



Universidad Tecnológica Nacional
Facultad Regional Buenos Aires

Gestión de Datos

Trabajo Práctico
2º Cuatrimestre 2011

SUBE

Enunciado V1.0



Índice

Índice	2
Introducción.....	3
Objetivos generales.....	3
Descripción general	3
Componentes del TP.....	4
Base de Datos	4
Aplicación Desktop	4
Modelo de datos	4
Requerimientos.....	5
General.....	5
Base de Datos	5
Aplicación Desktop	5
Implementación	12
General.....	12
Base de Datos	12
Aplicación Desktop	13
Restricciones de la solución	15
Condiciones de aprobación.....	15
Testing	15
Modelo de Datos.....	16
Consultas SQL.....	16
Respetar Guía de ABMs	16
Aplicación Desktop	16
Fecha de entrega y condiciones	16
Sobre los grupos	17
Ayuda y contacto	17
Donde aprender C#.....	18
Sobre la elección de C#	18
Obtención de herramientas	19
Formato de entrega	19
Lugar de envío.....	19
Estructura del archivo zip	20
Readme.txt:.....	21
Estrategia.pdf:.....	21
\src:	21
\data:	21

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

Se le solicitó a la Universidad Tecnológica Nacional desarrollar un software que para la gestión de un sistema único de boleto electrónico el cual permitirá reemplazar el sistema actual de ventas de pasajes en los subtes además de extender dicha funcionalidad añadiendo cualquier tipo de empresas que quieran realizar el cobro de sus operaciones por este medio de pago.

El software debe permitir gestionar la carga de las tarjetas, la facturación y además generar los pagos a los diferentes beneficiarios. A su vez el sistema deberá permitir la generación de reportes y análisis de los datos para su posterior estudio por parte de la gerencia del sistema.

Actualmente la empresa cuenta con un sistema el cual permite el cobro de las líneas de subte, por lo cual será necesario ampliar los beneficiarios del sistema.

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

La cátedra provee un script que permite crear una base de datos en el motor SQL Server 2005. Esta base de datos incluye una única tabla 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.

Aplicación Desktop

La cátedra provee un proyecto C# a modo de template sobre el cual deberá desarrollarse una aplicación Desktop que interactúe con la base de datos. 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.

Modelo de datos

Todos los datos provistos por la cátedra se encuentran en una única tabla de la base de datos de SQL Server, llamada “maestra”.

Los datos de esta tabla maestra pertenecen a un dominio de expendio y cobro de boletos de subte, con información de líneas de subte, estaciones, tarjetas, etc.

El sistema a desarrollar será utilizado por 3 tipos de usuarios distintos: administradores, gestores y clientes.

Los administradores tienen permisos para realizar todas las operaciones que permite el sistema.

Los gestores son los encargados de realizar tareas administrativas del sistema tales como altas de tarjetas, altas de nuevos beneficiarios del sistema (empresas), etc.

Los clientes o puntos de venta son los encargados de realizar el cobro mediante el pago con las tarjetas, y realizar las recargas de las mismas.

En base a este análisis de datos establecen las nuevas políticas de marketing y negocio.

Los datos de la tabla maestra responden a las siguientes reglas de negocio:

- Un cliente deberá poseer no más de una sola tarjeta.
- Los datos de la tabla maestra corresponden a los servicios de subte.

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.

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.
- 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 puede inventar información, por ejemplo crear una venta que nunca sucedió.

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.

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

ABM de Tarjetas

Funcionalidad que permite crear modificar y dar de baja una tarjeta del sistema.

Dar de alta una tarjeta implica el ingreso de los siguientes datos:

- Nro de tarjeta.
- Fecha de alta.
- Cliente (Selección por buscador)

Todos los datos mencionados anteriormente son obligatorios. Los números de tarjetas son únicos en el sistema por ende no deben existir dos tarjetas con el mismo número.

El sistema deberá controlar esta restricción e informar debidamente al usuario.

Para elegir cliente al cual se le asignará la tarjeta se debe presentar un buscador con listado, que permita filtrar simultáneamente por alguno o todos los siguientes campos:

- Nombre.
- Apellido.
- DNI.
- Mail.
- Telefono.
- Dirección calle, nro piso y depto.

En la modificación solo se pueden cambiar el cliente y el nro de tarjeta.

La eliminación de una tarjeta implica una baja lógica de la misma. La tarjeta se inhabilita pero los datos de la tarjeta siguen existiendo en la base de datos. Una tarjeta inhabilitada no podrá ser utilizada para realizar una transacción, tampoco se le puede asignar un cliente. Se debe poder volver a habilitar una tarjeta deshabilitada desde la sección de modificación.

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

- Nro de tarjeta.
- Fecha de alta.
- Cliente (Selección por buscador)

ABM de Rol

Funcionalidad para poder crear, modificar y eliminar roles 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. No puede poseer funcionalidades repetidas.

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

- ABM de Tarjetas
- ABM de Rol
- ABM de Usuario
- ABM de Cliente
- ABM de Beneficiarios/Empresas
- Carga de crédito
- Efectuar Compra
- Pago a empresas
- Inactividad de tarjetas
- Clientes Premium

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

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 que ya posee así como también permitir agregarle nuevas funcionalidades.

La eliminación del rol implica una baja lógica del mismo. El rol se inhabilita. No puede asignarse un rol inhabilitado a un usuario. 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 que rol se desea modificar o eliminar se debe mostrar un listado con todos los roles existentes en el sistema.

ABM de Usuario

Funcionalidad para poder crear, modificar y eliminar usuarios del sistema.

Crear un empleado implica cargar los siguientes datos:

- Username
- Password
- Listado de roles (selección especial)

Todos los datos mencionados anteriormente son obligatorios.

El username debe ser único. 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 deben poder asignar un conjunto de roles. Un usuario no puede tener asignado dos veces un mismo rol. Se debe mostrar una tabla visual con todos los roles disponibles del sistema. Por cada rol debe existir una casilla que indique si el usuario actual posee o no ese rol.

En la modificación de un usuario solo se puede modificar el password y el listado de roles. De la lista completa de roles del sistema, esta funcionalidad debe mostrar para

cada rol, indicando si está asignado al usuario o no. Y debe permitir modificar esta asignación.

En la modificación de un usuario también podrá rehabilitarse, en caso de que éste se encuentre inhabilitado por una acción anterior.

La eliminación de un usuario implica una baja lógica del mismo. El usuario se inhabilita pero sus datos siguen existiendo en la base de datos. Un usuario inhabilitado no podrá acceder al sistema.

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

- Username (texto libre)

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, 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.

ABM de Cliente

Funcionalidad para poder crear, modificar y eliminar clientes del sistema.

Crear un cliente implica cargar los siguientes datos:

- Nombre
- Apellido
- DNI
- Mail
- Teléfono
- Dirección calle, número y piso/dpto.
- Provincia (selección acotada)

Todos los datos mencionados anteriormente son obligatorios.

No pueden existir en el sistema dos clientes con el mismo DNI. La aplicación deberá controlar esta restricción e informar debidamente al usuario. Tampoco puede repetirse el DNI con un empleado.

En la modificación de un cliente se pueden alterar todos sus datos a excepción del DNI.

La eliminación de un cliente implica una baja lógica del mismo. El cliente se inhabilita pero sus datos siguen existiendo en la base de datos. Un cliente inhabilitado no podrá realizar una compra. Se debe poder volver a habilitar un cliente inhabilitado 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)
- Provincia (selección acotada)

ABM de Beneficiarios

Esta funcionalidad permite poder crear, modificar y eliminar los beneficiarios o empresas, las cuales serán los que reciban los pagos por las ventas realizadas por este nuevo sistema de pago.

Dar de alta un nuevo beneficiario implica cargar los siguientes datos:

- Razón social o nombre de la empresa
- Dirección calle, nro, piso y depto
- Post-net registrados(selección por buscador)
- Rubro (selección acotada)

Todos los datos mencionados anteriormente son obligatorios, excepto piso y depto. No pueden existir en el sistema dos beneficiarios con la misma razón social.

En la modificación de un beneficiario se pueden alterar todos sus datos.

La eliminación de un beneficiario implica una baja lógica del mismo. El beneficiario se inhabilita pero sus datos siguen existiendo en la base de datos. Un beneficiario no podrá ser utilizado en operaciones de facturación. Se debe poder volver a habilitar al beneficiario inhabilitado desde la sección de modificación.

Para elegir que beneficiario 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 o nombre de la empresa
- Dirección calle, nro, piso y depto
- Post-net registrados(selección por buscador)
- Rubro (selección acotada)

Facturación

Funcionalidad que permite realizar una venta.

Se debe mostrar una pantalla que permita elegir:

- Beneficiario (selección por buscador)
- Post-net (selección acotada)
- Nro de tarjeta
- Fecha
- Monto

Para elegir que beneficiario se desea cargar en el campo beneficiario, se debe presentar un buscador con listado, que permita filtrar simultáneamente por alguno o por todos los siguientes campos:

- Razón social o nombre de la empresa
- Dirección calle, nro, piso y depto
- Rubro (selección acotada)

Una vez cargado el campo beneficiario, se deberá cargar los post-net de el beneficiario anteriormente seleccionado.

Al realizar la venta se debe generar una transacción o movimiento. Esta transacción debe contener todos los datos propios de la venta.

El sistema deberá validar que la tarjeta posea crédito al momento de realizar la operación. Una vez finalizada la operación el sistema mostrará el crédito restante al usuario.

Carga de crédito

Funcionalidad que permite registrar la carga de saldo a una determinada tarjeta.

Se debe mostrar una pantalla que permita elegir:

- Nro de tarjeta
- Fecha
- Monto

Al realizar la carga de crédito se debe generar una transacción o movimiento. Esta transacción debe contener todos los datos propios de la carga.

Pago a Beneficiarios

Funcionalidad que permite realizar el pago de las ventas a sus correspondientes empresas o beneficiarios.

Se debe mostrar una pantalla que permita elegir:

- Fecha de inicio.
- Fecha de fin
- Empresa o beneficiario(selección por buscador)

El sistema generará un listado con las ventas dentro del periodo elegido y para ese beneficiario en particular. El listado deberá incluir los siguientes datos:

- Fecha
- Apellido del cliente
- Nombre del cliente
- Nro de tarjeta
- Código post-net
- Marca del post-net
- Modelo del post-net
- Monto

Luego de realizar el listado correspondiente, el sistema deberá preguntar al operador si este desea registrar el pago en el sistema.

Cientes Premium

Funcionalidad que permite visualizar los clientes que más facturaron.

Se debe mostrar una pantalla que permite seleccionar:

- Año a analizar

El año debe ser un valor comprendido entre 1995 y 2100.

En base a los valores elegidos se debe mostrar un listado con los 30 clientes que más compraron, en monto. El listado se debe ordenar en forma descendente por monto.

Se deben mostrar las siguientes columnas:

- Nombre del cliente.
- Apellido del cliente.
- DNI del cliente.
- Nro de tarjeta del cliente.
- Monto total acumulado.
- Fecha de la última compra registrada.

Inactividad de tarjetas

Funcionalidad que permite dar de baja a las tarjetas que posean un periodo de inactividad mayor a 30 días corridos, a partir de la última transacción registrada.

Se deben mostrar las siguientes columnas:

- Nombre del cliente.
- Apellido del cliente.
- DNI del cliente.
- Monto total acumulado.
- Fecha de la última compra registrada.
- Beneficiario de la última compra.
- Código de post-net de la ultima compra.

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 2005 con las siguientes consideraciones:

- El nombre de la instancia del motor de base de datos a instalar debe llamarse “SQLSERVER2005”. 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 2005. 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 “GD1C2011”.

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: “gd”
 - Autenticación de SQL Server
 - Contraseña: “gd2011”
 - Base de Datos Predeterminada: GD2C2011.
 - El resto de los parámetros respetar sus valores default.
- Solapa “Funciones del Servidor”:
 - Seleccionar “sysadmin”
- Solapa “Asignación de Usuarios”:
 - Seleccionar asignar a “GD2C2011”
- 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: “gd2011”.

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\SQLSERVER2005 -U gd -P gd2011 -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 “BoletoElectronicoDesktop”. 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\SQLSERVER2005
- Utilizar autenticación de SQL Server:
 - Nombre de Usuario: gd
 - Password: gd2011
- Nombre de la base de datos: GD2C2011

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 2005. Tanto la versión Express como la full sirven para realizar el trabajo. No podrá ser utilizada la reciente versión 2008.

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# “BoletoElectronicoDesktop” 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 tenida 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 tenida 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 reutilizable 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

El TP puede entregarse a partir del día **10/10/2011** a las 12 del mediodía.

La fecha final de recepción de trabajos es el día **18/11/2011** hasta las 12 del mediodía. No se recibirán trabajos prácticos después de la fecha y hora final, sin excepción.

El TP se entrega una vez y si no se aprueba puede presentarse una vez más, siempre y cuando no se llegue a la fecha de entrega final. En total solo existen **dos** oportunidades para entregar el TP, siempre y cuando el alumno no se pase de la fecha de entrega final.

La cantidad de funcionalidad de cada entrega no varía.

La nota decrece con cada entrega realizada.

Los trabajos prácticos entregados deberán contar con la totalidad de las funcionalidades pedidas. Solo tendrán posibilidad de reentrega aquellos TPs

desaprobados que tuvieron problemas o bugs en alguna funcionalidad particular que se consideraba terminada. Los TPs que sean entregados con funcionalidades sin terminar, o a medio hacer, serán rechazados sin continuar su evaluación y sin posibilidad a reentrega.

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”.

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/11cPz0LhFvEw9L7I9svu_aqvCb-VSIEsAZ3vjUGZb7B-U/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.programacion.net/tutorial/csharp/> de José Antonio González Seco

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 2005 Express. Puede ser descargado de la siguiente dirección:

<http://www.microsoft.com/Sqlserver/2005/en/us/express-down.aspx>

Es necesario descargar e instalar dos componentes:

- Install Microsoft SQL Server 2005 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:

TP2C2011<curso><nombreGrupo>

: espacio en blanco

Ejemplos:

TP2C2011 k9999 LOS_MEJORES

(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:

TP2C2011 k9999 LOS_MEJORES.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 “BoletoElectronicoDesktop”
- ⇒ \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.

\src:

Dentro de este directorio se encuentra la solución entera de Visual Studio del proyecto “BoletoElectronicoDesktop”. 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.

\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.