



2do Parcial 100/100

Explique muy brevemente:

- a. Transacción distribuida. Indicar elementos involucrados y explicar cada uno.

Es una transacción plana o anidada que accede a objetos gestionados por múltiples servidores. Cuando una transacción llega a su fin, la propiedad de atomicidad requiere que todos los servidores completen su transacción o que todos aborten.

Coordinador: El coordinador es el servidor que abrió la transacción y es el responsable final de consumirla o abortarla.

Participante: cada uno de los servidores que gestione un objeto al que accede la transacción. Los participantes son responsables de cooperar con el coordinador para sacar adelante el protocolo de consumación.

- b. Tres protocolos y/o elementos involucrados en el desarrollo e implementación de Servicios Web.

XML: eXtensible Markup Language, es un lenguaje extensible de etiquetas el cual se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas. Se puede usar en bases de datos, editores de texto, hojas de cálculo y casi cualquier aplicación en representación de datos. Ofrece un formato de datos estándar, flexible y extensible.

SOAP: Simple Object Access Protocol, es un protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse mediante el intercambio de datos XML. Proporciona un framework estándar y extensible para el empaquetado e intercambio de mensajes XML.

WSDL: Web Service Description Language, es un lenguaje para describir servicios web. Es un lenguaje de interfaz pública para los servicios web de los

requisitos funcionales necesarios para establecer una comunicación con los servicios Web.

UDDI: Universal Description, Discovery and Integration, es un estándar para registrar y descubrir los Web Services.

c. Transacción.

Define una secuencia de operaciones que se realiza por el servidor y se garantiza por el mismo que sea atómica, ya sea en presencia de múltiples usuarios e incluso caídas de servidor.

d. Interbloqueo fantasma.

Un interbloqueo que se “detecta” pero que realmente no lo es se conoce como interbloqueo fantasma. En la detección de interbloqueos distribuidos, la información acerca de las relaciones espera por se transmite de un servidor a otro. Ya que este procedimiento tarda un cierto tiempo, existe la probabilidad de que una de las transacciones que mantiene un bloqueo se haya liberado, en cuyo caso ya no existe.

e. Propiedad de Aislamiento de las transacciones.

Cada transacción debe ser realizada sin interferencia de otras transacciones, es decir, los efectos intermedios de una transacción no deben ser visibles para los demás.

f. Registro histórico.

El registro histórico es una técnica de recuperación de transacciones, en la que el archivo de recuperación representa a un registro que contiene el historial de todas las transacciones realizadas por el servidor. El historial consta de los valores de los objetos, las entradas del estado de la transacción y las listas de intenciones de las transacciones.

Complete con la expresión adecuada:

- a. En el protocolo de consumación en dos fases, la operación "puedeConsumar?" es llamada desde el coordinador a participante para preguntar si puede consumir una transacción.
- b. Cuando una subtransacción finaliza, hace una decisión independiente sobre si consumarse provisionalmente o abortar. Su decisión de abortar es final.
- c. El código de estado 500 del protocolo HTTP significa Internal Server Error.
- d. Una operación idempotente es una operación que puede ser llevada a cabo repetidamente con el mismo efecto que si hubiera sido ejecutada exactamente una sola vez.
- e. Cuando decimos que un par de operaciones en transacciones tienen conflictos queremos decir que su efecto combinado depende del orden,
- f. Si una transacción T ha realizado ya una operación de lectura en un objeto particular, entonces una transacción concurrente U no debe escribir ese objeto hasta la consumación de T, o que aborte.
- g. Un método para resolver los bloqueos indefinidos es el uso de timeouts.

Desarrolle en lenguajes diferentes para el servidor y para el cliente, escriba el código fuente necesario para realizar una votación online según lo siguiente:

- a. El servidor debe:
 - Retornar los candidatos habilitados, con un código de candidato.
 - Recibir las votaciones de los ciudadanos.
 - Debe retornar un objeto compuesto como resultado de cada método web.
- b. El cliente debe:
 - Listar los candidatos habilitados.
 - Elegir el candidato.
 - Enviar su votación.
 - Para cada invocación, utilizar los atributos del objeto de respuesta que indique el éxito o error de dicha invocación.

Defina conceptualmente.

a. Sesgos y deriva de reloj.

Sesgo es la diferencia instantánea entre las lecturas de dos relojes cualquiera.

Deriva de reloj es el fenómeno que describe la diferencia de tiempos entre dos o más relojes.

b. Estado global y depuración distribuida.

El estado global de un sistema distribuido se refiere al estado de todos sus procesos y mensajes en tránsito. Un algoritmo para capturarlo es el de la instantánea de Chandy/Lamport.

La depuración distribuida determina las posibles transiciones entre estados globales, para identificar la posibilidad o certeza de que un determinado evento ocurra.

La depuración distribuida consiste en determinar el predicado que queremos evaluar, especificar un método para construir una red o historial de estados globales consistentes, y evaluar si nuestro predicado se cumple en algún momento.

Depurar un sistema distribuido requiere registrar su estado global, para poder hacer evaluaciones de predicados en dichos estados. Generalmente, la evaluación trata de determinar si el predicado cumple con la condición "posiblemente" o "sin duda alguna".

c. Qué es una transacción?

Una transacción define una secuencia de operaciones que realiza el servidor y que el mismo garantiza que sea atómica, ya sea en presencia de múltiples usuarios e incluso caídas del servidor.

d. Qué hacen las transacciones anidadas?

Las transacciones anidadas extienden el modelo de transacción permitiendo que las transacciones estén compuestas de otras. Desde una transacción se pueden arrancar varias transacciones.

e. WSDL.

Web Service Description Language, es una interfaz descrita en un formato procesable por máquina usada por los servicios web.

f. Control optimista de la concurrencia.

Esta alternativa se basa en la observación de que, en la mayoría de las aplicaciones, la posibilidad de que las transacciones de dos clientes acceden al mismo objeto es baja. Por lo que se permite que las transacciones procedan como si no hubiera posibilidad de conflicto.

Fases:

- Fase de trabajo: durante esta fase, cada transacción tiene una versión tentativa de cada uno de los objetos que actualiza.
- Fase de validación: cuando se recibe la solicitud cierraTransacción, se valida la transacción para establecer si sus operaciones en los objetos entran en conflicto o no con las operaciones en otras transacciones sobre los mismos objetos. Si la validación falla, se debe utilizar alguna forma de resolución de conflictos o abortar alguna transacción:
- Fase de actualización: Si una transacción es validada, todos los cambios registrados en sus versiones provisionales se hacen permanentes.

g. Equivalencia secuencial.

Un solapamiento de las operaciones en las que el efecto combinado es el mismo que si las transacciones hubieran sido realizadas una cada vez en el mismo orden es un solapamiento secuencialmente equivalente.

h. Qué es y cómo se produce un interbloqueo fantasma?

Un interbloqueo que se “detecta” pero que realmente no lo es se conoce como interbloqueo fantasma. En la detección de interbloqueos distribuidos, la información acerca de las relaciones espera por se transmite de un servidor a otro. Ya que este procedimiento tarda un cierto tiempo, existe la probabilidad de que una de las transacciones que mantiene un bloqueo se haya liberado, en cuyo caso ya no existe.

Cite y Explique

a. Modos en que se sincronizan los servidores NTP.

- Multidifusión: pensado para su uso en una LAN de alta velocidad. Uno o más servidores reparten periódicamente el tiempo a los servidores que se ejecutan en otros computadores conectados en la LAN, que fijan sus relojes suponiendo un pequeño retardo.
- Llamada a procedimiento: similar al algoritmo de Cristian. Un servidor acepta solicitudes de otros computadores, a los cuales responde con su marca de tiempo.
- Simétrico: está aplicado en los estratos más bajos. Se usa entre servidores NTP para sincronizarse entre sí, como mecanismo de respaldo cuando no pueden alcanzar el servidor NTP externo.

b. Características posibles de un predicado SD.

- Estabilidad: el valor del predicado no varía con nuevos sucesos.
- Seguridad: el predicado tiene valor falso para cualquier estado alcanzable desde S_0 .
- Vitalidad: el predicado tiene valor verdadero para algún estado alcanzable desde S_0 .

c. Propiedades de las transacciones.

- Atomicidad: Asegura que una transacción se realice por completo o no se realice en lo absoluto.

- Consistencia: Garantiza que las transacciones lleven al sistema de un estado válido a otro también válido, cumpliendo con las reglas establecidas en la base de datos.
- Aislamiento: Asegura que cada transacción se ejecute de forma independiente de otras, sin interferencias.
- Durabilidad: Asegura que una vez que se confirma una transacción, sus cambios permanecen, incluso si hay un fallo en el sistema o se apaga el servidor.

d. Problemas conocidos del control de concurrencia.

- Actualizaciones perdidas: Cuando una actualización de una transacción se pierda por ejemplo cuando las operaciones de ambas transacciones dependen de cierto valor el cual ambos leen a la vez, por lo que, cuando una de las transacciones culmina la transacción, la otra transacción está trabajando con el valor antiguo, por lo que el resultado final sería incorrecto.
- Recuperaciones inconsistentes: Se da cuando una transacción intenta acceder a un objeto en medio de dos operaciones sobre este objeto.

e. Medidas de tolerancia a fallos utilizados por la semántica de invocación "Como máximo una vez".

- Retransmisión de mensajes: se retransmite el mensaje de petición hasta que, o bien se recibe una respuesta o se asume que el servidor ha fallado.
- Filtro de duplicados: Cuando se emplean retransmisiones, se descartan las peticiones duplicadas en el servidor.
- Retransmisión de respuesta: Se mantiene un historial de los mensajes de respuesta para permitir retransmitir las respuestas perdidas sin reejecutar las operaciones en el servidor.

f. Pasos de los algoritmos de caza de arcos.

Es una aproximación distribuida para la detección de interbloqueos. En esta aproximación, el grafo espera por global no se construye, sino que cada uno de los servidores implicados posee información sobre algunos de sus arcos. Los servidores intentan encontrar ciclos mediante el envío de mensajes denominados sondas, que siguen los arcos de un grafo.

Tres pasos:

- **Iniciación:** Cuando un servidor percibe que una transacción T comienza a buscar otra transacción U, y esta a su vez está esperando a otra transacción, iniciará la detección enviando una sonda que contiene el arco $T \rightarrow U$ al servidor del objeto por el cual está esperando U.
- **Detección:** La detección consiste en recibir sondas y decidir si se ha producido un interbloqueo o si hay que enviar nuevas sondas.
- **Resolución:** Cuando se detecta un ciclo, se aborta una transacción en el ciclo para romper el interbloqueo.

g. Operaciones dentro del protocolo de consumación en dos fases.

- puedeConsumar?(trans): llamada desde el coordinador al participante para preguntar si puede consumir una transacción. El participante responde con su voto.
- Consuma(trans): llamada desde el coordinador al participante para decirle que consuma su parte de la transacción.
- Aborta(trans): llamada desde el coordinador al participante para decirle que aborte su parte de la transacción.
- heConsumado(trans, participante): llamada desde el coordinador al participante para decirle que aborte su parte de una transacción.
- dameDecisión(trans): llamada desde el participante al coordinador para preguntarle por la decisión sobre una decisión tras haber votado Sí aunque no ha obtenido respuesta tras cierto periodo de tiempo. Se utiliza para recuperarse de la caída de un servidor o de mensajes con retraso.

h. Roles dentro de OAuth2.

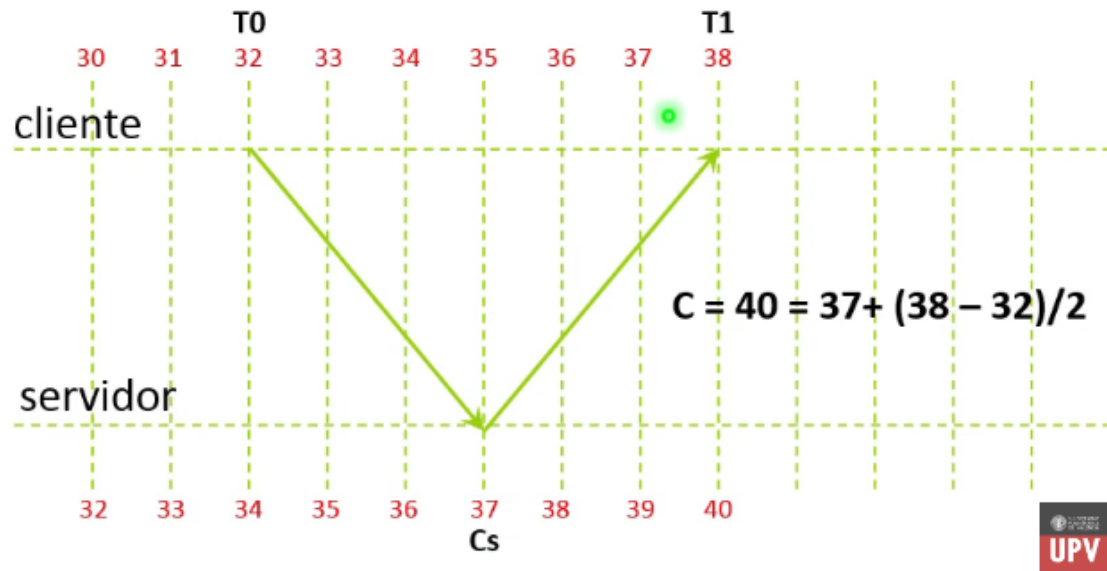
- Propietario del recurso: una entidad capaz de otorgar acceso a un recurso protegido.
- Servidor de recursos: el servidor que aloja los recursos protegidos, capaz de aceptar y responder a solicitudes de recursos protegidos mediante tokens de acceso.
- Cliente: una aplicación que realiza solicitudes de recursos protegidos en nombre del propietario del recurso y con su autorización.
- Servidor de autenticación: el servidor que emite tokens de acceso al cliente después de autenticar con éxito al propietario del recurso y obtener la autorización.

Explique

a. El método Cristian para que sirve, describa. Incluya gráficos y fórmulas.

El método Cristian se utiliza para la sincronización de un sistema síncrono, utilizando un servidor de tiempo.

1. Un cliente solicita al servidor el valor del reloj, en un tiempo T_0 .
2. El servidor contesta con el valor de su reloj, T_s .
3. La respuesta llega al cliente en un tiempo T_1 .
4. Aquí se calcula cual debe ser el nuevo valor del reloj del cliente, utilizando la formula $T = T_s + (T_0 + T_1)/2$



b. Explique el método Lamport utilizando los eventos de la figura.

Un reloj lógico de Lamport es un contador software que se incrementa monótonamente, cuyos valores no necesitan tener ninguna relación particular con un reloj físico. Cada proceso mantiene su propio reloj lógico, que se utiliza para aplicar las llamadas marcas de Lamport a los sucesos.

Las marcas de Lamport se actualizan de la siguiente manera:

Siendo $Li(e)$ la marca de tiempo del suceso e en pi

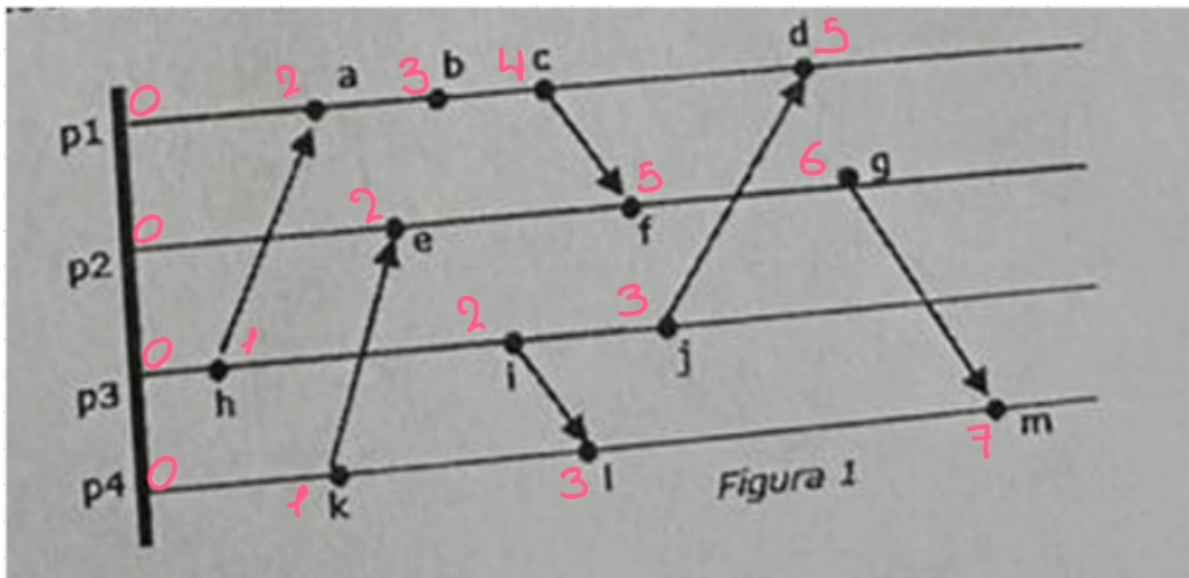
RL1: Se incrementa antes de emitir cada suceso en el proceso pi : $Li = Li + 1$

RL2:

a. Cuando un proceso envía un mensaje m , acarrea en m el valor de $t = Li$.

b. Al recibir (m, t) , cada proceso pi calcula $Lj = \max(Li, t) + 1$

Por ejemplo:



c. Identifique los relojes vectoriales.

Los relojes vectoriales se desarrollaron para vencer la deficiencia de los relojes de Lamport que de $L(e) < L(e')$ no podemos deducir que $e \rightarrow e'$.

Un reloj vectorial para un sistema de N procesos es un vector de N enteros. Cada proceso mantiene su propio reloj vectorial V_i .

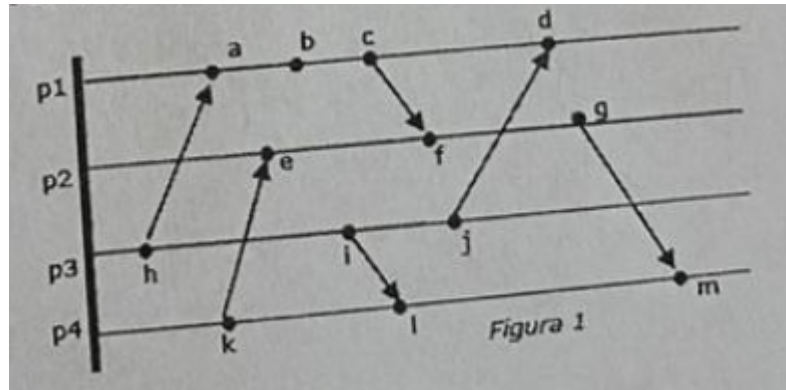
Reglas:

RV1: Inicialización. $V_i[j] = 0$ para $i, j = 1, 2, \dots, N$

RV2: Para cada suceso, $V_i[i] = V_i[i] + 1$.

RV3: p_i incluye el valor $t = V_i$ en cada mensaje que envía.

RV4: Cuando p_i recibe una marca de tiempo t , establece $V_i[j] = \max(V_i[j], t[j])$ para $j = 1, 2, \dots, N$.



- d. Explique e indique el nombre de la primera fase del "protocolo de consumación en dos fases".

Fase 1 - Fase de votación:

El coordinador envía una petición de puedeConsumar? a cada participante en la transacción.

El participante responde con su voto. Antes de votar Sí, se prepara para consumir, guardando los objetos en un dispositivo de almacenamiento permanente. Si el voto es no, el participante aborta inmediatamente.

Fase 2 - Finalización en función al resultado de la votación:

El coordinador recoge los votos

Si no hay fallos y todos los votos son Sí, el coordinador envía peticiones de Consuma a cada uno de los participantes.

En otro caso, el coordinador decide abortar la transacción y envía peticiones Aborta a todos los participantes que votaron Sí.

- e. Explique en el lenguaje Java, un Webservice que contenga 2 métodos web. Además, escriba la representación XML del request y response para la invocación de uno de sus métodos.

```
import javax.jws.WebMethod;
import javax.jws.WebService;
```

```

@WebService
public interface MiWebService {

    @WebMethod
    String saludar(String nombre);

    @WebMethod
    int suma(int a, int b);
}

@WebService(endpointInterface = "MiWebService")
public class MiWebServiceImpl implements MiWebService {

    @Override
    public String saludar(String nombre) {
        return "Hola, " + nombre + "!";
    }

    @Override
    public int suma(int a, int b) {
        return a + b;
    }
}

```

```

//Ahora, publicamos el Web Service usando Endpoint
public class ServidorWebService {
    public static void main(String[] args) {
        MiWebServiceImpl implementacion = new MiWebServiceImpl();
        String url = "http://localhost:8080/MiWebService";
        Endpoint.publish(url, implementacion);
        System.out.println("Web Service publicado en " + url);
    }
}

```

```
//Request XML para el método saludar
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/s
  <soapenv:Header/>
  <soapenv:Body>
    <web:saludar>
      <nombre>Juan</nombre>
    </web:saludar>
  </soapenv:Body>
</soapenv:Envelope>
```

```
//Response XML del método saludar
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/s
  <soapenv:Header/>
  <soapenv:Body>
    <ns2:saludarResponse xmlns:ns2="http://example.com/">
      <return>Hola, Juan!</return>
    </ns2:saludarResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

- f. Escriba 2 transacciones cada una con operaciones sobre objetos a, b y c, posteriormente describa dichas transacciones para un solapamiento secuencialmente equivalentes. Graficar.

Transaction T:		Transaction U:	
<i>balance = b.getBalance()</i>		<i>balance = b.getBalance()</i>	
<i>b.setBalance(balance*1.1)</i>		<i>b.setBalance(balance*1.1)</i>	
<i>a.withdraw(balance/10)</i>		<i>c.withdraw(balance/10)</i>	
<i>balance = b.getBalance()</i>	\$200	<i>balance = b.getBalance()</i>	\$220
<i>b.setBalance(balance*1.1)</i>	\$220	<i>b.setBalance(balance*1.1)</i>	\$242
<i>a.withdraw(balance/10)</i>	\$80	<i>c.withdraw(balance/10)</i>	\$278

g. Explique el método de Versiones Provisionales.

Para que un servidor de objetos recuperables participe en las transacciones, debe estar diseñado de forma que las actualizaciones de los objetos puedan ser eliminadas si la transacción aborta. Para esto, todas las actualizaciones realizadas durante una transacción se realizan sobre versiones provisionales de los objetos en memoria volátil. Las versiones provisionales son transferidas a los objetos solo cuando una transacción se consuma, en cuyo caso ellas serán también registradas en memoria permanente. Cuando una transacción aborta, sus versiones provisionales se borran.

Escriba las clases y/o interfaces en JavaRMI para definir:

- Un objeto remoto con su implementación.
- Un servidor RMI que publique el objeto remoto definido.

```
import java.rmi.Remote;
import java.rmi.UnicastRemoteObject;
import java.rmi.RemoteException;

public interface Interfaz extends Remote{
    ...
}

public class Objeto extends UnicastRemoteObject implements Inter
    ...
}
```

```
import java.rmi.registry.LocateRegistry;
import java.rmi.registry.Registry;

public class ServidorRmi{
    public static void main(String[] args){
        try{
            Objeto objeto = new Objeto();

            Registry registry = LocateRegistry.createRegistry(1099);
            registry.rebind("objeto", objeto);
        } catch (Exception e){
            ...
        }
    }
}
```