

Universidad Tecnológica de Panamá
Facultad de Sistemas Computacionales
Asignatura: Desarrollo Lógico y Algoritmo
Examen Parcial2

Profesor: Napoleón Ibarra

Valor: 100 puntos

Estudiante: Rafael Rivera- Olmedo Alvarado

Cédula:

Fecha Inicio: 11/11/2025. Hora: 3:20 PM

Fecha Entrega: 18/11/2025. Hora 4:10 PM

Procedimiento:

- ✓ 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:

Crterios	Puntos (Mnimo=1, Mximo=5)	Porcentaje
Desarrollo	5	70 %
Responsabilidad	5	15 %
Sustentacin	5	15 %

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 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.

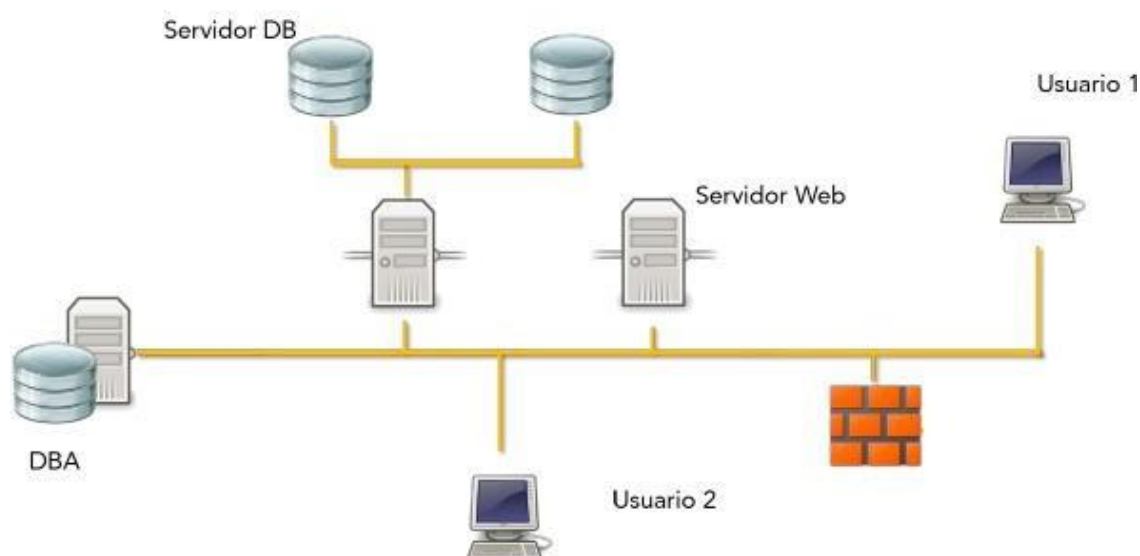


Figura 1. Esquema propuesto

¿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 su respuestas.
2. Adecuación, configuración de diagrama entidad-relación, registros, tablas, base de datos (DB). Identifique, gestione el Servidor MYSQL Server, Servidor de Archivos, Servidor Web.
3. Gestione un Sistema POS (Punto de Venta) PYTHON. Contemple procesos necesarios cotidianos (Sección Ventas, Proveedores, Inventario, Factura).
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.
5. Conexión entre el prototipo (IDE) desarrollado y los servidores (archivo, principal). Es decir intercambio de datos y/o registros, acceso de administración.
6. Diagrama de Flujos y Pseudocódigo.

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.

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.

BUENA SUERTE

Desarrollo:

Parte 1.

A. Arquitectura SGBD Recomendada:

Se recomienda la arquitectura Cliente/Servidor (Client/Server).

Porque, esta arquitectura es superior a la de "Servidor de Archivo" porque el Servidor MySQL (el motor de la base de datos) es el único que accede al almacenamiento físico. Los clientes (PCs, Servidores, y el POS en Python) solo envían comandos SQL y reciben resultados, no necesitan transferir archivos completos por la red.

Además, presenta ventajas como, la garantía de la Integridad de los Datos al gestionar la concurrencia y es fundamental para la Escalabilidad del sistema POS en las dos sucursales.

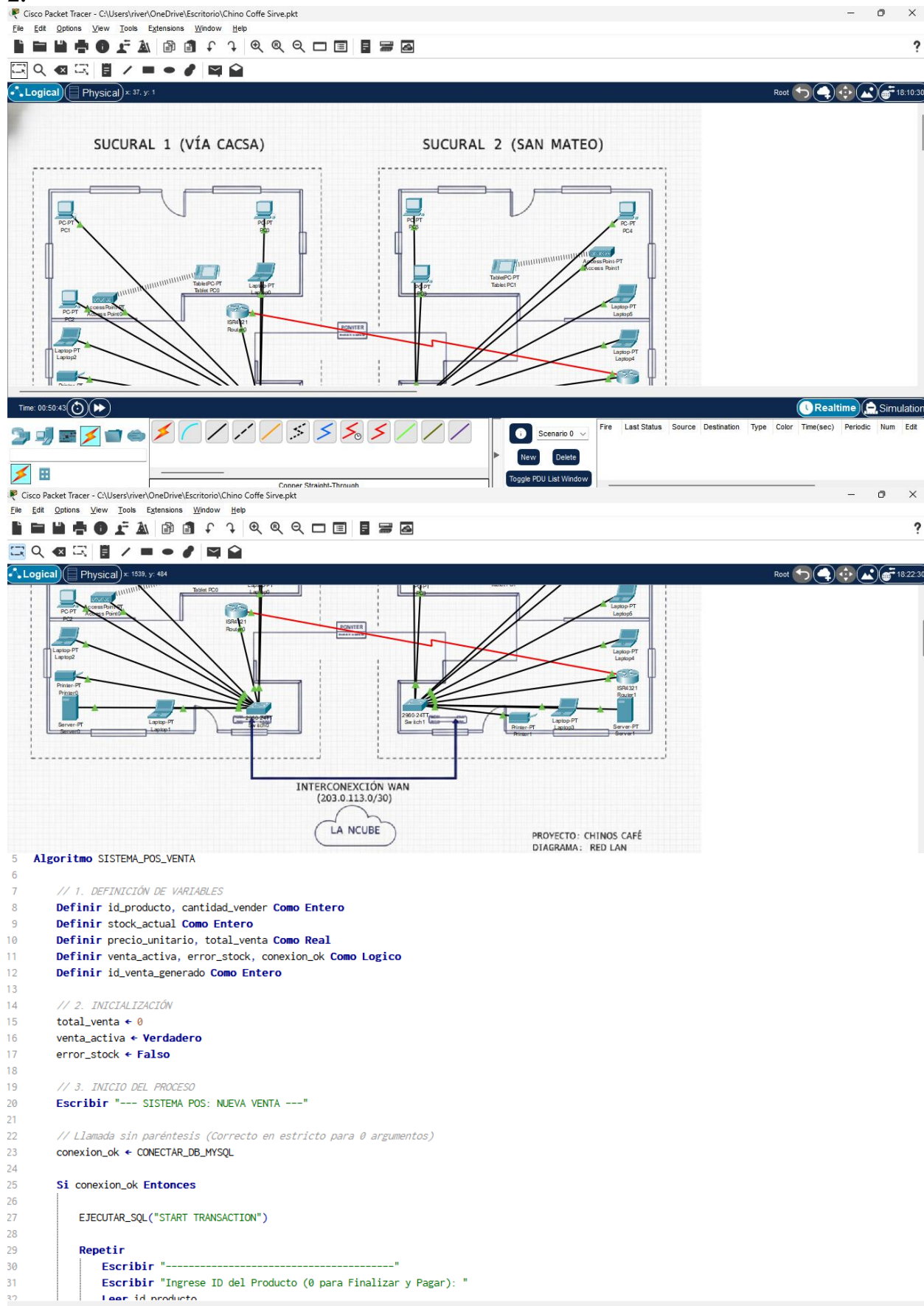
B. SGBD Multihilo

Sí, es necesario que el SGBD sea multihilo (multithreaded).

MySQL opera de manera multihilo. Cuando un cajero en la Sucursal 1 consulta el inventario y, simultáneamente, un administrador en la Sucursal 2 genera un reporte, el SGBD asigna un hilo (thread) separado a cada solicitud.

Beneficio Técnico: Esto permite el procesamiento concurrente y evita que las operaciones largas (como generar un reporte) bloqueen las operaciones críticas y rápidas (como registrar una venta), maximizando el uso de la CPU y mejorando la experiencia del usuario.

2.



```
34      Si id_producto ≠ 0 Entonces
35      |   Escribir "Ingrese Cantidad a vender: "
36      |   Leer cantidad_vender
37
38      |   // Llamadas con argumentos
39      |   stock_actual ← OBTENER_STOCK_DB(id_producto)
40      |   precio_unitario ← OBTENER_PRECIO_DB(id_producto)
41
42      |   // CORRECCIÓN: Usamos + y ConvertirATexto para evitar errores 233 y de Tipos
43      |   Escribir " -> Producto detectado. Stock: " + ConvertirATexto(stock_actual) + " | Precio: $" + ConvertirATexto(precio_unitario)
44
45      |   Si stock_actual ≥ cantidad_vender Entonces
46      |   |   total_venta ← total_venta + (cantidad_vender * precio_unitario)
47
48      |   |   // CORRECCIÓN: Concatenación estricta
49      |   |   Escribir " [SQL] UPDATE productos SET stock = stock - " + ConvertirATexto(cantidad_vender)
50      |   |   Escribir " -> Agregado al carrito. Subtotal actual: $" + ConvertirATexto(total_venta)
51      |   Sino
52      |   |   Escribir " [!!!] ERROR CRÍTICO: Stock insuficiente."
53      |   |   error_stock ← Verdadero
54      |   |   venta_activa ← Falso
55      |   FinSi
56
57      Sino
58      |   venta_activa ← Falso
59      FinSi
60
61 Hasta Que venta_activa = Falso
62
```