

Diseño de una Base de Datos basada en Coca-Cola

Amador Bassaure Elizabeth, González Cárdenas Verónica, Camacho Peralta Karla Jimena, Macedo Franco Emmanuel

Resumen— este proyecto se basa en el diseño y desarrollo de una base de datos que tomo a la empresa Coca-Cola como modelo de negocios, dicha base de datos se enfoca en resolver problemas tales como la inserción, actualización y borrado dentro de las tablas de la base de datos, así como su administración adecuada, implementando la generación de una ruta efectiva con el uso de Google Maps y una interfaz para el fácil manejo del mismo.

I. INTRODUCCIÓN

Una interfaz gráfica de un manejador de Base de Datos es una herramienta de uso sencillo y eficaz con un diseño interactivo y fácil de manejar con distintos tipos de usuarios que se enlistarán a continuación:

1. Administrador. - Puede hacer uso de todas las funcionalidades de la aplicación, mismo que se explicará posteriormente.

2. Usuario. – Es la persona que usando una ventana gráfica en el sistema operativo, Windows 7, recupera datos de la base, para seleccionar una ruta de entregas de pedidos de alguna empresa o fábrica, en este caso se tomó el ejemplo de Coca-Cola por practicidad, el programa mostrará con la ayuda de Google Maps, un mapa en pantalla donde se presenta el mejor camino a seguir para completar la ruta de entrega a diferentes tiendas de dicha ruta.

Se decidió hacer este tipo de interfaz ya que con ello se podía crear una experiencia más personal con los usuarios, creando una aplicación que facilite la planeación y manejo de una empresa ya sea pequeña o grande, ayudando así a tener una mejor calidad de servicio.

II. MARCO TEÓRICO

Se detallarán las partes por las que se compone una base de datos, así como las herramientas que se utilizaron para llevar a cabo este proyecto:

Una base de datos se compone por 4 partes:

- **Utilerías.** - Acceso a la base de datos visualizar gráficamente.
- **Usuarios.** – Existen 4 tipos de usuarios: los que acceden a la información contenida en la base de

datos, Programadores de APP, Usuarios finales y DBA

- **DBMS.** - Manejador de la base de datos
- **Datos.** - Consultar, recopilar datos organizados protegidos por el manejador deben de ser integrados y compartidos

Los objetivos de una base de datos son evitar la duplicación de datos, y debe cumplir estándares de como nombrar usuarios, campos y tablas de datos, todo esto por medio del lenguaje SQL (Structured Query Language) el cual es un lenguaje de programación estándar e interactivo para la obtención y actualización de información desde una base de datos. El diseño de la misma es importante ya que cada tabla corresponde a una sola entidad o relación, las tuplas de la tabla corresponden a una única ocurrencia de la entidad. Se tiene una llave primaria es una identificación individual única de las ocurrencias de una entidad para evitar que se encime la información y se pierdan datos importantes.

La siguiente parte importante del proceso al crear una base de datos es la normalización que es el proceso de aplicar principios de diseño a las estructuras de datos para que ajusten las expectativas y con esto prevenir que los datos se repitan, problemas de actualización y las dependencias de datos sean lógicas.

Una vez que se tiene la idea se plasma en el diagrama llamado Entidad-Relación, en él se tienen tablas de la base de datos y sus relaciones.

Para llenar las tablas se pretende activar un desencadenador para el momento en el que se realizan cambios en una base de datos, mientras que una aplicación en línea que utiliza la base de datos se está aplicando o actualizando con una redefinición basada en la edición. El cuerpo de un trigger o disparador está diseñado para manejar estos cambios, para que se puedan aplicar de forma adecuada después de que se completen los cambios en el código de la aplicación.

Se utilizó la herramienta Java que es la base para prácticamente todo tipo de aplicaciones en red y es el estándar mundial para desarrollar y entregar software empresarial, contenido web, juegos y aplicaciones móviles. Java disfruta de un ecosistema grande y maduro con un fuerte soporte de herramientas. Java ofrece portabilidad de aplicaciones y un rendimiento robusto en muchos entornos informáticos.

Los componentes de Java SE son Java Development Kit (JDK), Java Runtime Environment (JRE) y Java SE interfaz de programación de aplicaciones (API). JDK contiene el JRE y los compiladores y depuradores necesarios para el desarrollo de applets y aplicaciones. JRE proporciona las bibliotecas, la Máquina Virtual Java (JVM) y los componentes necesarios para ejecutar aplicaciones y applets Java (Figura 1).

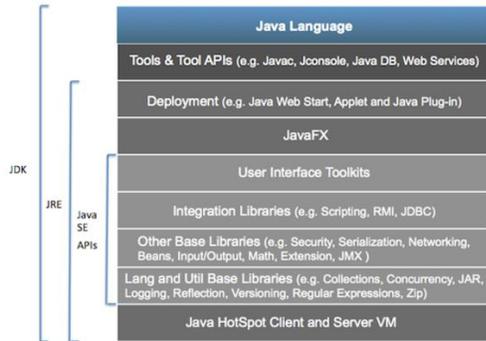


Figura 1. Componentes de Java.

Para la parte gráfica la “GUI” es un tipo de interfaz de usuario que permite a los usuarios interactuar con dispositivos electrónicos a través de iconos gráficos e indicadores visuales tales como notación secundaria, en lugar de interfaces de usuario basadas en texto. Las acciones en una interfaz gráfica de usuario generalmente se realizan mediante la manipulación directa de los elementos gráficos. Esto sirve para que el usuario pueda interactuar con la información que se requiere.

Se utilizó Google Maps que es un servicio de mapas web desarrollado por Google. Ofrece imágenes de satélite, mapas de calles, vistas panorámicas de calles de 360 ° (Street View), condiciones de tráfico en tiempo real (Google Traffic) y planificación de rutas para viajar a pie, en automóvil, en bicicleta (en beta) o en transporte público.

Google Maps API es una interfaz de programación de google que permite incorporar los mapas de este desarrollador a una aplicación ya sea WEB, IOS, Android. La utilizada en este proyecto fue Matrix API que computa la distancia y la duración de viajes entre varios orígenes y destinos según determinados modos de viaje.

Este servicio no devuelve información detallada sobre rutas. La información sobre rutas, incluidas las polilíneas y las indicaciones textuales, puede obtenerse pasando el origen y el destino deseados al servicio de indicaciones.

El acceso al servicio de matriz de distancia es asincrónico, ya que la Google Maps API debe realizar una llamada a un servidor externo. Por esta razón, a fin de procesar los resultados, se utiliza un método *callback* para la ejecución al completarse la solicitud.

En la respuesta de matriz de distancia se incluyen un código de estado para la respuesta en conjunto y un estado para cada elemento.

Para optimizar la distancia se recurrió al algoritmo “Waypoints optimize” que permite optimizar la distancia recorrida en un viaje y obtener el tiempo en el que es recorrido. Debe incluir el modo de la ruta de viaje y dos waypoints consecutivos coordinados desde la respuesta hasta los Waypoints Sequence Extensión API.

La ventana de tiempo constraint define el tiempo en que abre o tiempo límite para recibir el producto en los puntos o citas específicas en cada waypoint.

El servicio de tiempo indica la longitud de tiempo entre el waypoint de llegada y salida.

Los tiempos de servicio son tomados en cuenta para tomar la ruta más óptima en la secuencia de waypoints.

Para facilitar la llegada a los clientes se agregó la herramienta “Google street view” que es una presentación de Google Maps y de Google Earth que proporciona panorámicas a nivel de calle (360 grados de movimiento horizontal y 290 grados de movimiento vertical), permitiendo a los usuarios ver partes de las ciudades seleccionadas y sus áreas metropolitanas circundantes.

III. DESARROLLO

En esta sección se detalla la realización del programa para una empresa como Coca-Cola, creando tablas que respeten las formas normales, por medio de tablas que cumplen con los estándares de normalización con sus respectivas tablas históricas y llaves primarias.

Utilizando la plataforma JAVA con el IDE Netbeans para la interfaz gráfica, Matrix API de Google para la implementación del mapa y SQL Developer para el Script.

A continuación, se enlistan los datos que contendrá la base de datos (Figuras 2,3 y 3.1):

- 1.- Direcciones de clientes.
- 2.- Plantas, sucursales y sucursales propias.
- 3.-El nombre y teléfono de contacto de cada cliente.
- 4.-La nómina, pagos, quincenas y seguros de los empleados.
- 5.-Los productos creados, llevados y asegurados por la propia empresa.
- 6.-Los camiones surtidores con sus conductores, acompañantes y destinos.
- 7.-Los convenios con las empresas clientes al igual que con las aseguradoras para los trámites de los empleados.

Resaltando para cada uno de los datos anteriores lo siguiente: **Coca-Cola.** - tabla principal que contiene los datos de razón social, persona de contacto, teléfono y dirección

Razón Social. - descripción de cada razón social

Planta. -contiene número de planta, clave de dirección, clave de nombre de contacto, y clave de Coca-Cola. Para poder controlar el número de plantas que tiene cada empresa Coca-Cola.

Corporación. - descripción de la corporación y clave de la razón social.
 Para poder saber si las tiendas a surtir son de la misma corporación. (Ejemplo: Sanborns, Grupo Inbursa)

Empresa. - clave de nombre de contacto, clave de dirección, clave de teléfono, clave del corporativo, clave de la razón social.
 Para poder registrar el nombre de las empresas cliente o con las que se tiene algún convenio.

Empleado. - puesto, clave persona, clave planta, clave nómina, clave coca.
 Como empresa se debe tener un registro de los empleados.

Conductor. - clave conductor y clave de empleado.
 Saber que empleados son conductores de qué camión.

Acompañante. - clave de acompañante y clave de empleado.
 Si el conductor del camión tiene acompañantes o entrega solo.

Camión. - número de camión, placa, litros de gas, clave de conductor, clave de acompañante, numero de diablitos, clave planta.
 Saber que camión se mueve con qué conductor, al igual saber de qué planta salió.

Origen Destino. - distancia, clave camión, duración.
 Tabla que ayudará a guardar los datos obtenidos de Google Maps para dar una ruta eficiente.

Negocio. -nombre de negocio, clave de nombre de contacto, clave de teléfono, clave de dirección.
 Si el cliente no tiene razón social o no es una empresa como tal, se registrará como un negocio.

Sucursal. -número de sucursal, nombre de la sucursal, clave empresa, clave dirección, clave nombre de contacto.
 Si una empresa tiene más de 1 sucursal, saber a dónde se enviarán los productos.

Producto. - nombre producto, precio fabrica, precio comercial, contenido en ml, fecha elaboración y fecha caducidad.
 Tener un control de los productos que se elaboran en la fecha y lugar.

Cliente. - tipo de cliente, clave persona, clave dirección, clave teléfono, clave máquina, clave sucursal, clave negocio.
 Poder saber qué tipo de cliente es (Formal o Informal), saber si se le dio máquina expendedora.

Máquina expendedora. - dinero ingresado, dinero recibido, dinero final, folio.
 Registrar el dinero ingresado y el folio que se le asignó.

Convenio. - fecha inicio, fecha fin, descripción del convenio, monto y clave del producto perteneciente al convenio.

Saber si una empresa tiene un convenio, saber cuándo inició y de que trata el convenio.

Como complemento a estas tablas y para generar una base más dinámica y completa se decidió incluir:

- Triggers de autoincremento de llave primaria, y colocación de datos en históricos al insertar, actualizar y borrar registros de las tablas.
- Validación en campos de llaves primarias y foráneas. El campo fecha como llave alterna.
- Cifrado utilizando el método Hashing en el campo contraseña para garantizar la seguridad del usuario. Si por algún motivo los datos de la base son robados, los datos del usuario están protegidos alno poder ser leídos.
- Auditoria a las tablas que son puntos clave para las llamadas de pedido, ruta, cliente y planta, las cuales son consideradas de mayor importancia por ser las que se ocupan para la aplicación gráfica.
- Transacciones para garantizar la consistencia de la base de datos, la ruta calculada por google Maps (Figura 8) que no se guardará hasta que se confirme.
- Interfaz gráfica creada en un entorno de programación ya conocido por los alumnos para manejar dinámica y eficazmente los datos con el lenguaje de java

Utilizar los datos de la tabla origen, destino, para poder crear una ruta óptima con la inclusión de Google Maps para que la aplicación muestre el mapa en tiempo real y el usuario pueda seleccionar las tiendas a las cuales se debe surtir, arrojando los datos de distancia y duración del trayecto (Figura 4); con estos datos se podrá crear la mejor ruta.

IV. RESULTADOS

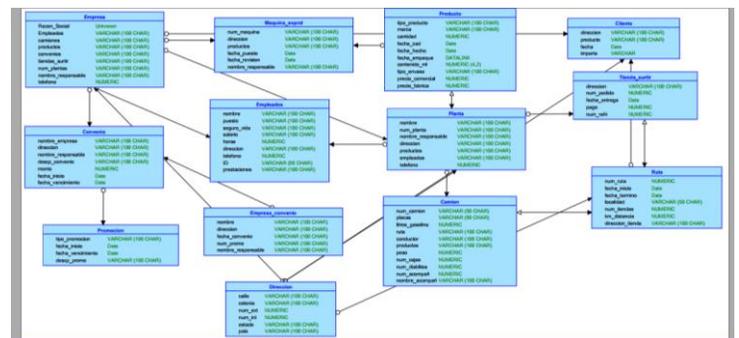


Figura 2. Modelo Lógico.

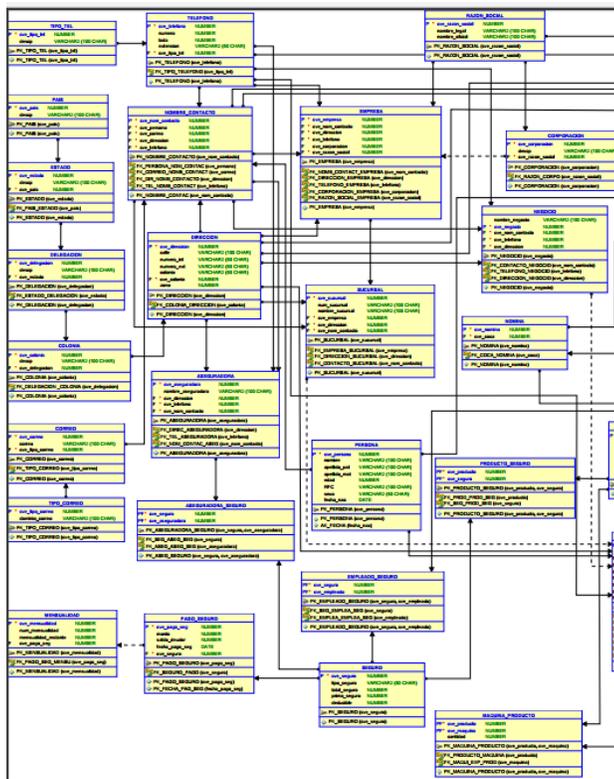


Figura 3. Modelo Físico.

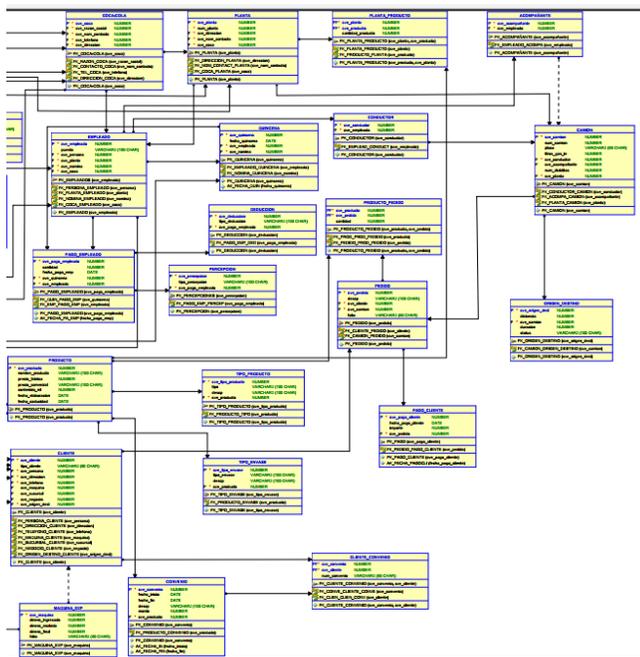


Figura 3.1 Modelo Físico.

Se muestra la pantalla principal (Figura 4) de la aplicación de “Rutas”, esta aplicación fue diseñada para los choferes de los camiones. Las rutas son elegidas al azar, sin darle oportunidad al chofer de elegir las direcciones de su preferencia.

Una vez iniciada la aplicación coloca las direcciones origen, destino y la de los clientes (Figura 5). Al oprimir el botón de “Calcular Ruta”, nos muestra el tiempo y distancia total de la ruta(Figura 4), si la ruta no es del agrado del usuario puede oprimir el botón de Crear nueva ruta, generando direcciones nuevas al azar.

A continuación, se muestran las indicaciones que el chofer debe seguir (Figura 6) para llegar a cada uno de los destinos (clientes).

La aplicación ofrece un servicio de Street View (Figura 7) para facilitar las entregas y evitar que el chofer se pierda o llegue a una dirección equivocada (Figuras 7 y 8).

Para el cálculo de la ruta, se usó un algoritmo de optimización, eligiendo la ruta más corta y rápida.

Este algoritmo está basado en el problema del viajero comerciante, el cual menciona que de una cantidad N de ciudades, el viajero debe pasar por cada una de ellas, sin repetir, en el menor tiempo posible.

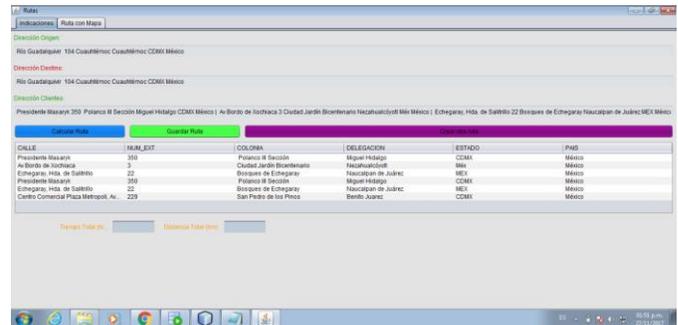


Figura 4. Tiempo y distancia

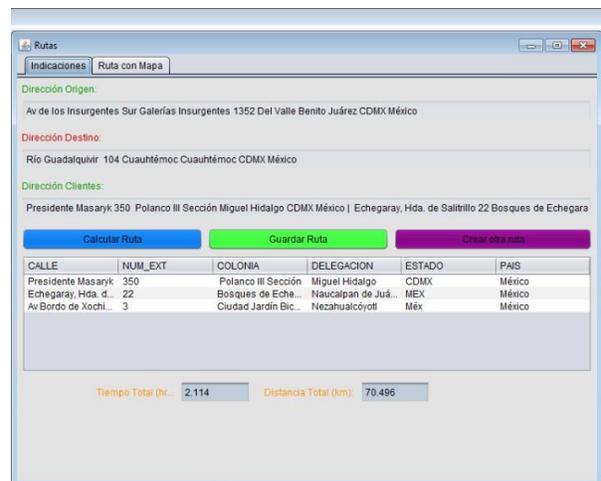


Figura 5. Nueva ruta

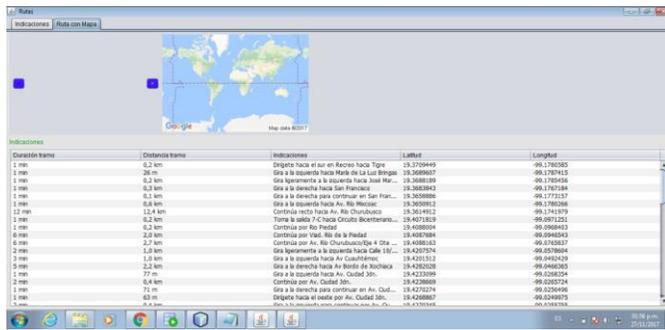


Figura 6. Mapa

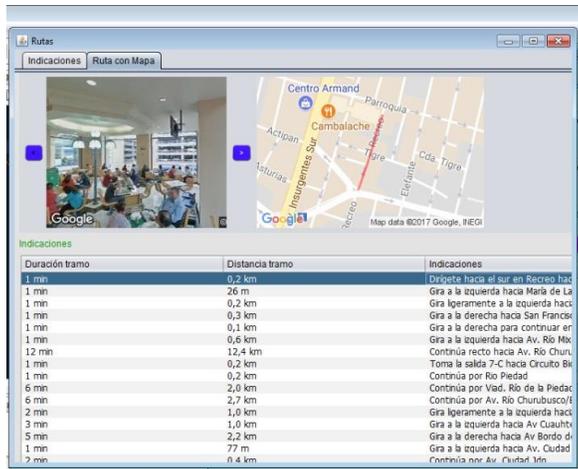


Figura 7. Street View

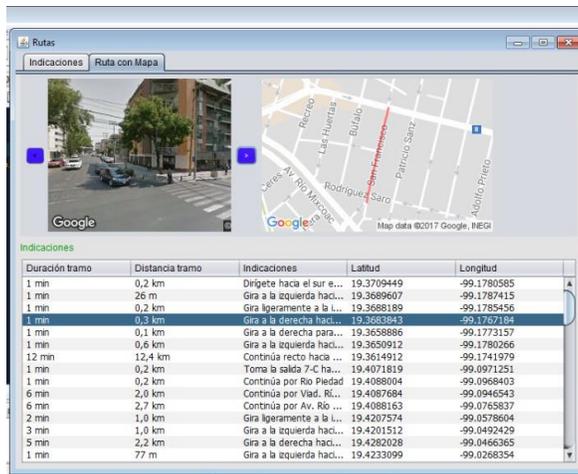


Figura 8. Direcciones

V. CONCLUSIONES

Se logró realizar la conexión para poder manejar la base de datos por medio de la interfaz gráfica al igual que la matriz de Google Maps. El objetivo que no se logró fue el hacer una sola interfaz que tuviera la opción de la creación de rutas y el dar de alta los productos ya que hasta el momento sólo funcionan por separado. Se encontraron muchos desafíos al realizar este proyecto ya que no se contaba con el conocimiento de implementar Google Maps en una interfaz creada, al igual que conectar una base de datos de Oracle con

una aplicación y lograr que los 3 puntos funcionaran armónicamente.

Se ha entendido que llevar el control de una empresa de producción-entrega es complejo ya que hay muchos rubros que cumplir y tener bien controlado ya que en cualquier momento se puede salir de control y perderse mucho dinero, clientes y prestigio.

Este proyecto puede servir para grandes o medianas empresas que quieran llevar un control de sus productos, empleados y rutas que se tenga para la entrega de la mercancía. Este sistema es menos propenso a fallas ya que en tiempo real se modifican los productos que se tiene.

Se quiere en un futuro tener solo una interfaz que realice ambos roles mencionados e implementar el uso de una clave y contraseña al ingresar a la aplicación para un mejor control y seguridad.

REFERENCIAS

- [1] Oracle. (2008, 2017). SQL Developer Data Modeler User's Guide. Noviembre 11, 2017, de Oracle Sitio web: <https://docs.oracle.com/database/sql-developer-17.3/DMDUG/toc.htm>
- [2] Oracle. (2008,2017). SQL Developer User's Guide. Noviembre 11 de 2017, de Oracle Sitio web: <https://docs.oracle.com/database/sql-developer-17.3/RPTUG/toc.htm>
- [3] Oracle. (2008,2017). SQL Developer Command-Line Quick Reference. Noviembre 11, 20117, de Oracle Sitio web: <https://docs.oracle.com/database/sql-developer-17.3/SQCQR/toc.htm#SQCQR-GUID-6736ADCE-ABD9-49B8-91E5-7973221DC434>.
- [4] https://docs.oracle.com/cd/B28359_01/server.111/b28337/tdpsg_auditing.htm#TDPSG50051
- [5] http://csrg.inf.utfsm.cl/~jfuentes/_build/html/lectures/week6/lecture24.html
- [6] http://docs.oracle.com/cd/B19306_01/server.102/b14200/statements_7004.htm#sthref7885
- [7] F.J.Ceballos Sierra, Java 2: Interfaces Graficas, 2, Alfaomega Grupo Editor
- [8] P.Deitel, H,Deitel, Como Programar en Java, 8, Pearson
- [9] Programación Java, <http://puntoconmoesunlenguaje.blogspot.mx/2012/12/arralist-en-java.html>