



**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 2 TECPlane

Base de datos NoSQL – Bases de datos orientadas a documentos

### 1. Introducción

Actualmente, viajar se ha convertido en una de las metas más soñadas por las personas. Muchas personas ahorran para lograr hacerlo y las agencias de viajes formulan ofertas para atraer más clientes.

Las personas constantemente adquieren boletos de avión para viajar por diferentes motivos y contar con una manera automatizada para adquirirlos hace que las personas tengan una mejor percepción del servicio brindado por las distintas aerolíneas.

El TEC ha decidido implementar un sistema para registrar la información de aeropuertos con las respectivas aerolíneas que operan en cada uno y los vuelos disponibles para los pasajeros, y ustedes han sido seleccionados para realizar dicha implementación según lo indicado en este documento.

### 2. ¿Qué se busca con este proyecto?

- Mejorar las habilidades de resolución de problemas.
- Aumentar el conocimiento del estudiante sobre bases de datos distribuidas.
- Aumentar el conocimiento del estudiante sobre bases de datos NoSQL
- 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

Se le ha encargado el diseño y la implementación de un sitio web para registrar la información de vuelos que será utilizado por distintos aeropuertos.

El sistema consta de dos módulos: el módulo administrativo y el módulo de gestión de boletos.

#### 3.1. Módulo administrativo

El módulo administrativo debe permitir registrar la siguiente información (hacer las operaciones de CRUD sobre dicha información):

##### documento **Aeropuerto**

Para cada aeropuerto se guarda la siguiente información: código del aeropuerto, nombre, lugar donde está ubicado, información de contacto, dirección del sitio web.

##### documento **Funcionario**

En cada aeropuerto los pasajeros pueden ser atendidos por diferentes funcionarios, para cada funcionario se almacena la identificación (cédula), nombre completo, tipo de funcionario (Administrativo, Técnico, Operario), fecha de ingreso a la institución y área en la que trabaja (Counter, Salas de abordaje, Carga, Mantenimiento). Las áreas pueden aumentar y variar.

##### documento

##### **Aerolíneas**

En esta sección se registra la información de las diferentes aerolíneas que operan en un determinado aeropuerto, dichas aerolíneas tienen registrada la siguiente información: identificador, nombre, países en los que realiza vuelos.

##### documento **Vuelos**

El catálogo de vuelos disponibles tiene registrada la siguiente información: identificador, nombre, origen, destino, itinerario, precio, restricciones (pueden variar), características del vuelo (servicios brindados a los pasajeros), estado (a tiempo, retrasado, cancelado), capacidad máxima.

#### 3.2. Módulo de gestión de boletos

El módulo de información debe permitir registrar la siguiente información:

documento **Pasajeros**

Los pasajeros son las personas que serán atendidas en los diferentes aeropuertos, la información que se debe almacenar es: identificación (cédula), nombre completo, fecha de nacimiento, nacionalidad, lugar de residencia, teléfono (puede tener varios teléfonos registrados), correo electrónico.

acción:  
update **Compra de boletos = Vuelo**

Los pasajeros pueden comprar sus boletos a través del sistema, para comprar un boleto deben indicar el origen y destino y un rango de fechas en las que desea viajar, el sistema le mostrará una lista con los vuelos que cumplen con los datos proporcionados por el pasajero y el pasajero podrá seleccionar uno de los vuelos e indicar el número de boletos que requiere para dicho vuelo, cantidad de maletas registradas, además pueden indicar alguna observación adicional que consideren importante (esta información es opcional). Al finalizar esta funcionalidad el sistema debe asignar a los boletos el estado de comprado.

acción:  
update **Check-In**

Esta funcionalidad permite que los pasajeros realicen el check-in del vuelo, para esto deben indicar su número de identificación y el código del vuelo y posteriormente el sistema les asignará el número de asiento en el vuelo y cambiará el estado del boleto a chequeado.

acción:  
update **Abordaje**

Los funcionarios de la sala de abordaje pueden ingresar al sistema e indicar el número de vuelo y número de cédula del pasajero y cambiar el estado del boleto a utilizado.

Se debe implementar un esquema de seguridad simple: Cada usuario debe tener un código de usuario y una contraseña. Para la contraseña se debe buscar alguna forma de encriptación de tal manera que si consulta la base de datos físicamente no se pueda obtener el valor real de la contraseña.

Se deben crear tres tipos de usuarios (pasajero, administrador, funcionario) y el sistema debe garantizar que estos usuarios pueden tener acceso a las siguientes funciones:

Usuario	Función
Administrador	Tienen acceso a todas las funcionalidades del módulo de administración. Puede ejecutar los siguientes reportes: <ul style="list-style-type: none"><li>Para todas las aerolíneas debe mostrar todos los vuelos junto con la cantidad de boletos vendidos en cada uno y el monto total correspondiente a los boletos vendidos.</li><li>Rango de boletos comprados por cada pasajero. El rango va del menor al mayor número de boletos adquiridos por un pasajero. Por ejemplo, si Ana ha comprado boletos para 5 vuelos, y se identifica que en el vuelo que menos boletos compró, adquirió uno y en el que más boletos compró, adquirió tres, entonces su rango será [1,3]</li></ul>

	<ul style="list-style-type: none"> <li>• ¿Cuáles son los destinos más visitados? Se debe mostrar el nombre de cada destino y la cantidad de pasajero que han comprado vuelos para ese destino.</li> <li>• Cantidad de operaciones de compra de boletos registradas en el sistema, esta información se puede filtrar por pasajero, por rango de fechas, por estado de vuelo. También mostrar los tres pasajeros con más vuelos adquiridos.</li> </ul>
<b>Paciente</b>	<ol style="list-style-type: none"> <li>1. Registrar información de paciente (CRUD)</li> <li>2. Compra de boletos</li> <li>3. Check-in</li> <li>4. Ejecutar los siguientes reportes: <ul style="list-style-type: none"> <li>• Vuelos asociados al pasajero: se muestra un listado con la información de los vuelos para los que un pasajero ha adquirido boletos. Se puede filtrar por rango de fechas, por estado del vuelo. Para cada vuelo se debe mostrar el detalle correspondiente.</li> </ul> </li> </ol>
<b>Funcionario</b>	<ol style="list-style-type: none"> <li>1. Verificar información de check-in</li> <li>2. Consultar información de pasajeros</li> <li>3. Ejecutar función de abordaje</li> <li>4. Ejecutar los siguientes reportes: <ul style="list-style-type: none"> <li>• Consultar información de un pasajero.</li> <li>• Vuelos registrados en el sistema: se muestra un listado con la información de los vuelos registrados. Se puede filtrar por rango de fechas, por estado, por nombre de pasajero. Para cada vuelo se debe mostrar el detalle.</li> </ul> </li> </ol>

#### 4. Por hacer

1. Implementar una base de datos distribuida utilizando una base de datos orientada a documentos, a fin de almacenar la información indicada en el punto 3. Se trata de una base de datos distribuida y se trabajará con 3 nodos.
2. Implementar un sitio web que funcione de manera distribuida, utilizando la información almacenada en la base de datos del punto anterior. Debe contar con todas las funcionalidades antes descritas.
3. El sitio web deberá estar disponible en los tres nodos y podría haber usuarios utilizando el sistema de manera simultánea en cada uno de ellos.
4. El usuario puede ingresar a realizar consultas desde cualquier nodo.
5. Debe existir alta disponibilidad de la información, es decir, si un nodo deja de funcionar sus datos deben seguir disponibles.
6. Los resultados de las consultas deberán verse en la aplicación web con un formato agradable al usuario. Es decir, evite presentar sólo códigos.

## 5. Puntos a ser evaluados

1. **Correctitud** de la solución computacional
2. **Robustez y resiliencia** de la solución computacional
3. **Implementación de una base de datos NoSQL distribuida**
4. **Implementación de aplicación** con sus respectivos módulos.
5. **Conocimiento** de la solución generada.
6. **Desarrollo** de documentación efectiva.
7. **Uso de mejores prácticas** de programación en su código. Puede revisar los principios del programador y olores de software (Clean Code).

## 6. Aspectos técnicos

- Para la implementación del sistema debe utilizar MongoDB.
- Puede utilizar el lenguaje de programación de su preferencia para implementar la interfaz de usuario, el único requisito es que la interfaz sea web.

## 7. Evaluación

Rubro	Puntaje
<b>Base de datos NoSQL</b>	
Implementación de una base de datos orientada a documentos	10
Disponibilidad de la información	10
<b>Requerimientos de la aplicación*</b>	
Funcionamiento de las aplicaciones en los tres nodos simultáneamente en los diferentes roles	10
Operaciones CRUD del módulo de administrador	5
Funcionalidades asociadas al pasajero	10
Funcionalidades asociadas al funcionario	10
Consultas del administrador	
• Vuelos	5
• Rango de boletos comprados	10
• Destinos más visitados	10
• Cantidad de operaciones de compra de boletos	10
<b>Documentación</b>	
• Aspectos generales	1
• Diseño de la base de datos	5
• Pruebas de funcionalidad completas	4
<b>Total</b>	<b>100</b>

\* Para todas las funcionalidades se debe garantizar que exista una alta disponibilidad de la información, aún si un nodo deja de funcionar

## 8. Aspectos administrativos

1. Entregar **un documento** con al menos los siguientes apartados. Los apartados DEBEN entregarse en ese orden:
  - a. Portada.
  - b. Aspectos generales: indicar en este apartado cuáles herramientas se utilizaron para implementar el sistema.  
Debe incluir también el estado actual del proyecto: si es 100% funcional o bien indicar qué no se pudo implementar y por qué.
  - c. Diseño de la base de datos: Deberá describirse detalladamente cómo se representó la información mediante documentos. Debe describir cada campo contenido en un documento.
  - d. Pruebas de funcionalidad: se deben incluir pruebas de funcionalidad de todos los requerimientos descritos. Debe describir en qué consisten las pruebas y agregar un *screenshot* donde se demuestre la funcionalidad. Si alguna funcionalidad queda sin documentar, no será tomada en cuenta durante la evaluación y el grupo de trabajo perderá el puntaje respectivo.
2. Debe crear un archivo .zip (“PP2.zip”) que contenga únicamente un archivo **info.txt** y 2 carpetas llamadas **documentación**, **fuentes**, en la primera deberá incluir el documento solicitado en formato pdf y en la segunda el código fuente de la aplicación (todos sus componentes). El archivo **info.txt** debe contener la siguiente información (calidades):
  - a. Nombre del curso
  - b. Número de semestre y año lectivo
  - c. Nombre del Estudiante x4
  - d. Número de carnet x4
  - e. Número de proyecto programado
  - f. Fecha de entrega
  - g. Estatus de la entrega (definido por el responsable de la implementación de la tarea, debe ser **CONGRUENTE** con la solución entregada):  
[Deplorable|Regular|Buena|MuyBuena|Excelente|Superior]
3. Deberá subir el archivo antes mencionado al TEC Digital en el curso de BASES DE DATOS AVANZADOS GR 0X, en la asignación llamada “P02” debajo del rubro de “Proyectos programados”. El proyecto debe entregarse a más tardar el **domingo 27 de octubre a las 11:55pm**
4. Dentro de la carpeta de **aplicacion**, deberá incluir un archivo .txt (**PrimerNombreMiembro1.PrimerNombreMiembro2.PrimerNombreMiembro3.PrimerNombreMiembro4.txt**) que contenga todo el texto de la solución del o de los archivos presentados (las implementaciones). Este archivo puede ser revisado en el sistema de Control de Plagio del TEC Digital. **Todo el código de cada proyecto debe ser 100%**

**original, y no se va a tolerar el plagio.** Este archivo también deberá ser subido al TEC Digital en el curso de BASES DE DATOS AVANZADOS GR 0X, en la asignación llamada “P02 (archivo TXT)” debajo del rubro de “Proyectos”. Se deberá subir el archivo siguiendo los mismos rangos de tiempo que los descritos en el punto anterior.

5. Los proyectos deberán ser revisadas con el profesor o el asistente. Todos los miembros del grupo deberán participar de la revisión, de lo contrario a la nota obtenida grupalmente se le descontará 15%. Además, si durante la revisión el estudiante no puede demostrar conocimiento de la solución presentada, se le descontará hasta un 15% de la nota grupal.
6. El proyecto se hará en grupos de 4 personas, solamente, pueden mezclarse integrantes de ambos grupos.
7. El proyecto vale un 15% de la nota del curso.
8. El día de la asignación del proyecto se pasará una hoja en la que deben anotar los integrantes de cada grupo de trabajo.

## 9. Condiciones

Este proyecto programado se rige por las siguientes condiciones:

1. El desarrollo del proyecto es estrictamente en grupos de 4 integrantes.
2. Debe cumplir con todo lo indicado en esta especificación.
3. Deberá entregarse en tiempo y forma según el plazo establecido por la profesora al momento la lectura de este documento.
4. La aplicación debe poder ejecutarse desde cualquiera de los tres nodos simultáneamente inclusive.
5. No 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 nota de cero.**
6. Si se detecta copia o plagio se procederá según lo indicado en el Reglamento del Régimen de Enseñanza Aprendizaje del ITCR.
7. Las citas de revisión del proyecto se asignarán después de la entrega del mismo.
8. Para la revisión, el grupo de trabajo deberá llevar los equipos necesarios para poder mostrar el sistema distribuido en funcionamiento.
9. No se harán revisiones previas de los diseños.

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.