

DASE: Actividades Tema 4

ACTIVIDAD 1

Sea un sistema distribuido homogéneo formado por N nodos idénticos. Para equilibrar la carga en estas máquinas se utiliza un algoritmo centralizado. Cualquier proceso puede ser migrado de una máquina a otra en cualquier momento, pues todas ellas tienen copias idénticas de los recursos necesarios. En uno de los ordenadores se ejecutará un proceso que coordina este algoritmo. Cada máquina le envía cada medio minuto sus valores de carga actuales. Para ello, en el mensaje correspondiente se incluye esta información:

- Número de procesos activos, programa que ejecuta cada uno de ellos y lista de los recursos que necesita cada proceso.
- Utilización media de la CPU durante ese último medio minuto.
- Ancho de banda utilizado durante esos 30 últimos segundos en esa máquina, incluyendo la lista de procesos que han accedido a la red y el ancho de banda utilizado por cada uno.
- Porcentaje de memoria principal y memoria de intercambio libre en el momento de enviar el mensaje, así como el consumo de memoria de cada uno de los procesos.

El proceso coordinador almacena todos los mensajes recibidos por parte del resto de nodos. Cuando los ha recibido todos, pasa a comparar las cargas de cada uno y, si las diferencias entre el nodo más sobrecargado y el más ligero superan cierto umbral, procede a migrar un proceso de uno a otro. Una vez programada una migración, reordena los procesos en función de su carga prevista tras la migración y determina si se necesita alguna migración más. El equilibrado finalizará tan pronto como la comparación realizada en la última iteración no llegue al umbral mencionado.

Responda a las siguientes cuestiones:

- a) Proponga un algoritmo descentralizado para realizar esa misma tarea de equilibrado de carga.

- b) ¿Qué cree que ocurrirá en el algoritmo original si en alguno de los intervalos de evaluación y migración varios nodos han tenido que aceptar varios procesos locales nuevos o alguno de los que estaba previamente en ejecución ya ha finalizado? ¿Se soportaría mejor esa situación en el algoritmo descentralizado que ha generado en el apartado anterior?

- c) Si todos los procesos en ejecución en ese sistema fueran servicios altamente disponibles y la carga a soportar fuera constante, ¿habría que preocuparse por la situación mencionada en el apartado anterior?

- d) ¿Qué ventajas e inconvenientes plantea una versión descentralizada frente a una versión centralizada en la solución de este problema de equilibrado?

- e) ¿Cómo debería gestionarse este problema de equilibrado en un sistema que facilitara servicios distribuidos elásticos y donde la carga variara con alta frecuencia? Sugiera al menos dos o tres principios a respetar en la solución utilizada. No es necesario que describa en detalle una solución concreta.

ACTIVIDAD 2

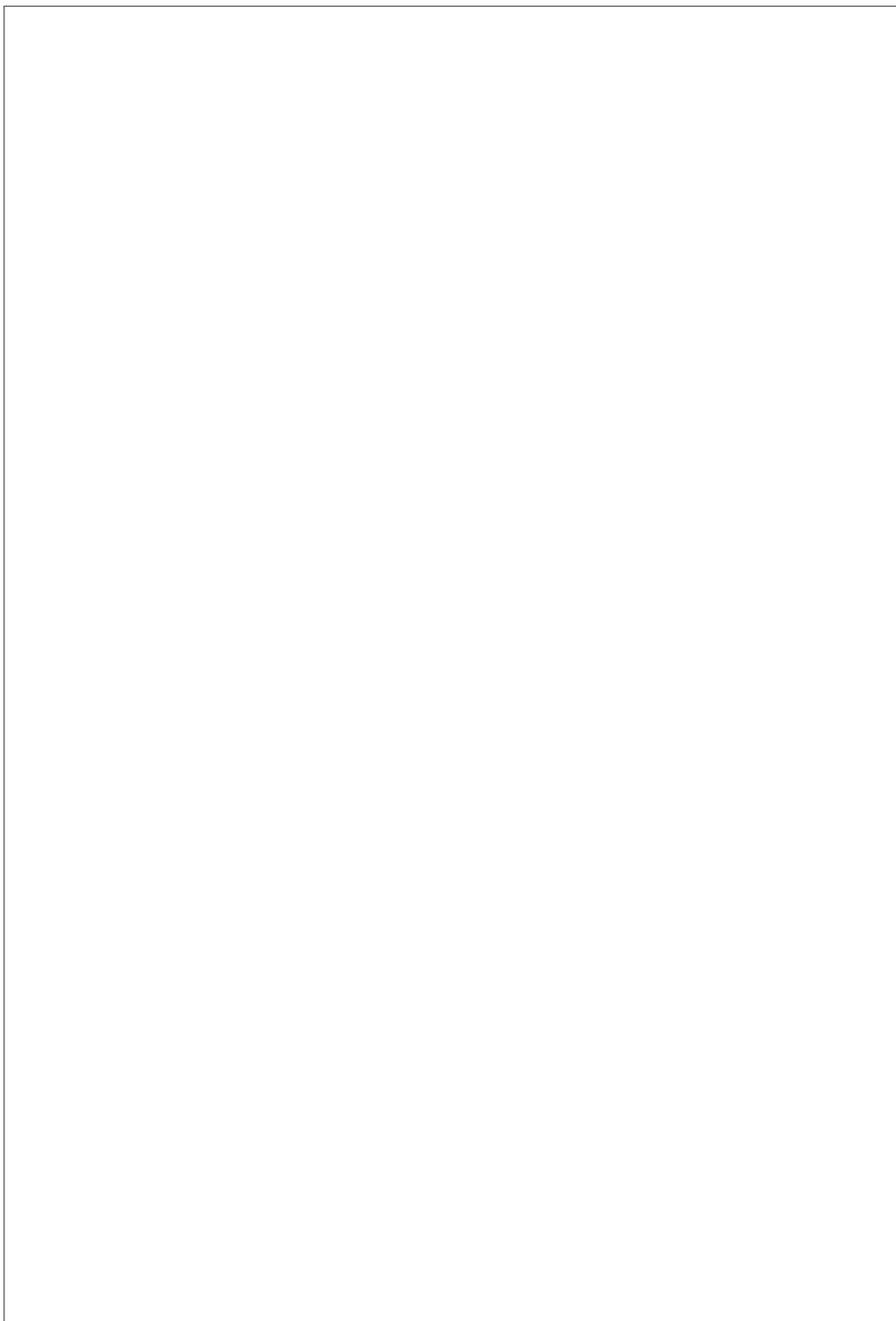
Un servicio escalable suele estar implantado mediante componentes replicados. Estos componentes deben utilizar algún mecanismo de comunicación para propagar la información entre sus diferentes réplicas y hacia otros componentes de la aplicación. En el caso de la comunicación entre réplicas de un mismo componente, suele ser necesario el uso de algún servicio de difusión de mensajes.

En función del modelo de consistencia que deba implantarse, estas difusiones pueden ser de diversos tipos:

- a) De orden total FIFO, para lograr consistencia secuencial. Existirá un middleware de comunicación. El emisor puede utilizar comunicación asincrónica, sin bloquearse al enviar el mensaje. El middleware puede utilizar un algoritmo basado en secuenciador. Todos los mensajes, enviados por cualquier nodo, son propagados a un proceso secuenciador. El secuenciador los etiqueta con su número de secuencia y los redifunde al grupo de réplicas. Cada réplica recibe cada mensaje y los entrega en el orden de sus números de secuencia.
- b) De orden total FIFO, para lograr consistencia secuencial. En este caso se organizan todos los nodos en un anillo lógico y existe un token que circula por ese anillo. El proceso/réplica que posea el token puede difundir a todos los demás todos los mensajes que tenga en su cola de envío, numerándolos secuencialmente (tomando como origen el último mensaje difundido por el anterior poseedor del token). Cuando vacíe su cola de envío, propagará el token al siguiente proceso en el anillo. Mientras un proceso no tenga el token, va depositando sus mensajes en la cola de envío local. De nuevo, la acción de envío puede considerarse asincrónica. El proceso emisor podrá tener un hilo adicional encargado de esperar el token y difundir los mensajes de la cola, permitiendo que los hilos emisores continúen con su trabajo sin necesidad de bloquearse.
- c) De orden causal, para lograr consistencia causal. En cada mensaje difundido “m” se incluyen también todos los mensajes previos dentro del orden causal para los que todavía no se tenga confirmación de recepción por parte de los demás procesos del grupo. Cada proceso receptor examinará esa lista de mensajes previos e irá entregando todos aquellos que no hubiese recibido anteriormente. Una vez hecho esto, pasará a entregar “m”. La primera versión del algoritmo CBCAST del sistema Isis utilizó este principio. Observe que para construir esa colección de mensajes que acompañan a “m” basta con manejar información local. No se necesita preguntar nada a ningún otro proceso.

Responda a las siguientes cuestiones:

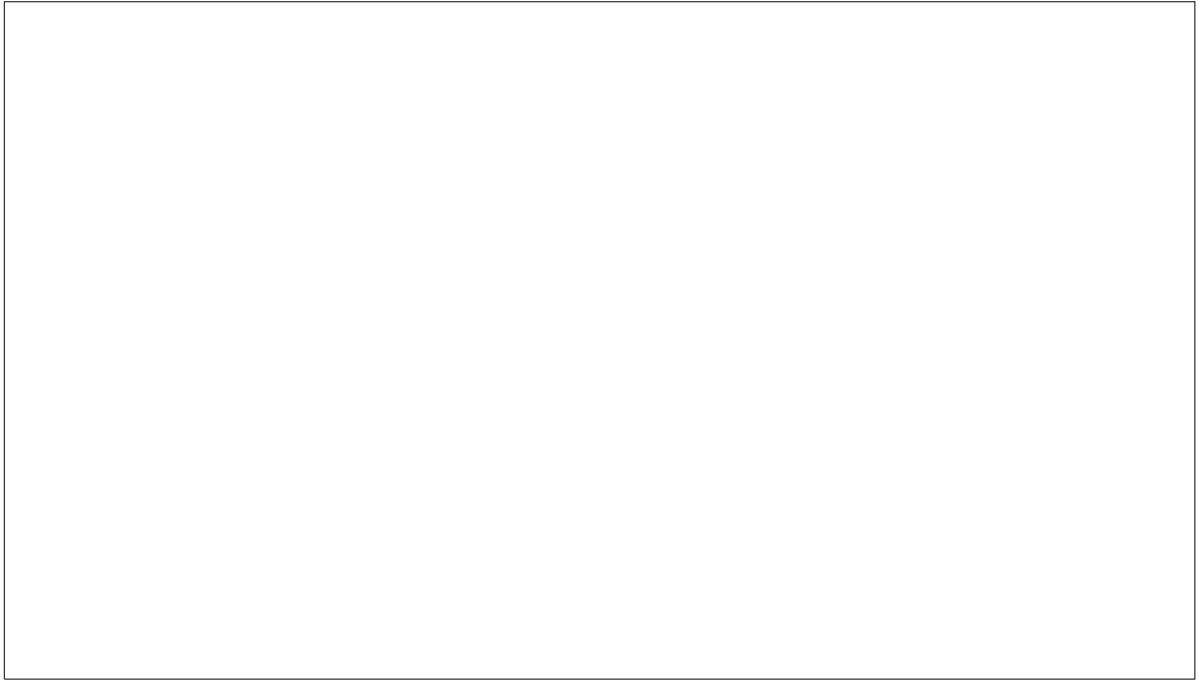
1. Analice qué propiedades de los algoritmos descentralizados cumple cada uno de los algoritmos de difusión presentados en estos apartados.

A large, empty rectangular box with a thin black border, occupying the central portion of the page. It is intended for students to write or draw their responses to the activities listed in the footer.

2. Indique qué extensiones se necesitaría realizar en cada algoritmo para soportar las situaciones de fallo en el grupo de réplicas que implantan un componente. Asuma fallos de parada que solo afecten a una de las réplicas.

3. Indique qué algoritmo de los tres que se han presentado llegaría a ser más eficiente en un sistema formado por únicamente tres nodos. Justifique su respuesta.

4. Ídem para un sistema formado por 300 nodos.

A large, empty rectangular box with a thin black border, intended for the student's answer to question 4.

5. Según lo visto en los dos apartados anteriores... ¿qué algoritmo podría considerarse más escalable?

A large, empty rectangular box with a thin black border, intended for the student's answer to question 5.

ACTIVIDAD 3

La biblioteca ZeroMQ permite realizar diferentes tipos de comunicación distribuida basada en intercambio de mensajes bajo un modelo de interacción asincrónica. Esta biblioteca no proporciona comunicación persistente y no utiliza ningún servidor de comunicación: solo intervienen los procesos que utilicen la biblioteca. Uno de los patrones de interacción que implanta es el “publish/subscribe”.

Compare ZeroMQ con la mejor alternativa seleccionada en el apartado 5 de la actividad anterior por lo que respecta a sus grados de asincronía en las interacciones, escalabilidad potencial y capacidad para gestionar las situaciones de fallo (en función del rol que ejecute el nodo parado: publicador, suscriptor...).

ACTIVIDAD 4

Ya que el principal inconveniente de las transacciones utilizadas en un SGBD relacional son los bloqueos prolongados que pueden surgir al utilizar concurrentemente varias transacciones conflictivas, un programador decide modificar las transacciones de una aplicación que está escribiendo. Para ello se plantea diversas alternativas:

1. Cambiar el nivel de aislamiento de las transacciones. En lugar de utilizar el nivel “*serializable*” que siempre se recomendó al estudiar la carrera, decide emplear el nivel “*read uncommitted*” pues la guía de su SGBD explica que en ese caso no se utilizará ningún tipo de “*lock*” para acceder a los datos.
2. Descomponer cada una de las transacciones que tenía la aplicación, de manera que tras cada una de las sentencias se realice el “*commit*”. Así las transacciones serán más breves y los intervalos de bloqueo más cortos.
3. Dejar las transacciones tal como las había escrito inicialmente, sin realizar ningún cambio. Esto supone seguir las reglas “tradicionales” y respetar las cuatro propiedades ACID.

Explique qué ventajas en cuanto a rendimiento podría comportar cada una de las alternativas, justificando por qué se generan esas mejoras. Desgraciadamente esas ventajas no son “gratuitas”. Explique también qué situaciones “incorrectas” podrían surgir al realizar esas “optimizaciones” y qué propiedades ACID se estarían perdiendo.

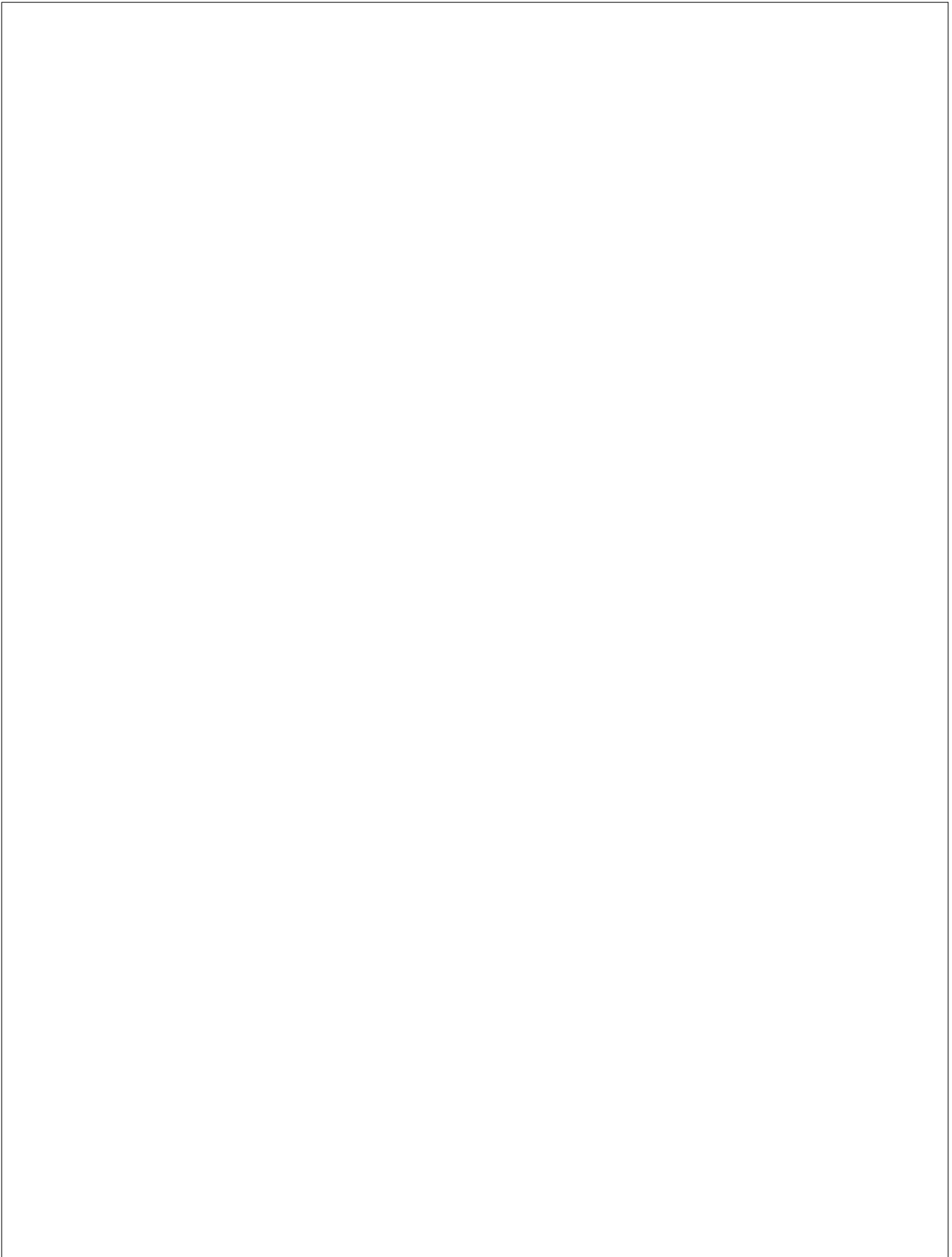
Para centrarlo en un caso concreto, asuma una aplicación de comercio electrónico y dos tipos de transacciones (T1 para que un usuario compre X unidades de un elemento Y; T2 para que un usuario devuelva X unidades de un elemento Y). Una carga habitual para esta aplicación estaría formada por un 80% de transacciones de tipo T1 y un 20% de transacciones de tipo T2, atendiendo un total de 100 transacciones por segundo sobre un almacén con 300 clases de elementos, pero donde el 70% de los accesos se realiza sobre un conjunto con solo 15 clases de elementos (los de mayor éxito de ventas). El “pseudocódigo” de estas transacciones sería:

TRANSACCIÓN T1:

```
UY := Unidades en almacén del elemento Y; // Lectura de la BD.
Si UY - X < 10 entonces incrementar precio de Y un 10%; // No afecta a la BD.
// Presentar coste al usuario y esperar confirmación.
Si cliente_confirma() y X < UY entonces UY := UY - X;
// Actualizar valor de UY en la BD.
// Gestionar cobro.
Si cobro_correcto() entonces Confirmar transacción
    sino Abortar transacción.
```

TRANSACCIÓN T2:

```
UY := Unidades en almacén del elemento Y; // Lectura de la BD.
// Presentar abono al usuario y esperar confirmación.
Si cliente_confirma() entonces UY := UY + X;
// Actualizar valor de UY en la BD.
// Gestionar pago al usuario.
Si orden_pago_correcta() entonces Confirmar transacción
    sino Abortar transacción.
```

A large, empty rectangular box with a thin black border, occupying the central portion of the page. It is intended for the user to write or draw their activities related to the topic.