



Universidad Tecnológica Nacional
Facultad Regional Buenos Aires

Gestión de Datos

Trabajo Práctico

2º Cuatrimestre 2012

Cuponete utn

Enunciado V1.1



Índice

Índice	1
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.....	14
Base de Datos	14
Aplicación Desktop	15
Restricciones de la solución	17
Condiciones de aprobación.....	17
Testing	17
Modelo de Datos.....	18
Consultas SQL.....	18
Respetar Guía de ABMs	18
Aplicación Desktop	18
Fecha de entrega y condiciones	18
Sobre los grupos	19
Ayuda y contacto	19
Donde aprender C#.....	20
Sobre la elección de C#	20
Obtención de herramientas	21
Formato de entrega	21
Lugar de envío	21
Estructura del archivo zip	22
Readme.txt:.....	23
Estrategia.pdf:.....	23
\src:	23
\data:	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 migración y remoción de un viejo sistema de cupones y ofertas que ha quedado obsoleto.

Es para ello que 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.

En general, se basa en el dominio de los sitios de compra de cupones mas conocidos como Groupon, Pez urbano, Lets Bonus, Groupalia, Coponica, Big Deal, etc.

Dado el ambiente académico, se ha simplificado el domino para que el mismo se adecue a los tiempo de cursada.

Es por ello que la implementación será bajo una aplicación escritorio y no un entorno WEB.

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 2008. 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 y publicación de cupones

El sistema a desarrollar será utilizado por 3 tipos de usuarios distintos: administradores, proveedores 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 2008 y Framework de .NET 3.5.

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 2008 con Framework .NET versión 3.5. 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.

Las funcionalidades existentes en el sistema son todas las que el TP exige desarrollar en la aplicación Desktop. El listado completo es el siguiente:

1. Login y seguridad
2. ABM de Rol
3. Registro de Usuario
4. ABM de Cliente
5. ABM de Proveedor
6. Cargar Crédito
7. Comprar GiftCard
8. Comprar Cupón
9. Pedir Devolución
10. Historial de Compra de Cupones
11. Armar Cupón
12. Registro de consumo de Cupón.
13. Publicar Cupón
14. Facturación a Proveedor.
15. 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 3 roles, Proveedor, Administrativo, 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 una las funcionalidades como así también agregar nuevas funcionalidades a rol que se está modificando.

La eliminación del rol implica una baja lógica del mismo. El rol debe poder inhabilitarse. No permitido 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

Al ejecutar la aplicación el usuario no podrá acceder a ninguna funcionalidad del sistema hasta completar el proceso de Login.

El proceso de Login pedirá al usuario su Username y su Password. Si Login es correcto, el usuario podrá acceder al sistema. Al acceder al mismo, la aplicación solo deberá generar y mostrar las entradas de menú disponibles para este usuario, según los roles del mismo. El usuario no debe ni siquiera ver las funcionalidades a las que no posee acceso.

Si el Login es incorrecto el usuario no podrá acceder al sistema. 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 puede ser asignada a un rol. Todos los usuarios tienen la capacidad de utilizar el Login.

3. Registro de Usuario.

Funcionalidad que se encuentra disponible al momento de loguearse el usuario al sistema. Si es la primera vez que ingresa al mismo, el alta deberá realizarse por medio de esta opción (solo para usuarios del tipo Cliente o Proveedor)

Se deberá tener en cuenta que un cliente o proveedor no puede tener más de un usuario y una vez que un cliente o proveedor se registró en el sistema, no puede cambiar su tipo de usuario.

Si el usuario que se registra por primera vez es un cliente, este deberá elegir cuales son sus preferencias, con esto nos referimos a las ciudades a las cuales se suscribe para ver ofertas.

Crear un Usuario implica cargar los siguientes datos:

- Username
- Password
- Rol asignado (selección especial)
- Datos identificatorios según el tipo de usuario (cliente o proveedor). Ver en el ABM correspondiente al tipo

Todos los datos mencionados anteriormente son obligatorios.

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.

A un usuario se le puede asignar un solo rol según corresponda a su tipo. Esto no implica que el usuario administrativo pueda modificar el rol de un usuario ya creado, para dicho caso se descartarán los datos identificatorios guardados con anterioridad (exceptuando el username y password), debiéndose ingresar nuevamente dichos datos en función del nuevo tipo de rol asignado.

Debe tenerse en cuenta que se pueda modificar el password. Tanto por el propio usuario como por el usuario administrativo.

También debe contemplarse de alguna manera, que un administrativo pueda dar de baja un usuario como así también el propio usuario. Esto es independiente a la inhabilitación por fallas de ingreso. **IMPORTANTE:** Toda baja debe realizarse en forma lógica.

4. ABM de Clientes

Funcionalidad que permite a un administrativo crear, modificar y dar de baja un cliente del sistema. Estos clientes son aquellos que realizan compras de cupones por medio de la aplicación.

Dar de alta un cliente implica el ingreso de los siguientes datos:

- Nombre.
- Apellido.
- DNI.
- Mail.

- Teléfono.
- Dirección calle, nro piso, depto y localidad
- Código Postal
- Fecha de Nacimiento.
- Ciudades de preferencias para la visualización de cupones disponibles.

Todos los datos mencionados anteriormente son modificables. El teléfono es un dato único, por ende no pueden existir 2 clientes con el mismo teléfono. El sistema deberá controlar esta restricción e informar debidamente al usuario ante alguna anomalía.

El alumno deberá determinar un procedimiento para evitar la generación de clientes “gemelos” (distinto nombre de usuario, pero igual datos identificatorios según se justifique en la estrategia de resolución).

Toda creación de cliente nuevo, implica una carga de dinero de bienvenida de \$10.

La eliminación de un cliente implica la baja lógica del mismo. Un cliente inhabilitado no podrá comprar cupones ni cargarse crédito bajo ninguna forma. Se debe poder volver a habilitar el cliente deshabilitado desde la sección de modificación.

Para elegir que cliente se desea modificar o eliminar se debe presentar un buscador con listado, que permita filtrar simultáneamente por alguno o todos los siguientes campos:

- Nombre (texto libre)
- Apellido (texto libre)
- DNI (texto libre exacto)
- Email (texto libre)

5. ABM de Proveedor

Funcionalidad que permite a un administrativo crear, modificar y dar de baja un proveedor del sistema. Estos proveedores son aquellos que arman cupones para ser ofrecidos a los clientes por medio de la aplicación de venta de cupones y que son subidos a la red por un usuario administrativo.

Dar de alta un proveedor implica el ingreso de los siguientes datos:

- Razón Social.
- Mail.
- Teléfono.
- Dirección calle, nro piso, depto y localidad
- Código Postal
- Ciudad
- CIUT
- Rubro en el cual se desempeña
- Nombre de Contacto

Todos los datos mencionados anteriormente son modificables. La razón social y cuit son datos únicos, por ende no pueden existir 2 proveedores con la misma razón

social y cuit. El sistema deberá controlar esta restricción e informar debidamente al usuario ante alguna anomalía.

La eliminación de un proveedor implica la baja lógica del mismo. Un proveedor inhabilitado no podrá armar ofertas ni registrar el consumo de un cupón entregado por un cliente determinado. Se debe poder volver a habilitar a un proveedor deshabilitado desde la sección de modificación.

Para elegir que proveedor se desea modificar o eliminar se debe presentar un buscador con listado, que permita filtrar simultáneamente por alguno o todos los siguientes campos:

- Razón Social (texto libre)
- CUIT (texto libre exacto)
- Email (texto libre)

6. Carga de crédito.

Esta funcionalidad permite la carga de crédito a la cuenta de un cliente para poder operar en este nuevo sistema en la compra de cupones. Si el cliente no posee crédito en la cuenta no podrá realizar ningún tipo de operación.

Al momento de efectuarse la carga de dinero, el sistema tomará la fecha de día. **Dicha fecha no será extraída automáticamente del sistema.** Sino que la misma será tomada del archivo de configuración de la aplicación.

El monto a cargar, será un importe fijo (entero) y mayor a \$15.

Una vez que se determina el monto a cargar, será necesario que se elija el tipo de pago. En caso de tratarse de un pago con tarjeta de crédito o débito, será obligatorio que se registren los datos necesarios para poder identificar la tarjeta en cuestión de manera completa. No será necesario validar si dicha tarjeta posee dinero disponible o si la misma se encuentra habilitada. Siempre que se ingrese una tarjeta se procesará el pago.

Para dar de alta una nueva carga implicará registrar los siguientes datos:

- Fecha
- Cliente
- Tipo de pago (selección acotada)
- Monto
- Datos de la tarjeta (a determinar por los alumnos)

7. Comprar GiftCard

Esta funcionalidad permite a un determinado usuario (cliente) comprar una suma determinada de dinero y asignársela a otro cliente.

Cabe destacar que la suma de dicha GiftCard será un monto fijo (entero), dentro de un rango acotado y previamente determinado. Se debe tener en cuenta que dichas cantidades puedan variar en función del tiempo, pero para ello no será necesario un abm.

Al momento de registrar la compra de un GiftCard, se tomará la fecha indicada en el archivo de configuración y no la fecha actual que tenga el sistema.

Cuando la compra de dicha Gift se registra, se actualizará automáticamente el saldo del usuario que recibe el regalo. Se debe tener en cuenta que el usuario

destinatario del gift debe ser existir y estar habilitado. No está permitido que usuario comprador sea el receptor del regalo.

Datos a Registrar obligatoriamente:

- Fecha
- Monto (selección Acotada)
- Cliente Origen
- Cliente Destino (selección por búsqueda)

8. Comprar Cupón

Esta funcionalidad permite a un cliente comprar un cupón de descuento ofrecido por los diferentes proveedores. Los cupones se encuentran distribuidos por zona, por ende, en función de la preferencia del usuario, solo se podrán ver los cupones correspondientes a las zonas de interés del usuario. Además de dicho filtro, se debe tener en cuenta que el usuario solo podrá ver los cupones habilitados para el día de hoy, la cual se tomará del archivo de configuración y no la que brinda el sistema. No se podrá tener acceso a cupones pasados o futuros.

Cuando un cliente compra un cupón no sabe cuanto de ellos hay en stock, pero el sistema deberá informar al usuario de la falta de stock de los cupones.

Al momento de realizar la compra el sistema deberá validarse que el saldo del comprar sea superior al precio del cupón que se intenta compra

Cuando un cliente adquiere un cupón, se le deberá informar el código de dicho cupón y en la compra se deberá validar que la adquisición no supere la cantidad máxima de cupones (de ese mismo tipo) permitida por usuario.

Los datos mínimos a registrar son los siguientes:

- Fecha de compra
- Cupón o promoción
- Nro de cupón
- Cliente que realizó la compra

9. Pedir Devolución

Esta funcionalidad permite a un cliente pedir la devolución de crédito por un cupón que compró y todavía no canjeó.

El usuario ingresará al sistema el código del cupón, este deberá validar la existencia del mismo como también que este mismo cliente sea el dueño (haya comprado) ese mismo cupón. En caso afirmativo el sistema mostrará todos los datos del cupón.

Este proceso permitirá el ingreso de un cupón a la vez.

Si para el cupón a pedir devolución ha expirado su fecha límite de consumo, la devolución se cancelará.

Si el proceso de devolución cumple con los requisitos necesarios, se procederá a concretar la operación y a actualizar el saldo del cliente.

Será necesario que se registre un motivo por el cual se hace la devolución, esto será determinado por los alumnos según el criterio de implementación que ellos crean conveniente.

Esta funcionalidad solo puede ser utilizada por usuarios del tipo cliente y al efectuarse la devolución es necesario que se actualice el stock de dicho cupón.

Los datos a registrar para una devolución son los siguientes:

- Fecha de devolución (fecha actual, no modificable)
- Cliente
- Código de cupón
- Motivo de devolución

10. Historial de Compra de Cupones

Esta funcionalidad permite al cliente poder consultar que cupones fue adquiriendo a lo largo del tiempo, para lo cual solo deberá completar un intervalo de fechas. Si determinan que fuese necesario el ingreso de más filtros, queda a criterio de los alumnos.

Los datos a mostrar, como mínimo deberían ser los cupones comprados y el estado de los mismo, por ejemplo comprado, devuelto, etc

11. Armar Cupón

Este caso de uso es utilizado por los proveedores para armar las promociones que luego serán publicados por los administrativos de la empresa.

El proveedor podrá ir cargando promociones con diferentes fechas, lo que no se permite es que se carguen promociones con fechas anteriores a la que tiene el sistema.

Un cupón consta de 2 precios, que son determinados por el proveedor:

- El precio real de venta
- El precio ficticio del cupón a vender

Los cupones cuentan con 2 fechas de vencimientos bien diferenciadas, una es la fecha de vencimiento de la oferta y la otra es la fecha de vencimiento para su canje o consumo.

Al momento de armar un cupón, el proveedor determinará el stock de cupones que brindará, que está dispuesto a ofrecer al público.

Cada uno de estos cupones o promociones poseen la restricción de que tienen un máximo de unidades compradas por cliente.

El armado de cupones pide obligatoriamente el registro de estos datos:

- Descripción del cupón
- Fecha de publicación
- Fecha de vencimiento de la oferta
- Fecha de vencimiento para el consumo o canje
- Precio Real
- Precio Ficticio
- Proveedor

- Cantidad disponible
- Cantidad max de comprar por usuario
- Zonas en las cuales se puede adquirir el cupón

12.Registro de Consumo cupón

Funcionalidad que permite a un proveedor dar de baja un cupón entregado por un cliente al momento de realizarse el canje.

Este proceso tiene como restricciones que un cupón no puede ser canjeado más de una vez, si el cupón se venció tampoco podrá ser canjeado y validarse que dicho cupón entrega corresponda al proveedor.

Para dar de baja un cupón disponible para consumir es necesario que se registre:

- Fecha de consumo
- Código de cupón
- Cliente

13.Publicar Cupón.

Funcionalidad que permite al administrativo publicar los cupones armados por los proveedores. Esta tarea se realiza diariamente y no existe ningún proceso automático que publique los cupones. El administrativo deberá chequear que cupones tiene para publicar. La tarea de publicación de deberá realizar de la siguiente manera:

El sistema mostrará el día en cuestión, el cual no es modificable y será tomado de del archivo de configuración. Luego de ello se podrá realizar un filtro opcional por proveedor, en caso de no utilizarse se listarán todas las promociones disponibles para la fecha. El administrativo determinará que promociones publicar y cuales no según su criterio.

Debe quedar registrado de alguna manera en la base de datos que cupones fueron publicado y cuales no.

14.Facturación a Proveedor

Esta funcionalidad permite a un administrativo facturar a un proveedor todos los cupones canjeados por los clientes.

Para ello ingresará el período de facturación por intervalos de fecha, se deberá seleccionar el proveedor y a continuación se listarán todos los cupones que fueron utilizados por los clientes. De manera de no facturar un cupón que no fue utilizado.

Una vez que se tiene dicho listado, se informará el importe de la factura y el número correspondiente de la misma.

De más esta decir que este proceso debe quedar registrado en la base datos, dado que no se permite volverá facturar un cupón que ya fue facturado.

15.Listado Estadístico

Esta funcionalidad nos debe permitir consultar el TOP 5 de:

- Top 5 porcentaje de devolución de los cupones, por proveedor por semestre.
- Top 5 por usuario a los cuales se le acreditó GiftCards por semestre.

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 por monto.

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 2 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 2008 con las siguientes consideraciones:

- El nombre de la instancia del motor de base de datos a instalar debe llamarse “SQLSERVER2008”. 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 2008. 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 “GD2C2012”.

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: “gd2012”
 - Base de Datos Predeterminada: GD2C2012.
 - El resto de los parámetros respetar sus valores default.
- Solapa “Funciones del Servidor”:
 - Seleccionar “sysadmin”
- Solapa “Asignación de Usuarios”:
 - Seleccionar asignar a “GD2C2012”
- 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: “gd2012”.

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\SQLSERVER2008 -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 2008 con el Framework de .NET 3.5. La versión Express posee la funcionalidad necesaria como para desarrollar el Trabajo Práctico.

La aplicación template se denomina “GrouponDesktop”. 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\SQLSERVER2008
- Utilizar autenticación de SQL Server:
 - Nombre de Usuario: gd
 - Password: gd2012
- Nombre de la base de datos: GD2C2012

La aplicación siempre debe conectarse a localhost. En caso de que el alumno se conecte a otra dirección, deberá cambiarlo a la hora de entregar su TP para corregir.

La aplicación deberá contar con un usuario de sistema ya creado 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 3.5. Cualquier otra implementación que no haya sido desarrollado en éste lenguaje será rechazada, sin excepción.

El entorno de desarrollo debe ser Microsoft Visual Studio 2008 o Microsoft Visual C# Studio Express 2008. No podrá ser utilizada la reciente versión 2010.

El motor de base de datos deberá ser Microsoft SQL Server 2008. 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# “GrouponDesktop” 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.
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 3 veces más (entregas que no tienen fecha fija).

La cantidad de funcionalidad de cada entrega no varía.

- *Entrega unica:*
 - *Día:* 15/11/2012 hasta las 12:00 del mediodía (GMT -3:00 Buenos Aires).
Los TPs entregados ese día, después de las 12:00 del mediodía se consideran que no son entregados en fecha y le restan 2 oportunidades más de entrega, como se menciona anteriormente.

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=dG16aEltMHclX2hPN3U2YTVoVGxfeUE6MA>

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 15/10/2012 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 y de recibido el mail con el número de grupo, la conformación de los mismos será inalterable hasta la finalización del cuatrimestre. Cualquier cambio de integrantes, sea por el motivo que fuese, deberá realizarse antes de esa fecha.

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.

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/1DDumUck5jea8srx3pXNvGfNcN-Phtxj4AIAdHikAg1Y/edit>

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 la 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 2008. No puede utilizarse la versión 2010:

1. **Microsoft Visual Studio Professional 2008:** 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# 2008 Express Edition:** Existe una versión gratuita del IDE llamada *Visual C# 2008 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 2008 Express. Puede ser descargado de la siguiente dirección:

<http://www.microsoft.com/Sqlserver/2008/en/us/express-down.aspx>

Es necesario descargar e instalar dos componentes:

- Install Microsoft SQL Server 2008 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:

TP2C2012<curso><nombreGrupo><Nro de grupo>

: espacio en blanco

Ejemplos:

TP1C2012 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:

TP2C2012 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 “GrouponDesktop”
- ⇒ \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 del modelo de datos creado con una explicación detallada de cada entidad, relaciones, claves primarias y foráneas, índices, stored procedures, triggers, vistas, 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 “GrouponDesktop”. 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 reentrega.