



Universidad Nacional Autónoma de México
FACULTAD DE CIENCIAS

Computación Concurrente

Práctica 05

Aplicaciones de la Computación Concurrente

Profesor:

Salvador González Arellano

Integrantes:

Contreras Ibarra Francisco

Marín Parra José Guadalupe de Jesús

Ortega González José Ethan

Ramírez López Alvaro

1. Teoría:

Problema 1.

Zeratún en primer lugar en su pagina web tiene el problema, de que 2 personas (o más) pueden comprar el mismo tanque de gas pues los tiene en exhibición porque los decora entonces si importa que estén en exhibición, de estos pueden tener varios entonces lo que deben solucionar es lo siguiente:

1. Qué solo una persona compre el tanque de Gas.
2. Que cumpla con la propiedad de justicia, es decir, que se lo quede la primer persona que le dio click.
3. Al momento de hacer el cobro (o el incremento de ganancias) este correcto, pues anteriormente teníamos una condición de carrera.
4. Investiga de que manera se podría implementar, es decir, como se haría la pagina web que implementa esto (puedes usar un framework y poner la anotación en caso de usarse o lo puedes hacer sobre la BD o incluso sobre el Front), si quieren pueden llegar a hacer un mini implementación de esto.

Solución:

Lo primero que se puede comentar es que el problema que tiene Zerátún es un problema del tipo cliente servidor, el cliente que en este caso serán las personas que van a comprar los tanques de gas van a realizar peticiones HTTP al servidor, este servidor contendrá el Backend y el Frontend de la pagina web de Zerátún, por el momento la parte a la que le daremos mas énfasis sera al Backend de la pagina.

Una solución propuesta para el problema que tiene Zerátún con la venta de gas seria que en la lógica (Backend) de su pagina web se integrara el algoritmo de Szymański, este algoritmo al ser de exclusión mutua, cumple que solo una persona compre el tanque de gas, también como el algoritmo contiene una sala de espera en la cual estarán todos los procesos (en este caso podemos decir que los procesos serian las peticiones HTTP que manda a hacer el cliente), al momento de que los procesos entrarían a la sala de espera, estos pasarían a la sección critica (que en este caso es la compra) de forma ordenada, esto podríamos lograrlo asignando identificadores para cada proceso para saber cual es el que entro primero y después el segundo y así sucesivamente, esto solucionaría los problemas planteados anteriormente.

Lo que se tiene en mente en cuestión de implementar una solución seria basándonos en el Stack **MERN: (MongoDB, Express JS, React JS y Node JS)**, repetimos de nuevo, podemos validar que se cumpla la propiedad de justicia que se cumpla desde el Backend, pero es fundamental que se realice el diseño de los modelos de MongoDB para que se realice una implementación del código adecuada.

Problema 2:

A Zerátún le encanta la película de “Shrek 2”, por lo que decidió innovar en sus tanques de gas por lo que necesitan de su ayuda, en primer lugar , sus clientes le mandan diseños, pero cuando llegan se almacenan en una base de datos, pero lamentablemente cuando se les asigna un ID se pueden a llegar a sobrescribir y perder la referencia, por lo que les pide 2 cosas:

1. Proponer una solución para que no se sobrescriba, de tal manera que se conserven ambas imágenes.

Solución: Primero que Zerátún debe dejar ser tacaño para que pueda contratar a programadores que puedan hacer un buen trabajo en la página web de venta de tanques de gas donde para resolver el problema de que se sobrescriba el ID de los diseños que les mandan los clientes los programadores contratados podrían usar el algoritmo de el panadero donde cada vez que un cliente consuma un endpoint el cual haga que se guarde en la base de datos el diseño que dieron donde en la parte del service que es la encargada del procesamiento de la información, se tendrá que tener que tener un control de las peticiones se que hagan para agregar una imagen del diseño del cliente donde cada petición va a tener un turno de ser atendida conforme al tiempo en que se haya hecho la petición para lo cual la petición que quiera elegir un turno se van a usar unas banderas para decirle a las demás peticiones que se está escogiendo turno donde el número de turno que se le asigna a una petición será el número de turno más grande que tengan

las demás peticiones más 1 para lo cual después la petición que quiera guardar su imagen se comparara con las otras peticiones que hayan llegado antes que ella donde vera que si la petición n está eligiendo turno si es así se espera a que termine de elegir mediante una espera ocupada hasta que haya terminado para después se le preguntara que si el número de turno de la petición n es diferente de 0 si es 0 entonces el peticion n no tiene turno por lo que ya no nos interesa ya que esta petición logro guardar en la base de datos su imagen, ahora sí vemos que es diferente de 0 preguntamos que si el número de turno que tiene la petición n es menor que el de la petición que esté preguntando a las demás peticiones si si estaremos en una espera ocupada y si la petición que está preguntando es menor que la entonces seguira preguntado a la siguiente petición, pero si el de turno no era ni mayor ni menor entonces la petición n y la petición con la que se está comparando el turno entonces tienen el mismo número de turno por lo que se le aplicará un criterio de desempate donde se dejará avanzar al que tenga un hora menor en el momento en que se hizo la petición donde la petición que salga de la espera ocupada se irá comparando con las demás peticiones hasta que esta petición sea la que tiene el turno con el menor número que las otras peticiones para después pueda hacer el guardado de la imagen en la base de datos mediante el método que se haya definido en la parte del repository ya que esta parte es la encargada de toda la parte la base de datos para consultar, eliminar o actualizarla y cuando acabe de guardar la imagen se debe poner el número de turno de la petición que logró guardar la imagen en 0.

2. Realizar un dibujo para enviársela como propuesta a Zeratulun.

Solución: Zeratulun enojado por que contrato a programadores que si les paga.

