

Plan de Simulación

Sistemas de Bases de Datos II

Elaborado por:
Tomás Guzmán, CI 21.615.008
César Guzmán, CI 28.387.170
Sebastian González, CI 27.234.142

Contenidos

Desarrollo del plan de simulación	2
Módulo I: Compras y Despachos	2
Módulo II: Unidades	3
Módulo III: Registro de Proveedores	3
Módulo IV: Colapso por concurrencia	4
Requerimientos técnicos	5
Manejador de base de datos	5
Herramienta de reportes	5
Referencias	6

Desarrollo del plan de simulación

Los datos que serán constantes en la Base de Datos serán los usuarios, las aplicaciones de Delivery y sus oficinas y las empresas aliadas (su sector) y sus sucursales.

Para comprender de mejor manera como funciona el día a día de las empresas de Delivery, tomamos como referencia algunos de los datos de una de las aplicaciones de delivery más reconocidas en nuestro país. Para Octubre del 2020 se tiene que, para la aplicación con más descargas en Venezuela (Yummy), los datos son los siguientes:

- Descargas > 50.000
- Usuarios registrados \approx 10.000
- Usuarios que hacen pedidos al día = 200 a 350
- Promedio de pedidos al mes por usuario = 0 a 3
- Pedidos realizados = 12.000 mensuales (más de 350 diarios).
- Cantidad de unidades de delivery = 200 (en total contando todas las zonas en las que opera).
- Tiempo de entrega promedio < 45 min
- Locales registrados en la aplicación \geq 125

Con base a estos datos se implementará en PL/SQL las ocurrencias de las situaciones por medio de proporciones (uso de porcentajes) de los datos suministrados en un principio, de manera que la simulación sea más realista.

A continuación se definen los módulos a implementar para la simulación en este proyecto:

Módulo I: Compras y Despachos

En este módulo se simula todo el proceso de realización de pedidos por parte de los usuarios. De manera aleatoria algún usuario hace un pedido a alguna empresa aliada por medio de alguna de las aplicaciones en la que está registrado. Se sigue la siguiente lógica:

- A diario, el porcentaje de usuarios que realizan pedidos es de un 3%, de entre todos los registrados en la aplicación
- Se selecciona aleatoriamente el/los productos de una empresa aliada (que trabaje con la aplicación del usuario) para ser solicitados por el usuario (se verifica que el producto solicitado

pertenezca al mismo sector de la empresa aliada) y se selecciona aleatoriamente de entre las direcciones del usuario cual es el destino del pedido. Se procede a buscar la sucursal que esté más próxima: en misma zona, municipio o ciudad (en caso de no existir no se puede efectuar al pedido).

- Una vez hecho el pedido, se asigna a la unidad disponible más cercana el envío del paquete. Se calcula el tiempo desde la ubicación de la unidad hasta la sucursal para recoger el paquete.
- Una vez recogido el paquete (haya transcurrido el tiempo de llegada a la sucursal), se guarda la fecha de inicio del envío y se calcula el tiempo estimado de llegada a la dirección del cliente, se guarda dicho tiempo en la fecha final. En caso de que la fecha final sea después de la fecha de ejecución (el envío aún no haya llegado) entonces se tiene que la unidad aún está llevando el producto.
- Cuando llegue el envío a la dirección del usuario, este procede a calificar la entrega de 0 a 5 estrellas (se guarda una valoración aleatoria donde hay una probabilidad mayor de que sea una nota aprobatoria) y se registra el momento en el que lo recibió.

Módulo II: Unidades

En este módulo se simula lo referente al estado de las unidades de Delivery:

- Por semana, la cantidad de unidades que se estropean oscilan entre un 2-10% del total de una aplicación.
- Mientras que de un 10 a un 20% del total de unidades fuera de servicio son reparadas a la semana.
- Adicionalmente, si una aplicación alcanza una cifra del 30% de unidades en reparación, entonces automáticamente deberá comprar más unidades para reponer el inventario, es decir, se agregan tantas unidades como las que hayan dañadas.

Módulo III: Registro de Proveedores

En este módulo se simula todo lo referente al registro de los proveedores de servicios (aplicaciones de delivery). La empresa proveedora se encargará de ofrecer un servicio a los distintos comercios, tratando a éstos como sus aliados. A continuación se muestra la lógica del módulo:

-
- Una aplicación de delivery define sus acuerdos de servicios, los cuales están delimitados por la cantidad de pedidos fijados en un rango de tiempo establecido y tienen un precio específico. Una aplicación puede tener uno o varios acuerdos, se generan de manera aleatoria.
 - Un comercio (aliada) se encargará de buscar un proveedor de servicio que se adapte a sus necesidades y que opere en la misma zona (tenga oficinas en dicha zona, municipio o ciudad). Seleccionando uno de los acuerdos de servicio de alguna empresa de Delivery.
 - El acuerdo de servicio queda enmarcado bajo un contrato el cual posee una fecha de vencimiento, y una vez cumplida esa fecha el cliente puede decidir crear otro contrato o simplemente prescindir del servicio.

Módulo IV: Colapso por concurrencia

En este módulo se simulan los colapsos de servicios de Delivery de una aplicación para una zona en la que ya no hay unidades disponibles, es decir, todas las unidades están ocupadas en el transporte de otro envío. En este caso se van a inhabilitar los pedidos de los usuarios a las empresas de Delivery que no tengan unidades disponibles al momento.

La simulación hará lo siguiente:

- Se realizan una cierta cantidad de pedidos a través de una misma aplicación y sus distintas aliadas, de manera que la fecha y hora de todos esos pedidos solapen, es decir, todas las unidades de una aplicación en una ubicación estén realizando un envío en ese momento.
- Cuando otro usuario quiera realizar un pedido en ese momento, no se le permitirá ya que todas las unidades están ocupadas.
- Se mostrará un mensaje donde se indica al usuario: "Inténtelo más tarde".

Requerimientos técnicos

Manejador de base de datos

En este proyecto se utilizará el manejador de base de datos de Oracle. Esta compañía ofrece su extensión PL/SQL para su manejador de bases de datos, lo cual ofrece funcionalidad extra como stored procedures, tipos de datos abstractos, triggers y funciones. Los mismos permiten la automatización y reutilización de código para lograr distintos objetivos a partir de la base de datos. A su vez, posee una amplia documentación y un servicio en vivo en el cual cualquier usuario registrado podrá ejecutar código PL/SQL.

Para la mayor parte del manejo en sí de Oracle se utiliza SQLDeveloper, de la misma compañía. Esta herramienta da funciones visuales al programador sobre la base de datos, simplificando el trabajo.

Además de todos los beneficios que nos ofrece Oracle, es utilizado en este proyecto ya que el profesor lo solicita como requisito obligatorio para la implementación del mismo.

Herramienta de reportes

El equipo decidió trabajar con el software TIBCO Jaspersoft Studio de JasperReports ya que uno de los integrantes posee experiencia con esta herramienta y el resto del grupo aceptó la idea porque conocen de la versatilidad y facilidad que se presenta a la hora de trabajar con el mismo.

Jaspersoft presenta una interfaz muy fácil de entender, la parametrización se puede realizar de manera sencilla, soporta datos de tipo BLOB, nos ofrece herramientas para dibujar mapas, crear buenos diseños y tablas; herramientas que pueden facilitar el cumplimiento de los requerimientos de los reportes del proyecto. Además realiza fácilmente la conexión con Oracle, el manejador de bases de datos a utilizar.

En términos generales, según experiencias pasadas creemos que es la herramienta correcta para realizar todos los reportes solicitados de manera efectiva y práctica, es por ello que será nuestra elección.

Referencias

- Vicente Zavarce, el visionario detrás de Yummy Venezuela.
<https://eldiario.com/2020/10/22/visionario-yummy-venezuela/>
- App de delivery Yummy amplía su cobertura a más ciudades del país.
<https://www.bancaynegocios.com/app-de-delivery-yummy-amplia-su-cobertura-a-mas-ciudades-d-el-pais/#:~:text=Es%20por%20ello%20que%20Yummy,que%20cuentan%20hoy%20en%20d%C3%Aa>