

Alumno/a: \_\_\_\_\_

CI: \_\_\_\_\_

**Tema 1 (20p):** Explique brevemente:

- a) Transacción,
- b) Interbloqueo fantasma,
- c) Propiedad de Aislamiento de las transacciones,
- d) Registro histórico.

**Tema 2 (30p):** Cite:

Propiedades de las transacciones: (12p):

- a) ..... **Atomicidad** ..... c) ..... **Duración** .....
- b) ..... **Consistencia** ..... d) ..... **Isolation** .....

Operaciones dentro del protocolo de consumación en dos fases (9p):

- a) ..... **Consumo** ..... c) ..... **Puede consumir?** .....
- b) ..... **Abortar** .....

Pasos en los algoritmos de caza de arcos (9p):

- a) ..... **Imaginación** ..... c) ..... **Resolución** .....
- b) ..... **Protección** .....

**Tema 3 (20p):** Complete con la expresión adecuada:

- a) Cuando decimos que un par de operaciones en transacciones tienen conflictos queremos decir que su efecto combinado depende del **OPEN DE SEÑAL**.
- b) Cuando un subtransacción finaliza, hace una decisión independiente sobre si consumarse provisionalmente o abortar. Su decisión de **ACOMPLIR** es final.
- c) Si una transacción T ha realizado ya una operación de **LEER** en un objeto particular, entonces una transacción concurrente U no debe **ESCRIBIR** ese objeto hasta la consumación de T, o que aborte.
- d) Un método para resolver los bloqueos indefinidos es el uso de **TIMEOUT**.

**Tema 4 (30p).** Resuelva.

- a) Escriba 2 transacciones cada una con operaciones sobre objetos "a", "b" y "c", posteriormente rescriba dichas transacciones para un solapamiento secuencialmente equivalentes. OBS: Graficar cada transacción en una columna, tal como fue analizada en clase (10p)
- b) Explique el método de "Versiones Provisionales". (6p)
- c) Explique la fase 1 del protocolo de consumación en dos fases referente a transacciones distribuidas. (6p).
- d) En la figura 1, explique las operaciones *wait* y *notify* dentro del contexto del método *put* (8p)

Carrera: Ingeniería Informática  
Examen: Segundo parcial

Profesor: Ing. Fernando Mancia  
Fecha: 01/06/2015

Alumno/a: Mario Antonio Rivas Ojeda

CI: 3.958.174

Tema 1 (20p). Defina / Responda. *(Encuentre errores en XML para definir servicios web)*

- ✓ a) WSDL. *WEB SERVICE DESCRIPTION LANGUAGE*
- ✓ b) Control optimista de la concurrencia. *SIN BLOQUEOS*
- ✓ c) Equivalencia secuencial. *PROGRAMA DE SISTEMA DISTRIBUIDO*
- ✓ d) ¿Qué es y cómo se produce un interbloqueo fantasma?

Tema 2 (30p): Cite y explique brevemente.

- ✓ a) Tres métodos de sincronización utilizados en los servidores NTP.
- ✓ b) Cuatro propiedades deseables de las transacciones. *ACID*
- ✓ c) Tres pasos de los algoritmos de captura de arcos o caza de arcos.  
*INICIO DE RESERVA DE TIEMPO → SE ADQUIRE TIEMPO → SE LIBERA TIEMPO*

-MULTIFUSIÓN  
-LLAMADA A PROCEDIMIENTO  
-SIMÉTRICO

Tema 3 (36p): Resuelva:

- a) Explique el método Cristian [1989] para sincronizar relojes. Incluir fórmula/s (6p). *UTILIZA SERVICIOS DE TCP PARA TIEMPO + TRABAJO EN TIEMPO.*
- b) Escriba, en el lenguaje Java, un WebService que contenga 2 métodos web. Además escriba la representación XML (una aproximación) del request y response para la invocación de uno de sus métodos. (14p).
- c) Identifique y complete las marcas temporales de Lamport de los eventos de la Figura 1. (6p).
- d) Identifique los relojes vectoriales en la Figura 1 para los eventos d, e, g, i y m. (5p).
- e) Explique e indique el nombre de la primera fase del "protocolo de consumación en dos fases". *FASE DE VOLTAJE*

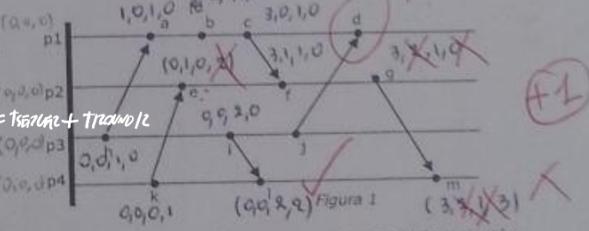


Figura 1

Tema 4 (14p): Complete con la o las palabras correctas.

- a) En JavaRMI, el método *Bind* del Registry (`java.rmi.registry.Registry`) es utilizado para publicar/disponibilizar un objeto remoto desde el servidor; y el método *Lookup* también del Registry (`java.rmi.registry.Registry`) es utilizado en el cliente para encontrar el objeto remoto.
- b) Uno de los problemas que evita el control de concurrencia es *PROBLEMA DE ACCESO CONCURRENTE A PISTAS*.
- c) Una forma de detectar un bloqueo indefinido es *TIMEOUT (TRAMPAZON)*.
- d) Al implementar mecanismos de bloqueo, 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 aborde.

Alumno/a:

Tema 1 (30p): Complete con la expresión adecuada:

- En el algoritmo de instantánea de Chandy y Lamport, la regla \_\_\_\_\_ obliga a los procesos a enviar un marcador después de haber registrado su estado, pero antes de que envíen cualquier otro mensaje. (5p)
- TIEMPO DE DERIVA es el cambio en la compensación entre el reloj y un reloj de referencia nominal perfecto por unidad de tiempo, normalmente expresado en partes por millón. (5p)
- Cuando un subtransacción finaliza, hace una decisión que puede ser CONSUMA o ABANDONA. Su decisión de ABANDONAR es final. (5p)
- Un método para resolver los bloqueos indefinidos es el uso de TIMEOUTS que se basa en TEMPORIZACIÓN CON UN TIEMPO DEFINIDO DE ESPERA. (5p)
- El LEER/ESCRIBIR ocurre cuando dos transacciones leen el valor antiguo de una variable y la utilizan para calcular el nuevo valor. (5p)
- En transacciones distribuidas, cada uno de los servidores que gestione un objeto al que accede la transacción es un PARTE de la transacción. (5p)

Tema 2 (20p): Complete con verdadero o falso, justifique los falsos:

- La atomicidad asegura que una transacción se ejecute parcialmente si ocurre un error. (F)  
*Asegura que la transacción vuelve a un estado válido anterior al de su inicio sin errores.*
- Un bloqueo indefinido ocurre cuando varias transacciones esperan por recursos de otras, creando un ciclo de espera. (V)
- Un evento es la ocurrencia de una única acción que un proceso realiza durante su ejecución que puede ser una acción de comunicación o una transformación del estado. (V)
- En el algoritmo de Berkeley, el coordinador debe tener acceso a un reloj extremadamente preciso para sincronizar los relojes de la red. (F)  
*SE UTILIZA UN NODO MAESTRO, NO SEPARAR*

- e) En Java, los métodos synchronized permiten que múltiples hilos accedan simultáneamente al mismo objeto. (F)  
*PROTEGE QUE MÁS DE UN HILLO ACCEDA A UN RECURSO.*

Tema 3 (18p): Cite y explique brevemente:

- Elementos o campos que se almacenan en una estructura JWT (9p): JSON WEB TOKEN
  - HEADER : ESPECIFICA EL TIPO DE TOKEN
  - PAYOUT : INFORMACIÓN A TRANSMITIR
  - FIRMA : PARA VERIFICAR QUE EL MENSAJE NO FUÉ ALTERADO.
- Primitivas de comunicación en el Protocolo de Petición-Respuesta (9p):
  - DO OPERATION : INICIA OPERACIÓN
  - GET REQUEST : ESPERA PETICIÓN EN EL SERVIDOR
  - SEND REPLY : MANDA MENSAJE DE REPUESTA

Tema 4 (18p)  
a) Propiedad de aislamiento de las transacciones.

b) Protocolo Oauth-2. Especificación, funcionamiento, gráfico explicativo.

PROTOCOLO DE AUTORIZACIÓN, PERMITE ACCEDER A RECURSOS PROTEGIDOS SIN NECESSIDAD DE REVELAR CREDENCIALES. EL CLIENTE APP QUE QUIERE ACCEDER A LOS RECURSOS SE DIRIGIDO AL SERVICIO DE AUTORIZACIÓN, SI SE CONCEDE ALUEO, SE LE PROPORCIONA UN CÓDIGO DE AUTORIZACIÓN, LUEGO SE OBTIENEN TOKEN.

c) Realice un gráfico donde se visualice un interbloqueo distribuido. Debe contar con 4 servidores, 6 objetos y 4 transacciones.

Tema 5 (14p). Escriba una clase Java Springboot del tipo Controlador (Controller) que utilice los elementos/anotations/estructura/etc. para definir al menos 2 métodos web REST de consulta y 2 métodos de escritura. Los servicios deben corresponder al sistema del aeropuerto Silvio Pettrossi que necesita para interoperar con los demás aeropuertos.

```
@RequestMapping("/api/v1/aeropuerto")
public class AeropuertoController {
    @Autowired
    private VueloService vueloService;
    @Autowired
    private AeronaveService aeronaveService;

    // Métodos de consulta (GET)
    // Consultar todos los vuelos programados
    @GetMapping("/vuelos")
    @ResponseStatus(HttpStatus.OK)
    public List<Vuelo> obtenerTodosVuelos() {
        return vueloService.obtenerTodosVuelos();
    }

    // Consultar un vuelo específico por su código
    @GetMapping("/vuelos/{codigoVuelo}")
    @ResponseStatus(HttpStatus.OK)
    public Vuelo obtenerVueloPorCodigo(@PathVariable("codigoVuelo") String codigoVuelo) {
        Optional<Vuelo> vuelo = vueloService.obtenerVueloPorCodigo(codigoVuelo);
        return vuelo.orElseThrow(() -> new RuntimeException("Vuelo no encontrado"));
    }

    // Métodos de escritura (POST)
    // Registrar un nuevo vuelo
    @PostMapping("/vuelos")
    @ResponseStatus(HttpStatus.CREATED)
    public Vuelo registrarVuelo(@RequestBody Vuelo vuelo) {
        return vueloService.registrarVuelo(vuelo);
    }

    // Registrar una nueva aeronave
    @PostMapping("/aeronaves")
    @ResponseStatus(HttpStatus.CREATED)
    public Aeronave registrarAeronave(@RequestBody Aeronave aeronave) {
        return aeronaveService.registrarAeronave(aeronave);
    }
}
```

1. En general, los URLs de HTTP son de la forma: ~~esquema://dirección del recurso/recurso~~(p)
2. El término ~~sistema~~ se emplea para referirse al código que puede ser enviado desde un computador a otro y ejecutarse en éste. (1p)
3. Se dice que un sistema es ~~escalable~~ si conserva su efectividad cuando ocurre un incremento significativo en el número de recursos y el número de usuarios.(1p)
4. El nivel de hardware y las capas más bajas de software se denominan ~~plataforma~~ para sistemas distribuidos y aplicaciones.(1p)
5. La ~~encriptación~~ y la ~~autenticación~~ se emplean para construir canales seguros en forma de capa de servicio sobre los servicios de comunicación existentes.(1p)
6. Si un proceso para permanece parado, y otros procesos pueden no ser capaces de detectar este estado estamos hablando de la clase de fallo ~~OMISIÓN~~ que afecta a un proceso.(1p)
7. En aquellos servidores que necesitan retransmitir las respuestas sin tener que volver a ejecutar las operaciones es imprescindible la utilización de un ~~histórico~~.(1p)
8. El protocolo TCP utiliza ~~seq~~ para la transmisión de datos, mientras que el protocolo UDP utiliza ~~cholegramas~~.~~publicación-suscripción~~ (1p)
9. Los sistemas distribuidos basados en eventos emplean el paradigma ~~publicación-suscripción~~, en el que un objeto que genera eventos publica el tipo de eventos que ofrece para su observación por otros objetos.(1p)
10. En RMI, los módulos de comunicación cooperantes realizan el protocolo ~~petición-respuesta~~ que retransmite los mensajes entre el cliente y el servidor(1p)
11. En un sistema distribuido, un estado global consistente es uno que corresponde a un ~~completo~~ consistente.(1p)
12. En el protocolo NTP la sincronización en modo ~~unicast~~ está pensado para su uso en una LAN de alta velocidad.(1p)

#### B- CITA (17p) (9)

1. 4 (cuatro) métodos que soporta el HTTP (4p)

~~POST~~ ~~DELETE~~

b) ~~PUT~~  
c) ~~GET~~  
d) ~~HEAD~~

Universidad Nacional de Asunción  
Facultad Politécnica  
Sistemas Distribuidos

Alumno/a: \_\_\_\_\_ Profesor: Ing. Fernando Mancia  
Examen: 1er final Fecha: 25/06/2016

**Tema 1 (15p):** Defina: ~~Interfaz~~ ~~Protocolo~~; Protocolo de comunicación utilizado para intercambio de ~~información~~ entre los sistemas Punto a Punto, que sigue ~~normas~~ y ~~funciones~~

a) Protocolo HTTP: ~~Protocolo de comunicación utilizado para intercambio de información entre los sistemas Punto a Punto, que sigue normas y funciones~~

b) Sockets: ~~Protocolo de comunicación que permite la fragmentación de datos entre diferentes sistemas~~

c) Servidor proxy y caché: ~~Servidor proxy: intercambiar entre cliente y servidor uno o más datos~~

**Tema 2 (23p):** Cite: ~~Protocolo de conexión entre cliente y servidor web~~

a) Tres modelos fundamentales (9p). ~~Interacción, Fallo, Seguridad~~

b) Dos alternativas a la representación externa de datos y empaquetado (6p). JSON, XML

c) Dos desventajas de utilizar bloques para el control de concurrencia (8p). ~~Interbloqueo, Latencia~~

**Tema 3 (28p):** Complete con la expresión adecuada:

a) Uno de los motivos principales para construir sistemas distribuidos es el de ~~Paralelismo o Explotamiento~~. (4p)

b) Los sistemas distribuidos pueden fallar de formas diferentes, en general se engloban como "fallos independientes" a los fallos ~~En la red~~ y ~~Copia de datos~~. (8p)

c) Los servidores NTP se sincronizan entre sí en uno de estos tres modos: ~~Unidadaria~~ a ~~Platónico~~ y modo simétrico. (8p)

d) El protocolo DNS, está implementado sobre el protocolo ~~UDP~~ de la capa de transporte. (4p)

e) La clase Java que representa un paquete UDP es ~~DATAGRAMPACKET~~. (4p)

**Tema 4 (34p).** Explique detalladamente.

a) Modelo "Peer to peer", incluir gráfico. (7p)

b) Arquitecturas en capas, ejemplifique de 2 y 3 capas. (7p)

c) Algoritmo de instantánea de Chandy y Lamport (10p)

d) Complete el reloj vectorial correspondiente a la figura (10p)

Figura 1

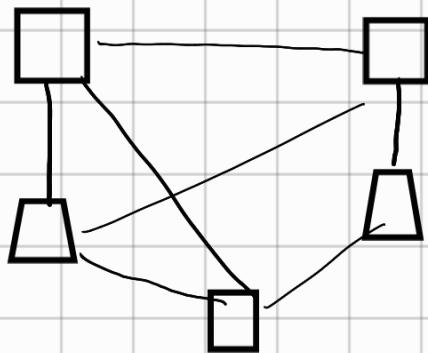
4. 3 (tres)-aspectos sobre los cuales el middleware ofrece transparencia. (3p)
- a) ~~Transparencia de ubicación~~  
b) ~~Lenguaje de implementación~~  
c) ~~Plataforma~~
5. 2 (dos) responsabilidades del módulo de referencia remota en RMI (2p)
- a) ~~manejear una tabla de referencia por cada objeto remoto~~  
b)  ~~establecer relaciones entre el objeto remoto y su referencia al objeto local~~

## tema 4 (34p). Explique detalladamente.

- Modelo "Peer to peer", incluir gráfico. (7p)
- Arquitecturas en capas, ejemplifique de 2 y 3 capas. (7p)
- Algoritmo de instantánea de Chandy y Lamport (10p)

### a) MODELO PEER TO PEER, EXPLICAR Y GRÁFICAR.

MODELO EN EL CUAL CADA NODO ES "PEER", TIENE LA MISMA CAPACIDAD, NO HAY SERVIDOR (FUTRO QUE CONTROLA LA RED)



### b) ARQUITECTURAS EN CAPAS EJEMPLIFIQUE DE 2 Y 3 CAPAS

#### • ARQUITECTURA EN DOS CAPAS

- CAPA CLÍENTE: INTERACTÚA CON EL USUARIO FINAL
- CAPA SERVIDOR: ALMACENA, RECOPILA Y MANIPULA DATOS

EJEMPLO: APLICACIÓN DE FICHEROS QUE SE CONECTA CON BASE DE DATOS SQL

- ARQUITECTURA EN 3 CAPAS
  - FRONTEND: CAPA QUE INTERACTUA CON USUARIO
  - BACKEND: PROBLEMA LOGICO DE APLICACION
  - CAPA DE DATOS: DONDE SE ALMACenan DATOS

Ejemplo: APLICACIÓN WEB COMERCIO ELECTRÓNICO

### c) ALGORITMO DE INSTANCIA CHANDY Y COOPER

ALGORITMO UTILIZADO EN SID PARA CAPTURAR UNA INSTANCIA CONSISTENTE DENTRO DE UN SISTEMA  
SE DEFINE MEDIANTE DOS REGLAS.

- 1- REGLA DE RECEPCIÓN DE MARGADOR: Obliga a (o) PROCESO A ENVIAR UN MARGADOR
- 2- REGLA DEL ENVÍO DEL MARGADOR: Obliga a UN PROCESO QUE NO HA REGISTRADO SU ESTADO A HACERLO.

### QUE ES MIDDLEWARE?

SOFWARE QUE ACTUA COMO PUENTE DENTRO DE UN SISTEMA DISTRIBUIDO FACILITANDO LA COMUNICACIÓN Y EL INTERCAMBIO DE DATOS PROPORCIONA:

- TRANSPARENCIA
- PROTOCOLO DE COMUNICACIÓN
- SISTEMAS OPERATIVOS