



Universidad Tecnológica Nacional
Facultad Regional Buenos Aires

Gestión de Datos

Trabajo Práctico

2° Cuatrimestre 2015

AEROLINEA FRBA

Enunciado V1.0



Índice

Índice	2
Introducción.....	3
Objetivos generales.....	3
Descripción general	3
Componentes del TP.....	4
Base de Datos y Modelo de Datos.....	4
Aplicación Desktop	4
Requerimientos	4
General.....	4
Base de Datos	4
Aplicación Desktop	5
General.....	13
Base de Datos	13
Aplicación Desktop	14
Restricciones de la solución	16
Condiciones de aprobación.....	16
Testing	16
Modelo de Datos.....	17
Consultas SQL.....	17
Respetar Guía de ABMs.....	17
Aplicación Desktop	17
Fecha de entrega y condiciones	17
Sobre los grupos	18
Ayuda y contacto	18
Donde aprender C#.....	19
Sobre la elección de C#	19
Obtención de herramientas	20
Formato de entrega	21
Lugar de envío	21
Estructura del archivo zip.....	22
Readme.txt:.....	22
Estrategia.pdf:.....	22
\src:	22
\data:	22
10 Tips para desaprobación Gestión de Datos	23

Introducción

Objetivos generales

El presente trabajo práctico persigue los siguientes objetivos generales

- Promover la investigación de técnicas de base de datos.
- Aplicar la teoría vista en la asignatura en una aplicación concreta.
- Desarrollar y probar distintos algoritmos sobre datos reales.
- Utilizar un lenguaje de programación de última generación.
- Fomentar la delegación y el trabajo en grupo.

Descripción general

Mediante este trabajo práctico se intenta simular la implementación de nuevo sistema para una Aerolínea, sus terminales kiosco o autoservicio. Además a dicha solución se le debe agregar el programa de pasajero frecuente, del cual no había registro anteriormente.

Para simplificar la resolución, se debe tener en cuenta que los datos migrados, de cada pasaje o encomienda se traten como una compra independiente, es por eso, que una compra hace referencia a una compra de pasaje o a una encomienda.

Otra consideración a tener en cuenta es que una aeronave parte de su origen si o si llega a su destino.

La implementación de dicho sistema requiere previamente realizar la migración de los datos que se tenían hasta el momento.

Se necesita que se reformulen los procesos y el diseño de la base de datos que cumpla con las nuevas restricciones y se adecue a la situación actual.

Componentes del TP

El alumno recibirá dos componentes ya hechos del sistema y, en base a estos deberá crear uno nuevo e implementar nuevas funcionalidades. Los componentes a recibir son:

Base de Datos y Modelo de Datos

La cátedra provee un script que permite crear una base de datos en el motor SQL Server 2012. Esta base de datos incluye el modelo de una única tabla, llamada maestra, que es cargada con datos provistos por la cátedra. Los datos de esa tabla se encuentran desorganizados y no poseen ningún tipo de normalización.

El alumno deberá estudiar los datos recibidos y confeccionar un modelo de datos que siga todos los standards de desarrollo de bases de datos explicados durante la cursada.

Los datos de esta tabla maestra pertenecen a un **dominio de compra de pasaje y encomiendas**.

El sistema a desarrollar será utilizado por **2 tipos de usuarios** distintos: **administradores** y **clientes**.

Parte de la lógica del negocio a resolver deberá ser inferida por el alumno, en base a las columnas y valores presentes en los datos. De todas maneras es recomendable consultar al grupo de Google de la materia antes de tomar decisiones incorrectas.

Aplicación Desktop

La cátedra provee un proyecto C# a modo de template, sobre el cual deberá desarrollarse una aplicación Desktop que interactúe con la nueva base de datos, cuyo diseño estará a cargo de los alumnos. La aplicación deberá ser del tipo Desktop desarrollada sobre C# con **Visual Studio 2012** y **Framework de .NET 4.5**.

Esta aplicación tendrá diversas pantallas, reportes y formularios que permitirán interactuar, cargar y visualizar la información de la base de datos de SQL Server.

Requerimientos

General

El alumno deberá crear todos los componentes de base de datos e implementar todas las funcionalidades pedidas para la aplicación Desktop, cumpliendo con las siguientes pautas:

Base de Datos

El alumno deberá crear un modelo de datos que **organice y normalice** los datos de la única tabla provista por la cátedra. Este modelo de datos incluye:

- Creación de nuevas tablas y vistas.

- Creación de claves primarias y foráneas para relacionar estas tablas.
- Creación de constraints y triggers sobre estas tablas cuando fuese necesario.
- Creación de los índices para acceder a los datos de estas tablas de manera eficiente.
- Migración de datos: Cargar todas las tablas creadas utilizando la totalidad de los datos entregados por la cátedra en la única tabla del modelo. Para este punto deberán utilizarse Stored Procedures sobre la base de datos. No podrá realizarse la migración de datos utilizando la aplicación Desktop ni ninguna otra herramienta auxiliar.

El alumno deberá entregar un único archivo de Script que al ejecutar realice todos los pasos mencionados anteriormente, en el orden correcto. Todo el modelo de datos confeccionado por el alumno deberá ser creado y cargado correctamente ejecutando este Script una única vez, antes de empezar a testear la aplicación Desktop.

Todas las columnas creadas para las nuevas tablas **deberán respetar los mismos tipos de datos** de las columnas existentes en la tabla principal. A su vez el alumno podrá crear nuevas columnas, claves e identificadores para satisfacer sus necesidades. Pero nunca se podrá inventar información, por ejemplo crear un cliente que nunca existió.

Aplicación Desktop

El alumno deberá crear una aplicación Desktop en C# sobre Visual Studio 2012 con Framework .NET versión 4.5. Esta aplicación deberá contar con formularios, reportes y tablas. Todos estos componentes deberán respetar los lineamientos planteados en el documento “Guía de ABMs”. Es recomendable leerlo en este punto antes de continuar con el enunciado.

En su primera fase de prototipo el sistema contará con el siguiente listado que se detalla a continuación:

1. ABM de Rol
2. Login y seguridad
3. Registro de Usuario
4. ABM de Ciudad
5. ABM de Ruta Aérea
6. ABM de Aeronave
7. Generación de Viaje.
8. Registro de Llegada a Destino.
9. Compra de pasaje/encomienda
10. Devolución/Cancelación de pasaje y/o encomienda.
11. Consulta de millas de pasajero frecuente.
12. Canje de millas
13. Listado Estadístico

El listado total de funcionalidades del sistema es fijo y no varía.

La funcionalidad de la aplicación deberá responder a los siguientes requerimientos de negocio:

1. ABM de Rol

Funcionalidad para poder crear, modificar y eliminar el acceso de un usuario a una opción del sistema.

Crear un rol implica cargar los siguientes datos:

- Nombre
- Listado de Funcionalidades (selección acotada)

Todos los datos mencionados anteriormente son obligatorios.

Un rol posee un conjunto de funcionalidades y las mismas no pueden estar repetidas dentro de un rol en particular.

Debe tenerse en cuenta, que **actualmente existen 2 roles**, uno **Administrativo** y otro **Cliente**.

En la modificación de un rol se pueden alterar ambos campos: el nombre y el listado de funcionalidades. Se deben poder quitar de a una las funcionalidades como así también agregar nuevas funcionalidades al rol que se está modificando.

La eliminación del rol implica una baja lógica del mismo. **El rol debe poder inhabilitarse. No está permitida la asignación de un rol inhabilitado a un usuario, por ende, se le debe quitar el rol inhabilitado a todos aquellos usuarios que lo posean.**

Se debe poder volver a habilitar un rol inhabilitado desde la sección de modificación. Esto no implica recuperar las asignaciones que existían en un pasado.

Para elegir el rol que se desea modificar o eliminar se debe mostrar un listado con todos los roles existentes en el sistema.

2. Login y Seguridad

Como bien se dijo en el punto anterior, solo existen 2 roles. Los únicos usuarios que se loguean en el sistema son aquellos que tengan el perfil Administrador, con acceso a todas la funcionalidades del sistema. Esto es así dado que **los usuarios “cliente” acceden al sistema por medio de las terminales kiosco o terminales autoservicio**. En caso de que un cliente se acercase a un punto de venta de la aerolínea (oficina comercial o el mismo aeropuerto), un administrativo se encargará de tomar el pedido, dicho administrativo se logueará en el sistema y realizara la operación de compra que el cliente solicite.

Al ejecutar la aplicación el usuario cliente no podrá acceder a ninguna funcionalidad del tipo administrativa.

El proceso de Login pedirá al usuario su **Username** y su **Password**. Si el Login es correcto, el usuario administrativo podrá acceder a las funcionalidades del sistema.

Bajo ningún concepto un usuario cliente se puede enterar/ver las funcionalidades propias de un administrativo.

Si el Login es incorrecto el usuario no podrá acceder al sistema bajo el **perfil administrativo**, se debe volver a mostrar el Login para que intente nuevamente. El sistema debe llevar un registro de cantidad intentos fallidos de login. Luego de 3 intentos fallidos en cualquier momento, el usuario debe ser inhabilitado. Si al realizar un

logueo satisfactorio, el sistema deberá limpiar la cantidad de intentos fallidos que tenga ese usuario en dicho momento.

El Login se considera una funcionalidad de características especiales. No se considera una funcionalidad que puede ser asignada a un rol. Solo los usuarios administrativos tienen la capacidad de utilizar el Login.

3. Registro de Usuario

Para reducir el tiempo de confección del TP y así beneficiar al alumno se determinó que no será necesario que se implemente/codifique el ABM de usuarios. Pero deberán tener en cuenta que al momento de realizar la entrega de dicho TP, deben entregar un set de usuarios con perfil Administrador para que pueda ser probada la funcionalidad de login. ESTO NO JUSTIFICA BAJO NINGUNA CIRCUNSTANCIA NO MODELAR LA ENTIDAD USUARIO.

Los alumnos determinarán cuál será el username para cada uno de los usuarios entregados (que deberán ser detallados en la estrategia del TP) y la password a asignar será obligatoriamente **w23e** a fin de unificar criterios y sencillez de pruebas.

El username debe ser único en un todo el sistema. La aplicación deberá controlar esta restricción e informar debidamente al usuario.

El password deberá almacenarse encriptado de forma irreversible bajo el algoritmo de encriptación **SHA256**.

4. ABM de Ciudades

Funcionalidad que permite a un administrativo crear, modificar y dar de baja una ciudad del sistema. Estas ciudades son posibles orígenes y destinos que un cliente podrá elegir al momento de realizar su viaje.

En caso de querer dar de baja una ciudad, será necesario que se chequee si existen viajes ya programados para esa ciudad. Si el viaje tiene pasajes vendidos, se tendrá que realizar la devolución correspondiente a cada uno de los clientes.

Para reducir el tiempo de elaboración a los alumnos, la cátedra determina que no será necesaria la implementación de dicha funcionalidad.

5. ABM de Ruta Aérea

Esta funcionalidad es la que permite a un administrativo crear, modificar, o dar de baja las diferentes tuplas origen y destino que pueden realizar las aeronaves habilitados.

Una aeronave tiene, una ciudad de inicio y otra de fin de viaje.

Cada ruta se caracteriza, además de tener la ciudad de origen y destino, por indicar el tipo de servicio que se brinda (por ej. turista, ejecutivo, primera clase). Para un mismo origen y destino puede existir más de un tipo de servicio. *Para un mayor entendimiento sobre el tipo de servicio leer atentamente el punto 6.

Hay que tener en cuenta que el costo de un pasaje se calcula de la siguiente manera: una ruta tiene un precio base entre la ciudad de origen y destino, en función de dicho precio base, se le aplica un porcentaje adicional en función del tipo de servicio que se esté brindando. Para el caso de las encomiendas el precio base se reemplaza por el concepto de precio base por kilo transportado.

En caso de que se necesite dar de baja un determinado recorrido, la misma será en forma lógica y se cancelarán todos los pasajes y/o encomiendas que fueron vendidos

Mínimamente una ruta debe registrar los siguientes campos de manera obligatoria:

- Código de ruta
- Tipo de Servicio
- Ciudad de Origen
- Ciudad de Destino
- Precio base por Kg
- Precio base por pasaje

6. ABM de Aeronaves

Esta funcionalidad permite a un administrativo incorporar, modificar o dar de baja una aeronave. Una aeronave se caracteriza por tener una cantidad X de butacas, que en este trabajo práctico, serán TODAS del mismo tipo. Esto quiere decir que no existe la posibilidad que dentro de un mismo avión haya más de un tipo de servicio, esto que se comenta es para simplificar la solución a desarrollar. También es necesario determinar para cada butaca si la misma es pasillo o ventanilla. No existiendo butacas para tipificadas como salida de emergencia.

Se debe tener en cuenta que cada avión tiene un espacio reservado para el envío de encomiendas, dicho espacio es independiente del espacio que el avión tiene para transportar las valijas de los pasajeros. Este espacio (para facilitar su implementación) estará medido en KG, solo preocupándonos por esta magnitud y no por el tamaño de dicha encomienda, importándonos solamente cuánto pesa el paquete. Se garantiza que si se cumple con la restricción de pesaje, el espacio físico también se cumplirá.

Otra cuestión a tener en cuenta es que un avión pertenece a un fabricante particular.

Se debe prever la posibilidad de dar de baja una aeronave (mediante baja lógica), en la cual dicha aeronave no volverá a ser utilizada en el servicio de pasajeros, dado que se completó su vida útil. También se deberá tener en cuenta otro tipo de “baja”, la que hace referencia a la no utilización de una aeronave ya que la misma está fuera de servicio por problemas técnicos y no está apto momentáneamente para el transporte de pasajeros. Esta situación momentánea (fuera de servicio) se tendrá que poder revertir indicando la fecha en la que la aeronave volverá a estar disponible para su utilización.

Para la implementación de estas 2 situaciones anteriormente descriptas, se tiene que tener en cuenta lo siguiente: Si la aeronave ya tenía uno o más rutas programados con pasajes/encomiendas vendidos/as, se debe dar la posibilidad de cancelar dichos pasajes/encomiendas o de suplantar la aeronave actual por otra de la flota (del mismo tipo y fabricante y mismo tipo de servicio) que tenga la capacidad de cumplir con los recorridos ya programados de la aeronave a reemplazar. Si la inactividad es por fin de la vida útil se reemplazan todos los viajes futuros, si la inactividad es por fuera de servicio, solo se lo reemplaza por los días que no esté activa la aeronave en cuestión. En caso de reemplazar una aeronave, los viajes no son cancelados y es totalmente transparente para el cliente.

Si se llegase a dar el caso que ninguna aeronave pueda reemplazar a otra, será necesario que se de alta una nueva aeronave. Se aclara que la decisión final sobre si se cancelan o no los viajes, sea por el motivo que fuese, corre por cuenta de la empresa que es quien decide según su criterio y políticas financieras, indicándole y dándole la directiva al administrativo de que es lo que debe hacer, si reemplaza la aeronave o cancelar todos los pasajes ya comprados (pasajes y encomiendas). En función de lo que se le haya comunicado, el administrativo realizará la acción que corresponda.

Mínimamente se debe poder registrar de manera obligatoria (según corresponda) los siguientes campos:

- Fecha de alta.
- Número de aeronave
- Modelo
- Matrícula
- Fabricante
- Tipo de servicio
- Baja por fuera de servicio
- Baja por vida útil
- Fecha de fuera de servicio (en el caso que corresponda)
- Fecha de reinicio de servicio(en el caso que corresponda)
- Fecha de baja definitiva(en el caso que corresponda)
- Cantidad de butacas y su tipo (Ventanilla/Pasillo).
- Cantidad de Kgs disponibles para realizar encomiendas.

Cabe aclarar que se debe validar que la matrícula sea única.

7. Generar Viaje

Esta funcionalidad permite al usuario administrativo confeccionar una ruta de viaje disponible para ser vendida, en donde se asociará un determinado recorrido con una aeronave en particular, en una fecha futura determinada. Se debe validar que la fecha sea mayor a la fecha actual.

Una vez generado esto, se encontrará disponible la venta de pasajes/encomiendas para la nueva ruta confeccionada.

Cuando se esté generando dicho viaje se tendrá que controlar que el servicio de la ruta concuerde con el tipo de servicio que brinda la aeronave seleccionada.

También se debe tener en cuenta que al momento de seleccionar una aeronave esta deba estar disponible y no estar asignada previamente a otro viaje en la fecha que se está ingresando.

El tiempo estimado que tarda una aeronave en llegar a cualquier ciudad de destino nunca es mayor a 24hs.

Datos a Registrar mínimamente en forma obligatoria:

- Fecha Salida (con horas, minutos y segundos).
- Fecha Llegada (con horas, minutos y segundos).
- Fecha Llegada estimada (con horas, minutos y segundos).
- Aeronave.
- Ruta Aérea.

8. Registro de Llegada Destino

Esta funcionalidad permite a un administrador ingresar la fecha y hora en la que efectivamente una aeronave llegó a su destino.

Para el desarrollo de esta funcionalidad será necesario que se ingrese la matrícula de la aeronave en cuestión, junto con el aeropuerto de donde proviene y el aeropuerto de

donde arribó. El sistema informará los datos de la aeronave y validará si dicho avión efectivamente debía llegar al aeropuerto al cual arribó.

Este registro no se realiza en el momento de arribo de una aeronave, sino que en algún momento del día se realiza la carga de todos los aviones que fueron arribando en un determinado momento.

9. Compra de Pasaje/Encomienda

Esta funcionalidad permite a un cliente/administrativo realizar la compra de un pasaje/encomienda. Para realizar la compra de un pasaje, se deberá ingresar la **fecha de viaje** junto con la **ciudad de origen** y la **ciudad de destino**. El sistema informará si hay viajes disponibles para esa fecha, junto con las cantidades de butacas libres y kgs disponibles para encomiendas como así también el tipo de servicio que brinda la aeronave. Luego de ello será necesario que se ingrese la cantidad pasajes a comprar y/o kgs a enviar vía encomienda.

Tener en cuenta que no existe el concepto de reserva. Solo pueden venderse recorridos que tengan asignadas aeronaves habilitadas.

Luego de seleccionar un determinado viaje, se procederá a ingresar los datos propios de los pasajeros y/o encomienda.

El sistema deberá contar con una base de datos de clientes en la cual se irán guardando por lo menos los siguientes datos:

- Nombre y apellido
- Dni con su número de documento
- Dirección
- Teléfono
- Mail (opcional)
- Fecha de nacimiento

Si el pasajero ya viajó alguna vez con la Aerolínea FRBA, al momento de ingresar su número de documento, el sistema se encargará automáticamente de cargar el resto de los datos personales. Se debe permitir que luego de dicha carga, los datos puedan ser actualizados. Si el pasajero nunca viajó por la Aerolínea se ingresarán uno a uno los campos solicitados.

El sistema deberá permitir la selección de la butaca a comprar por pasajero. Si el proceso de compra no concluye la/s butaca/s seleccionada/s deberán ser liberadas.

Al momento de seleccionar la butaca, será necesario que se informe si la misma corresponde a pasillo o ventanilla.

Si dentro de la compra se necesita enviar una encomienda, los datos a ingresar son los mismos que para un pasajero, adicionándole la cantidad de Kgs que se pretenden enviar.

Todos los pasajes/encomiendas que se venden en una compra hacen referencia a la misma ciudad de destino.

Una vez cargados los datos conforme al pasaje y/o encomienda el sistema pedirá que se ingrese los datos necesarios para hacer efectiva la compra.

Si la compra se está realizando desde una terminal kiosco, el único medio de pago será por tarjeta de crédito (TC), para ello se deberá determinar el número de documento de quien realiza el pago, que puede ser o no uno de los pasajeros ingresados anteriormente. Si dicho documento se encuentra en la base de datos se cargaran todos sus datos personales con posibilidad de modificación, luego de ello será necesario que

ingrese el número de la tarjeta de crédito, su código de seguridad, su fecha de vencimiento (MMAA) y tipo de tarjeta de crédito. Dependiendo del tipo de tarjeta se podrá realizar la compra en cuotas. Finalizada esta carga se informará al cliente su PNR (Passenger Name Record) o código de compra, con dicho número se deberá acercar el día del viaje a canjear sus pasajes y/o entregar el paquete encomienda.

Si la compra se está realizando en un punto de venta (oficinas comerciales o aeropuerto), el administrativo será quien cargue estos datos, permitiendo además que el medio de pago sea en efectivo, pero de igual manera se deberá dejar registro de quien fue la persona que realizó el pago, de la misma forma que una venta con TC.

Finalizada la carga, el sistema informara el monto a abonar, generara un número de compra o PNR, entregando al cliente un comprobante de dicha transacción junto con los pasajes y/o recepción del paquete encomienda según corresponda.

Para ambos tipos de compra será necesario que se registre la fecha de compra y un código para cada pasaje (persona/encomienda), las mismas también deben estar impresas en el pasaje.

Una validación a tener en cuenta es que un pasajero no puede viajar a más de un destino a la vez.

10. Cancelación/Devolución de pasaje y/o encomienda

Cuando un administrativo hace uso de esta funcionalidad le permitirá cancelar un pasaje y/o encomienda por expreso pedido del cliente que realizó la compra.

Esta devolución lo que hace es liberar la butaca de un viaje en cuestión o la liberación de Kgs utilizados en la bodega para encomiendas. La liberación puede ser en forma parcial o total, puede devolverse uno o más pasajes que fueron comprados como así también la devolución completa de la encomienda (no está permitido la liberación de Kgs parciales, si mi encomienda pesa 10Kg se liberan 10Kg) . Si la cancelación pose más de un ítem, todos ellos compartirán el mismo código de devolución.

El proceso de cancelación además de liberar lo anteriormente expuesto, procede a no computar los posibles puntos que se hubiesen acumulado en el programa de pasajero frecuente. Los puntos se hacen efectivos una vez registrada la llegada de la aeronave a destino. El proceso de cancelación pide como mínimo la registración de los siguientes datos.

- Fecha de devolución
- Número de compra o PNR
- Código de pasaje y/o encomienda.
- Motivo de cancelación.

La devolución de dinero se realizará en función de cómo se haya realizado la compra. Si fue en efectivo será en efectivo y si fue con tarjeta de crédito será vía tarjeta de crédito.

11. Consulta de millas de pasajero frecuente

Este caso de uso es utilizado por los clientes para consultar cuantas millas tienen acumuladas en su cuenta de pasajero frecuente. En ella se irán acumulando las millas por los viajes realizados y/o paquetes enviados por encomienda.

Las millas se obtienen por cada \$10 del valor de los viajes realizados y/o encomiendas realizadas, aunque el pasaje o encomienda no hubiese sido pagado por él.

Las millas tiene una validez de 1 año, (365 días).

Para realizar la consulta, solo se deberá ingresar el dni del cliente.

La consulta de puntos incluirá un listado en donde se detalle cómo se fueron acumulando esas millas, incluyendo los canjes realizados.

12. Canje de millas de pasajero frecuente

Funcionalidad que permite a un cliente cambiar sus millas de pasajero frecuente por beneficios. Simplemente el cliente elegirá una recompensa que estará al alcance de sus millas. Las mismas no serán viajes, ni crédito a favor para futuros viajes, las recompensas serán productos y se debe validar que la cantidad en stock del producto elegido permita realizar el canje.

Los productos del programa serán determinados por los alumnos, como así también las millas necesarias para realizar el canje.

Una vez que el stock de un determinado producto se agota no se vuelve a reponer.

El canje solo podrá ser realizado por un administrativo quien es el encargado de validar si el número de documento se corresponde con la persona que desea realizar el canje. Para hacerse efectivo el canje, será necesario que se registre:

- Número de documento
- Producto elegido
- Cantidad del producto elegido
- Fecha de canje.

13. Listado Estadístico

Esta funcionalidad nos debe permitir consultar el TOP 5 de:

- Top 5 de los destinos con más pasajes comprados.
- Top 5 de los destinos con aeronaves más vacías.
- Top 5 de los Clientes con más puntos acumulados a la fecha
- Top 5 de los destinos con pasajes cancelados.
- Top 5 de las aeronaves con mayor cantidad de días fuera de servicio.

Dichas consultas son a nivel semestral, para lo cual la pantalla debe permitirnos selección el semestre a consultar.

El listado se debe ordenar en forma descendente en función de las cantidades según corresponda. Además de ingresar el año a consultar, el sistema nos debe permitir seleccionar que tipo de listado se quiere visualizar.

Cabe aclarar que los campos a visualizar en la tabla del listado para las consultas no son los mismos, y al momento de seleccionar un tipo de listado solo deben visualizar las columnas pertinentes al tipo de listado elegido.

Las columnas del listado para cada una de las consultas quedan a cargo del alumno y dichas columnas deben ser lo suficientemente descriptivas para poder brindar un informe detallado.

Implementación

General

El alumno deberá desarrollar dos componentes: un script de base de datos SQL Server y una aplicación Desktop C#.

A continuación se detalla la implementación de cada componente:

Base de Datos

El alumno debe instalar el motor de base de datos SQL Server 2012 con las siguientes consideraciones:

- El nombre de la instancia del motor de base de datos a instalar debe llamarse “SQLSERVER2012”. No utilizar el nombre “Default” para la instancia. Instalar como instancia con nombre (“Named Instance”).
- La autenticación debe ser por “Modo Mixto”.
- El usuario administrador de la base de datos deberá tener la siguiente configuración:
 - Username: “sa”
 - Password: “gestiondedatos”

Una vez instalado el motor de base de datos se deberán instalar las herramientas cliente de trabajo: “Microsoft SQL Server Management Studio Express” para SQL Server 2012. Ejecutar esta aplicación e ingresar los datos del usuario “sa” creado anteriormente (en modo “Autenticación de SQL Server”).

Dentro del “Management Studio” crear una nueva base de datos con los parámetros default y nombre de base “GD2C2015”

Crear un nuevo “Inicio de Sesión”, desde el item “Seguridad” perteneciente al servidor de Base de Datos general. El inicio de sesión debe poseer las siguientes características:

- Solapa “General”:
 - Nombre de inicio de sesión: “gd”
 - Autenticación de SQL Server
 - Contraseña: “gd2015”
 - Base de Datos Predeterminada: GD2C2015.
 - El resto de los parámetros respetar sus valores default.
- Solapa “Funciones del Servidor”:
 - Seleccionar “sysadmin”
- Solapa “Asignación de Usuarios”:
 - Seleccionar asignar a “GD2C2015”
- Para el resto de los parámetros respetar sus valores default.

Salir del “Management Studio” como usuario “sa” y volver a ingresar con el nuevo usuario “gd” creado. Es probable que informe que la contraseña ha caducado. Cambiar la contraseña ingresando exactamente la misma que antes: “gd2015”.

Una vez que tenemos la base de datos creada y configurada con el usuario, necesitamos ejecutar dos scripts. Para ello debemos ejecutar un comando de consola de SQL Server llamada “sqlcmd”. Este comando debe ejecutar en orden los siguientes dos archivos:

- `gd_esquema.Schema.sql`: Este archivo genera un esquema llamado “gd_esquema” dentro de la base de datos y lo asigna al usuario “gd”.
- `gd_esquema.Maestra.Table.sql`: Este archivo crea la tabla principal del trabajo práctico y la carga con los datos correspondientes. El archivo posee un volumen significativo y no puede ser ejecutado desde el “Management Studio”.

La cátedra provee un archivo BATCH para ejecutar esta operación, denominado “EjecutarScriptTablaMaestra.bat”. Haciendo doble clic sobre el mismo se ejecutan ambos archivos (“gd_esquema.Schema.sql” y “gd_esquema.Maestra.Table.sql”) a través del modo consola. El Script necesita aproximadamente 40 minutos para finalizar su ejecución.

sqlcmd -S <Servidor\Instancia> -U <Nombre_de_usuario> -P <Password> -i <Nombre_del_archivo1>,< Nombre_del_archivo2> -a 32767

Ejemplo:

sqlcmd -S localhost\SQLSERVER2012 -U gd -P gd2012 -i gd_esquema.Schema.sql,gd_esquema.Maestra.Table.sql -a 32767 -o resultado_output.txt

Luego de cargados todos los datos de la tabla maestra, el alumno deberá crear su propio esquema dentro de la base de datos. El nombre del esquema deberá ser igual al nombre del grupo registrado en la materia (el proceso de registración se explica más adelante). El nombre del esquema debe ser en mayúsculas, sin espacios y separado por guiones bajos. Ejemplo “Los mejores” debe ser “LOS_MEJORES”.

Todas las tablas, stored procedures, vistas, triggers y otros objetos de base de datos nuevos que cree el alumno deberán pertenecer a este esquema creado. Sin la solución entregada posee objetos de base de datos por fuera del esquema con el nombre del grupo, el TP será rechazado sin evaluar su funcionalidad.

Con esta configuración el alumno está listo para empezar la implementación de la parte de base de datos.

Aplicación Desktop

La cátedra provee una aplicación Desktop en C#, a modo de template, sobre la cual se debe desarrollar la aplicación del Trabajo Práctico.

Para ejecutar esta aplicación es necesario instalar Visual Studio 2012 con el Framework de .NET 4.5. La versión Express posee la funcionalidad necesaria como para desarrollar el Trabajo Práctico.

La aplicación template se denomina “AerolineaFrba”. Cuenta con un formulario principal y un formulario para cada funcionalidad visual que hay que implementar en el trabajo. El alumno debe depositar su código respetando esta estructura.

Más allá de estas indicaciones, el alumno puede modificar a su criterio la aplicación template. Ante cualquier consulta sobre lo que se puede modificar consultar al grupo de Google de la materia.

La aplicación Desktop deberá conectarse a la base de datos con los siguientes parámetros:

- Origen de datos: Microsoft SQL Server (SqlClient)
- localhost\SQLSERVER2012
- Utilizar autenticación de SQL Server:
 - Nombre de Usuario: gd
 - Password: gd2015
- Nombre de la base de datos: GD2C2015

La aplicación siempre debe conectarse a localhost, pero al menos una vez debe ser probado en una red, dado que la corrección se realizara en red. En caso de que el alumno se conecte a otra dirección, deberá cambiarlo a la hora de entregar su TP para que sea corregido.

La aplicación deberá contar con un usuario de sistema ya creado (además del set que fue pedido en puntos anteriores) y listo para ser utilizado, con las siguientes características:

- Username: admin
- Password: w23e
- Rol:
 - Nombre: Administrador General
 - Funcionalidades: todas las existentes

Este usuario de aplicación debe ser generado en forma automática dentro del archivo “script_creacion_inicial.sql” y quedar listo para ser utilizado por la aplicación Desktop.

Restricciones de la solución

El lenguaje de programación utilizado deberá ser únicamente C# utilizando el Framework .NET 4.5. Cualquier otra implementación que no halla sido desarrollado en éste lenguaje será rechazada, sin excepción.

El entorno de desarrollo debe ser Microsoft Visual Studio 2012 o Microsoft Visual C# Studio Express 2012. No podrá ser utilizada la reciente versión 2010.

El motor de base de datos deberá ser Microsoft SQL Server 2012. Tanto la versión Express como la full sirven para realizar el trabajo. No podrá ser utilizada la reciente versión 2010.

No podrá utilizarse ninguna herramienta auxiliar que ayude a realizar la migración de datos. Tampoco podrá desarrollarse una aplicación personalizada para la migración de datos. La misma deberá ser efectuada en código T-SQL en el archivo de script “script_creacion_inicial.sql”.

Condiciones de aprobación

Testing

El alumno deberá entregar dos componentes:

- Un único script de base de datos (script_creacion_inicial.sql) con todo lo necesario para crear su modelo y cargarlo con datos.
- La aplicación C# “AerolineaFrba” con la funcionalidad pedida.

La cátedra probará el Trabajo Práctico en el siguiente orden:

1. Disponer de una base de datos limpia igual a la original entregada a los alumnos.
2. Ejecutar el archivo script_creacion_inicial.sql. Este archivo debe tener absolutamente todo lo necesario para crear y cargar el modelo de datos. Toda la ejecución debe realizarse en orden y sin ningún tipo de error ni warning.
3. Se ejecuta la aplicación Desktop y se prueban las funcionalidades pedidas.

El archivo “script_creacion_inicial.sql” debe contener todo lo necesario para crear el modelo de datos y cargarlo. Si el alumno utilizó alguna herramienta auxiliar o programa customizado, el mismo no será utilizado por la cátedra.

Si el script de base de datos ejecuta con errores, el trabajo práctico será rechazado sin continuar su evaluación.

Todos los objetos de base de datos nuevos creados por el usuario deben pertenecer a un esquema de base de datos creado con el nombre del grupo. Si esta restricción no se cumple el trabajo práctico será rechazado sin continuar su evaluación.

Modelo de Datos

El modelo de datos creado por el alumno deberá respetar las buenas prácticas de programación y diseño de bases de datos explicados durante la cursada de la materia.

También deberán ser considerados criterios de performance a la hora de crear relaciones e índices en las tablas.

Consultas SQL

Todas las consultas SQL que haga la aplicación serán evaluadas de acuerdo al standard de programación SQL explicados en clase. La performance de las mismas será tomada en cuenta a la hora de fijar la nota.

Respetar Guía de ABMs

Todo el código y las pantallas creadas en la aplicación Desktop deberá respetar a la perfección los lineamientos especificados en el documento “Guía de ABMs”. Aquellos TPs que no respeten las indicaciones en forma total serán rechazados, por más que cumplan la funcionalidad pedida.

Aplicación Desktop

La calidad y orden del código fuente será tomada en cuenta a la hora de fijar la nota. Es obligatorio que existan comentarios de código en todas las secciones principales de implementación.

Deberán crearse componentes de código reusable para aquellas porciones de código ejecutadas en muchas secciones de la aplicación. Todo tipo de configuración o parametrización de la aplicación deberá estar centralizado en un solo punto. Aquellos TPs que no respeten estos puntos mencionados serán rechazados, sin continuar su evaluación (ej: los parámetros de conexión a la base de datos).

Fecha de entrega y condiciones

Existe una sola fecha de entrega posible para el TP, pudiendo presentarse hasta 2 veces más (entregas que no tienen fecha fija).

La cantidad de funcionalidad de cada entrega no varía.

- *Entregas:*
 - *Día:* 12/11/2015 hasta las 12:00 del mediodía (GMT -3:00 Buenos Aires). Los TPs entregados fuera del día y horario anteriormente estipulado se consideran como no entregados en fecha perdiendo dicha instancia de corrección, quedando disponible solo 2 oportunidades de entrega, una primera entrega y 2 instancias de recuperación. Esta primera entrega no necesariamente tiene que constar del TP completo pero para que se corrija mínimamente debe contener el DER de la solución propuesta. Se considera

que la fecha final de entrega es el día 12/12/2015. El periodo de corrección del TP entregado es de 7 días en función de la cantidad de TP entregados en ese momento, factor que deberán tener en cuenta en función de la fecha final para la entrega del trabajo.

Sobre los grupos

Deberán estar compuestos de no más de cuatro integrantes. Cada grupo debe tener un representante que será el único que podrá enviar mails con el TP para su corrección. Los grupos pueden estar compuestos por alumnos de distinto curso. Los alumnos deben registrar su grupo en un sitio de registración especial, especificando un nombre único que identifique al grupo. La URL del sitio de registración es la siguiente:

<https://spreadsheets0.google.com/viewform?formkey=dG16aEltMHc1X2hPN3U2YTVoVGxfeUE6MA>

Al registrarse es necesario especificar un nombre de grupo. El nombre debe ser en mayúsculas, sin espacios y separado por guiones bajos. Ejemplo “Los mejores” debe ser “LOS_MEJORES”.

Luego, el 12/10/2015 se enviarán los mail correspondientes con la confirmación de los grupos y se les asignará un número de grupo además del nombre que debidamente eligieron. Luego de esa fecha, la cátedra enviará al grupo **OFICIAL** la conformación de los mismos que será inalterable hasta la finalización del cuatrimestre. Cualquier cambio de integrantes, sea por el motivo que fuese, deberá realizarse antes de esa fecha. No aceptándose ninguna modificación pasada dicha fecha. Es obligación de los alumnos ingresar al grupo de la cátedra para obtener dicha información.

Ayuda y contacto

El sitio oficial de la materia es el siguiente:

<https://sites.google.com/site/gestiondedatosutn>

También existe un grupo de Google en donde se podrán plantear dudas sobre el TP. Su dirección es la siguiente:

<http://groups.google.com/group/gestiondedatos>

Todos los mensajes referentes al trabajo práctico deberán contener la etiqueta [TP] antes del asunto. Ej: “[TP] consulta sobre base de datos”.

Es obligación del alumno revisar el grupo periódicamente y mantenerse informado sobre actualizaciones, cambios de consignas, modificaciones del programa, cambios de fecha, etc.

La cátedra no asigna ayudantes específicos a cada grupo. Todas las consultas deberán hacerse a través del grupo de Google.

Es obligación de los alumnos ingresar periódicamente al grupo para informarse sobre cuestiones del TP. El grupo OFICIAL es el único medio de comunicación con los alumnos sobre cuestiones del trabajo práctico. La cátedra no se hará responsable si existen grupo paralelos y que no sea el que se detalló en el siguiente enunciado.

Cualquier tipo de información sobre el trabajo práctico que haya sido brindada por los docentes, tendrá que ser validada con los ayudantes.

A lo largo de la cursada pueden ir surgiendo dudas particulares sobre el Trabajo Práctico que sean útiles para el resto de los alumnos. Para ello la cátedra cuenta con un documento denominado “Apéndice del Enunciado” en el cuál se agregan consideraciones generales de manera online. Su dirección es la siguiente:

https://docs.google.com/document/d/1_RZeJNLaiyvSu7YNQ0k_UAGtaGcEsDzCYK4s1J_ZMrt4/edit?usp=drive_web

Inicialmente el Apéndice se encuentra vacío. A medida que vayan surgiendo dudas sobre el desarrollo del Trabajo Práctico, la cátedra evaluará agregar consideraciones generales al documento. Es obligación del alumno revisar este documento periódicamente.

Donde aprender C#

Si bien para resolver el TP solo se necesita conocer una pequeña parte de la totalidad del lenguaje C#, es recomendable aprender los conceptos básicos mediante algún libro o tutorial. Recomendamos el siguiente tutorial:

<http://www.devjoker.com/asp/~/gru/Tutorial-C/TUCS/Tutorial-C.aspx>

En cuanto a las colecciones que posee .Net, recomendamos la siguiente documentación:

Documentación de MSDN en español

<http://msdn2.microsoft.com/es-es/library/7y3x785f%28VS.80%29.aspx>

Sobre la elección de C#

El lenguaje C, tradicionalmente usado en las cátedras de la facultad, ha demostrado tener cierta dificultad a lo hora de su uso. El uso de punteros y procesamiento de cadenas muchas veces resulta complicado, sin mencionar la dificultad de encontrar un error en tiempo de ejecución. También hemos notado que se invierte mucho tiempo tratando de crear interfaces amigables mediante consola de texto.

Es por esto que creemos que C# al igual que otros lenguajes de última generación, como Java, permiten invertir más tiempo en cuestiones algorítmicas y de estructura de datos, dejando de lado las cuestiones de sintaxis propias del lenguaje C.

Por nombrar algunas ventajas de C#:

- Facilidad en la depuración en tiempo de ejecución: Nos permite inspeccionar el valor de las variables durante la ejecución del programa, incluso visualizar estructuras de datos recursivas.
- Las sintaxis está totalmente normalizada.
- El IDE permite autocompletar código.
- Provee métodos simples para el manejo de E/S.

Podemos decir que gran parte de la eficiencia de un programa depende no del lenguaje en el cual es implementado, sino de las estructuras de datos y algoritmos elegidos para resolverlo.

Por último consideramos que el paradigma orientado a objetos puede brindarnos muchas ventajas que a esta altura ya resultan evidentes y al mismo tiempo permite aplicar los mismos esquemas algorítmicos que los lenguajes estructurados.

Obtención de herramientas

El TP puede ser desarrollado con dos versiones del IDE Microsoft Visual Studio 2012. No puede utilizarse la versión 2010:

1. **Microsoft Visual Studio Professional 2012:** esta versión puede ser obtenida con licencia universitaria completa, gracias a un convenio de Microsoft con la UTN. Dirigirse al laboratorio de Microsoft ubicado en la sede Medrano (planta baja, hacia la derecha, mirando desde la entrada de la facultad hacia adentro). La versión ofrecida es en español y contiene la ayuda completa (MSDN). Para encargarla es necesario llevar un DVD y completar un formulario. Es posible que el programa halla que encargarlo y pasarlo a buscar otro día.
2. **Microsoft Visual C# 2012 Express Edition:** Existe una versión gratuita del IDE llamada *Visual C# 2012 Express Edition*, la cual posee todas las herramientas necesarias para realizar el TP. Esta se encuentra disponible en:

<http://www.microsoft.com/express/Downloads/>

El motor de base de datos a utilizar es SQL Server 2012 Express. Puede ser descargado de la siguiente dirección:

<http://www.microsoft.com/Sqlserver/2012/en/us/express-down.aspx>

Es necesario descargar e instalar dos componentes:

- Install Microsoft SQL Server 2012 Express Edition
- SQL Server Management Studio Express

Formato de entrega

Lugar de envío

La entrega debe realizarse por mail el antes de las fechas estipuladas en el documento de enunciado

La dirección del mail es:

gestiondedatos.entregas@gmail.com

El asunto del mail debe cumplir con el siguiente formato:

TP2C2015<curso><nombreGrupo><Nro de grupo>

: espacio en blanco

Ejemplos:

TP1C2015 k9999 LOS_MEJORES 10

(Respetar los 2 espacios en blanco existentes)

Se debe adjuntar el trabajo práctico en un archivo del tipo zip con el mismo nombre que el asunto del mail.

Por cuestiones de seguridad Gmail rechaza todos los adjuntos que contengan archivos zip con .exe y .dll en su interior, por lo que es necesario limpiar la solución como se detalla más adelante y el resultante archivo zip sin extensión.

Por ejemplo:

TP2C2015 k9999 LOS_MEJORES 10 (Respetar los 2 espacios en blanco existentes)

No enviar adjuntos de más de 20 MB. La casilla de mail rechazará mails que superen esta restricción.

El cuerpo del mail debe contener lo siguiente:

Grupo:

Curso:

Integrantes: <apellido>, <nombres> - <legajo>

Nota: En caso de que haya integrantes de cursos distintos, se debe poner el curso de la persona elegida como representante

En caso de que algún alumno del grupo haya dejado de cursar o se haya cambiado de grupo, deberá ser aclarado en el mail de la entrega del TP.

Solo debe enviarse la entrega desde el mail del representante del grupo.

Los alumnos deberán registrar su grupo en la dirección mencionada anteriormente. No se aceptarán TPs que no estén registrados.

Estructura del archivo zip

El archivo zip (sin extensión) debe contener la siguiente estructura de directorios:

```
⇒ \  
⇒ Readme.txt  
⇒ Estrategia.pdf  
⇒ \src  
    ⇒ Solución entera de Visual Studio de ""  
⇒ \data  
    ⇒ Archivo de script de base de datos "script_creación_inicial.sql".
```

Readme.txt:

Es un archivo de texto plano con los siguientes datos:

- Curso
- Número de grupo
- Nombre y legajo de todos los integrantes
- Email del integrante responsable del grupo.

Estrategia.pdf:

Archivo PDF en donde se deberá explicar en forma detallada y extensa la estrategia utilizada para desarrollar el TP. Debe incluir una descripción de todas las estructuras de datos relevantes utilizadas en el algoritmo, explicando la razón de la elección de dichas estructuras. Cualquier consideración tomada o asumida deberá ser aclarada en este documento.

Se debe incluir un DER (legible y entendible) del modelo de datos creado con una explicación detallada de cada entidad, relaciones, claves primarias y foráneas, índices, stored procedures, triggers, vistas, etc, que será estrictamente necesario para la corrección del modelo, de ser posible también entregar el DER en formato de imagen PNG, JPEG, etc.

El archivo de estrategia deberá ser en formato PDF obligatoriamente, con carátula e índice. En caso de no cumplir esta condición, el TP será rechazado sin evaluar su funcionalidad.

Sin este archivo y un DER de la base la entrega no será tomada como válida.

\src:

Dentro de este directorio se encuentra la solución entera de Visual Studio del proyecto "". Evitar enviar archivos de SVN, CVS u otros. No enviar archivos ejecutables ni DLLs. Limpiar la solución desde Visual Studio (Proyecto => Limpiar Solución) antes de enviarla.

Además dentro del source deberá existir un archivo de configuración en donde se encuentren los parámetros de conexión a la base de datos que de ser **UNICA** para todo el TP, (si existiese más de un punto de conexión a la base de datos el TP no será corregido), **la fecha que tomará el sistema para funcionar** (se utiliza este criterio para simplificar al alumno el uso de fechas y así evitar el cambio de fechas del sistema operativo).

Sin ese archivo de configuración la entrega no será tomada como válida.

\data:

Archivo "script_creación_inicial.sql" con toda la creación del modelo de datos. El archivo debe poder ejecutar perfectamente de una sola vez, sin ningún tipo de error. Todas las sentencias deben estar perfectamente ordenadas para ejecutar correctamente. Cada sentencia debe estar comentada explicando su intención.

10 Tips para desaprobar Gestión de Datos

- ```

/***** Script for SelectTopNRows command from SSMS *****/

insert into [GAGA].[PUBLICACION] (
 DESCRIPCION, STOCK, FECHA_INICIO, FECHA_FIN, PRECIO, ID_VISIBILIDAD,
 ESTADO, TIPO, PUBLICADO_POR, PREGUNTAS, VISIBILIDAD_FACTURADA)
values (
 'cosa', 10, convert(datetime,'2015/01/01 00:00:00',121),
 convert(datetime,'08/01/2015 12:00:00',121), 100, 10006, 1, 1, 2, 1,0)

insert into [GAGA].[PUBLICACION] (
 DESCRIPCION, [Invalid object name 'gaga.PUBLICACION'], FECHA_FIN, PRECIO, ID_VISIBILIDAD,
 ESTADO, TIPO, PUBLICADO_POR, PREGUNTAS, VISIBILIDAD_FACTURADA)
values ('cosa', 10, '2015/01/01 00:00:00', '08/01/2015 12:00:00 a.m.', 100, 10006, 1, 1, 2, 1,0)

```
- Messages
- (1 row(s) affected)
- Msg 241, Level 16, State 1, Line 10  
 Error al convertir una cadena de caracteres en fecha y/u hora.

- 23