



**CARRERA DE TECNOLOGÍA DE LA INFORMACIÓN  
INSTRUMENTO DE EVALUACIÓN  
DE SISTEMAS DISTRIBUIDOS 2025-2**

**Docente:** Ing. Willian Zamora Mero., PhD.

**Paralelo A - B**

**Fecha:** 10 de noviembre de 2025.

**Grupo (ESCRIBIR APELLIDOS Y NOMBRES ORDENADOS A->Z):**

**CALIFICACIÓN EXAMEN: 6 PTS GRUPAL + 4 PTS TEORÍA (INDIVIDUAL)**

**EVALUACIÓN TEÓRICA SEGÚN LOS TEMAS TRATADOS Y REALIZADOS EN CLASES 4 PTS.**

**RESOLVER EL SIGUIENTE PROYECTO 6 PTS:**

Usando la práctica de sockets se le solicita que realice un programa que cumpla con la arquitectura que se muestra en la figura 1. La idea general es simular una solución cliente – servidor que retorne datos a través de una base de datos. En particular, el sistema cliente (múltiples clientes) consiste en solicitar datos mediante un mensaje, este puede ser la cédula de identidad. Por el otro lado, el socket server recibe la petición y él se encarga de procesar, y consultar a una base de datos devolviendo la información requerida. La información por devolver es: Apellidos, Nombres, y un Saldo. Para esto el servidor debe tener acceso a una base de datos donde se encuentre el registro de dicha persona. La base de datos no necesariamente puede estar alojada en el mismo computador. Lo mismo ocurre con el cliente y servidor. Adicionalmente, se les solicita que la solución permita actualizar, agregar saldo al cliente, y disminuir saldo al cliente, según el número de cedula. La solución propuesta puede ser desarrollada usando bases de datos, MongoDB, SQL Server, MariaDB, PostGreSQL y otras. El lenguaje de programación es independiente.

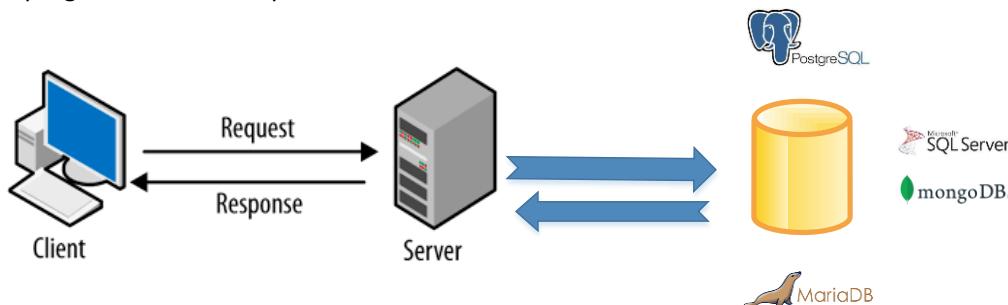


Figura 1.- Arquitectura propuesta usando socket.

Elaborado por	Revisado y aprobado en área	Recepción Comisión Académica
Ing. Willian Zamora Mero, PhD Docente	Ing. Patricia Quiroz, PhD. Colectivo disciplinario Software	Ing. Panchana Flores Mg. Comisión Académica



**RÚBRICA DE CALIFICACIÓN PARA EL PROYECTO**

	INCOMPLETO	REGULAR	PARCIAL	COMPLETO	CALIFICACIÓN
<b>DISEÑO DE LA SOLUCIÓN</b>	La solución propuesta he mostrada en grupo y su funcionalidad cumple en un 25% de lo esperado	La solución propuesta he mostrada por el grupo y su funcionalidad cumple en un 50% de lo esperado	La solución propuesta he mostrada en grupo y su funcionalidad cumple en un 75% de lo esperado. Además, muestra ciertos códigos aplicados en clases.	La solución propuesta cumple en un 100% lo esperado. Además, los recursos que consume se encuentran alojados en computadores distintos y en su codificación usa los conceptos aplicados en los talleres prácticos	
	<b>1. pts.</b>	<b>2 pts.</b>	<b>4 pts.</b>	<b>6 pts.</b>	
	ÉXITOS TOTALES				

**FORMATO DE ENTREGA**

**SOLUCIÓN SE REVISARÁ EL DÍA DEL EXAMEN.**



### **Relación entre sockets y sistemas concurrentes, hilos y bloqueos:**

Si bien el código en el server no implementa directamente la concurrencia explícita o el uso de hilos, sí se puede observar la posibilidad de aplicar estos conceptos para mejorar el rendimiento y la escalabilidad del servidor.

#### **1. Sistemas concurrentes:**

Un sistema concurrente es aquel que permite que múltiples tareas se ejecuten simultáneamente, ya sea en paralelo o de forma entrelazada. En el contexto del servidor de sockets, la concurrencia podría ser útil para manejar múltiples conexiones de clientes de manera eficiente. Por ejemplo, si se implementara un sistema concurrente, el servidor podría atender a varios clientes al mismo tiempo, en lugar de atender a un solo cliente a la vez.

#### **2. Hilos:**

Los hilos son una unidad de ejecución dentro de un proceso que permite la ejecución simultánea de múltiples tareas dentro del mismo proceso. En el ejemplo del servidor de sockets, la implementación de hilos podría permitir atender a múltiples clientes de manera más eficiente. Por ejemplo, cada cliente conectado podría ser manejado por un hilo separado, lo que permitiría al servidor procesar las solicitudes de los clientes de forma más fluida.

#### **3. Bloqueos:**

Los bloqueos son mecanismos de sincronización que permiten controlar el acceso a recursos compartidos entre diferentes hilos o procesos. En el ejemplo del servidor de sockets, los bloqueos podrían ser necesarios para evitar conflictos de acceso a la variable data cuando se reciben y envían datos a múltiples clientes.

### **Identificar métodos que utilizan estas características:**

Para identificar métodos en el código que podrían utilizar estas características, se debe analizar el código en detalle y considerar los siguientes aspectos:

- **Puntos de acceso a recursos compartidos:** Identificar las variables o datos que podrían ser accedidos por múltiples clientes o hilos al mismo tiempo.
- **Operaciones de larga duración:** Identificar operaciones que podrían bloquear el proceso principal del servidor durante un tiempo considerable, como la recepción o envío de grandes cantidades de datos.
- **Posibles cuellos de botella:** Identificar puntos en el código donde el rendimiento podría verse afectado por la falta de concurrencia o la ausencia de mecanismos de sincronización.

Si se identifican estos aspectos en el código, es probable que existan oportunidades para mejorar el rendimiento y la escalabilidad del servidor mediante la implementación de sistemas concurrentes, hilos y bloqueos.

### **Ejemplo de implementación de concurrencia:**

Una forma de implementar concurrencia en el ejemplo del servidor de sockets sería utilizar un módulo de concurrencia como `threading` en Python. Con este módulo, se podrían crear hilos separados para atender a cada cliente conectado. El código principal del servidor se encargaría de aceptar nuevas conexiones y crear un nuevo hilo para cada cliente nuevo. Cada hilo se encargaría de recibir y enviar datos del cliente correspondiente.