



Universidad Tecnológica Nacional
Facultad Regional Buenos Aires

Gestión de Datos

Trabajo Práctico

2° Cuatrimestre 2010

Calificación Bancaria

Enunciado V1.0



Índice

Introducción.....	3
Objetivos generales	3
Descripción general.....	3
Componentes del TP	4
Requerimientos	5
Implementación.....	10
Restricciones de la solución.....	13
Condiciones de aprobación.....	13
Fecha de entrega y condiciones	14
Sobre los grupos.....	14
Ayuda y contacto	14
Donde aprender C#	15
Sobre la elección de C#	15
Como obtener el visual C# 2005.....	15
Formato de entrega.....	16

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

La Cámara Nacional de Entidades Financieras de la República Argentina solicita a la Universidad Tecnológica Nacional desarrollar un software que permita evaluar la calidad crediticia de diversos clientes de Bancos del País.

El software debe permitir recopilar la información de diversas entidades bancarias del país y agruparla para poder obtener conclusiones comunes.

Se ha recopilado información bancaria de distintas unidades pero la misma se encuentra en forma desordenada.

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 2005 y Framework de .NET 2.0.

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 pertenecen a un dominio de negocio bancario, con información sobre clientes, bancos, cuentas, transacciones, etc.

La aplicación a desarrollar y el modelo de datos están orientados a que los utilice un usuario con rol de Operario Bancario o Administrador de Cuentas interno. Este operario bancario puede administrar cuentas de diversos clientes de diversas sucursales y bancos. La aplicación no tiene que contemplar que un cliente externo pueda acceder a los datos de la misma.

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

- La base de datos posee información sobre distintos bancos, cada uno con distintas sucursales.
- Cada banco posee sus propios clientes. Si una persona está en más de un banco, se consideran como clientes diferentes.
- Un cliente de un banco puede tener varias cuentas abiertas en varias sucursales.

- Los clientes pueden hacer transacciones (depósitos o extracciones) entre cuentas. Las transacciones pueden ser entre la misma cuenta, entre cuentas suyas o entre cuentas de distintos clientes.
- Un cliente puede tener varias tarjetas de crédito.
- Una cuenta de un cliente puede tener deuda en un determinado periodo de tiempo.
- Una cuenta de un cliente puede librar cheques contra otras cuentas.
- Un cliente puede tener un plazo fijo en una sucursal particular de un banco. Los mismos pueden estar activos actualmente o ya haber finalizado.
- Todas las operaciones realizadas se realizan sobre un tipo específico de moneda, que debe ser especificado en la transacción o la cuenta.
- Todas las bajas de la aplicación deben ser lógicas. Los registros físicos nunca deben eliminarse por completo.
- Debe quedar un registro histórico de todo lo sucedido. Por ejemplo si un cliente pertenecía a una sucursal de un banco y ya no pertenece más, debe quedar algún registro de aquella existencia en el pasado.

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.

Aplicación Desktop

El alumno deberá crear una aplicación Desktop en C# sobre Visual Studio 2005 con Framework .NET versión 2.0. Esta aplicación deberá contar con formularios, reportes y tablas que permitan cumplir los siguientes requerimientos de negocio:

ABM de Cliente

Poder crear, modificar y eliminar lógicamente a un cliente de la base de datos.

Crear un cliente implica especificar todos sus datos personales:

- Nombre
- Apellido
- DNI
- Mail

Luego se le deberá asignar un Banco al Cliente y se le deberá especificar una Sucursal del mismo.

En la modificación se deberá poder cambiar solo los datos personales del Cliente, sin posibilidad de cambiar el Banco. Se deberá poder agregar nuevas sucursales de Bancos al Cliente, pero no eliminar las existentes.

Al eliminar un cliente se deberá efectuar solo una baja lógica del mismo (sin borrar el registro real). Ese Cliente ya no estará disponible en el Banco ni en ninguna de sus sucursales.

Para elegir que cliente se desea modificar o eliminar se debe presentar un buscador con listado, que permita filtrar por:

- Nombre y Apellido del Cliente
- DNI del Cliente
- Banco
- Sucursal

ABM de Cuenta

Poder crear y eliminar lógicamente cuentas bancarias a un Cliente.

Crear una nueva Cuenta Bancaria implica:

- Generar su número de cuenta
- Sucursal del Banco a la cual pertenece la cuenta.
- Monto inicial de la cuenta
- Tipo de moneda que soporta la cuenta.

Solo se pueden agregar cuentas a clientes activos de una sucursal de un Banco.

Eliminar una cuenta implica realizar una baja lógica sobre la misma (sin borrar el registro real). El cliente ya no podrá realizar transacciones sobre esa cuenta.

El monto inicial deberá impactarse en la cuenta como si fuera una transacción de depósito (ver funcionalidad “Transacción entre cuentas”).

Para elegir que agregar una cuenta a un cliente se debe presentar un buscador con listado, que permita filtrar por:

- Nombre y Apellido del Cliente
- DNI del Cliente
- Banco
- Sucursal

Si el Cliente ya tiene cuentas, se debe mostrar el listado de cuentas activas que posee.

Transacción entre cuentas

Permitir realizar una transacción (depósito o extracción) desde una cuenta de un cliente hacia otra cuenta de otro cliente. Las cuentas pueden ser del mismo Banco o entre distintas entidades. Si son dentro del mismo Banco pueden ser de sucursales distintas. Se pueden hacer extracciones de una cuenta ajena porque estas transacciones son realizadas por un usuario con rol de administrador bancario, y no por un cliente.

Realizar una transacción implica:

- Especificar la cuenta de origen. La misma se debe poder especificar con su número de identificación o también a través de una búsqueda por Cliente, Banco y Sucursal.
- Especificar la cuenta destino, con los mismos criterios de selección que la cuenta origen.
- Especificar el tipo de operación: depósito o extracción.
- Especificar el monto de la transacción a realizar.
- Tipo de moneda sobre la cual se realiza la transacción.

Solo se pueden hacer transacciones entre cuentas activas.

Especificar la cuenta destino debe ser opcional. En caso de no realizarlo se considera que el destino de la transacción es la misma cuenta de origen.

Está permitido extraer de una cuenta un monto superior al que posee en el momento de la operación.

La fecha de operación de la transacción debe corresponder al momento en el cual se efectúa.

Consulta de plazos fijos

Listar los plazos fijos activos e históricos que tiene un cliente en las distintas sucursales de los bancos.

El listado debe mostrar las siguientes columnas:

- ID de cliente
- Nombre y Apellido del cliente
- DNI del cliente
- ID de cuenta
- Banco
- Sucursal
- Monto del plazo fijo convertido a pesos.
- Fecha de inicio del plazo fijo
- Fecha de fin de plazo fijo
- Columna que indica si el plazo fijo se encuentra actualmente vigente o ya finalizó.
- Interés ganado por el plazo fijo.

Para calcular el interés ganado se deben aplicar las siguientes tasas:

<i>Duración del plazo fijo en meses</i>	<i>% de interés a aplicar por cada mes transcurrido</i>
1-3	5%
4-6	12%
6-12	17%
12 o más	23%

El interés debe aplicarse por cada mes transcurrido. Por ejemplo si un plazo fijo se aplicó durante dos meses, se deberá sumar un 5% del monto (categoría 1-3) dos veces, uno por cada mes del plazo fijo.

Si el periodo no es exacto, es decir que quedan días sin formar un mes entero, por cada día extra se deberá sumar un 0.57%. Por ejemplo si un plazo fijo tiene 3 meses y 15 días, se deberá aplicar 3 veces 5% y 15 veces 0.57%.

La pantalla debe mostrar un buscador con filtros de búsqueda por los siguientes campos:

- Nombre y Apellido del Cliente
- DNI del Cliente
- Banco
- Sucursal
- Fecha desde y Fecha hasta de la creación del plazo fijo, ambas inclusive.

Auditoría de Tarjetas de Crédito

Periódicamente los Operadores Bancarios realizan ciertas modificaciones de datos directas sobre las transacciones efectuadas con Tarjetas de Crédito. Estas modificaciones se realizan mediante comandos de UPDATE y DELETE en forma directa contra la base de datos.

Es necesario implementar algún mecanismo que permite registrar todas estas modificaciones efectuadas contra las transacciones de tarjetas de crédito, a efectos de que luego puedan ser sancionadas por una auditoría general.

Proceso de calidad sobre clientes

La aplicación deberá calcular un número que clasifique la calidad de los clientes de los bancos. Por cada cliente de cada banco se deberá calcular este número que ayudará a definir la calidad financiera de estos clientes.

El número de calidad es un valor entero que puede tomar valores entre 0 y 100, siendo 100 la calificación para el cliente con las mejores prestaciones.

Este proceso debe poder ejecutarse desde la aplicación Desktop. El resultado del mismo no es mostrado en pantalla, sino que deberá volcarse en una tabla de la base de datos. Esta tabla se denomina “proceso_calidad_clientes” y tiene las siguientes columnas:

- ID de Cliente
- ID de Banco
- Fecha del proceso
- ID del proceso
- Valor de calidad

Como fecha del proceso se deberá tomar el momento en el cual se decide ejecutar el mismo desde la aplicación.

El número de calidad se calcula en base a varios parámetros que deben ser tenidos en cuenta de la operatoria del cliente. Los mismos se listan a continuación:

- Se suman 3 puntos por cada cuenta que posea el cliente que a la fecha del proceso no posean deudas.
- Se suman 4 puntos por cada 100 transacciones entre cuentas que el cliente haya efectuado con alguna de sus cuentas como origen.
- Se suma 6 puntos por cada 30 transacciones entre cuentas en las que alguna cuenta del cliente aparezca como destino, que pertenezcan al último año según la fecha del proceso.
- Se suman 5 puntos por cada 100 operaciones de tarjeta de crédito que haya realizado el cliente en el último año de la fecha del proceso, y 3 puntos por cada 100 operaciones de tarjeta en los demás años.
- Se suman 2 puntos por cada \$2000 (pesos) del monto total de todas las transacciones de cheques en las que aparezca una cuenta del cliente como origen.
- Se suman 9 puntos si el monto total de todas las transacciones de tarjeta de créditos de los últimos 3 años es mayor que el monto total de todas las transacciones entre cuentas que tengan una cuenta de origen de este cliente de los últimos 3 años.
- Se suman 4 puntos si el promedio de monto en operaciones de tarjeta de crédito en los últimos 5 años es superior a los \$1000 (pesos).

Todos los cálculos deben hacerse en pesos. Si las monedas de las transacciones son distintas se deberá hacer la conversión correspondiente.

Si el puntaje del cliente supera los 100 puntos, se deberá truncar en 100.

Al ejecutar el proceso se debe generar un ID único del proceso. El mismo debe informarse en pantalla.

Proceso de otorgación de préstamos

La aplicación deberá calcular el máximo monto en pesos de préstamo que se estaría dispuesto a entregar a un cliente.

Por cada cliente de cada banco se deberá calcular este monto que ayudará a definir el máximo préstamo que un banco debería prestar a un cliente.

Este proceso debe poder ejecutarse desde la aplicación Desktop. El resultado del mismo no es mostrado en pantalla, sino que deberá volcarse en una tabla de la base de datos. Esta tabla se denomina “proceso_otorgacion_prestamos” y tiene las siguientes columnas:

- ID de Cliente
- ID de Banco
- Fecha del proceso
- ID del proceso
- Valor de calidad

Como fecha del proceso se deberá tomar el momento en el cual se decide ejecutar el mismo desde la aplicación.

El monto se calcula en base a varios parámetros que deben ser tenidos en cuenta de la operatoria del cliente. Los mismos se listan a continuación:

- Sumar el 60% del promedio anual de los montos de transacciones entre cuentas del cliente que tienen alguna cuenta del cliente como destino (depósitos en cuenta).
- Sumar el 45% del promedio anual de los montos de con cheques que tienen alguna cuenta del cliente como destino (depósitos de cheques).
- Restar un 7% del promedio anual de todas las operaciones hechas con tarjetas de crédito del cliente.
- Sobre el total acumulado hasta el momento (mediante los porcentajes anteriores), sumar un monto extra en base al tiempo durante el cual cada una de las cuentas del cliente tuvo deuda, utilizando la siguiente tabla de porcentajes:

<i>Meses</i>	<i>% extra a aplicar sobre el monto total calculado hasta el momento</i>
0	20%
1-3	10%
4-6	5%
7 o más	0%

Este porcentaje debe aplicarse por cada cuenta activa del cliente.

Todos los cálculos deben hacerse en pesos. Si las monedas de las transacciones son distintas se deberá hacer la conversión correspondiente.

Si el monto total a prestar es menor a cero, deberá truncarse en cero.

Al ejecutar el proceso se debe generar un ID único del proceso. El mismo debe informarse en pantalla.

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 “GD2C2010”.

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: “gd2010”
 - Base de Datos Predeterminada: GD2C2010.
 - El resto de los parámetros respetar sus valores default.
- Solapa “Funciones del Servidor”:
 - Seleccionar “sysadmin”
- Solapa “Asignación de Usuarios”:
 - Seleccionar asignar a “GD2C2010”
- 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: “gd2010”.

Una vez que tenemos la base de datos creada y configurada con el usuario, necesitamos ejecutar dos scripts. Para ello abrir una hoja de “Nueva Consulta” dentro de la base de datos “GD2C2010”:

- 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 (aprox. 1GB) y no puede ser ejecutado desde el “Management Studio”. Debe ejecutarse desde la consola de comandos de Windows mediante la siguiente expresión:

sqlcmd -S <Servidor\Instancia> -U <Nombre_de_usuario> -P <Password> -i <Nombre_del_archivo>

Ejemplo:

sqlcmd -S localhost\SQLSERVER2005 -U gd -P gd2010 -i gd_esquema.Maestra.Table.sql

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

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 2005 con el Framework de .NET 2.0. La versión Express posee la funcionalidad necesaria como para desarrollar el Trabajo Práctico.

La aplicación template se denomina “CalificacionBancariaDesktop”. 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: gd2010
- Nombre de la base de datos: GD2C2010

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.

Restricciones de la solución

El lenguaje de programación utilizado deberá ser únicamente C# utilizando el Framework .NET 2.0. Cualquier otra implementación que no halla sido desarrollado en éste lenguaje será rechazada, sin excepción.

El entorno de desarrollo debe ser Microsoft Visual Studio 2005 o Microsoft Visual C# Studio Express 2005. No podrá ser utilizada la reciente versión 2008.

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 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# “CalifacionBancariaDesktop” 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.

Modelo de Datos

El modelo de datos creado por el alumno deberá respeta 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 los mismos será tomada en cuenta a la hora de fijar la nota.

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.

Fecha de entrega y condiciones

El TP puede entregarse a partir del día **04/10/2010** a las 12 del mediodía.

La fecha final de recepción de trabajos es el día **08/11/2010** 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 hasta 2 veces más, siempre y cuando no se llegue a la fecha de entrega final.

La cantidad de funcionalidad de cada entrega no varía.

La nota decrece con cada entrega realizada.

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.

Ayuda y contacto

Cualquier duda que pueda surgir sobre el enunciado del TP, podrá ser planteada en el siguiente grupo de *GOOGLE*:

<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.

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 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.

Como obtener el visual C# 2005

El TP puede ser desarrollado con dos versiones del IDE Microsoft Visual Studio 2005:

1. **Microsoft Visual Studio Professional 2005:** 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# 2005 Express Edition:** Existe una versión gratuita y en español del IDE llamada *Visual C# 2005 Express Edition*, la cual posee todas las herramientas necesarias para realizar el TP. Esta se encuentra disponible en:

<http://www.microsoft.com/spanish/msdn/vstudio/express/VCS/default.mspix>

De acuerdo al sitio:

*“Visual C# Express Edition es funcional durante 30 días. Para extender este periodo de forma indefinida debes registrar de forma gratuita tu copia desde el mismo equipo en el que está instalado el software. Como agradecimiento por registrar el producto recibirás beneficios especiales
La instalación típica de un producto Express requiere 500MB de espacio en disco. La instalación completa, incluyendo documentación, requiere 1.3GB de espacio en disco”*

También puede ser descargada la versión Express en inglés del siguiente sitio:

<http://www.microsoft.com/express/2005/download/default.aspx>

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:

TP2C2010<curso>G<numeroGrupo>

: espacio en blanco

Ejemplos:

TP2C2010 k9999 G99

(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:

TP2C2010 k9999 G99.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 siguiente dirección:

<https://spreadsheets.google.com/viewform?authkey=CIqchacC&hl=en&formkey=dHRPY3dMSE5qZklkNjlRRVB3VVpBemc6MA#gid=0>

Al registrarse cada grupo obtendrá un número de grupo que será necesario para enviar su TP a corrección. El número de grupo no se muestra al instante, sino que la cátedra luego publicará una lista de los números de grupos para todos los alumnos.

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 “CalificacionBancariaDesktop”
- ⇒ \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.

\src:

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