

20220802

1) ¿Qué diferencia una arquitectura Multithreading de una Multiprocessing? Indique ventajas.

Multithreading

- Comparten memoria
- Más fáciles de usar
- Comunicación sencilla
- Alto acoplamiento

Multiprocesos

- | . No comparten memoria
- | . Comunicación a través de IPC
- | . Bajo acoplamiento
- | . más escalables y estables
- | . Si un proceso falla no afecta al resto

2) Explique cómo garantizar un corte consistente al capturar un estado global de cierto sistema distribuido. Indique pasos principales del algoritmo propuesto.

S es el estado global en un tiempo t

$$S = S_1 \cup S_2 \cup \dots \cup S_n$$

Un corte es un subconjunto de historias de todos los procesos del sistema hasta un evento k

$$C = h_1 \cup h_2 \cup \dots \cup h_n$$

$$= (e_1, \dots, e_k) \cup \dots \cup (e_1', \dots, e_k')$$

Cada historia es una secuencia de eventos.

Para que un corte sea consistente, por cada evento que contiene, debe contener a los que ocurren antes.

El algoritmo Chandy - Lamport obtiene un snapshot del estado global con los siguientes pasos:

- 1) El proceso observador guarda su estado
- 2) Este envía un marcador a todos los procesos
- 3) Los procesos que reciben el marcador por primera vez envían su propio estado y propagan el marcador en el resto de sus mensajes
- 4) Si un proceso recibe un mensaje sin un marcador que ya recibió, significa que ese mensaje fue creado anterior y debe formar parte del snapshot

3) Explique qué estrategias puede implementar un MOM para garantizar tolerancia a fallos.
Ejemplifique con algún MOM que conozca o haya implementado.

Un MOM es un message oriented middleware. Para garantizar la tolerancia a fallos se pueden usar estrategias como

- ACK para asegurar su entrega
- Replicación de colas
- Persistencia de colas

4) Explique DSM (Distributed Shared Memory) desarrollando al menos una alternativa de implementación. Indique ventajas y desventajas de dicha arquitectura.

DSM consiste en que al menos dos procesos/nodos comparten una memoria.

Náive

Una estrategia posible es con un servidor que centraliza los pedidos para acceder a esta memoria.

Esta estrategia implica que cada proceso envíe un request al centralizador y si el recurso no está siendo utilizado lo devuelva.

Esta estrategia es muy segura pero de baja performance.

La ventaja de usar DSM es que es muy intuitivo y fácil compartir información entre nodos.

La desventaja es que se pierden algunas de las beneficios de un SD, y agrega un punto único de fallo.

5) Describa al menos 2 patrones de carga de aplicaciones Web e indique cómo el escalamiento "Horizontal" permite mitigar efectos negativos.

Predictable Burst: La carga aumenta de manera repentina pero predecible

Periodic Peaking: La carga se mantiene constante en ciertos períodos

En el primer caso, el escalamiento horizontal nos permite agregar más nodos dentro de un load balancer

En el segundo caso podemos agregar nodos en los períodos de carga

6) Indique las diferencias entre los modelos de Replicación Activa y Pasiva. Indique ventajas de cada uno.

En la pasiva, se modifica una base de datos y luego se actualiza una replica.

Solo la principal realiza acciones de compute. Si se cae el nodo activo, se debe reemplazar por uno pasivo.

En la activa, se actualizan todos (o replicar al mismo tiempo).

Todos realizan el compute y le responden al load balancer.

7) Explique un algoritmo de consenso para que varios procesos puedan llegar a un acuerdo sobre cierta variable que ellos definen. Indique cuántas réplicas son necesarias para garantizar que el algoritmo funcione si: a) se cae 1 proceso, b) se caen 2 procesos.

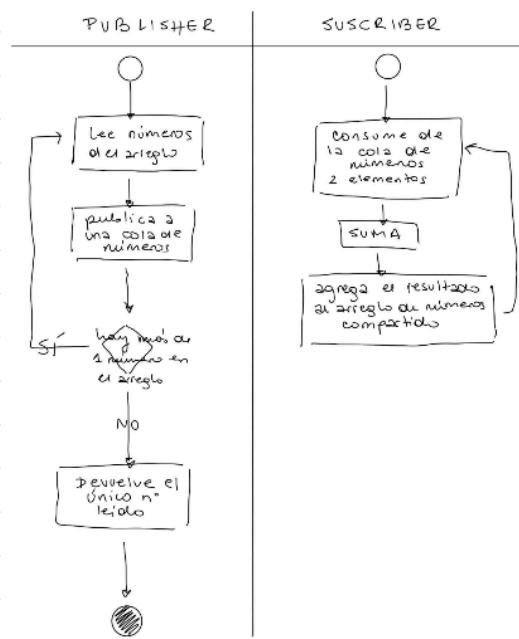
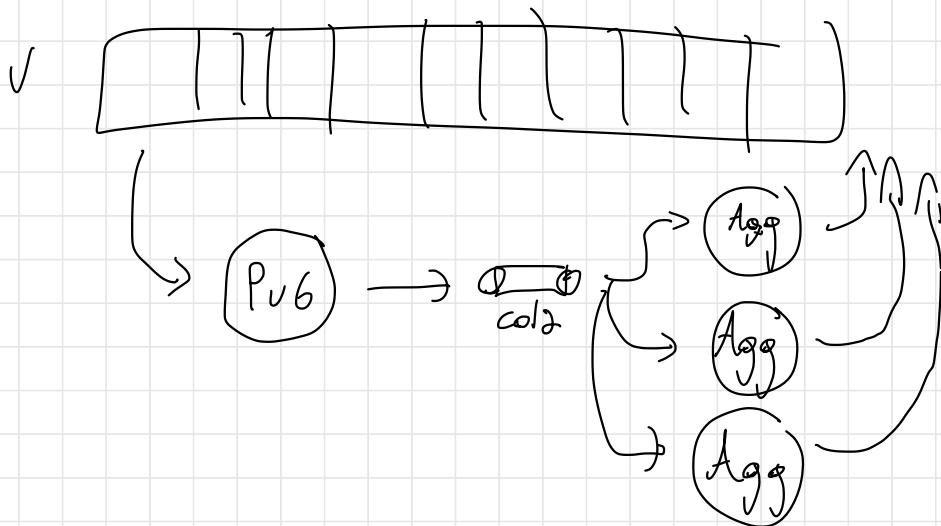
$$N \geq 2f + 1 \quad \text{PAXOS}$$

- a) Si se cae 1 proceso ($f=1$), se necesitan $N \geq 3$ nodos
- b) $N \geq 5$

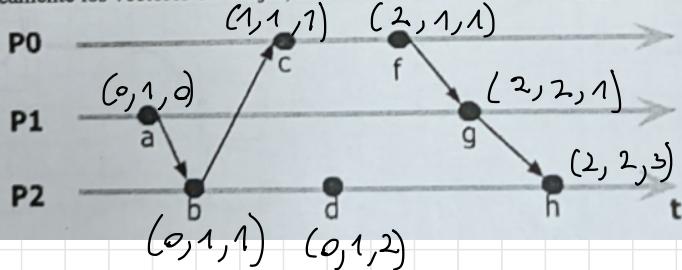
Paxos es un algoritmo de consenso para este tipo de cosas. Funciona de esta manera:

- 1) Cliente realiza un request (v)
- 2) Propositor recibe el request y envía un propose (N) a los Acceptors, donde N es un ID
- 3) Acceptors envían un promise (N, v) con el ID más grande que hayan recibido
- 4) Si Propositor recibe promise de la mayoría de los Acceptors, envía un propose (N, v) a los Acceptors
- 5) Si la promesa se mantiene se envían accept a los Learners y al Propositor
- 6) Learner le responde al cliente

8) Utilizando una arquitectura Publisher-Subscriber, calcule la suma de todos los enteros incluidos en un vector de tamaño N, optimizando el uso de P procesos conocidos como disponibles. Utilice pseudocódigo o diagrama UML (secuencia, colaboración o actividades).



9) Dado el siguiente diagrama de tiempo, calcule el valor del vector de relojes lógicos para cada evento. Utilizando únicamente los vectores de relojes, establezca la relación: entre b y g, entre d y g. Justifique.



$$B \rightarrow G \rightarrow C(B) < C(G)$$

$$(0,1,1) < (2,2,1) ?$$

$$0 < 2 \quad \checkmark \quad 1 < 2 \quad \checkmark \quad 1 < 1$$

los eventos son concorrentes

$$D \rightarrow G \Rightarrow (0,1,2) < (2,2,1) ?$$

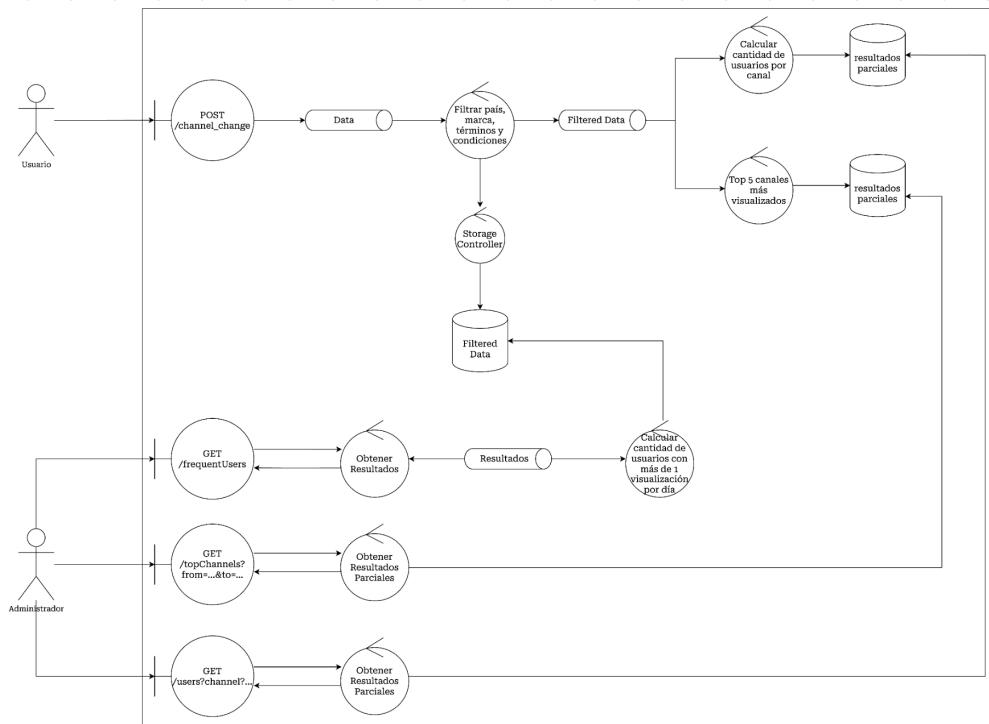
$$0 < 2 \quad \checkmark \quad 1 < 2 \quad \checkmark \quad 2 < 1$$

los eventos son concorrentes

10) Diseñe una arquitectura que asegure la escalabilidad para un sistema con los siguientes requisitos:

- Recibir usuario, timestamp y cambios de canales realizados en Argentina y Brasil por usuarios que poseen Smart TV de la marca LG y aceptó los términos y condiciones que incluyen monitoreo de uso.
- Consultar la cantidad de usuarios actualmente sintonizando cierto canal.
- Consultar los 5 canales más visualizados por usuarios (mayor cant. de minutos) en cierto rango de días.
- Consultar cantidad de usuarios con más 1 visualización por día.

Resuelva el problema mediante un diagrama de vista física y su explicación.



20220809

- 1) Indique las ventajas y desventajas de utilizar comunicación por sockets TCP frente al pasaje de mensajes por colas mediante MOM en un sistema distribuido típico.

TCP

- + P2P - To - peer
- + Respuesta rápida
- + Control
- + Poco overhead
- Escalabilidad
- Alto acoplamiento

Colas mom

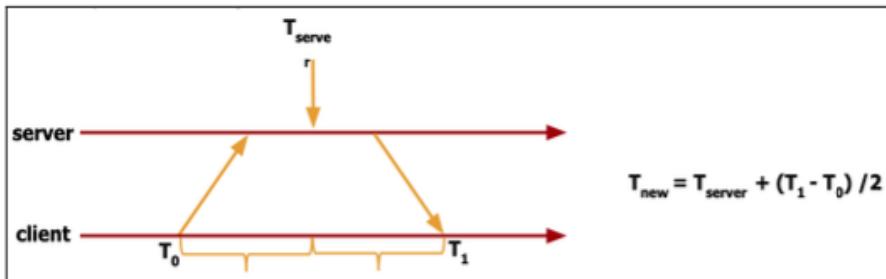
- + Publisher - Subscriber
- + Escalable
- + Tolerancia a fallas
- + Persistencia
- + Balanceo
- + Control de entrega
- Mucho overhead
- Mayor latencia

2) ¿A qué se refiere el concepto de drift de un reloj? Explique algún algoritmo para mitigar su efecto.

Quieres decir que con el paso del tiempo, un reloj puede desincronizarse con la hora real, ya sea por falta de precisión, errores, etc.

El algoritmo de Cristian sirve para corregir el drift.

Consiste en enviarle un mensaje a un servidor pidiéndole la hora, y cuando recibe una respuesta puede calcular el delay y ajustar su reloj.



3) En qué se diferencia la exigencia de Orden Total vs Orden Causal para una serie de eventos distribuidos. Dé un ejemplo donde se respete el primero y a su vez se viole el segundo.

En el orden causal, todo mensaje que implique la generación de otro mensaje es entregado manteniendo la secuencia.

En el orden total, si dos o más procesos envían un mensaje al mismo tiempo, todos los receptores deben recibir estos mensajes con el mismo orden.

Si $\text{delivery_1} \rightarrow \text{delivery_2}$ pero se impone un orden total de forma que

receptor_1 recibe message_2 y message_1

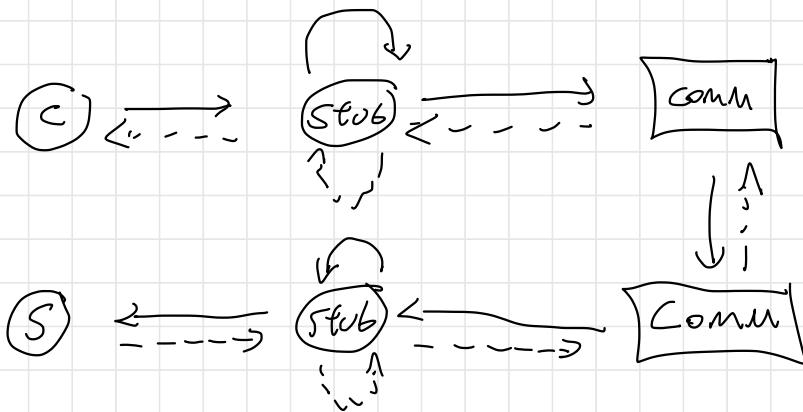
receptor_2 recibe message_2 y message_1

se viola el orden causal.

4) Explique cómo funciona el modelo RPC para arquitecturas distribuidas incluyendo el flujo de comunicaciones para un ejemplo propuesto.

El modelo RPC consiste en un modelo cliente-servidor en el cual el cliente realiza una llamada a un procedimiento y el servidor responde con el resultado.

Es transparente para el usuario y para ello requiere el uso de IDL.



Tanto el cliente como el servidor cuentan con un stub y un módulo de comunicaciones. El stub se encarga del marshall y el módulo atencore al stub de la comunicación.

5) ¿Qué componentes requiere una arquitectura para brindar elasticidad? Describa una situación donde este comportamiento sea deseable.

- Simple
- Fácil de escalar
- Autoescalación
- Load Balancing
- Monitoreo automático

Sistema de procesamiento de pagos con una colas

6) Considerando un escenario de particionamiento de datos en un sistema distribuido, explique al menos dos alternativas de enruteamiento de una consulta para poder ubicar la información deseada.

Particionamiento horizontal

Se dividen las filas en distintas particiones

element_id , fecha, precio

1	..	10]	Partición 1
1	..	20		
2	..	15]	Partición 2
2	..	100		

Vertical : Se dividen las columnas

element_id , fecha, precio

1	..	10]	Partición 1
1	..	20		
2	..	15		
2	..	100		Partición 2

Alternativa 1

Request a una partición , y este redirigir a la partición correcta

Alternativa 2

Request a un router , luego a la partición

7) Explique un algoritmo de exclusión mutua distribuida indicando sus ventajas y desventajas.

Centralizado

- Único punto de falla

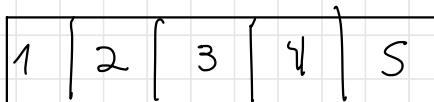
- Cuello de botella

+ Fácil

+ Intuitivo

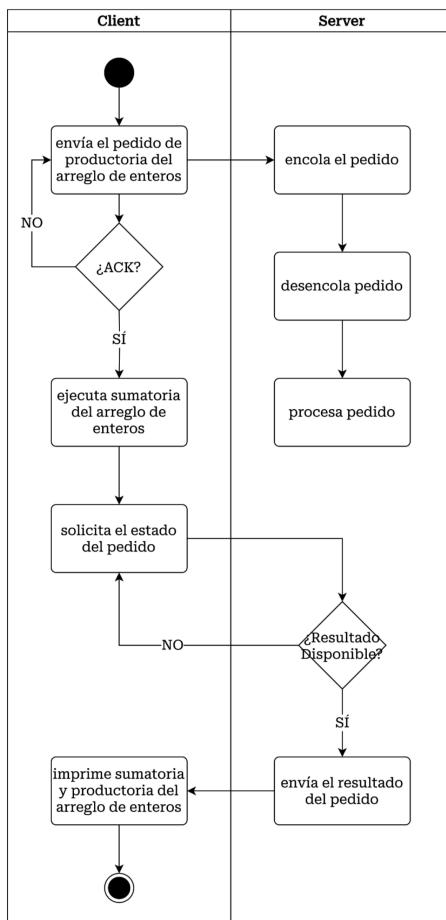
+ Pocas conexiones

- 8) Utilizando una arquitectura Request-Reply, ejecute de forma remota la productoría de un arreglo de enteros de manera asíncrona. Mientras la ejecución transcurre, calcule la sumatoria. Por último, imprima ambos resultados. Utilice pseudocódigo o diagrama UML (secuencia, colaboración o actividades).



$$\text{Server} \leftarrow 1 + 2 + 3 + 4 + 5 = 15$$

$$\text{Client} \leftarrow 1 + 2 + 3 + 4 + 5 = 15$$



9) Ejemplifique el uso de Direct Acyclic Graphs (DAGs) al diseñar el procesamiento de datos de todas las ventas diarias de una cadena de tiendas de ropa. Cada venta posee ID de tienda y un arreglo con: monto unitario, cantidad y código de producto. Se pretende obtener los 10 códigos de producto más vendidos, el monto promedio de venta y la cantidad de ventas con más de 3 productos.

Venta = ID tienda , monto , cantidad , código de producto

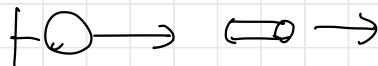


10) Diseñe una arquitectura que asegure la escalabilidad para un sistema con los siguientes requisitos:

- Recolectar la información de ventas de pasajes de trenes de larga distancia de toda Argentina, desde las distintas estaciones que funcionan como puntos de venta para los servicios que prestan.
- Recibir listado actualizado de servicios a prestar indicando origen, destino, fecha y Cant. de asientos.
- Recibir pedido de compra con destino, fecha de partida, nombre de cada pasajero y foto de cada DNI.
- Permitir consultar fotos de DNI de pasajeros incluyendo origen y destino si se posee la fecha de partida.

Resuelva el problema mediante un diagrama de vista física y su explicación.

Post /venta



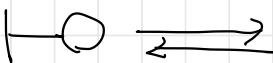
ESTACIÓN
DE VENTA

GET /servicios



filter

GET /pedido



GET /fotos



AGENTE



vertical



servicios



fotos

20230725

- 1) Explique la necesidad del uso de protocolos binarios y no-binarios para la comunicación en un sistema distribuido. De un ejemplo de protocolos conocidos en estas dos categorías.

Binarios

- Son rápidos
- Pegan poco
- Tamaño fijo
- Recomendable si se intercambian muchos mensajes
- Requiere un decoder

Protocolo

No-binarios

- JSON, XML
- Más fácil de usar
- Más lento

2) ¿Qué es un grupo de comunicación? ¿Qué utilidad presenta este concepto a la hora de diseñar un sistema escalable?

Es un conjunto de procesos comunicados entre si. Es una abstracción.

Es útil porque son dinámicos, un proceso se puede suscribir o desuscribir de un grupo.

Para la comunicación se utilizan los siguientes métodos

. Multicast: a todo el grupo

. Anycast: a uno del grupo

. Broadcast: a todo el sistema

. Unicast: 1 - 1

3) Defina los conceptos de Estado Local y Estado Global de un sistema distribuido. Explique un algoritmo para realizar cortes consistentes de estado global.

El Estado Global es el comportamiento de varios estados locales que hacen referencia a un proceso del sistema

$$S = S_0 \cup S_1 \cup \dots \cup S_n \quad \text{para } n \text{ procesos}$$

Para que un corte sea consistente, para cada estado dentro del corte, se debe encontrar también los estados anteriores.

El algoritmo de Chandy y Lamport sirve

Explicado arriba

4) Explique HDFS (Hadoop Distributed File System) y desarrolle los conceptos clave utilizados como factores de diseño.

HDFS es un sistema de archivos distribuido diseñado para hardware de bajo costo. Se lo considera un almacenador de datos en vez de file system porque no implementa POSIX.

- Tolerancia a fallos : se adopta en vez de prevenir
- Volumen y latencia : favorece el streaming
- Portabilidad : hardware de bajo costo
- Performance : favorece operaciones de lectura

5) Explique un algoritmo de exclusión mutua distribuida. Indique ventajas y desventajas del mismo en un caso concreto.

CENTRALIZADO

o TOKEN Ring

6) ¿Qué propone el teorema CAP? Según el teorema, ¿Es posible garantizar la tolerancia a fallos de un sistema distribuido?

Consistency: consistencia de las respuestas

Availability: capacidad para responder pedidos

Partition tolerance: capacidad seguir funcionando si un grupo de nodos falla.

Según el teorema solo se pueden asegurar 2 de los 3.

7) Explique los términos falla (fault), error y avería (failure) y cómo estos se relacionan con el concepto de tolerancia a fallos.

Fault (falla) → condición no cumplida

error → estado incorrecto en el sistema

Failure → comportamiento incorrecto de cara al usuario

Falla → error → failure
Error → avería

Un rayo cósmico cae sobre un equipo y causa una **FALLA** en el sistema, cambiando un bit de la memoria RAM. El cambio de este bit provoca un **ERROR** pues hace que una variable del heap apunte a una dirección inválida en memoria. El sistema se **AVERÍA**, muestra un cartel de bloqueo al usuario y deja de funcionar.

8) Defina con pseudocódigo los pasos para extraer \$1000 de una cuenta bancaria de número CA-1234 y acreditarlos en otra de número CA-5673 asumiendo la existencia de una plataforma de Distributed Objects. Realice un gráfico que exemplifique la arquitectura de Distributed Objects y estado de la operación luego de ejecutarla.

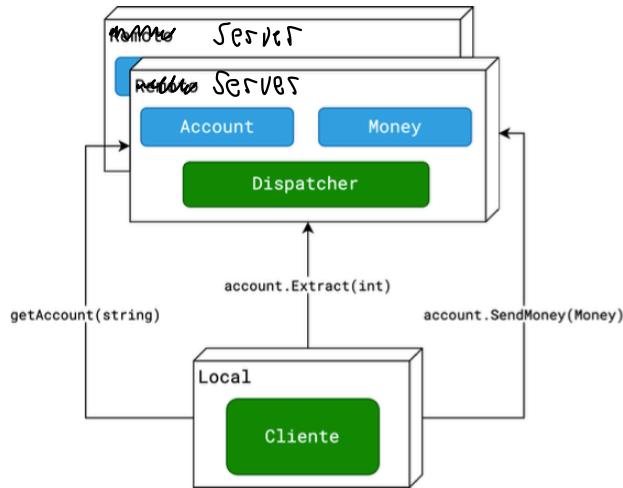
```
distributedBankService.go

srv = getBankService()
account1 = srv.getAccount(CA-1234)

extracted, err = account1.Extract(31000)
if (err != nil) {
    throw("No hay suficiente dinero en la cuenta de origen")
}

account2 = srv.getAccount(CA-5673)

success = account2.SendMoney(extracted)
if (!success) {
    account1.SendMoney(extracted)
    throw("No se pudo realizar el depósito en la cuenta de destino")
}
```



- 9) Utilizando map-reduce para distribuir la carga, procese los eventos de log de una aplicación Web mediante un listado de tuplas (email, url_navegada, fecha) y calcule:
- Cantidad de navegaciones detectadas por día de semana (lu, ma,..., do)
 - Listado de emails con más de 10 navegaciones registradas.
- Utilice pseudo-código e indique supuestos de ser necesario.

a.

```
def map(list values):  
  
    // day → MON, TUE, WED, THU, FRI, SAT, SUN  
  
    // value → (email, url_navegada, fecha)  
  
    for v in values:  
  
        day = getDay(v.fecha)  
  
        emitIntermediate(day, 1)  
  
def reduce(string key, list values)  
  
    // key → MON || TUE || WED || THU || FRI || SAT || SUN  
  
    // values → lista de 1s asociados a la key  
  
    emit(key, len(values))
```

b.

```
def map(list values):  
  
    for v in values:  
  
        emitIntermediate(v.email, 1)  
  
def reduce(string key, list values)  
  
    // key → xxx@xxx.com  
  
    if len(values) >= 10  
  
        emit(key, len(values))
```

10) Diseñe una arquitectura que asegure la escalabilidad para un sistema con los siguientes requisitos:

- Recibir mensajes whatsapp incluyendo campos con tipo (texto o audio), contenido (string o binario), teléfono origen y nombre de usuario para procesar las consultas del canal oficial de Anses.
 - Procesar todos los mensajes de audio para obtener el texto a partir del audio recibido.
 - Identificar palabras clave y usar tablas p/calc. área, siendo 1) Jubilaciones y Pensio., 2) Asignaciones Fam., 3) Otras
 - Crear un ticket con el contenido mediante POST en sistema externo y retornar ID al usuario.
 - Permitir la consulta de tickets por número, con cierta palabra clave y por área en cierto rango de fechas.
- Resolver el problema mediante análisis de volumen, endpoints y vista física con su explicación.

2021 0108

1) ¿En qué consiste el modelo de Work-Span para análisis de paralelización de tareas y camino crítico?

El modelo work-span parte de la premisa que NO todos los tareas paralelizables se pueden ejecutar al mismo tiempo.

Propone una cota superior e inferior para el speedup

T_1 : Tiempo de ejecución con un proceso

T_{∞} : Tiempo de ejecución con infinitos procesos

$$T_p \geq \max \left(\frac{T_1}{p}, T_{\infty} \right)$$

Sirve para evaluar que tan paralelizable ej. un algoritmo

2) ¿Qué se entiende por "Atomicidad de Mensajes" en el contexto de grupos de comunicación? ¿Con qué estrategias es posible garantizarla?

Atomicidad quiere decir que un mensaje tiene la garantía de ser entregado a todos o a ninguno.

Para garantizar la atomicidad:

- Uso de ACK
- Reinicios ante caídas o fallos de ACK

3) Defina 2 protocolos distintos para la transmisión de paquetes considerando un registro que posee un int y un arreglo de flotantes. Compare ventajas y desventajas de cada protocolo.

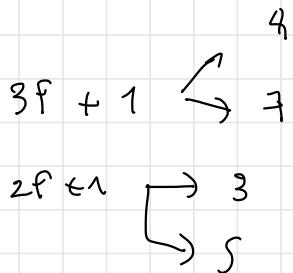
BINARIO :

Texto :

5) Describa un algoritmo de exclusión mutua distribuida que no utilice un servidor central.

Token ring

- Se construye un anillo entre los procesos, cada uno se conecta a dos procesos
- Cuando un proceso tiene el token puede acceder a la sección crítica.
- Cuando termina lo pasa al siguiente



8) Detalle un algoritmo de consenso que permita que 3 procesos concuerden en el valor de altitud de un dron en pleno vuelo. Muestre paso a paso cómo funciona el algoritmo considerando que los procesos A,B miden 10m mientras que C mide 9m.

9) Dada una librería basada en request-reply, realice un pequeño sistema que realice la tarea de calcular el hash de un archivo binario leyendo sus bloques de 4 bytes y aplicando sucesivamente la operación XOR a medida que se leen más bloques. Distribuya la carga de cómputo entre N servidores conocidos. Utilice pseudocódigo o diagrama UML (secuencia, colaboración o actividades) para explicar la solución e indique supuestos de ser necesario

10) Diseñe una arquitectura que asegure la escalabilidad para un sistema con los siguientes requisitos:

- Permitir la asignación de Aires Acondicionados inteligentes en LATAM para la captura de datos: fecha última medición, minutos en uso desde última medición, temperatura medida, ubicación aproximada (lat, long).
- Asegurar la captura de datos con una frecuencia de 10 minutos.
- Permitir la consulta de un reporte con ciudades conocidas y temperatura min, max, avg medidas en las últimas 24hs.
- Permitir la generación de una alarma y reporte por email al dueño del dispositivo si se detecta uso excesivo: más de 12 horas de uso durante las últimas 24 horas.

Resuelva el problema mediante análisis de volumen, endpoints y vista física con su explicación.

20240716

- 1) Suponga que debe implementar un protocolo de health-check de server a clientes mediante sockets
¿Qué condiciones se deberían cumplir para que sea más conveniente utilizar UDP frente a TCP?

- Logica de reintentos
- Esperar un ACK
- No es necesario mantener una conexión abierta entre cliente y servidor

2) Explique la utilidad de los algoritmos de relojes lógicos de Lamport y de vectores de relojes (lógicos) explicados en clase. ¿Para qué sirve cada uno?

Los relojes de Lamport y vectores de relojes sirven para saber en qué orden ocurren los eventos.

En el caso de Lamport, se utilizan escalares en cada proceso.

Los vectores permiten mapear todo el sistema y comprender entre dos o más procesos, qué evento vio primero.

Lamport no sirve para concurrencia porque captura poca información.

3) Describa cómo implementaría un middleware que garantice la comunicación de grupo con las siguientes condiciones: a) sólo existen tres grupos: x, y, z. b) los miembros de los grupos se pueden unir en cualquier momento para recibir mensajes. c) sólo se permite enviar grupos a "todos" los miembros. (*tu grupo*)
mensajes

Un MOM centralizado como rabbit.

4) Explique el beneficio de la replicación de datos y desarrolle una estrategia de implementación.

- Backup en caso de fallos
- Distribución de carga
- Distribución geográfica para disminuir latencia

LEADER BASED

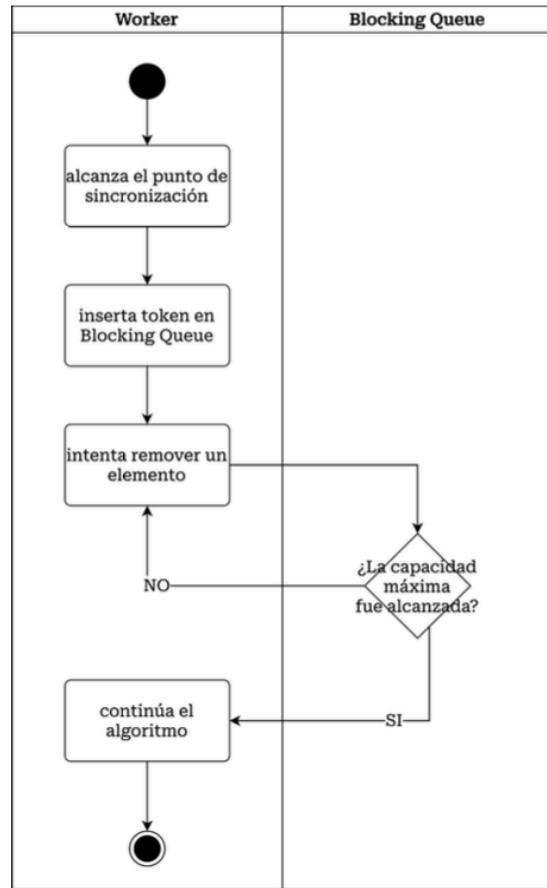
- Writes ocurren en la DB líder
- Se replica en los replicas
- Read en los replicas

7) El estudio de sistemas confiables (*dependable*) implica la revisión de varias propiedades que impactan en la tolerancia a fallos del sistema. Detalle los conceptos clave de al menos una de ellas.

- Availability : Siempre está disponible para responder
- Safety : Capacidad para prevenir errores
- Maintainability : Capacidad para ser mantenido
- Reliability : Confianza ante fallos

8) Utilizando **blocking-queues**, implemente una barrera para que 3 procesos llamados workers puedan sincronizarse en cierto punto del algoritmo que ejecutan. Detalle supuestos de ser necesario. Utilice pseudocódigo o diagrama UML (secuencia, colaboración o actividades).

Una blocking queue puede tener una capacidad máxima, y cuando llega a esa capacidad no te deja insertar elementos.



2029.12.19

4) Asumiendo la existencia de un equipo de hasta 10 agentes de atención al cliente en una empresa telefónica, defina cómo implementar un **middleware distribuido** que permita a los agentes llamar al siguiente número en fila. ¿Dónde almacenaría la información relacionada al último número llamado?

Utilizaría un middleware con sockets. Cada agente tiene una cola adicional y el load balancer que recibe las requests las distribuye.

El último numero de llamado se puede guardar dentro del middleware en un storage controller.

8) Utilizando **request-reply**, implemente una barrera para que 3 procesos llamados workers puedan sincronizarse en cierto punto del algoritmo que ejecutan. Detalle supuestos de ser necesario. Utilice pseudocódigo o diagrama UML (secuencia, colaboración o actividades).

Tres procesos envían un request a un server y quedan bloqueados hasta recibir una respuesta.

Request - reply sincronización

20250213

1. ¿Qué es el **throttling** o descarte de paquetes? ¿Tiene efectos negativos?

El throttling ocurre cuando hay una sobrecarga en un sistema, y entonces este limita la cantidad de paquetes procesados descartando algunos o bajando el flujo de datos.

- Se puede perder información
- Puede tardar más en llegar las respuestas

2. Describa los conceptos clave de Network Time Protocol (NTP)

usado para sincronización de relojes en sistemas distribuidos. ¿Por qué se dice que este protocolo es escalable?

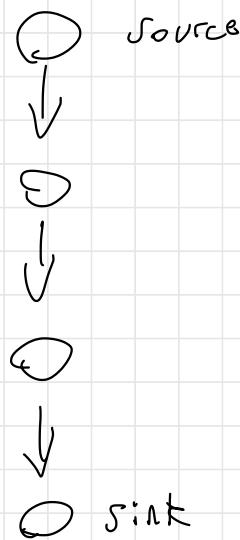
Es el protocolo que se utiliza para sincronizar relojes.

El estrato 0 es el master clock. Y la información se propaga hacia abajo.

El protocolo es escalable porque se pueden sumar clientes a un estrato y sincronizarse con los de arriba.

5. Describa el modelo de Pipeline-Filters para procesamiento de datos. Describa cómo funcionan los modelos de Worker por Filter y Worker por Dato.

Un pipeline implica una serie de workers conectados.



Donde cada worker funciona como filtro.
Realiza una tarea
y se lo pasa al siguiente

En worker por ítem, cada worker realiza todas las tareas para un solo ítem.

6. Describa cómo se implementa la redundancia para la tolerancia a fallas. Dé un ejemplo en donde esta estrategia garantice tolerancia a fallos y un ejemplo en donde no se garantice.

La redundancia puede ser

- físico → HW
- de información → replicas
- de tiempo → reintentos

20250220

3) El contrato de una interfaz puede estar orientado a entidades o a procesos. Describa ambos modelos y dé un ejemplo de al menos dos endpoints de cada estilo para una API.

Orientado a Entidades:

- Expone los datos como recursos accesibles mediante operaciones CRUD (Create, Read, Update, Delete)
- **Modelo más cercano a la estructura de la base de datos** y representa directamente las entidades del dominio
- Ejemplos
 - GET /clientes/{id}
 - POST /pedidos

Orientado a Procesos:

- **Expone acciones o flujos de negocio que reflejan cómo se utilizan los datos en un contexto real**
- Ejemplos
 - POST /pedidos/{id}/confirmar
 - GET /clientes/{id}/compras?desde=2024-01-01&hasta=2024-12-31

4) ¿Cómo se calculan los 9s de disponibilidad de un sistema con 2 de frontend idénticos, accedidos por los usuarios y que se comunican con 2 nodos de backend idénticos. Asuma $P(\text{availability}) = 0,95$ para cada nodo y que el balanceo de carga es transparente con load balancers que no se caen.

$$P(\text{av}) = 1 - P(\text{failure})$$

$$\begin{aligned} P(\text{failure_back}) &= P(\text{failure_back1}) \cdot P(\text{fail_back2}) \\ &\approx 0,05 \cdot 0,05 = 0,0025 \end{aligned}$$

$$P(\text{back_av}) = 1 - 0,0025 = 0,9975$$

$$P(\text{front_av}) = 0,9975$$

$$P(\text{global_av}) = 0,9975 \cdot 0,9975 \approx 0,9950 \approx 99,5\%$$

20250227

8) Detalle un algoritmo de consenso que permita que 4 servicios concuerden en el valor de tasa de cambio dolar a publicar luego de obtener mediciones independientes de distintas fuentes de datos.
¿Cómo funciona cada paso del algoritmo considerando mediciones: a. 1010, b. 1000, c. 1000, d. 1030?

En este caso conviene utilizar un algoritmo sincronico.

Paso 1: cada proceso envia su valor

Paso 2:

10) Diseñe una arquitectura que asegure la escalabilidad para un sistema con los siguientes requisitos:

- Recibir capturas de cámaras de ingreso/egreso en puertas de acceso de una cadena de Shoppings.

- Detectar ingreso o egreso de los empleados considerando una base con fotos de legajo previa.

- Determinar Cant. de minutos trabajados por cada empleado utilizando la info de ingreso/egreso.

- Cada 60 minutos se debe imprimir los legajos y nombres de todos los empleados presentes.

Resuelva el problema mediante lista de endpoints, análisis de volumen y vista física con su explicación.

POST /capturas

body: { tipo: string, foto: bytes[] }

GET /empleado /ingresos
/empleado /egresos

GET /empleado /minutos

GET /empleados

Análisis de volumen

- 100 empleados
 - 90% de asistencia → 90 empleados → 180 capturas
 - 60% come afuera → 60 empleados → 120 capturas
 - 20% fuma dos veces por día → 20 emp → 40 capturas
= 340 capturas
- ↓
- 340 m² de fotos

Vista física

