



UNIVERSIDAD DE TALCA
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN (DCC)

Informe Técnico

Proyecto N°2

Integrantes: Juan Cordero
Álvaro Elgueda
Nicolás Pradenas
Profesor: Rodrigo Pavez
Fecha: 11-03-2020

Solución General

Para la solución del problema, se dividió el problema en tres componentes principales: El Servidor principal de la casa matriz, un Distribuidor y un Surtidor.

En el caso del Servidor principal tiene dos hilos, uno encargado de estar escuchando y cargando los datos que vienen desde los distribuidores, mientras que el otro se encuentra mostrando los informes y realizando actualizaciones de precios en todos los distribuidores. El motivo de implementar hilos paralelos es para poder escuchar permanentemente al distribuidor, de esta manera realizar actualizaciones en tiempo real de las ventas en el sistema principal, por otro lado se debe permitir generar reportes de las ventas y actualizar los precios en los distintos distribuidores y cada surtidor respectivamente en el sistema.

En el caso del Distribuidor también posee dos hilos, uno que se preocupa de escuchar las actualizaciones de precios desde el servidor hacia todos los surtidores existentes, mientras que el segundo recibe los datos de las transacciones de los surtidores hacia el respectivo distribuidor y consecuentemente el servidor central. Ambos hilos generan una conexión con su respectiva base de datos. El principal problema que fue abordado, es la implementación de los distribuidores para que cada uno se conecte con el servidor y con sus respectivos surtidores de forma simultánea, por lo tanto, la necesidad de tener procesos que están dedicados era imprescindible para la solución planteada.

Por último para el caso de los Surtidores, posee un hilo que se encarga de realizar las ventas de los distintos tipos de combustibles (incluyendo una interfaz por consola), además de tener un socket que se encarga de recibir los datos desde el distribuidor para poder realizar el cambio de precios. Esto se justifica porque el surtidor siempre tiene que estar disponible para realizar cargas de combustible, pero también existe la necesidad de escuchar al distribuidor con el que está conectado para poder realizar los cambios en los precios.

Otra parte de la solución es el modelo de conexión a bases de datos, tanto el servidor como cada distribuidor cuentan con una propia. Las clases que se ven involucradas son `ConexionBDServidor` y `ConexionBDDistribuidor`, estas incluyen métodos para insertar registros de ventas, actualizar los precios de los distintos combustibles y la obtención de reportes.

Para poder ejecutar las distintas partes del proyecto en redes distintas, se utiliza Hamachi para crear una red LAN virtual, de esta forma se conecta el Servidor, Distribuidor y Surtidor cada uno con una ip independiente. Cada instancia se ejecuta

en un pc independiente los cuales se comunican por las direcciones ip generadas por Hamachi. A modo de facilitar su ejecución se agregan las tres ip de los integrantes en cada clase para su respectivo uso.

Protocolo de Transporte

En el caso del protocolo utilizado para la resolución del problema, se decidió optar por TCP, esto debido a que la comunicación debe ser orientada a la conexión, para asegurar que los mensajes lleguen a los destinatarios correspondientes. Además, la información que se envía hacia los distintos host no se debe perder, porque incluye información de actualización de precios y los registros de las ventas.

Motor de Bases de Datos

Para el motor de base de datos se optó por MySQL, esto debido a que incluirla al entorno de desarrollo es relativamente fácil, además permite realizar las operaciones de inserción, actualización y selección se hacen en las clases que se encargan de la conexión con la base de datos

Anexo (forma de ejecutar el programa)

- Antes de ejecutar el programa, hay que cambiar el String ip que se encuentran en las clases Servidor, Surtidor y Distribuidor.
 - La dirección IP en el Servidor debe ser la misma que en el TunelDistribuidor para establecer la conexión.
 - La dirección IP del Distribuidor debe ser la misma que en el Surtidor para establecer la conexión.
- Se debe crear una base de datos con los nombres que vienen en las clases ConexionBDServidor y ConexionBDDistribuidor.
 - Los scripts para las crear las tablas de cada base de datos están en la carpeta BDScript.
- El orden para ejecutar, primero se tiene que ejecutar el servidor, luego el distribuidor y los surtidores.
- Una vez que los programas se ejecutan, se puede probar las funcionalidades incluidas en los distintos programas.
 - En caso de querer agregar más distribuidores y surtidores, se tiene que crear un distribuidor con una nueva IP y a los surtidores se les debe asignar la dirección del distribuidor nuevo.