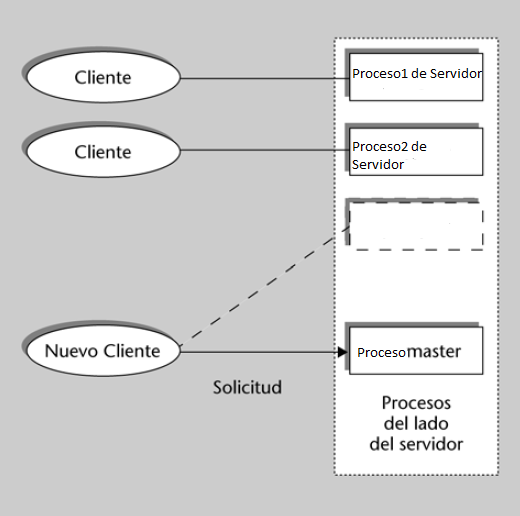
**Arquitectura de los Datos**

**Daniela, David, Efrain, Ingrid**



**1 Visión general**

La arquitectura a Usar se basa en una Arquitectura cliente servidor0. El programa servidor se llama pservidor y el cliente pcliente (cliente en modo texto).

Un proceso servidor puede atender exclusivamente a un solo cliente; es decir, hacen falta tantos procesos servidor pservidor como clientes haya. El proceso pmaster es el encargado de ejecutar un nuevo servidor para cada cliente que solicite una conexión.

El equipo anfitrión (host) almacena un conjunto de bases de datos de nuestro servidor. En un sitio se ejecuta solamente un proceso pmaster y múltiples procesos pcliente. Los clientes pueden ejecutarse en el mismo sitio o en equipos remotos conectados por TCP/IP.

Es posible restringir el acceso a usuarios o a direcciones IP modificando las opciones del archivo ips.conf, que se encuentra c:/misgbd/ips.conf. tiene mecanismos de autenticación que usará para verificar las credenciales de los usuarios.

**2 Almacenamiento y organización de los datos**

La data siempre se va a guardar en “disco” (esto puede no ser literalmente un HD).

Esto genera un intenso trabajo de I/O, cuando leemos la data la sacamos del “disco” para pasarla a la RAM, cuando escribimos la bajamos de la RAM al “disco”.

Los tamaños de sectores serán de 4kB

**3 Los Índices**

Para los índices **map de maps STL**, inplementados en c++ debido a que son más rápidos que un Btree.

**4 Query**

Los Query son funciones en c++.

**5 Concurrencia y gestión De bloqueos**

Cuando 2 transacciones trabajan sobre el mismo objeto y al menos uno de ellos incluye operaciones de escritura entonces se produce un “conflicto”.

Cuando 2 transacciones hacen exactamente lo mismo puede que sean “serializadas” para optimizar el acceso a la data

Antes de que una transacción pueda ejecutar un R/W sobre un objeto en la db debe obtener un “bloqueo”.

Este bloqueo puede ser **Compartido** (Share) o **Exclusivo** (Exclusive), estos son administrador por un “Lock Manager”**implementado en c++**.