



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

**Semestre:** II-2018  
**Valor:** 15%

## PROYECTO PROGRAMADO 3 TRIPTEC

### Bases de datos no SQL de grafos

#### 1. Introducción

Dada la vida tan ajetreada que llevan las personas en la actualidad, siempre buscan espacios de esparcimiento que los ayude a liberar estrés y que les permitan compartir con sus amigos, familiares, compañeros o seres queridos. Una forma de cumplir estos objetivos es realizando distintos viajes.

Para realizar los viajes muchas veces las personas buscan información anticipadamente, para organizar el viaje de una forma adecuada y así aprovechar el tiempo al máximo.

La empresa TEC startup ha identificado esta necesidad y desea realizar un sistema utilizando un API (acrónimo del inglés *Application Programming Interface*, en español Interface de Programación de Aplicaciones) el cual es un conjunto de funciones que ofrece cierta biblioteca para ser utilizadas por otros sistemas. Estas funciones, le permiten al desarrollador de una aplicación, abstraer ciertas funcionalidades de una manera efectiva y eficiente.

La empresa se está constituyendo como una agencia de viajes que tiene como objetivo mostrar contenidos diferentes para todas las personas, por lo decidió hacer una inversión inicial para desarrollar un sistema para realizar reservaciones llamado TRIPTEC y los ha contratado a ustedes para desarrollar dicho sistema. El usuario podrá acceder y gestionar información en este sistema referente a lugares de interés, viajes, entre otros. Y para ello, este sistema utilizará el API de Google Maps. En las secciones subsecuentes se describe con detalle las funcionalidades y requerimientos que debe desarrollar.

#### 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

TRIPTEC será un sistema de apoyo de una agencia de viajes, este sistema permitirá a la agencia gestionar la información de todos los sitios turísticos a los que puede llevar a un cliente, además permite gestionar la información de los clientes y los tours seleccionados por el cliente. Alguna de la información consultada será real, actualizada y deberá obtenerse de los servidores de Google Maps. Para lograr esto puede, 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.

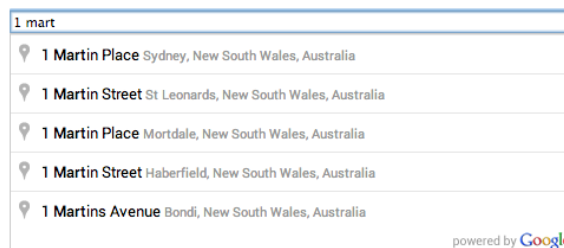
Se deberá implementar un programa para gestionar los destinos turísticos, los clientes y los tour seleccionados.

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

#### 3.1 Gestionar sitios turísticos

En esta funcionalidad, un empleado de la empresa TRIPTEC puede registrar la información de diferentes sitios turísticos que ofrece la compañía, 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, las actividades que se pueden realizar en este sitio y el precio de la estadía en este lugar por persona por día.

También se debe obtener del API la información tipo de lugar (restaurante, cine, museo, etc), 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.

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

### 3.2. Registro de clientes

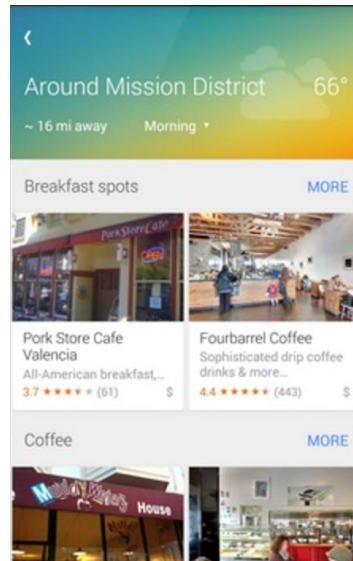
El sistema debe permitir que los clientes se registren la primera vez que ingresan al sistema, para cada cliente se debe registrar la cédula, el nombre, el teléfono, correo electrónico y la fecha de nacimiento, nombre de usuario y clave de acceso.

### 3.3. Realizar reservación

Los clientes podrán realizar reservaciones a través del sitio web. Las reservaciones se hacen para cualquier número de personas y para cualquiera de los sitios turísticos registrados en el sistema. El cliente debe ingresar al sitio con usuario y clave, si es la primera vez debe registrarse antes de hacer la reservación. Para las reservaciones se debe registrar la cantidad de personas para las que se hace la reservación, la fecha de llegada, la fecha de salida, además, opcionalmente, el cliente podría indicar alguna necesidad en particular, por ejemplo: acceso para personas con discapacidad, servicio de alquiler de cunas o coches para niños, comida vegana o vegetariana, entre otros. También, se puede indicar si se desea reservar servicios extras que brinda el sitio turístico como (caminatas, deportes acuáticos, deportes de aventura, entre otros). El usuario podría ver la información de sitios, y guardarlos en una “lista de deseos”, esta información debe estar disponible la siguiente vez que el cliente ingrese al sitio web.

El usuario podría consultar en el sistema información sobre sitios de interés (restaurantes, cines, doctores, museos y bares) cercanos a una locación específica del sitio turístico. El usuario puede seleccionar para cuál o cuáles de los sitios de interés requiere la información. El usuario deberá ingresar el radio en metros para realizar la búsqueda de los sitios de interés 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 a visitar”, al agregarlos el sistema los registrará en una lista relacionada con el sitio turístico y además se almacenará la distancia entre el sitio turístico y el posible lugar a visitar (esta información se tomará del API de Google Maps).

### 3.4. Ver historial de reservaciones

Un cliente ya registrado tendrá la opción de ver su historial de reservaciones, la página debe desplegar: cantidad de personas para las que se realiza la reservación, fecha de entrada y salida, necesidades registradas, servicios adicionales y el monto total de la reservación.

### 3.5. Migración de datos

Adicionalmente, para el administrador(a) de la agencia de viajes se requieren algunas consultas, sin embargo, la empresa 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 sitios, las reservaciones 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 reservaciones.
- Ver todos los sitios en los que los clientes han hecho reservaciones.
- Ver los 5 sitios para los que se han registrado más reservaciones.
- Dado un cliente en particular mostrar todos los demás clientes que hayan reservado al menos en un sitio en común con ese cliente.
- Dado un cliente y una reservación realizada, el sistema debe tomar los sitios almacenados en la lista de posibles sitios a visitar, y debe aplicar un algoritmo para obtener la ruta óptima para que el cliente pueda visitar los lugares cercanos al centro turístico, esta información se le debe enviar por medio de correo electrónico al usuario, en forma de recomendación para que aproveche al máximo su estancia en el centro turístico.

### 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 TRIPTEC 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 seleccionar.

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
<b>Requerimientos de la aplicación</b>	
Registro correcto y completo de clientes	4
Registro correcto y completo de sitios turísticos	8
Registro correcto y completo de las reservaciones	9
La lista de deseos del cliente está disponible en su siguiente acceso	3
Historial del cliente	6
Migración correcta y completa de los datos hacia la base de datos de grafos	15
<b>Consultas del usuario administrador/empleador de la agencia de viajes</b>	
<ul style="list-style-type: none"> <li>• Buscar un cliente en particular y mostrar todo su historial de reservaciones</li> </ul>	9
<ul style="list-style-type: none"> <li>• Ver todos los sitios en los que los clientes han hecho reservaciones.</li> </ul>	9
<ul style="list-style-type: none"> <li>• Ver los 5 sitios para los que se han registrado más reservaciones.</li> </ul>	9
<ul style="list-style-type: none"> <li>• Dado un cliente en particular mostrar todos los demás clientes que hayan reservado al menos en un sitio en común con ese cliente.</li> </ul>	9
<ul style="list-style-type: none"> <li>• Dado un cliente y una reservación realizada, el sistema debe tomar los sitios almacenados en la lista de posibles sitios a visitar, y debe aplicar un algoritmo para obtener la ruta óptima para que el cliente pueda visitar los lugares cercanos al centro turístico, esta información se le debe enviar por medio de correo electrónico al usuario, en forma de recomendación para que aproveche al máximo su estancia en el centro turístico.</li> </ul>	9
<b>Documentación</b>	
<ul style="list-style-type: none"> <li>• Aspectos generales</li> </ul>	1
<ul style="list-style-type: none"> <li>• Diseño de la base de datos</li> </ul>	4
<ul style="list-style-type: none"> <li>• Pruebas de funcionalidad completas</li> </ul>	5
<b>Total</b>	<b>100</b>

Se asignará puntaje extra por la utilización de Loopback (5% 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 0X, en la asignación llamada "Proyecto 3" debajo del rubro de "Proyectos". El proyecto debe entregarse el jueves 22 de noviembre del 2018 antes de las 08:30am (ambos grupos)

**La documentación externa se debe entregar a más tardar el día miércoles 21 de noviembre a las 8:00am, (las documentaciones que no se entreguen en esa fecha y hora tendrán una nota de 0). El archivo de la documentación deben subirlo al TEC Digital en el curso de BASES DE DATOS AVANZADOS GROX, en la asignación llamada "DocsP03" debajo del rubro de "Proyectos".**
3. Dentro de la carpeta de **solucion\_computacional**, deberá incluir un archivo .txt (**PrimerNombreMiembro1.PrimerNombreMiembro2.PrimerNombreMiembro3PrimerNombreMiembro4.txt**) que contenga todo el código fuente de la aplicación web.
4. Los proyectos deberán ser revisadas con el profesor 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, pero el proyecto se revisará el mismo día de la entrega, es decir, el jueves 22 de noviembre del 2018.
5. La tarea se hará en grupos de 4 personas, solamente.
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 turísticos, clientes y sus respectivas reservaciones.
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 agencia de viajes 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 turísticos, reservaciones 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.
14. Deben realizar entregar dos informes de avance del proyecto, los informes deben tener el siguiente formato

Semana	Del xxxx al xxxx
<b>Actividades desarrolladas durante la semana (Hacer un desglose de las actividades realizadas por cada integrante del grupo)</b>	
Integrante x	<ul style="list-style-type: none"> <li>• Act 1</li> <li>• Act 2</li> <li>• Act 3</li> </ul>
<b>Actividades planeadas para la próxima semana</b>	
Integrante x	<ul style="list-style-type: none"> <li>• Act 1</li> <li>• Act 2</li> <li>• Act 3</li> </ul>

Por favor trabajen con ética y en este informe indiquen con honestidad la distribución de tareas. Indiquen explícitamente quién no está trabajando o quién está trabajando menos.

Se deben entregar dos informes:

- a. El primero debe entregarse el lunes 5 de noviembre antes de las 11:55pm(Comprende el trabajo hecho desde el día de la asignación del proyecto hasta el domingo 4 de noviembre)
- b. El segundo debe entregarse el viernes 16 de noviembre antes de las 11:55pm (comprende el trabajo hecho desde el día lunes 5 de noviembre hasta el viernes 16 de noviembre)

Estos informes los debe enviar al correo electrónico [martavia@itcr.ac.cr](mailto:martavia@itcr.ac.cr) con el asunto [Avanzados]-Informe x, no es necesario que lo envíen todos los integrantes, con solo un integrante que lo envíe (deben copiar en ese mismo correo al resto de compañeros)

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.