Instituto Tecnológico de Costa Rica, Escuela de Computación, Bases de Datos II
Prof. Rodrigo Núñez, Proyecto I – 25%

Green Carrot

- La empresa ***Green Carrot*** es un startup que va a tener un lanzamiento paulatino en diversos países de centro y sur américa.
- **Green Carrot** se especializa en la fabricación de vehículos tienda acogedores, con atmósfera de tienda contemporánea. Dichos vehículos tienen una ruta predefinida en las ciudades, en dicha ruta se visitan diversos proveedores de ropa, artículos de belleza, zapatos, entre otros; es decir tiendas que se encuentran en dicha ruta, y cargan el carrot camión con mercadería de cada tienda que ha sido previamente solicitada por algún cliente.
- Cada camión dado la ruta, sabe cuales son los vecindarios o zonas que puede visitar sin afectar los tiempos de entrega, de tal forma que el camión realiza un anillo de recorrido cada cierto tiempo.
- Los clientes ingresan a un website, los cuales pueden buscar artículos de las tiendas a su disposición para comprar, tomando en cuenta solo aquellos artículos que están en una ruta dentro de la ubicación o lugar de entrega del cliente.
- Sabiendo el lugar de entrega del cliente, es posible saber si un camión pasa por es zona, y de esa forma saber las tiendas que ese camión o camiones visitan, por ende saber que artículos están disponibles para esa persona para comprar.
- Los proveedores por su lado entran a un módulo donde pueden subir artículos que ellos venden al sistema de Green Parrot, detallando nombre, descripción, tamaños, cantidad y fotografías de los productos. Pueden dar de baja productos insertados previamente. Ya el proveedor por su ubicación se conoce las rutas en las cuales hay carrot camiones.
- El consumidor decide buscar artículos sin importar la tienda, los agrega al carrito de compras y decide comprar, para lo cual paga un flete según el camión que puede traer los artículos.
- Los pedidos son recibidos por los camiones, los cuales montan dicha mercadería y otras mercaderías similares en el camión tienda, indicando en el sistema que el pedido ya está en el camión.
- Seguidamente el chofer del camión en su ruta, puede visualizar que clientes tiene cercanos que ya tenga sus pedidos cargados al camión. Alertando así al consumidor la ubicación del camión tienda cuando éste llega a un punto, siempre y cuando sus artículos ya estén ahí.
- El consumidor puede dirigirse caminando al camión tienda, y ver la mercadería que pidió, y otra mercadería adicional que encuentre en la tienda. Finalmente hace checkout de los artículos que desea dejarse, incluso de aquellos que agregó nuevos; se hace el cargo inmediatamente a la tarjeta que se uso para el pago del flete.
- En caso de que no se pueda hacer el cargo, el cliente debe pagar con tarjeta en el camión o bien cambiar la información de la tarjeta para realizar el pago en el webapp.
- La mercadería que no se compra, rota cierta cantidad de veces por la ruta, pero eventualmente es descargada en la tienda origen, generando una devolución.
- La intención de Green Carrot es poner las tiendas y proveedores a una distancia corta de los compradores, realizando dicha venta y compra por demanda según necesidades reales de los mismos.

Actividades a realizar

- Diseño e implementación de la base de datos usando SQL Server Azure. Diseñe la base de datos que permita manejar todas las transacciones de pedido y compra, además de los pagos, tarjetas y todo aquello de carácter transaccional y económico.
- Replique la base de datos en Azure en otro nodo cercano geográficamente separados. Puede ser en modo cluster o load balance.
- Diseñe una base de datos en MongoDB, la cual mantenga la información relacionada a la configuración de rutas, las paradas que debe hacer el camión tanto para visitar un proveedor como para parquear la tienda para recibir consumidores. Es decir, toda la planificación del recorrido. Dicha base de datos se apoya mucho en consultas y tipos de datos de geolocalización.

- Diseñe una base de datos en Cassandra, la cual mantenga la información relacionada a los inventarios en ruta, es decir, que mercadería está o no en el camión, en resumen, lleva el inventario interno de cada camión. Dejando claras las entradas, salidas y devoluciones de inventario.
- Haga un script de llenado para la base de datos en SQL server en T-SQL para que la base de datos sea funcional.
- Implementación de los stored procedures transaccionales:
    - Agregar-Editar un nuevo cliente
    - Agregar-Editar información de payment y merchant
    - Registrar un pago
    - Registrar una transacción
- Haga deploy de la base de datos en mongo configurada en un cluster de MongoDB que posea: 2 config servers, 2 routers, shard servers por ciudad, cada shard debidamente replicado, usando replicación por arbitro y un servidor secundario. Inicie a nivel de datos con 3 ciudades.
- Haga deploy de la base de datos en Cassandra sobre un cluster activo de 3 nodos.
- Implemente un programa en nodejs que realice las siguientes tareas:
    - Una sola vez haga el llenado inicial de la base de datos de rutas y la base de datos de productos
    - Cuando un cliente haga un pedido en Cassandra se agregue la planificación de rutas en MongoDB, teniendo en cuenta la recolección del proveedor y la parada para entregarle al cliente
    - Cuando un cliente haga una checkout de una compra final en Cassandra, se actualice el inventario y se realicen los cargos, pagos y transacciones en SQL Server
    - El programa debe realizar esas acciones en forma automática sin intervención de una persona, básicamente escuchando los registros que ingresan a la base de datos y realizando las acciones necesarias
- Debe tener un script, json, o instrucciones a mano para poder ingresar un pedido, recibir pedido, checkout del pedido, revisar transacciones, es decir, lo que sea necesario para probar un ciclo completo de pedido, recorrido, compra y pago, dado pues que no va existir UI.
- Una vez verificado la funcionalidad completa del sistema se procederá a probar:
    - Es posible apagar un SQL Server, tener un auto recovery y automáticamente que el sistema siga funcionando
    - Es posible apagar un servidor primario de un shard dentro de una replica y poder seguir viendo los datos de esa ciudad
    - Es posible apagar hasta 2 nodos de Cassandra y todo sigue funcionando correctamente

- El proyecto puede hacerse en parejas o individual
- Cualquier sospecha de copia anulará el trabajo
- SQL Server se debe montar en Azure, Mongo y Cassandra cluster sobre docker
- Habrá una revisión de diseños de bases de datos general que debe entregarse el 27 de Agosto
- Una revisión del diagrama general de arquitectura y flujo de los sistemas de base de datos para el viernes 31 de Agosto
- Los anteriores al correo vsurak@gmail.com
- Fecha de revisión del proyecto contra cita jueves 20 de setiembre