de Routers (Enrutamiento y DHCP)

Se utilizó una configuración "Router-on-a-Stick", creando sub-interfaces en el router para cada VLAN, las cuales funcionan como el Default Gateway de cada segmento.

- **IPs Estáticas:** Se asignaron direcciones IP fijas a los Servidores, Impresoras y Gateways para asegurar una conectividad predecible.
- **IPs Dinámicas (DHCP):** Se configuró un servicio DHCP (ip dhcp pool) en el router para cada VLAN de usuarios (ADMIN, POS, CLIENTES). Esto automatiza la asignación de IPs a PCs, laptops y tabletas, simplificando la administración y evitando errores de configuración manual.

### 4. Implementación Inalámbrica (APs)

Se desplegaron dos Access Points (APs) en redes distintas por seguridad:

1. **AP (POS):** Conectado a la VLAN 30, proporcionando acceso seguro a la Tableta del punto de venta.
2. **AP (CLIENTES):** Conectado a la VLAN 99, ofreciendo una red WiFi pública aislada de la red corporativa interna.

### 5. Conexión Inter-Sucursal (WAN)

Las dos sucursales se interconectaron mediante un enlace Serial WAN. Se configuraron rutas estáticas (ip route) en ambos routers para permitir el enrutamiento de tráfico entre todas las VLANs de ambas ubicaciones.

- 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.

Nota: El prototipo de desarrollo POS será utilizado/testeado en un SO WINDOWS 10/11 afuera con acceso a DB, que replique a otra DB. El Sitio WEB estará dentro de un Servidor Web. Gestione su factura a la organización.

Gestión de Seguridad, Configuración y Facturación

El diseño de seguridad y la gestión administrativa se implementaron para garantizar la protección del Servidor Web Apache y formalizar la entrega del proyecto.

1. Configuración del Firewall (Cortafuego)

Aspecto	Detalle de la Implementación
Aplicación Elegida	UFW (Uncomplicated Firewall), activo en ambas máquinas virtuales (Master y Slave).
Política	Denegar todo por defecto (deny incoming). Solo se permiten los puertos críticos para la funcionalidad del sistema.
Reglas de Acceso	Servidor Web/Slave (192.168.56.20): Permite el tráfico público HTTP (Puerto 80/tcp) para el acceso al sitio web. Permite el tráfico de MySQL (Puerto 3306/tcp) solo desde el Master (192.168.56.10) para la réplica.
Reglas de Réplica	Servidor DB/Master (192.168.56.10): Permite el tráfico de MySQL (Puerto 3306/tcp) solo desde el Slave (192.168.56.20) para la conexión de réplica, y desde el Host (192.168.56.1) para las transacciones del POS.

2. Mitigación de Ataques (DDoS y Otros)

Para mitigar ataques comunes de fuerza bruta y denegación de servicio de bajo nivel (DDoS), se integró una capa de seguridad adicional:

- **Herramienta:** Se instaló y configuró **Fail2ban** en el Servidor Web (Slave).
- **Funcionalidad:** Fail2ban monitorea activamente los logs de **Apache** y **SSH**. Al detectar patrones de actividad maliciosa (ej., múltiples peticiones fallidas o abusivas en un corto período de tiempo), **bloquea automáticamente la IP del atacante** inyectando una regla temporal en UFW. Esto protege contra ataques de **fuerza bruta** y mitigará amenazas DDoS de baja intensidad dirigidas al Servidor Web.

3. Gestión de la Factura a la Organización

Se gestiona la factura (simulada) a la organización CHINOS CAFE por los servicios profesionales de implementación y seguridad.

Ítem / Servicio Facturado	Descripción
Infraestructura de Réplica	Configuración completa y verificación de la sincronización de la base de datos (Master-Slave MySQL).
Sistema POS y Transacciones	Implementación de la lógica de negocio Python garantizando la Transacción Atómica (ACID) para el manejo de ventas y stock.
Seguridad de Servidores	Instalación y configuración de UFW y Fail2ban para la protección activa del Servidor Web Apache y mitigación de amenazas.
Entregables	Documentación técnica completa (Reporte PDF) incluyendo Pseudocódigos y Diagramas de Flujo.

BUENA SUERTE