

Examen parcial 2

Delanis Tejeira 4-829-1643, Samir Salas 4-833-843

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 respuesta.

aparte de la arquitectura de servidor de archivo, se recomienda una arquitectura cliente-servidor (de 2 o 3 capas) y la arquitectura de microservicios. Las arquitecturas cliente-servidor proporcionan una estructura robusta y escalable para aplicaciones web y empresariales, mientras que la de microservicios ofrece mayor flexibilidad y agilidad al dividir la aplicación en servicios más pequeños, cada uno con su propia base de datos; en este caso se recomienda una arquitectura Cliente-Servidor con MySQL Server.

Porque

1. **Centralización de datos:** Ambos establecimientos acceden a la misma base de datos central
2. **Seguridad:** El servidor controla accesos y permisos
3. **Escalabilidad:** Fácil agregar más sucursales en el futuro
4. **Backup centralizado:** Un solo punto para realizar copias de seguridad
5. **Consistencia de datos:** Todas las sucursales trabajan con información actualizada

Sí es necesario por:

Los múltiples usuarios concurrentes: Ya que el sistema POS atenderá varios clientes simultáneamente

Operaciones paralelas: Mientras se procesan ventas, otros usuarios pueden:

Consultar inventario

Generar reportes

Actualizar datos de proveedores

Dos sucursales operando: Ambas tiendas accederán a la base de datos al mismo tiempo

Servidor web: El sitio web procesará consultas de clientes mientras el POS opera

Mejor rendimiento: El multihilo permite optimizar el uso de recursos del servidor

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

```
1 ● CREATE DATABASE Chinos_Cafe_db;
2 ● USE chinos_cafe_db;
3
4 ● CREATE TABLE sucursales (
5     id_sucursal INT AUTO_INCREMENT PRIMARY KEY,
6     nombre VARCHAR(100) NOT NULL,
7     direccion TEXT,
8     telefono VARCHAR(20),
9     estado ENUM('activa', 'inactiva') DEFAULT 'activa'
10 );
11
12 ● CREATE TABLE usuarios (
13     id_usuario INT AUTO_INCREMENT PRIMARY KEY,
14     usuario VARCHAR(50) UNIQUE NOT NULL,
15     password VARCHAR(255) NOT NULL,
16     nombre VARCHAR(100),
17     rol ENUM('admin', 'cajero', 'gerente'),
18     id_sucursal INT,
19     estado ENUM('activo', 'inactivo') DEFAULT 'activo',
20     FOREIGN KEY (id_sucursal) REFERENCES sucursales(id_sucursal)
21 );
22
23 ● CREATE TABLE configuracion_sucursal (
24     id_config INT AUTO_INCREMENT PRIMARY KEY,
25     id_sucursal INT,
26     nombre_impresora VARCHAR(100),
27     direccion_ip VARCHAR(15),
28     telefono VARCHAR(20),
29     FOREIGN KEY (id_sucursal) REFERENCES sucursales(id_sucursal)
30 );
31
32 ● CREATE TABLE categorias (
33     id_categoria INT AUTO_INCREMENT PRIMARY KEY,
34     nombre_categoria VARCHAR(100) NOT NULL,
35     descripcion TEXT
36 );
```

```

38 ● ○ CREATE TABLE proveedores (
39     id_proveedor INT AUTO_INCREMENT PRIMARY KEY,
40     nombre VARCHAR(200) NOT NULL,
41     contacto VARCHAR(100),
42     telefono VARCHAR(20),
43     email VARCHAR(100)
44 );
45
46 ● ○ CREATE TABLE productos (
47     id_producto INT AUTO_INCREMENT PRIMARY KEY,
48     nombre VARCHAR(200) NOT NULL,
49     descripcion TEXT,
50     precio_compra DECIMAL(10,2),
51     precio_venta DECIMAL(10,2),
52     stock INT DEFAULT 0,
53     id_categoria INT,
54     id_proveedor INT,
55     FOREIGN KEY (id_categoria) REFERENCES categorias(id_categoria),
56     FOREIGN KEY (id_proveedor) REFERENCES proveedores(id_proveedor)
57 );
58
59 ● ○ CREATE TABLE clientes (
60     id_cliente INT AUTO_INCREMENT PRIMARY KEY,
61     nombre VARCHAR(200) NOT NULL,
62     email VARCHAR(100),
63     telefono VARCHAR(20),
64     fecha_registro DATE DEFAULT (CURDATE())
65 );
66
67 ● ○ CREATE TABLE ventas (
68     id_venta INT AUTO_INCREMENT PRIMARY KEY,
69     fecha_venta DATETIME DEFAULT CURRENT_TIMESTAMP,
70     total DECIMAL(10,2),
71     id_cliente INT,
72     id_sucursal INT,

```

```

72         id_sucursal INT,
73         FOREIGN KEY (id_cliente) REFERENCES clientes(id_cliente),
74         FOREIGN KEY (id_sucursal) REFERENCES sucursales(id_sucursal)
75     );
76
77 ● ⊖ CREATE TABLE detalle_venta (
78     id_detalle INT AUTO_INCREMENT PRIMARY KEY,
79     id_venta INT,
80     id_producto INT,
81     cantidad INT,
82     precio_unitario DECIMAL(10,2),
83     subtotal DECIMAL(10,2),
84     FOREIGN KEY (id_venta) REFERENCES ventas(id_venta),
85     FOREIGN KEY (id_producto) REFERENCES productos(id_producto)
86 );
87
88 ● ⊖ CREATE TABLE inventario (
89     id_inventario INT AUTO_INCREMENT PRIMARY KEY,
90     id_producto INT,
91     id_sucursal INT,
92     stock_actual INT,
93     FOREIGN KEY (id_producto) REFERENCES productos(id_producto),
94     FOREIGN KEY (id_sucursal) REFERENCES sucursales(id_sucursal)
95 );
96
97 ● ⊖ CREATE TABLE contactos_web (
98     id_contacto INT AUTO_INCREMENT PRIMARY KEY,
99     nombre VARCHAR(200) NOT NULL,
100     email VARCHAR(100),
101     telefono VARCHAR(20),
102     mensaje TEXT,
103     fecha_contacto DATETIME DEFAULT CURRENT_TIMESTAMP
104 );

```

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

C:\Users\USUARIO\Desktop\warframe\Nueva carpeta\gestiionn.py > gestiionn.py > Database

```
1 import mysql.connector
2 from mysql.connector import Error
3 import tkinter as tk
4 from tkinter import ttk, messagebox, simpledialog
5 from datetime import datetime
6 import os
7
8 class Database:
9     def __init__(self):
10         self.connection = None
11         self.connect()
12
13     def connect(self):
14         try:
15             self.connection = mysql.connector.connect(
16                 host='localhost',
17                 user='Connection gestion',
18                 password='1234,ssa2334',
19                 database='chinos_cafe_db',
20                 port=3306
21             )
22             print("✅ Conexión a MySQL exitosa")
23         except Error as e:
24             print(f"❌ Error de conexión: {e}")
25
26 class POSSystem:
27     def __init__(self, root):
28         self.root = root
29         self.root.title("CHINOS CAFE - Sistema POS Completo")
30         self.root.geometry("1200x800")
31
32         self.db = Database()
33         if not self.db.connection:
34             messagebox.showerror("Error", "No se pudo conectar a la base de datos")
35             return
36
37         self.connection = self.db.connection
```

>< ∞ Agents ⊗ 0 ▲ 0

