



Carrera: Licenciatura en Administración de Tecnología de Información
Curso: TI4601 Bases de datos avanzados
Prof.: Ing. María José Artavia Jiménez, MAE.

Semestre: II-2019
Valor: 15%

PROYECTO PROGRAMADO 3 TECMarket

Bases de datos no SQL de grafos

1. Introducción

La vida ajetreada actual, ha llevado al desarrollo de nuevos servicios y soluciones innovadoras que faciliten tareas cotidianas a las personas. Una de estos nuevos servicios en auge es la entrega a domicilio de cualquier producto de supermercados, para evitarle al cliente gastar tiempo desplazándose de un lugar a otro. Usted ha sido escogido para desarrollar un servicio que permita una entrega rápida a sus clientes con un servicio de calidad.

2. ¿Qué se busca con este proyecto?

- Mejorar las habilidades de resolución de problemas.
- Aumentar el conocimiento del estudiante sobre bases de datos NoSQL de grafos.
- Practicar la experimentación y la resolución de problemas (divide y vencerás).
- Ejercitar la toma de decisiones.
- Fomentar el trabajo **grupal**.
- Fomentar la **investigación** por parte del estudiante.

3. Contexto y las funcionalidades esperadas

TECMarket será un sistema de apoyo a una compañía de retail, la cual cuenta con varios supermercados. Este sistema permitirá a la compañía gestionar la información de todas las sucursales de los supermercados, de los productos disponibles, además permite gestionar la información de los

clientes y los **diferentes servicios solicitados por el cliente**.. Alguna de la información consultada será real, actualizada y deberá obtenerse de los servidores de Google Maps. Para lograr esto, puede utilizar la biblioteca Java Client for Google Maps Services accesible mediante este enlace: <https://github.com/googlemaps/google-maps-services-java>, en caso que utilice Java o seleccionar la Biblioteca cliente correspondiente al lenguaje de programación utilizado.

A continuación, se amplían los requisitos funcionales para el sistema:

3.1 Gestionar sucursal de supermercado

En esta funcionalidad, un empleado de la empresa TECMarket puede registrar la información de diferentes sucursales de supermercados, la información que se debe incluir en el sistema es la siguiente: latitud y longitud o la dirección exacta. Si el usuario ingresa la latitud y longitud, el sistema deberá identificar la dirección exacta del lugar y almacenarla. En caso contrario, si el usuario ingresa la dirección exacta, el sistema deberá identificar y almacenar la información de la latitud y longitud para esa ubicación. El usuario podría ingresar solamente una parte de la dirección, y no ingresar ningún otro dato, en este caso el sistema le mostrará todas las coincidencias. Después de esto, el usuario seleccionará una opción y el sistema le mostrará los datos correspondientes.

A modo de ilustración, véase la siguiente imagen:



Además, se debe registrar el nombre del sitio, una descripción del lugar

También se debe obtener del API la siguiente información: una imagen del sitio, teléfono internacional, el rating, el horario atención al público y el website. Tome en cuenta que, para algunos sitios, puede darse que el API no devuelva todos los datos anteriores. Para esta funcionalidad se buscará información de supermercados reales.

Adicionalmente se debe registrar la información de los productos que se ofrecen en este sitio, se debe indicar un código de producto, nombre, descripción, precio unitario y foto.

El sistema debe permitir que los usuarios (empleados y clientes) consulten todos los sitios registrados en el sistema. Los empleados podrán editar o eliminar lugares previamente creados.

3.2. Registro de clientes

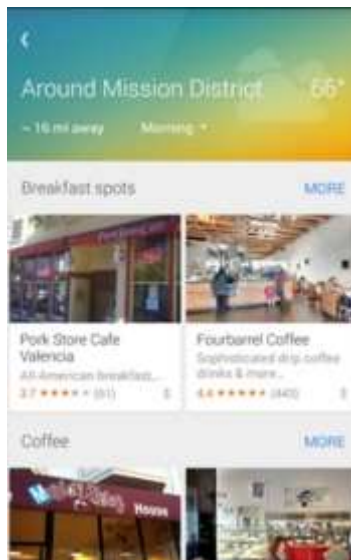
El sistema debe permitir que los clientes se registren la primera vez que ingresan al sistema, para cada cliente se debe registrar la cédula, el nombre, el teléfono, correo electrónico y la fecha de nacimiento, nombre de usuario y clave de acceso.

3.3. Realizar pedido

Los clientes podrán realizar pedidos a través del sitio web. El cliente debe ingresar al sitio con usuario y clave, si es la primera vez debe registrarse antes de hacer el pedido. Para los pedidos se debe registrar el producto solicitado, la cantidad de ese producto, el precio unitario del producto (se puede agregar uno o varios productos en el mismo pedido), fecha y hora del pedido, estado del pedido (Registrado, Asignado, En ruta, Entregado). Además, opcionalmente, el cliente podría indicar alguna necesidad en particular, por ejemplo: si el pedido es un producto frágil puede solicitar un empaque especial para el mismo, entre otros.

El usuario podría consultar en el sistema información sobre sucursales registradas y solicitar información de sitios cercanos a una sucursal (según el tipo restaurantes, cines, doctores, museos y bares). El usuario puede seleccionar para cuál o cuáles de los sitios requiere la información. El usuario deberá ingresar el radio en metros para realizar la búsqueda de los sitios cercanos, asimismo, podría reajustar el radio en cualquier momento para obtener más o menos sitios de interés según desee. Para cada sitio, el sistema debe mostrar: una imagen, el identificador del sitio, el teléfono internacional, el rating, el horario de atención al público y su website.

A modo de ilustración, véase la siguiente imagen:



El usuario podrá seleccionar uno o varios de los sitios cercanos mostrados por el sistema y agregarlos como “posibles lugares donde realizará pedidos”, al agregarlos el sistema los registrará en una lista relacionada con la sucursal original y además se almacenará la distancia entre ambos sitios (esta información se tomará del API de Google Maps).

3.4. Ver historial de pedidos

Un cliente ya registrado tendrá la opción de ver su historial de pedidos, la página debe desplegar: cantidad de paquetes solicitados, fecha de pedido, necesidades registradas y el monto total del pedido.

3.5. Migración de datos

Adicionalmente, para el administrador(a) de la compañía se requiere algunas consultas, sin embargo, la compañía desea hacer estas consultas sobre una base de datos de grafos, por lo cual el programa deberá tener una funcionalidad que permita lo siguiente:

- Tomar toda la información de los supermercados, los pedidos y los clientes y trasladarla a una base de datos de grafos. Sobre esta base de datos se realizarán las consultas.

3.6. Consultas

El sistema debe permitir realizar las siguientes consultas:

- Buscar un cliente en particular y mostrar todo su historial de pedidos.
- Ver todas las sucursales en las que los clientes han hecho pedidos.
- Ver las 5 sucursales para los que se han registrado más pedidos.
- Dado un cliente en particular mostrar todos los demás clientes que hayan realizado al menos un pedido en una sucursal en común con ese cliente.
- Dado un cliente en particular mostrar todos los productos que han comprado otros clientes en los supermercados donde ha comprado ese cliente, para mostrárselos como sugerencias para próximas compras.

3.7. Requisitos no funcionales

A continuación, se detallan algunos requisitos no funcionales:

- El sistema debe utilizar un entorno gráfico web para el desarrollo de la GUI.
- El sistema debe mantener la integridad ante la ocurrencia de errores como el ingreso de ubicaciones inválidas, cierres abruptos, entre otros.
- Dado que TECMarket es una empresa que desea ser global, todo el diseño de la base de datos, así como la interfaz debe estar en idioma inglés.

4. Aspectos técnicos

El grupo puede seleccionar el lenguaje de programación a utilizar tanto para la capa de lógica como para la capa de la interfaz de usuario.

La base de datos para las funcionalidades 3.1 a la 3.4 debe ser una base de datos NoSQL (se puede seleccionar entre una base de datos orientada a documentos o una base de datos key-value), el grupo puede seleccionar el motor de base de datos a utilizar.

La base de datos para las funcionalidades 3.5 y 3.6 debe ser una base de datos NoSQL orientada a grafos, el motor de base de datos a utilizar es Neo4j.

Debe utilizar el API de Google Maps para obtener la información indicada en la especificación.

Deben utilizar la plataforma GitHub o GitLab, estas plataformas sirven para desarrollar código de forma colaborativa y control de versiones del mismo, pueden obtener información y descargar las herramientas en los siguientes sitios:

- GitHub: <https://github.com/>
- GitLab: <https://about.gitlab.com/>

5. Documentación

La documentación es un aspecto de gran importancia en el desarrollo de programas, especialmente en tareas relacionadas con el mantenimiento de los mismos.

Para la documentación interna, deberán incluir comentarios descriptivos para cada función, con sus entradas, salidas, y restricciones.

La documentación externa debe contener lo siguiente:

- Aspectos generales
 - Software utilizado para desarrollar el proyecto (bases de datos y aplicación web)
 - Estado del proyecto: 100% funcional o bien indicar qué no se pudo implementar.
 - Indicar cómo se distribuyeron el trabajo entre los integrantes del grupo.
- Modelo de datos implementado
 - Deberá describirse detalladamente cómo se representó la información mediante documentos o estructuras key-value.
- Pruebas de funcionalidad
 - Pruebas de funcionalidad de todos y cada uno de los requerimientos descritos en este enunciado. Desde el registro de los usuarios, pruebas de disponibilidad de la información y cada una de las consultas y sus resultados. Todo ejecutado desde la aplicación web. Debe describir en qué consiste la prueba y agregar screenshots para demostrar la funcionalidad. Para las pruebas de las consultas, los resultados deben verse mediante la interfaz de la aplicación web.

6. Evaluación

Rubro	Puntaje
Requerimientos de la aplicación	
Registro correcto y completo de clientes	4
Registro correcto y completo de supermercados	8
Registro correcto y completo de los pedidos	9
La lista de posibles sitios para realizar pedidos	3
Historial del cliente	6
Migración correcta y completa de los datos hacia la base de datos de grafos	15
Consultas del usuario administrador/empleador de la cadena de supermercados	
<ul style="list-style-type: none">• Buscar un cliente en particular y mostrar todo su historial de pedidos	9
<ul style="list-style-type: none">• Ver todos los sitios en los que los clientes han hecho pedidos.	9
<ul style="list-style-type: none">• Ver los 5 sitios para los que se han registrado más pedidos.	9
<ul style="list-style-type: none">• Dado un cliente en particular mostrar todos los demás clientes que hayan pedido al menos en un sitio en común con ese cliente.	9
<ul style="list-style-type: none">• Dado un cliente en particular mostrar todos los productos que han comprado otros clientes en los supermercados donde ha comprado ese cliente, para mostrárselos como sugerencias para próximas compras.	9
Documentación	
<ul style="list-style-type: none">• Aspectos generales	1
<ul style="list-style-type: none">• Diseño de la base de datos	4
<ul style="list-style-type: none">• Pruebas de funcionalidad completas	5
Total	100

Se asignará puntaje extra por la utilización de Loopback (5 puntos extra)

7. Aspectos administrativos

1. Debe crear un archivo .zip ("PP3.zip") que contenga únicamente un archivo **info.txt** y 2 carpetas llamadas **documentacion** y **solucion_computacional**, en la primera deberá incluir el documento de *word* o *open office* (no pdf) solicitado y en la segunda los archivos y/o carpetas necesarias para la implementación de esta tarea. El archivo **info.txt** debe contener la siguiente información (cualidades):
 - a. Nombre del curso
 - b. Número de semestre y año lectivo
 - c. Nombre del Estudiante x4
 - d. Número de carnet x4
 - e. Número de tarea programada
 - f. Fecha de entrega
 - g. Estatus de la entrega (definido por el responsable de la implementación de la tarea):
[Deplorable|Regular|Buena|MuyBuena|Excelente|Superior]
2. Deberá subir el archivo antes mencionado al TEC Digital en el curso de BASES DE DATOS AVANZADOS GR 01, en la asignación llamada "Proyecto 3" debajo del rubro de "Proyectos". El proyecto debe entregarse el jueves 21 de noviembre del 2019 antes de las 11:55 pm
3. Dentro de la carpeta de **solucion_computacional**, deberá incluir un archivo .txt (**PrimerNombreMiembro1.PrimerNombreMiembro2.PrimerNombreMiembro3 PrimerNombreMiembro4.txt**) que contenga todo el código fuente de la aplicación web.
4. Los proyectos deberán ser revisadas con la profesora o el asistente. Todos los miembros del grupo deberán participar de la revisión, ya que de lo contrario no se les asignará el puntaje correspondiente. La nota de la revisión es individual, el resto de la nota es grupal. Las citas para revisión se darán a conocer con antelación.
5. La tarea se hará en grupos de 4 personas máximo.
6. La tarea vale un 15% de la nota del curso.
7. Los grupos del proyecto se formarán el día de la asignación de este proyecto.

8. Condiciones

Este proyecto programado se rige por las siguientes condiciones:

1. Es obligatorio el diseño e implementación de una base de datos NoSQL (base de datos orientada a documentos o base de datos key-value únicamente) para almacenar la información de los sitios, clientes y sus respectivos pedidos.
2. Se debe implementar un sitio web que utilice la información almacenada en la base de datos del punto anterior. Debe contar con todas las funcionalidades antes descritas.
3. Las consultas del usuario empleado de la compañía se realizarán mediante la misma aplicación web y deberá consultar únicamente la información contenida en la base de datos de grafos.
4. Los resultados de las consultas deberán verse en la aplicación con un formato agradable al usuario. Es decir, evite presentar sólo códigos de sitios, clientes, etc.
5. Puede utilizar cualquier motor NoSQL para bases de datos orientadas a columnas o key-value que esté disponible para la implementación de la base de datos de sitios, pedidos y clientes. Si esta funcionalidad se implementa con grafos, se perderá el puntaje correspondiente.
6. Para implementar la base de datos sobre la que se realizarán las consultas deberá utilizar la base de datos de grafos Neo4j. Si las consultas no se realizan sobre la base de datos de grafos se perderá el puntaje respectivo.
7. La aplicación no es distribuida.
8. El código fuente del proyecto y la documentación (archivo doc o docx) deberán incluirse en un archivo zip o rar, con dos carpetas: solución computacional, documentación. El archivo comprimido debe tener como nombre el nombre de los integrantes del grupo. Se penalizará con 5 puntos el incumplimiento de este punto.
9. **Bajo ninguna circunstancia** se recibirán proyectos fuera de la fecha/hora indicadas, tampoco se recibirán por correo electrónico ni otro medio que no sea el TECDigital. Todo trabajo entregado después de la fecha y hora indicada tendrá una penalización de 1 punto por cada minuto de retraso.
10. Todo el código fuente y el diseño de las bases de datos deben ser creación 100% original de cada grupo de trabajo.
11. Toda copia o plagio tratará según lo establecido en el Reglamento de Enseñanza Aprendizaje.
12. Para la revisión, el grupo de trabajo deberá llevar los equipos necesarios para poder mostrar el sistema en funcionamiento.

13. Todos los miembros del grupo de trabajo deberán estar presentes en la revisión, quien no asista perderá el puntaje del proyecto.

Nota: El incumplimiento de alguna condición implicará una calificación de cero.

IMPORTANTE: CONOCIMIENTO DE LA SOLUCIÓN PRESENTADA.

En la revisión del trabajo, los estudiantes deben demostrar un completo dominio de la solución implementada, tanto desde el punto de vista técnico (uso de herramientas) como de la funcionalidad del proyecto. La revisión se puede hacer individualmente o en grupos, examinando la solución o temas específicos aplicados en el proyecto. Todos los integrantes del grupo deben tener el mismo conocimiento de la solución presentada. Recuerde que, de no contestar las preguntas durante la revisión, se puede descontar hasta un 15% de la nota obtenida de forma grupal.