



Universidad Tecnológica Nacional
Facultad Regional Buenos Aires

Gestión de Datos

Trabajo Práctico

1° Cuatrimestre 2019

FRBA - Crucero

Enunciado V1.0



Índice

Índice.....	2
Introducción.....	3
Objetivos generales.....	3
Descripción general	3
Componentes del TP.....	3
Base de Datos y Modelo de Datos	3
Aplicación Desktop.....	4
Requerimientos.....	4
General	4
Base de Datos	4
Aplicación Desktop.....	5
General	12
Base de Datos	12
Aplicación Desktop.....	13
Restricciones de la solución	15
Condiciones de aprobación	15
Testing.....	15
Modelo de Datos.....	15
Consultas SQL.....	16
Respetar Guía de ABMs.....	16
Aplicación Desktop.....	16
Fecha de entrega y condiciones.....	16
Sobre los grupos	16
Ayuda y contacto	18
Donde aprender C#.....	19
Sobre la elección de C#.....	19
Obtención de herramientas.....	20
Formato de entrega	20
Lugar de envío	20
Estructura del archivo zip.....	21
Readme.txt:.....	22
Estrategia.pdf:.....	22

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 un nuevo sistema que permita la comercialización de tickets de cruceros por medio de terminales kiosco.

Una consideración a tener en cuenta es que cuando un crucero parte de origen si o si llega a su puerto de destino. (Como la información disponible en la tabla maestra corresponde a una simulación, puede darse el caso que algunas ciudades no dispongan de un “puerto” ej Bogotá)

La implementación de dicho sistema, requiere previamente realizar la migración de los datos que se tenían hasta el momento, para ello que es necesario que se reformulen los procesos y el diseño de la base de datos para que cumpla con las nuevas restricciones.

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 pasajes.

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 Puerto
5. ABM de Recorrido
6. ABM de Crucero
7. Generar Viaje.
8. Compra y/o reserva de viaje
9. Pago Reserva.
10. 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 solo 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, quedando dicho rol inhabilitado. 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.

Se debe pensar en la escalabilidad del sistema, de manera que en futuro se puedan agregar más roles y su asociación a las diversas funcionalidades que le correspondieran.

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. Los usuarios del tipo cliente acceden directamente al sistema sin pasar por el login.

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 extras del sistema.

Bajo ningún concepto un usuario cliente se puede visualizar las funcionalidades propias de un administrativo, aunque las mismas se encuentren deshabilitadas.

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. Al realizar un Login satisfactorio, el sistema deberá limpiar la cantidad de intentos fallidos.

El Login se considera una funcionalidad de características especiales. No se considera una funcionalidad que pueda 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 (más de uno) con perfil Administrador. NO codificar una solución no los habilita a que dejen tener en cuenta todas aquellas entidades dependientes de dicha funcionalidad, en síntesis, no codificar no es sinónimo de no modelar/diseñar.

Los alumnos determinaran cual será el username para cada uno de los usuarios entregados (que deberán ser detallados en la estrategia del TP) y asignarán **w23e** como password de usuario.

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 Puertos

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

En caso de querer dar de baja un puerto, será necesario que se chequee si existen recorridos ya programados para ese puerto. Si el recorrido tiene pasajes vendidos, se tendrá que realizar la devolución correspondiente a cada uno de los clientes.

Para reducir el tiempo de elaboración del TP, la cátedra determina que no será necesario que codifiquen dicha funcionalidad. Aplica el mismo criterio del punto 3.

5. ABM de Recorrido

Esta funcionalidad es la que permite a un administrativo crear, modificar, o dar de baja las diferentes rutas que pueden realizar los cruceros habilitados.

Un recorrido se caracteriza por conectar mínimamente 2 ciudades, nunca una sola porque el barco debe salir de puerto siempre. Un recorrido puede ir a la ciudad A a la ciudad B, luego de la ciudad B a la ciudad C y finalizar allí el mismo, ó retornar de la ciudad C a la ciudad A. No es necesario que todas las rutas retornen a la ciudad de origen.

Al tratarse de una modificación, pueden modificarse todas ciudades que lo componen, hasta incluso no pudiendo retornar a origen si es que inicialmente lo hacía, con respecto a los pasajes vendidos, no se podrá dar de baja un recorrido que tenga pasajes vendidos sin haber realizado el viaje.

En caso de que se necesite dar de baja un determinado recorrido, el mismo será en forma lógica y no se cancelarán los pasajes vendidos. En una segunda etapa se analizará que ocurrirá con dichos pasajes. Si debe tenerse en cuenta que si un recorrido es dado de baja, el mismo no debe ofrecerse más a la venta.

Mínimamente un recorrido debe registrar los siguientes campos de manera obligatoria:

- Código de recorrido (alfanumérico)
- Ciudad de Inicio (selección acotada)
- Ciudad de Destino (selección acotada)
- Precio del tramo

6. ABM de Cruceros.

Esta funcionalidad permite a un administrativo incorporar, modificar o dar de baja un crucero. Un crucero se caracteriza por tener una cantidad X de cabinas, las cuales tienen su tipo de servicio correspondiente.

Hay que tener en cuenta que el costo de un pasaje se calcula de la siguiente manera: un recorrido puede tener uno o más tramos, cada tramo tiene un costo, en función de la sumatoria de dichos tramos, se le aplica un porcentaje adicional en función del tipo de cabina que se haya elegido.

Otra cuestión a tener en cuenta es que un crucero pertenece a una empresa/marca determinada, se debe permitir modificar dicho valor de la manera más óptima posible.

Se debe prever la posibilidad de dar de baja un crucero (mediante baja lógica), en la cual dicho crucero no volverá a ser utilizado en el parque de servicios de pasajeros, dado que **se completó su vida útil**. También se tendrá que tener en cuenta otro tipo de “baja” que es la que hace referencia a la no utilización de un crucero ya que el mismo 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 el crucero volverá a estar disponible para su utilización.

Para la implementación de estas 2 situaciones anteriormente descritas se debe poder cancelar los pasajes vendidos, dejando un registro de dicha situación y su motivo. En caso de no querer cancelar los pasajes, si la inactividad es por fin de la vida útil se reemplazan todos los viajes futuros por otro crucero que pueda cumplir con el mismo recorrido que hacia el crucero reemplazado. Además, el crucero reemplazante debe disponer de fechas libres para cumplir con el recorrido en cuestión en tiempo y forma. Si la inactividad es por fuera de servicio, se reprogramarán de manera automática la fecha de TODOS los viajes implicados, para ello el administrador deberá ingresar la cantidad de días de corrimiento para todos los viajes.

Si se llegase a dar el caso que ningún crucero pueda reemplazar a otro, será necesario que se de alta un nuevo crucero. 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 de ventas, indicándole y dándole la directiva al administrativo de que es lo que debe hacer, si reemplazar el crucero o cancelar todos los pasajes ya comprados. 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.
- Crucero
- Modelo
- Identificador
- Marca (selección acotada)
- Tipo de servicio (selección acotada)
- 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 cabinas

- Determinación del tipo de cabina

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 un crucero en particular, en una fecha de inicio y otra de finalización. Se debe validar que la fecha de inicio sea mayor a la fecha actual.

Una vez generado esto, está disponible la venta de pasajes el nuevo recorrido confeccionado.

Se debe tener en cuenta que al momento de seleccionar un crucero este deberá estar disponible y no estar asignado previamente a otro viaje en la fecha que se está ingresando.

A fines prácticos, para simplificar el desarrollo, solo se tendrá en cuenta la fecha de inicio y la fecha de finalización del viaje, no se tendrá en cuenta o no se planificará en qué fecha atraca un crucero en un puerto determinado.

Datos a Registrar mínimamente en forma obligatoria:

- Fecha inicio del viaje (con horas, minutos y segundos).
- Fecha finalización del viaje (con horas, minutos y segundos).
- Crucero.
- Recorrido.

8. Compra y/o reserva de viaje

Esta funcionalidad permite a un cliente/administrativo realizar la compra y/o reserva de un viaje.

Una compra implica el pago total de un viaje, una reserva es un compromiso y pago en diferido (en otro momento) de un viaje.

Para realizar la compra y/o reserva del pasaje, se deberá ingresar la fecha de viaje junto con el puerto de origen y puerto destino. El sistema informará si hay viajes disponibles para esa fecha, junto con las cabinas (y sus tipos) disponibles del crucero en cuestión.

Luego de ello será necesario que se ingrese la cantidad pasajes a comprar.

Solo pueden venderse viajes que tengan cruceros asignados habilitados.

El cliente deberá pagar por la totalidad del recorrido que haga el crucero, si el crucero visita las ciudades A, B, C, D y retorno a A y el cliente quiere bajarse en la ciudad C, puede hacerlo pero deberá pagar por el total del costo del viaje, no existiendo la posibilidad de fraccionamiento de costos. Si el pasajero descende en X ciudad, la cabina que haya tomado al inicio seguirá “ocupada” hasta la finalización del viaje. Como así también, tampoco está permitido el ascenso de pasajeros en ciudades intermedias.

Luego de seleccionar un determinado viaje, se procederá a ingresar los datos propios del cliente comprador SOLAMENTE.

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 compañía FRBA CRUCERO, al momento de ingresar su número de documento, se cargará automáticamente el resto de los datos personales. Se debe permitir que luego de dicha carga, los datos puedan ser actualizados. Si nunca viajó se cargaran uno por uno desde el inicio.

Una vez cargados los datos conforme al viaje el sistema pedirá que se ingrese los datos para hacer efectiva la compra. Se deben poder determinar 2 o más medios de pagos, junto con el ingreso correspondiente de datos que sean necesarios para el registro del mismo. Dependiendo del tipo de tarjeta se podrá realizar la compra en cuotas. Finalizada la compra se informará al cliente el número de voucher de la compra juntos con TODOS los datos del viaje, pasajeros, días, destinos, crucero, etc.

Si la operación a realizar es una reserva, la operatoria es similar a la de una compra, con la diferencia que no hay pago en la misma transacción y de dicha reserva es pagada en diferido, se paga en los días posteriores. Una reserva tiene una validez de 3 días, al 4to día la reserva deja de tener efecto y la cabina vuelve a estar disponible para su venta o reserva.

El proceso de reserva finaliza informando al cliente dicho número de reserva (similar al voucher de compra, nada más que este es de reserva) para que luego pueda ingresarlo en la funcionalidad correspondiente para el pago.

Una validación a tener en cuenta es que un pasajero no puede viajar a más de un destino a la vez y tampoco realizar compras sobre viajes pasados.

9. Pago de reserva

Esta funcionalidad permite a un cliente/administrativo realizar el pago de una reserva de un viaje. Para ello se deberá ingresar el código de reserva brindado al momento de confeccionar la misma. Luego de dicho ingreso, se listará en pantalla todos los datos de la misma y se procederá al cobro de dicha reserva.

Tener en cuenta que para el pago corre las mismas condiciones de pago descriptas en el punto 8.

Si la reserva tiene más de 3 días de confección, la misma no puede ser pagada porque tuvo que haberse dado de baja al momento de realizar el login para los administradores, o al momento de ingresar a la funcionalidad de pago, si es un usuario de tipo cliente. (Se recuerda que para dar de baja las reservas que estuviesen vencidas, se chequearán las fechas de todas las reservas, se comparará con la fecha actual, y si la diferencia resultante es de 4 o más días, dicha reserva debe cancelarse. Tener en cuenta que esto es una simulación, en un caso real, ese proceso debería ejecutar en un momento X del día, como nosotros resolvemos esto académicamente, la única forma que tenemos para simplificar esto es según lo detallamos anteriormente.)

10. Listado Estadístico

Esta funcionalidad nos debe permitir consultar el TOP 5 de:

- Top 5 de los recorridos con más pasajes comprados.
- Top 5 de los recorridos con más cabinas libres en cada uno de los viajes realizados.
- Top 5 de los cruceros con mayor cantidad de días fuera de servicio.

Dichas consultas son a nivel semestral, para lo cual la pantalla debe permitirnos selección el semestral 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 solo deben visualizarse 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 “GD1C2019”.

Crear un nuevo “Inicio de Sesión”, desde el ítem “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: “gdCruceros2019”
 - Autenticación de SQL Server
 - Contraseña: “gd2019”
 - Base de Datos Predeterminada: GD1C2019.
 - El resto de los parámetros respetar sus valores default.
- Solapa “Funciones del Servidor”:
 - Seleccionar “sysadmin”
- Solapa “Asignación de Usuarios”:
 - Seleccionar asignar a “GD1C2019”
- 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: “gd2019”.

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 “FRBACrucero”. Cuenta con un formulario principal, una barra de menú 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: gdCruceros2019
 - Password: gd2019
- Nombre de la base de datos: GD1C2019

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 otras versiones

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 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# “FrbaCrucero” 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

Para cada entrega existe una sola fecha de entrega posible como límite. Para el caso de la entrega del sistema completo, se agregan hasta 2 presentaciones más para reentregar con correcciones (entregas que no tienen fecha fija).

La cantidad de funcionalidad de cada entrega deberá ser completa acorde a lo solicitado en la instancia.

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 02/05/2019 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.

Entrega obligatoria del DER:

En esta primera entrega deberá enviarse solamente el DER del sistema en un archivo formato imagen, preferentemente JPG, el cual debe estar realizado con una herramienta acorde y ser netamente legible, con todas sus relaciones y campos que componen la entidad. No se aceptarán imágenes de DER realizado a mano, en lápiz, birome, etc.

Solo se aceptarán entregas dentro de la fecha límite específicamente estipulada (14/05/2019) y pasada esa fecha no se aceptarán entregas bajo ninguna circunstancia. No habrá excepciones por entrega fuera de término ni motivos que lo justifiquen, ya que la entrega del TP es grupal y esto da la posibilidad de que la responsabilidad sea de todos los integrantes del grupo para llegar en fecha.

En caso de que el DER no sea correcto, los errores serán informados en la corrección y deberán ser resueltos para la entrega del sistema completo. Esto quiere decir que no hay reentrega del DER y la evaluación final será en base al sistema completo.

La motivación de esta entrega es la corrección de errores en el modelado de la base de datos antes de la construcción del sistema final, para evitar que en la siguiente entrega se encuentren con la necesidad de modificar gran cantidad de código por errores del modelado de la base de datos que pueden ser atacados previamente.

Si el contenido de la entrega está incompleto, aumentarán las posibilidades de que los agregados en el DER en la próxima entrega lleven a un gran retrabajo si es necesario reentregar.

Entrega del sistema completo

Día: 14/06/2019 hasta las 12:00hs del mediodía (GMT 3:00 Buenos Aires).

Los TPs entregados luego de las 12:00hs, se considerarán fuera de término perdiendo así una posibilidad de entrega y restandole solamente 2 instancias de presentación. Estas 2 instancias de reentrega no tienen fecha asignada y serán determinadas por el equipo para entregar cuando gusten, bajo responsabilidad de los alumnos.

Una vez entregado el TP, el periodo de corrección es aproximadamente de 10 días. Este factor puede variar dependiendo de la cantidad de TPs entregados en ese momento. Por lo cual, se recomienda tenerlo en cuenta para la fecha final de entrega del trabajo.

Si llegan a realizar una sola entrega del TP, cercana a la última fecha (menor a 10 días), es netamente responsabilidad del grupo, ya que se les da la posibilidad de tener

más de una entrega y solo hacen uso de una instancia (no hay posibilidad de reentrega), sea por el motivo que fuese., en caso de llegar a esa situación, solo contarán con una sola entrega habiendo perdido la chances anteriores de entrega.

Cualquier indicio de copia será penado con la pérdida de la materia, aun así tengan los parciales aprobado, se supone que el tp tiene carácter de parcial y es un producción propia del grupo.

La última fecha para recepción de TP es el día 14/07/2019

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 ya sea el Ing. López Matias Miguel o el Ing Ariosti Maximiliano.

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/1kkLxEQSoUCxPBfJw12d8J0P0wTV2KRHLX66Ksmgjh6U/edit?usp=sharing>

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:

TP1C2019<curso><nombreGrupo><Nro de grupo>

: espacio en blanco

Ejemplos:

TP1C2010 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 renombrar la extensión .zip a .zip123.

Por ejemplo:

TP1C2019 k9999 LOS_MEJORES 10.zip123
(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 halla 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 (.zip123) debe contener la siguiente estructura de directorios:

⇒ \
⇒ Readme.txt
⇒ Estrategia.pdf
⇒ \src
⇒ Solución entera de Visual Studio de “FrbaBusDesktop”
⇒ \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 “FrbaCruceros”. 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, **la fecha que tomará el sistema para funcionar** (se utiliza este criterio para simplificar al alumno el manejo de las 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.

Cualquier TP entregado que no cumpla con alguno de los requisitos mencionados en este documento, será rechazado sin ser evaluado, perdiendo una oportunidad de entrega.

10 Tips para desaprobar Gestión de Datos

1. No inscribirse en el grupo, si no tengo número de grupo, preguntar a otros alumnos como se enteraron y repetir el procesimiento.
2. No entregar el DER del modelado, y en caso de entregarlo, que el mismo no sea legible
3. No consultar periódicamente el grupo de la materia
4. Entregar el script de migración y/o solución de C# con errores, y que los mismos no puedan ser ejecutados
5. Entregar TP de años anteriores
6. Entregar ABMs sin validaciones propias de un AMB, por ej. permitir alta de registros duplicados.
7. Hacer mal uso de los formatos de fecha. (se recomienda uso de convert) y/o no usar el archivo config que se les pide. Ejemplo de que no hacer y que si hacer.

```

/***** 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 )

```

8. Entregar el TP faltando menos de 5 minutos para la fecha limite saturando la casilla de entrega o pasada la fecha límite para la recepción para su corrección. El tp es grupal y no individual, un integrante puede tener problemas, pero los 4 al mismo tiempo no.
9. Realizar una única entrega en una fecha cercana a la fecha límite sin la posibilidad de un feedback y si aun así deciden realizar una única entrega, deberán asumir el riesgo de Aprobación o Desaprobación
10. Si hay dudas, volver a leer los puntos anteriores o consultarlos con los ayudantes.