

# Sistemas de Bases de Datos Cliente/Servidor

M.C. Carlos Rojas Sánchez<sup>1</sup>

<sup>1</sup>Licenciatura en Informática  
Universidad del Mar - Puerto Escodido

[ Bases de Datos Distribuidas ]

# Objetivo

Aplicar los principios y características fundamentales de los sistemas de bases de datos distribuidas, considerando los aspectos básicos de diseño distribuido, procesamiento de consultas y transacciones distribuidas.

# Sistemas de Bases de Datos Distribuidas

- Unión: Sistemas de Bases de Datos y Redes de Computadoras
- Buscando “Integración”
- Evitando “Centralización”
- Procesamiento distribuido
  - Un numero de elementos de procesamiento autonomo (no necesariamente homogeneos) que estan interconectados por medio de una red y que colaboran en la realización de tareas.
  - Divide y venceras

# 1 Características de un sistema cliente/servidor

# Datos

- Surge en los 80's
- Idea: “dividir el trabajo y reducir la complejidad”
- Comunicación TCP/IP
- Se tiene un equipo (Cliente) que necesita un servicio de otro equipo (Servidor)
- Sirviente

# Características de un Sistema Cliente/Servidor

## Cliente

Normalmente maneja todas las funciones relacionadas con la manipulación y despliegue de datos

- Interactuar con el usuario
- Hacer validaciones locales
- Generar requerimientos para la Base de Datos
- Recibir resultados
- Dar formato a los resultados

# Características de un Sistema Cliente/Servidor

## Servidor

Es el proceso encargado de atender a múltiples clientes que hacen peticiones de algún recurso administrado por él

- Aceptar los requerimientos de bases de datos que hacen los clientes
- Procesar requerimientos de bases de datos
- Procesar la lógica de la aplicación y realizar validaciones a nivel de base de datos

# Bases de Datos en el Modelo Cliente Servidor

- Dos capas

- La aplicación del usuario (Cliente)
- La base de datos (Servidor)



## 2 Tipos de servidores

# Tipos de servidores

- dedicado: son aquellos que le dedican toda su potencia a administrar los recursos de la red, es decir, a atender las solicitudes de procesamiento de los clientes.
- no dedicado: son aquellos que no dedican toda su potencia a los clientes, sino también pueden jugar el rol de estaciones de trabajo al procesar solicitudes de un usuario local.

- 3 Arquitecturas en capas
  - Arquitectura de 2 capas
  - Arquitectura de 3 capas
  - Ventajas e inconvenientes

# Arquitectura de 2 capas

El cliente se comunica directamente con un servidor de bases de datos. La aplicación o lógica de negocio bien reside en el cliente, o en el servidor de base de datos en la forma de procedimientos almacenados

# Arquitectura de 3 capas

Añade una capa intermedia (middle tier). En la arquitectura de tres capas (en general, en la arquitectura multicapa) el cliente implementa la lógica de presentación, el servidor(es) de aplicación implementan la lógica de negocio y los datos residen en uno (o varios) servidor(es) de bases de datos.

# Arquitectura de 3 capas

- Un componente **front-end** que es el responsable de proporcionar la lógica de presentación.
- Un componente **back-end** que proporciona acceso a servicios dedicados, tales como un servidor de bases de datos.
- Un componente que hace las funciones de capa intermediaria (**middle tier**) que permite a los usuarios compartir y controlar la lógica de negocio mediante su aislamiento de la aplicación real.

# Arquitectura de 3 capas

- El sistema cliente interactúa con la capa intermedia vía un protocolo estándar como HTTP o RPC
- La capa intermedia interactúa con el servidor de datos (back-end) mediante protocolos de bases de datos estándar tales como SQL, ODBC y JDBC
- La arquitectura de 3 capas se puede extender a  $n$  capas cuando la capa intermedia soporta conexiones a diferentes tipos de servicios (no sólo servicios de almacenamiento de datos), integrándolos y acoplándolos al cliente y entre ellos

# Tarea 1

- Ejemplo de una arquitectura multicapa



# Ventajas

- Distribución de aplicaciones: concurrencia de procesos
- Permite emplear entornos heterogéneos y multivendedor:  
la plataforma no tiene por qué ser la misma (se comunican a través de un API conocido)
- Portabilidad
- Acceso independiente del lugar físico: empleo de un cliente “ligero” con requisitos mínimos de instalación

# Ventajas



## ■ Escalabilidad

- La capacidad del sistema informático de cambiar su tamaño o configuración para adaptarse a las circunstancias cambiantes
- Un sistema es escalable si al incrementar los recursos empleados (por ejemplo, nº procesadores), el rendimiento crece proporcionalmente
- Escalabilidad horizontal (scale out)
- Escalabilidad vertical (scale up)

# Inconvenientes

- Aumenta la comunicación (congestión tráfico de red)
- Falta de robustez (caída del servidor)

# Bibliografía I

-  M. Tamer Özsu and Patrick Valduriez.  
*Principles of Distributed Database Systems.*  
Springer, 2011.
-  Abraham Silberschatz, Henry F. Korth and S. Sudarshan.  
*Fundamentos de Bases de Datos.*  
McGRAW-HILL, 2002.