**CUESTIONARIO II**

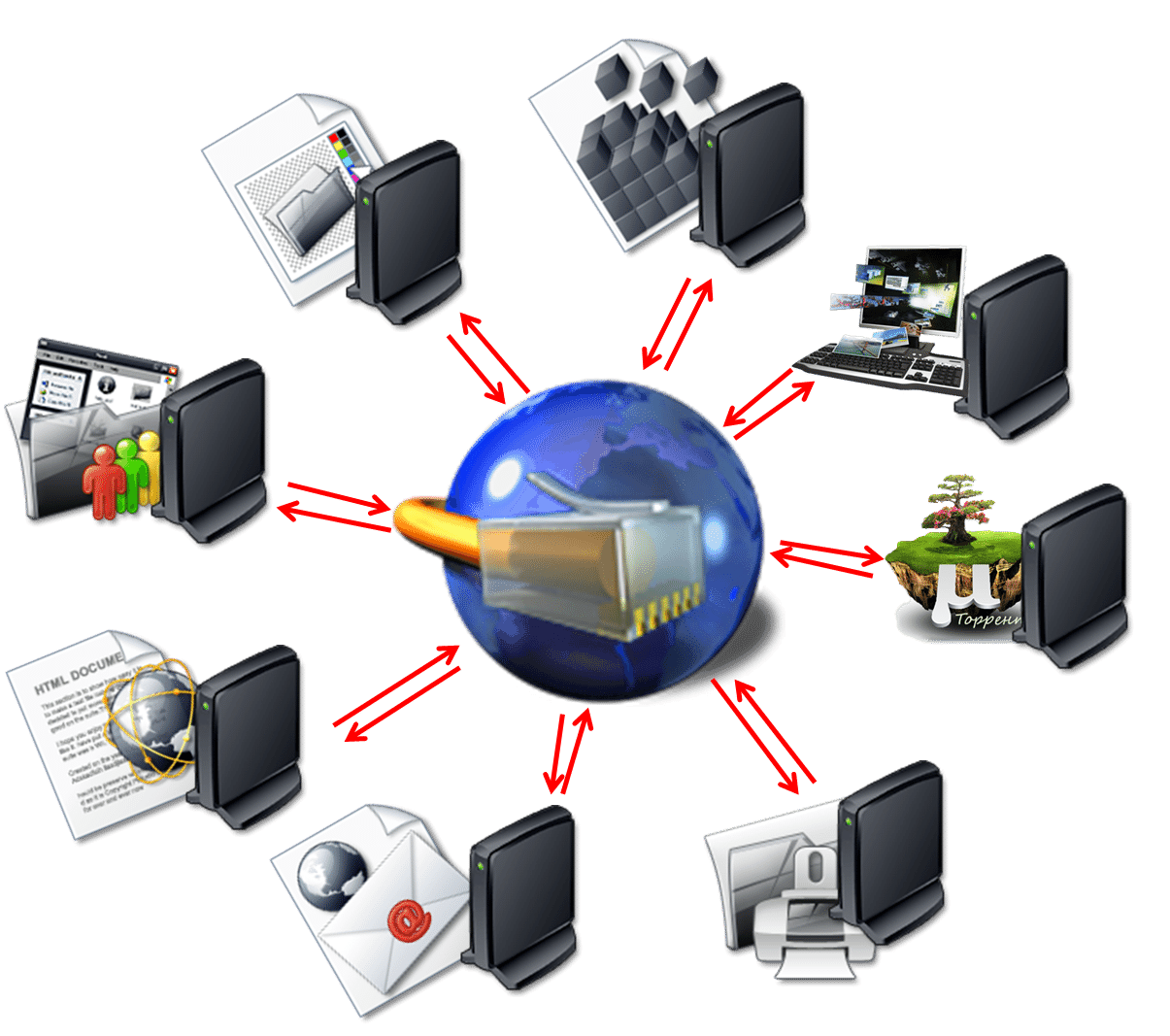
**DATOS DEL ALUMNO**

|  |
| --- |
| **NOMBRE**: Cuevas Olvera Ian Axel  **GRUPO**: 4CM2 |

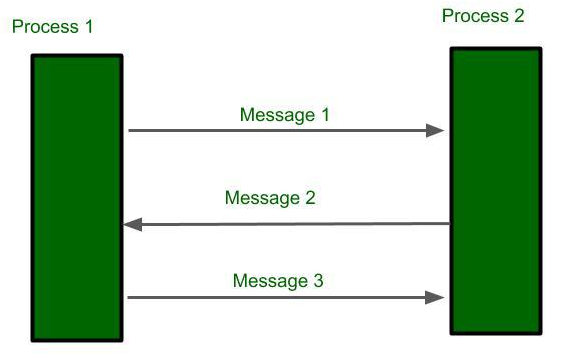
|  |
| --- |
| IMPORTANTE  Las siguientes preguntas no son deductivas, son de investigación, lo que implica que no debe responder sólo con lo que sabe o deduce. Responda con todos los argumentos necesarios, incluya diagramas y analogías. No responda de manera monosilábica. No responda con argumentos vagos o con base a creencias personales. Sustente su argumento con su investigación. |

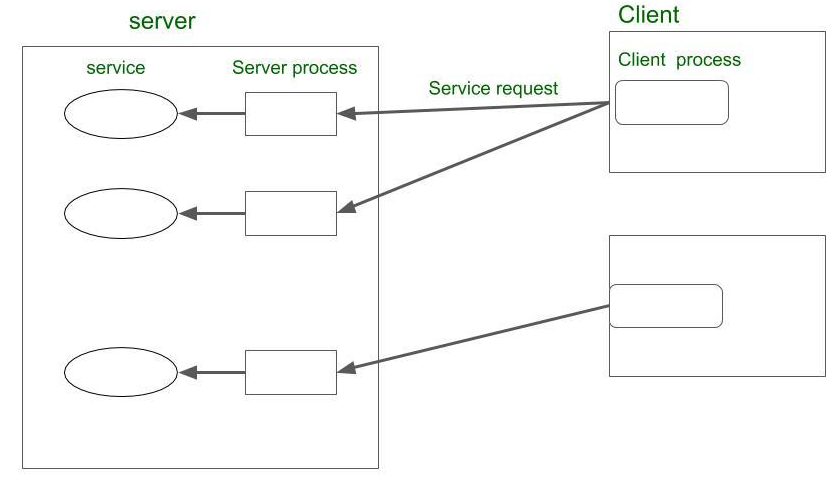
1-Indique las principales características de los sistemas distribuidos. Justifique Brevemente su respuesta.

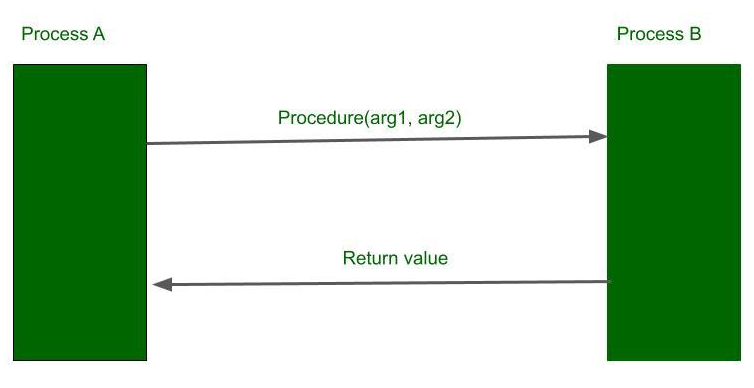
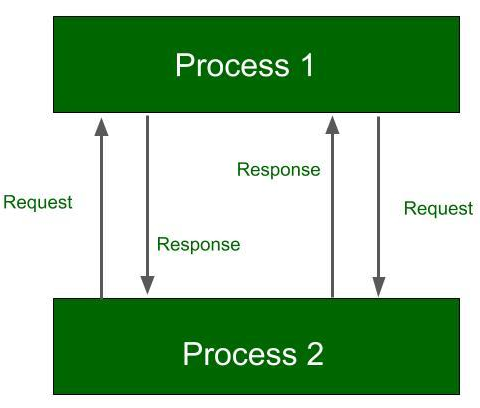
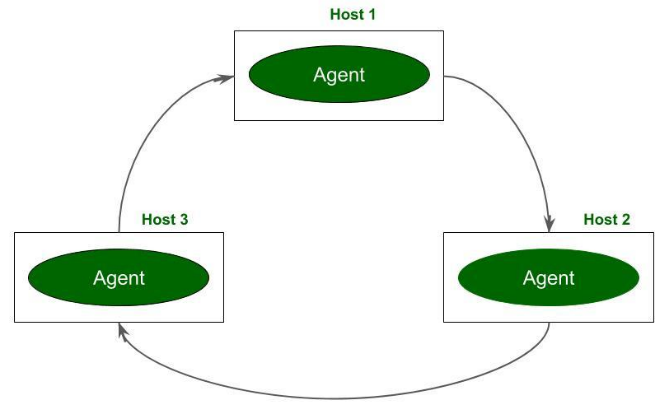
**RESPUESTA.**

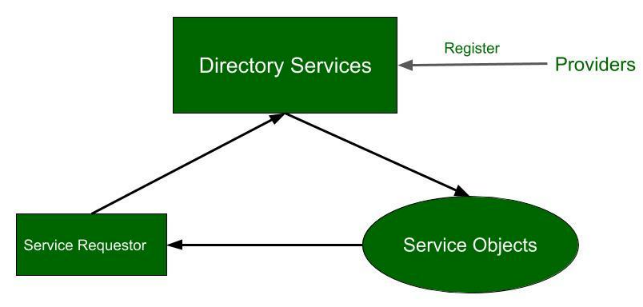
* **Los recursos son compartidos:** Significa que los recursos existentes son compartidos y a ellos pueden acceder distintas computadoras de manera remota.
* **Heterogéneos:** En los sistemas distribuidos los componentes pueden ser de una amplia gama de variaciones (Arquitectura, Red, Hardware, Sistemas Operativos).
* **Compatibles (Openness):** Tiene que existir compatibilidad con distintos tipos de Software o Hardware, para que cuando se pretenda expandir el sistema, este no tenga problemas.
* **Concurrencia:** Distintas actividades pueden ser ejecutadas al mismo tiempo, y estas ejecuciones pueden tomar lugar en diferentes componentes corriendo así en múltiples computadores.
* **Escalabilidad:** Consiste en la manera en cómo se maneja el crecimiento en el número de usuarios finales, normalmente se soluciona agregando más computadoras a la red, esto sin cambiar o afectar a ningún componente.
* **Tolerancia a fallos:** Dentro de un sistema distribuido puede existir un sinfín de fallos, el sistema debe estar diseñado para soportarlos en todo tipos y formas.
* **Transparencia:** Debe ser percibido por todo el mundo como una sola y sólida aplicación más que un conjunto de componentes.

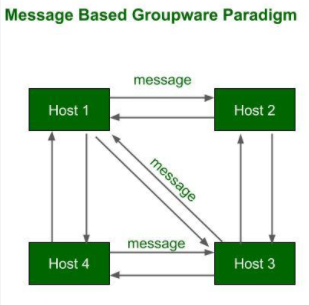
2.- Investigue los paradigmas de comunicación, incluya conceptos breves y diagramas:

* **Paso de mensajes:** Los datos se intercambian entre un remitente y un receptor, un proceso envía un mensaje representando el requerimiento. El receptor lo recibe y lo procesa, para finalmente enviar la respuesta.
* **Cliente-servidor:** En este tipo de comunicación, el servidor actúa como el proveedor, las peticiones las hace el cliente y esta espera a que sean respondidas por el servidor. Hasta que el cliente no hace llamadas al servidor no existe comunicación.



* Llamadas a procedimientos remotos: Sucede entre dos máquinas remotas, envuelve a dos procesos independientes que pueden residir en diferentes máquinas, el proceso A hace una petición al proceso B, el proceso B utiliza la información recibida para devolver una respuesta.
* Peer-to-peer: Es una comunicación directa ente procesos, en este no hay cliente ni servidor, cualquiera puede hacer peticiones a otros y obtener respuesta.
* **Objetos distribuidos:** La aplicación accede a los objetos distribuidos a través de la red, los objetos proveen métodos, con los cuales al ser invocados se obtiene acceso a los servicios. Existen diferentes tipos:
  + *Remote Method Invocation*
  + *Object Request Broker*
  + *Object Space*
* **Agentes móviles:** Este comienza con un host de origen y va transportando la información de host a host. En cada host, un agente puede acceder a los servicios o recursos para completar su misión.
* **Servicios en red:** Todos los objetos de servicio están registrados con un servicio de directorio global. Si un proceso requiere un servicio puede consultar el directorio mientras está corriendo, se le provee una referencia, la cual utiliza para poder interactuar con el servicio, los cuales tienen un identificador único.



* **Comunicación colaborativa o en grupo:** Un host puede hacer multicast para enviar información a todos los hosts en el grupo o parte del grupo de host.

**3.-** Describa el comportamiento de los siguientes tipos de servidores:

* **En función del número de peticiones:** Con esto se refiere al número de peticiones que puede atender un solo servidor al mismo tiempo, ya sea una sola petición o n peticiones.
* **Servidores con estado o sin estado:** El estado de una aplicación hace referencia a su condición o cualidad de existir en un momento determinado. El que un sistema tenga estado depende del tiempo durante el cual se registra interacción con él y de la forma en que se debe almacenar información.
  1. **Servidores sin estado:** Se refiere a los casos en que estos están aislados, no se almacena información sobre las operaciones anteriores ni se hace referencia a ellas. Cada operación se lleva a cabo desde cero, como si fuera la primera vez.
  2. **Servidores con estado:** Son aquellos a los que se puede volver una y otra vez, como un banco en línea o el correo electrónico. Se realizan con el contexto de las operaciones anteriores y la operación actual puede verse afectada por lo que ocurrió previamente.
* **Servidores orientados a conexión, servidores no orientados a conexión:** El hecho de que un servidor sea orientado a conexión o no lo sea, radica en el camino específico que se establece para intercambiar información, de tal manera que para un servidor **orientado a conexión** debe existir un “acuerdo” con el cliente, de tal manera que se establece un canal específico por el cual se va a enviar y recibir la información, un ejemplo de esto son los servidores que utilizan en protocolo TCP/IP.

Por otro lado, los servidores no **orientados a conexión** son aquellos en los cuales no se requiere de ese “acuerdo” con el cliente, de tal manera que los datos que son enviados puede que lleguen al cliente o puede que no, así como también se puede enviar a cualquier cliente que esté escuchando en ese puerto. Un ejemplo de esto es el protocolo UDP, en el cuál tampoco existen encabezados.

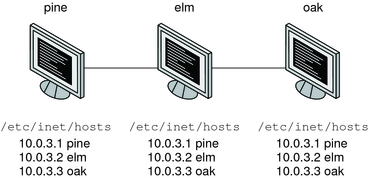
**4.-** ¿Qué es un servicio de nombres y para qué se utiliza? Indique qué servicios genéricos ofrece un servidor de nombres. ¿Qué tipos existen y cómo funcionan? Explique con diagramas y sus definiciones.

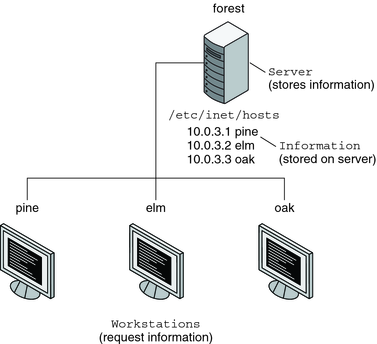
**RESPUESTA.** Un servicio de nombres almacena información en una locación central, la cual permite a los usuarios, máquinas y aplicaciones comunicarse a través de la red. Esta información incluye direcciones de hosts y usuarios, nombres de usuario, contraseñas, permisos de acceso, grupos, impresoras, etc.

Ofrece los siguientes servicios:

* Vincula nombres con objetos.
* Resuelve nombres hacia objetos.
* Elimina los vínculos.
* Enlista nombres.
* Renombramiento.

Existen algunos tipos de arreglo:

* **Red local:**  Este tipo de servicios funciona en una red pequeña o local, donde cada computador tiene asociado un nombre a su dirección ip, para que de esta forma los demás puedan referirse a él con este nombre.
* **Cliente-Servidor:** Toda la información se guarda en un servidor, el cuál se encarga de resolver las peticiones de los clientes, haciendo esto se reducen los errores, inconsistencias entre clientes, etc.



**5.-** Explique con sus propias palabras el Teorema de CAP Y teorema de PACELC, indique ejemplos de uno y otro.

**RESPUESTA.**

**CAP** (Consistency, Availability, Partition Tolerance): En este teorema se explica que en un sistema distribuido debe existir Consistencia, Disponibilidad y Tolerancia a particiones, de tal manera que de los tres debe tomarse uno como fijo, normalmente se recomienda que se asegure la tolerancia a particiones La tolerancia a particiones consta de cómo maneja los fallos de los distintos nodos de la red, de tal forma que el sistema siga funcionando, aunque algunas piezas no estén funcionando.

Lo que se hace aquí es jugar con el nivel tanto de consistencia como de disponibilidad que vas a usar, para que de esta forma se gestionen los recursos y mejore la experiencia del usuario.

**PACELC**: Este teorema es la “evolución” de CAP, ya que en este caso se toma a la tolerancia a particiones de manera fija, y lo que se pone en juego es la disponibilidad, la consistencia y se agrega un nuevo factor siendo este la latencia. Se recomienda dejar fija a la disponibilidad y jugar con las otras dos.

Si lo que se quiere es mayor consistencia, entonces la latencia empeora; por otro lado, si lo que se pretende es mejorar la latencia, esto pone en juego la consistencia del sistema.

La gran mayoría de los grandes sistemas distribuidos hoy en día trabajan bajo este teorema, ya que para mejorar la experiencia de usuario es indispensable que no exista ningún tipo de fallos y sobre todo que el sistema esté disponible en todo momento.

**6.-** Explique en sus términos que es la arquitectura de base de datos Sharding, indique diagramas para explicar. Indique en qué casos la utilizaría y donde no estaría recomendado no utilizar esta técnica.

**RESPUESTA.**

Es una técnica de almacenamiento de cómputo en la cual los grandes almacenamientos independientes son partidos en pequeñas unidades que son más fáciles de administrar. Lo que se obtiene de hacer esto es mejorar el rendimiento, aunque también consigo puede traer redundancia al sistema.

Es preferible usar esta técnica en sistemas que son muy grandes, ya que tener una enorme base de datos no es factible porque reduce la velocidad de la transacción de la información, ya que un sistema de esta magnitud manipula enormes cantidades de información.

Para sistemas pequeños y locales no es conveniente, ya que utilizar esta técnica implica gastos mayores que son totalmente innecesarios si la cantidad de información que se va a manejar no es excesiva.

# References

GeeksforGeeks. (2020). *Distributed Application Paradigms*. Retrieved from GeeksforGeeks: https://www.geeksforgeeks.org/distributed-application-paradigms/

IBM. (n.d.). *CAP Theorem*. Retrieved from IBM Cloud Learn Hub: https://www.ibm.com/cloud/learn/cap-theorem#:~:text=The%20CAP%20theorem%20applies%20a,'P'%20in%20CAP).

ORACLE. (2011). *What Is A Naming Service?* Retrieved from Oracle Solaris Administration: Naming and Directory Services : https://docs.oracle.com/cd/E23824\_01/html/821-1455/a00intro-21293.html