**UNIVERSIDAD DEL SURESTE**
**SAN CRISTOBAL DE LAS CASAS CHIAPAS**

**MATERIA:** SISTEMAS OPERATIVOS

**TAREA:** MAPA CONCEPTUAL

**NOMBRE DEL ALUMNO:** BALDOMERO SANTIZ GOMEZ

**SEMESTRE:** 3ER. CUATRIMESTRE

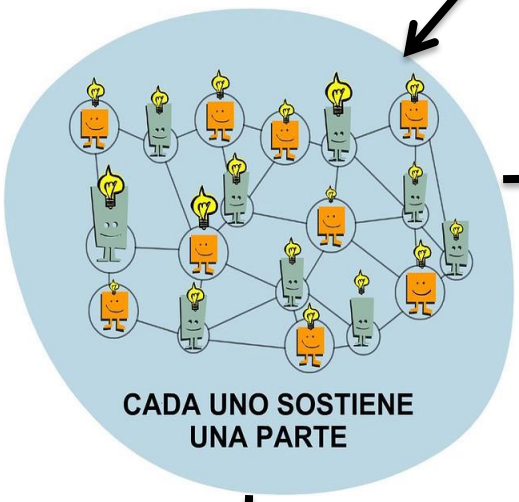
**GRUPO:** “A”

**CARRERA:** ING. EN SISTEMAS COMPUTACIONALES

**TEACHER:** ING. Eduardo Genner Escalante Cruz

**FECHA DE ENTREGA:** 17/06/2020

**SISTEMA DISTRIBUIDO**




Un sistema distribuido es un conjunto de ordenadores que trabajan juntos de forma coordinada, a través del intercambio de mensajes, para conseguir un objetivo.

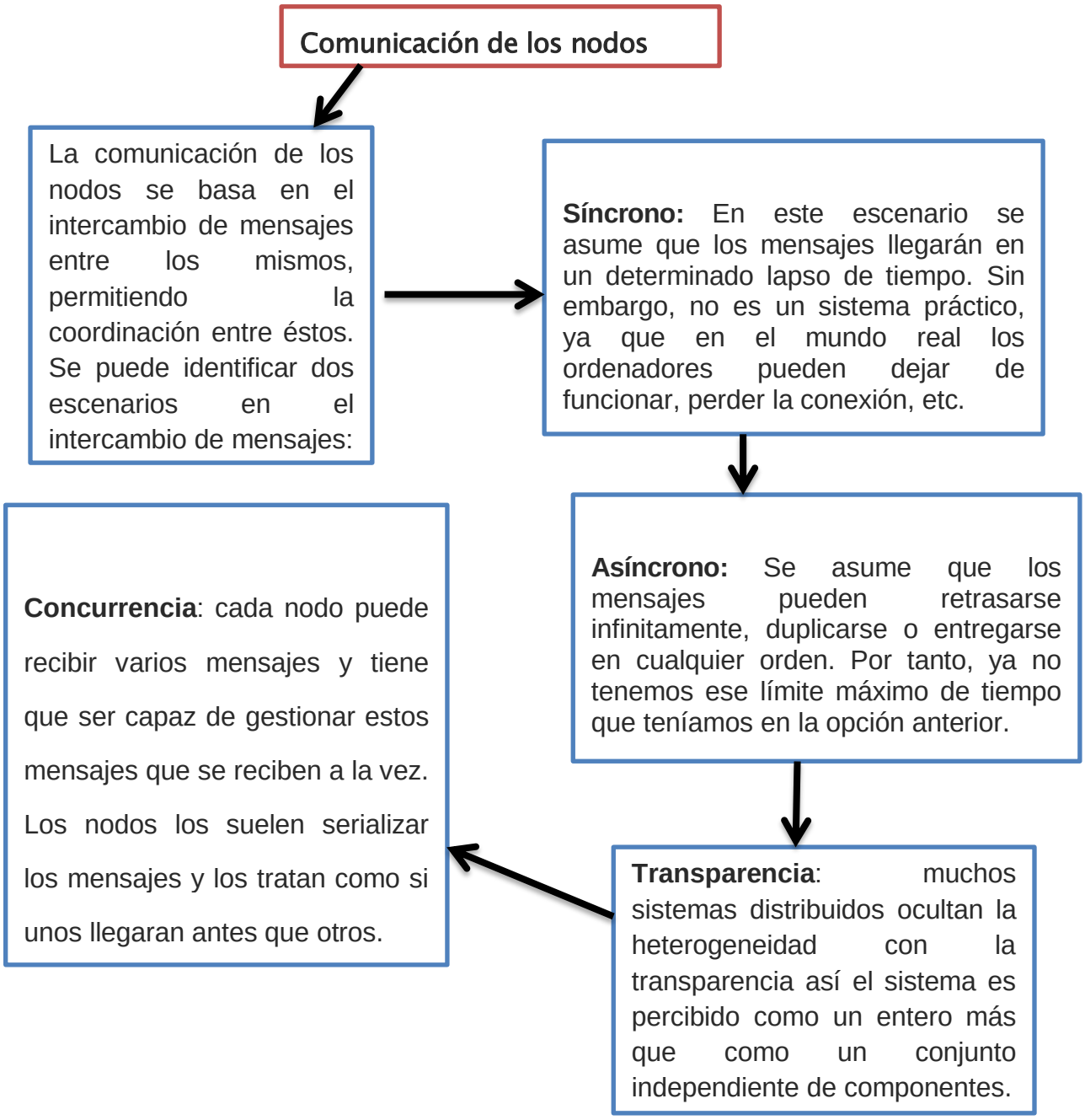
En dicho sistema, el estado y los programas se guardan en múltiples ordenadores. A pesar de que los procesos que tienen lugar están separados entre los diferentes participantes, para el usuario parece que está trabajando con un único ordenador.

**Caracteriza un sistema distribuido**

**Concurrencia:** Cada ordenador de la red ejecuta eventos de forma independiente y al mismo tiempo que los otros.

**Falta de un reloj global:** Para que el sistema distribuido funcione correctamente, necesitamos encontrar alguna forma de ordenar los eventos que tienen lugar.

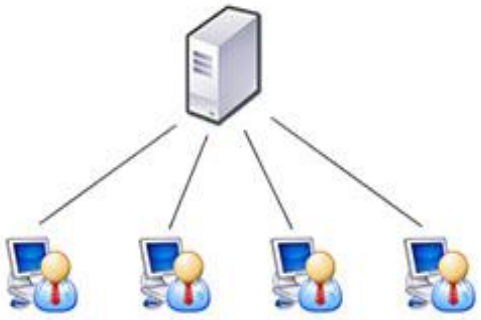
**Fallos independientes de los componentes:** Estos fallos los podemos clasificar en: **Crash**: los componentes dejan de funcionar. **Omisión**: un componente envía un mensaje, pero no es recibido por los demás. **Bizantino**: los componentes de la red empiezan a actuar de forma arbitraria.


Comunicación de los nodos
La comunicación de los nodos se basa en el intercambio de mensajes entre los mismos, permitiendo la coordinación entre éstos. Se puede identificar dos escenarios en el intercambio de mensajes:
Síncrono: En este escenario se asume que los mensajes llegarán en un determinado lapso de tiempo. Sin embargo, no es un sistema práctico, ya que en el mundo real los ordenadores pueden dejar de funcionar, perder la conexión, etc.
Asíncrono: Se asume que los mensajes pueden retrasarse infinitamente, duplicarse o entregarse en cualquier orden. Por tanto, ya no tenemos ese límite máximo de tiempo que teníamos en la opción anterior.
Transparencia: muchos sistemas distribuidos ocultan la heterogeneidad con la transparencia así el sistema es percibido como un entero más que como un conjunto independiente de componentes.
Concurrencia: cada nodo puede recibir varios mensajes y tiene que ser capaz de gestionar estos mensajes que se reciben a la vez. Los nodos los suelen serializar los mensajes y los tratan como si unos llegaran antes que otros.

# CLIENTE SERVIDOR

En dicho contexto, se llama **cliente** al dispositivo que requiere ciertos servicios a un servidor. La idea de **servidor**, por su parte, alude al equipo que brinda servicios a las computadoras (ordenadores) que se hallan conectadas con él mediante una red.

El concepto de **cliente servidor**, o **cliente-servidor**, refiere por lo tanto a un **modelo de comunicación** que vincula a varios dispositivos informáticos a través de una **red**. El cliente, en este marco, **realiza peticiones de servicios** al servidor, que se encarga de satisfacer dichos requerimientos.



Una de las ventajas menos aparentes de la organización en servidores y clientes es que la capacidad de procesamiento y memoria de estos últimos no debe ser tan grande como la de los primeros, lo cual beneficia al consumidor final permitiéndole usar un equipo relativamente.

CLÚSTER

La clusterización (clustering) divide una base de datos en grupos diferentes; la meta principal de realizar el proceso de clusterización es encontrar grupos que son diferentes de los otros, y que sus miembros sean similares entre sí.

El concepto de “Clúster” fue popularizado por el economista Michel Porter el año 1990, en su libro *The Competitive Advantage of Nations*. refiriédose a “un grupo geográficamente próximo de compañías interconectadas e instituciones asociadas, en un campo particular, vinculadas por características comunes y complementarias, incluyendo empresas de productos finales o servicios, proveedores, instituciones financieras y empresas de industrias conexas”.

**TIPOS DE CLUSTEHER**

**Cluster integrados verticalmente**. En ellos, las industrias se enlazan a través de la cadena de suministros

**Cluster integrados horizontalmente**. En que las industrias comparten una base común de conocimientos, un mercado similar por sus productos y utilización de tecnologías, de recursos humanos y/o recursos materiales similares.

Porter dijo que***” la competitividad de una región se basa en la competitividad de sus industrias que a su vez es mejorada si una industria está sumergida en una profunda red”.*** Es decir que los clúster son concentraciones geográficas de empresas e instituciones interconectadas que actúan en determinado campo.

RMI

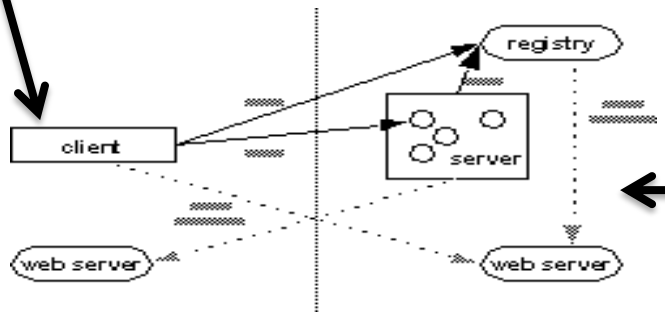
Las aplicaciones RMI normalmente comprenden dos programas separados: un servidor y un cliente. Una aplicación servidor típica crea un montón de objetos remotos, hace accesibles unas referencias a dichos objetos remotos, y espera a que los clientes llamen a estos métodos u objetos remotos.

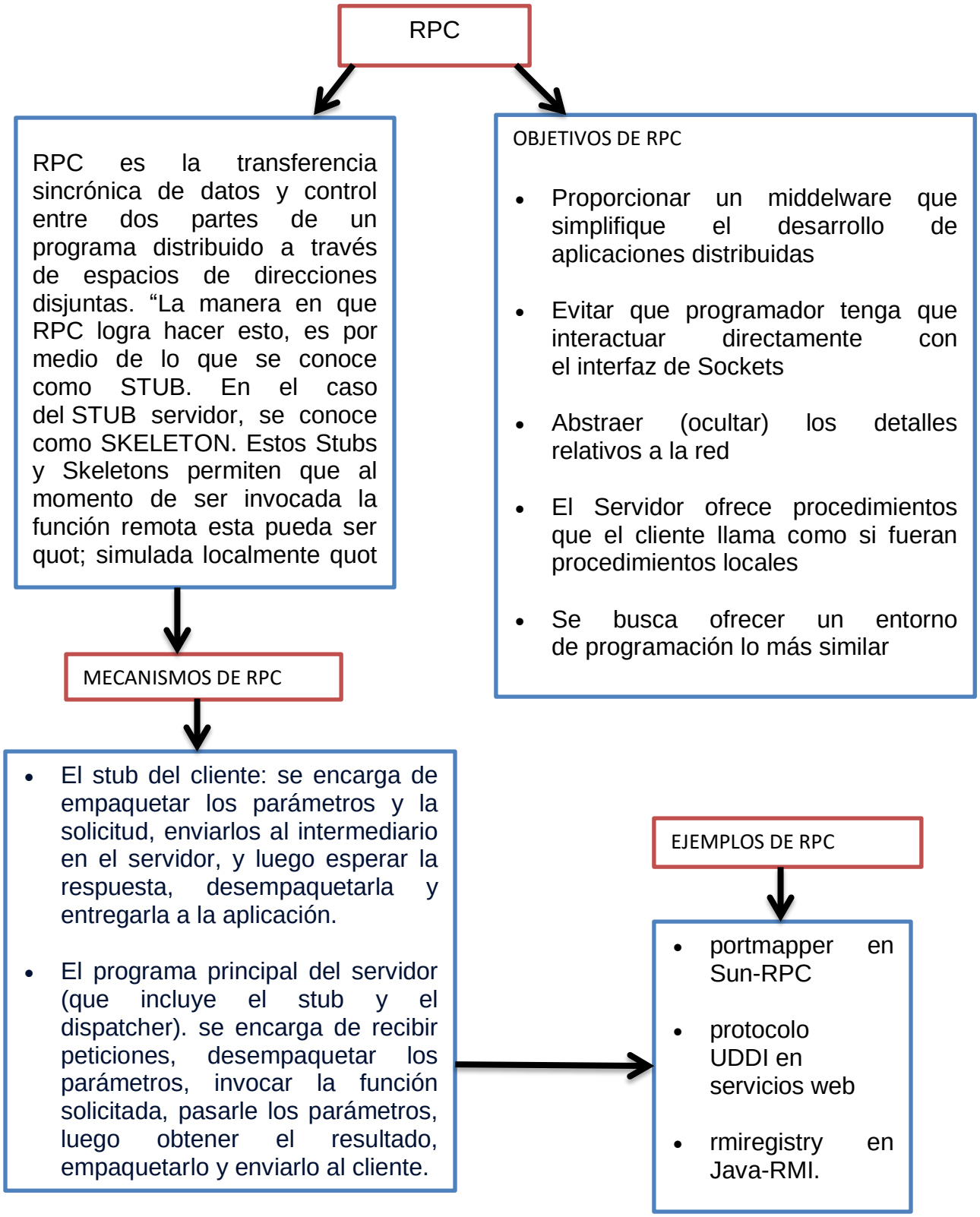
Una aplicación cliente típica obtiene una referencia remota de uno o más objetos remotos en el servidor y llama a sus métodos. RMI proporciona el mecanismo por el que se comunican y se pasan información del cliente al servidor y viceversa. Cuando es una aplicación algunas veces nos referimos a ella como **Aplicación de Objetos Distribuidos**.

**Localizar Objetos Remotos**

Las aplicaciones pueden utilizar uno de los dos mecanismos para obtener referencias a objetos remotos. Puede registrar sus objetos remotos con la facilidad de nombrado de RMI **rmiregistry**. O puede pasar y devolver referencias de objetos remotos como parte de su operación normal.

El cliente busca el objeto remoto por su nombre en el registro del servidor y luego llama a un método. Esta ilustración también muestra que el sistema RMI utiliza una servidor Web existente para cargar los bytecodes de la clase Java, desde el servidor al cliente y desde el cliente al servidor, para los objetos que necesita.




RPC
RPC es la transferencia sincrónica de datos y control entre dos partes de un programa distribuido a través de espacios de direcciones disjuntas. “La manera en que RPC logra hacer esto, es por medio de lo que se conoce como STUB. En el caso del STUB servidor, se conoce como SKELETON. Estos Stubs y Skeletons permiten que al momento de ser invocada la función remota esta pueda ser quot; simulada localmente quot
OBJETIVOS DE RPC
Proporcionar un middelware que simplifique el desarrollo de aplicaciones distribuidas
Evitar que programador tenga que interactuar directamente con el interfaz de Sockets
Abstraer (ocultar) los detalles relativos a la red
El Servidor ofrece procedimientos que el cliente llama como si fueran procedimientos locales
Se busca ofrecer un entorno de programación lo más similar
MECANISMOS DE RPC
El stub del cliente: se encarga de empaquetar los parámetros y la solicitud, enviarlos al intermediario en el servidor, y luego esperar la respuesta, desempaquetarla y entregarla a la aplicación.
El programa principal del servidor (que incluye el stub y el dispatcher). se encarga de recibir peticiones, desempaquetar los parámetros, invocar la función solicitada, pasarle los parámetros, luego obtener el resultado, empaquetarlo y enviarlo al cliente.
EJEMPLOS DE RPC
portmapper en Sun-RPC
protocolo UDDI en servicios web
rmiregistry en Java-RMI.