



Universidad Tecnológica Nacional
Facultad Regional Buenos Aires

Gestión de Datos

Trabajo Práctico

1° Cuatrimestre 2015

FRBA-Pago Electrónico

Enunciado V1.0



Índice

Índice	2
Introducción.....	3
Objetivos generales.....	3
Descripción general	3
Componentes del TP.....	4
<u>Base de Datos y Modelo de Datos</u>	4
<u>Aplicación Desktop</u>	4
Requerimientos	4
General.....	4
Base de Datos	4
Aplicación Desktop	5
General.....	13
Base de Datos	13
Aplicación Desktop	14
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	16
Ayuda y contacto	17
Donde aprender C#.....	17
Sobre la elección de C#	17
Obtención de herramientas	18
Formato de entrega	19
Lugar de envío	19
Estructura del archivo zip	20
Readme.txt:.....	20

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

El siguiente trabajo práctico se intenta simular la migración y renovación de un sistema de pago electrónico.

La base de este nuevo sistema se sustenta en tres operaciones principales y bien distintas una de otra:

- Depósito de dinero
- Transferencias entre cuentas.
- Retiro de dinero (vía cheques)

Además de estas 3 funciones principales, se deberán desarrollar las restantes funcionalidades de apoyo para que dicho sistema opere como un todo.

Dicho sistema deberá permitir el seguimiento de cada una de las cuentas que componen su universo, para ello será necesario que se administren los clientes que componen la plataforma como así también la facturación de un cargo por cada una de las transferencias de dinero que estos realicen.

El sistema se encargará de realizar cálculos necesarios para poder determinar si una cuenta posee o no saldo y cargo a cobrar en función del tipo de cuenta que se trate al momento de realizar una transferencia.

En primera instancia se desarrollará un prototipo bajo una aplicación escritorio que permita conectarse a la base y constará de 2 perfiles/roles bien definidos, uno será el de Administrador que posee acceso a todas las funcionalidades y otro perfil/rol de Clientes que tendrá un acceso limitado de funciones, se deberá tener en cuenta que estos roles pueden variar en el tiempo como así también la creación de nuevos roles en un futuro, de manera de descentralizar las funciones del administrador.

El principal objetivo de este nuevo sistema es mejorar la interfaz con los diferentes usuarios y brindarles nuevas funcionalidades que no estaban previstas en la versión anterior, de manera de captar nuevos clientes y potenciar su utilización incrementando los ingresos de la empresa.

Componentes del TP

El alumno recibirá dos componentes ya implementados 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, como por ej. Evitar que determinadas entidades no sean modeladas por más que las mismas no contentan datos que varíen en el ciclo de vida del sistema.

Los datos de la tabla maestra contienen las operaciones realizadas por el sistema anterior es decir: reservas, registro de ingreso y egreso de huéspedes, facturación de estadía, etc.

El sistema a desarrollar será utilizado por 2 tipos de usuarios distintos: administrador, 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, dado que ciertas cuestiones ya fueron resueltas y discutidas con anterioridad.

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 se pueden utilizar 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. Si el script genera algún error en su ejecución, la entrega quedará automáticamente desaprobada.

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. Se recomienda que todos estos componentes deban 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. ABM de Rol
2. Login y seguridad
3. ABM de Usuario
4. ABM de Cliente
5. ABM de Cuenta
6. Depósitos
7. Retiro de Efectivo
8. Transferencias entre cuentas
9. Facturación de Costos
10. Consulta de saldos
11. 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

Esta opción permite crear, modificar y eliminar el acceso de un tipo de usuario a una funcionalidad propia del sistema.

Crear un rol implica cargar los siguientes datos:

- Nombre del rol
- Listado de Funcionalidades (selección acotada)
- Estado del rol (activo/no activo)

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, de más está decir que una funcionalidad puede estar en más de un rol al mismo tiempo

Debe tenerse en cuenta que actualmente existen 2 roles:

- Administrador.
- Cliente.

En la modificación de un rol se pueden alterar todos los campos. Se deben poder quitar y agregar nuevas funcionalidades a rol que se está modificando.

La eliminación del rol implica una baja lógica del mismo, en ese caso pasaría a un estado no activo, por lo tanto, el rol debe poder inhabilitarse. El sistema debe poder volver a habilitar un rol que fue inhabilitado, esta acción debe llevarse a cabo desde la sección de modificación.

En el supuesto caso que un usuario tenga asignado un rol inhabilitado, el mismo no puede ser seleccionado al momento del login.

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, inicio del sistema, el usuario no podrá acceder a ninguna funcionalidad del sistema hasta haber completado el proceso de Login.

El/los administrador/es son los encargados principales de generar los usuarios y contraseñas de los posibles clientes junto con los datos restantes, aunque no se descarta que el formulario de login contenga un vínculo que permita realizar esta operación de alta nueva de cliente, a fin de agilizar la generación de usuarios. Bajo ninguna circunstancia, un potencial cliente puede acceder a otra funcionalidad que no sea la de creación de ID de usuario y su respectiva password.

Ya entrando en el proceso de logueo, el sistema pedirá al usuario su Username y su Password.

Si el Login es correcto, el usuario podrá acceder al sistema. Si el usuario tiene un solo rol asignado se ingresará al sistema directamente sin tener que seleccionar con cual rol o perfil quiere ingresar, pero si el usuario en cuestión tiene más de uno (porque un administrador también puede ser cliente de la plataforma), se deberá dar la posibilidad de seleccionar uno de los roles o perfiles. Luego de validar el login, la aplicación deberá generar y mostrar las entradas de menú disponibles para este usuario según sea su rol, ocultado aquellas funcionalidades a las que no tiene acceso.

Si el Login es incorrecto el usuario no podrá acceder a sus funcionalidades asignadas.

El sistema debe llevar un registro de cantidad intentos fallidos al querer acceder. Luego de 3 intentos fallidos en cualquier momento, el usuario debe ser inhabilitado. Deberá tenerse en cuenta que también deberá registrarse en una tabla de auditoría propia para el login todos los intentos de accesos sean efectivos como incorrectos, la cual seguirá un esquema del tipo log almacenando que usuario se trató de loguear, fecha y hora del mismo (deberán obtenerla en este único caso de la base de datos, luego deberán manejarse con el archivo de configuración), si el intento fue correcto o incorrecto y si se tratase de un intento fallido, registrar que número de intento fue. Si los alumnos creen pertinente registrar algo dato adicional pueden hacerlo.

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. ABM de Usuario.

Funcionalidad que solo se encuentra disponible exclusivamente para el administrador del sistema y no puede ser asignada a ningún otro tipo de usuario.

Un administrador puede dar de alta a otros administradores o clientes.

Crear un Usuario implica cargar los siguientes datos:

- Username
- Password
- Rol asignado (administrador/cliente) Tener en cuenta que en un futuro existan más roles o su descripción puede modificarse.
- Fecha de creación
- Fecha de ultima modificación
- Pregunta secreta
- Respuesta secreta

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 y la respuesta secreta deberán almacenarse encriptando de forma irreversible bajo el algoritmo de encriptación SHA256.

Debe tenerse en cuenta que el password sea modificable una vez que el usuario se encuentre logueado, también debe contemplarse que un administrador solo puede realizar modificaciones y dar de baja a usuarios, independiente a la inhabilitación por fallas de ingreso. **IMPORTANTE:** Toda baja debe realizarse en forma lógica.

Un dato a tener en cuenta, un cliente solo puede tener un usuario dentro del sistema, y un usuario solamente podrá estar ligado a un solo cliente. Si por algún motivo dicho cliente ha sido dado de baja y se intenta volver a crear ese Cliente con los mismo datos, se deberá informar esta situación de datos ya existentes y rehabilitarlo blanqueando sus contraseñas.

A fines prácticos, los alumnos deberán entregar un set de datos (más de uno) en donde dichos usuarios sean del tipo “administrador” con password w23e (únicamente esos 4 caracteres), además de generar (durante el proceso de migración) todos los usuarios pertenecientes a los clientes que se encuentran en la tabla maestra.

La implementación de este ABM de Usuarios no será necesaria que se realice.

4. ABM de Clientes

Funcionalidad que permite a un administrador crear, modificar y dar de baja un cliente del sistema.

Los clientes son aquellas personas que realizan depósitos, extracciones y transferencias de dinero a otros clientes.

Cuando se quiera dar de alta un nuevo cliente, se deberá tener en cuenta lo siguiente a fin de agilizar la operatoria; si un administrador quiere dar de alta un nuevo cliente deberá completar aquellos datos que son propios para identificar al usuario (detallados en el punto anterior) como así también los datos que le son propios como cliente, los cuales son:

- Nombre.
- Apellido.
- Número de identificación y su correspondiente tipo. (tener en cuenta que pueden ser extranjeros).
- Mail.
- País.
- Domicilio.

- Calle.
- Piso.
- Depto.
- Localidad.
- Nacionalidad.
- Fecha de Nacimiento.

Una vez completados los datos necesarios para la identificación del usuario como así también los que definen al cliente, se estará en condiciones de dar de alta un nuevo cliente, validaciones mínimas a tener en cuenta: se deberá controlar que el número y tipo de documento no estén duplicados, y mismo control para la dirección de email.

Al momento de querer realizar una modificación, solo se podrá modificar aquellos datos que identifican al cliente y no al usuario. Solamente al momento del alta pueden cargarse datos de usuario y cliente para realizar una carga más ágil y rápida.

Un cliente inhabilitado no podrá realizar depósitos, retiros de dinero y transferencias entre cuentas, por más que las mismas dispongan de saldo. Se debe poder volver a habilitar el cliente deshabilitado desde la sección de modificación de esta ABM.

Para seleccionar un cliente que se desea modificar o eliminar (inhabilitar) se deberá presentar un buscador con un listado de resultados obtenidos (grilla), que permita filtrar simultáneamente por alguno o todos los siguientes campos:

- Nombre (texto libre)
- Apellido (texto libre)
- Tipo de identificación (selección acotada)
- Nro. de identificación (texto libre solo números)
- Email (texto libre)

Debe tenerse en cuenta que también se podrán asociar diferentes tarjetas de crédito, dichas tarjetas deben ser propias del cliente y estar registradas a su nombre.

Dentro de este ABM también se podrá modificar los datos de las tarjetas asociadas como así también realizar el proceso de desvinculación, es importante que los 16 dígitos que conforman la tarjeta estén encriptados y solo puedan ser visible los últimos 4 dígitos. Los datos a persistir, adicionales, referente a las tarjetas de crédito será definido por los alumnos.

5. ABM de Cuenta

Funcionalidad que permite a un administrador o a un cliente crear, modificar y dar de baja una cuenta de dinero electrónico. Si un administrador quiere de dar alta/modificar o dar de baja una cuenta se deberá indicar a que usuario del sistema impactará dicha acción.

Dar de alta una nueva cuenta implicará el ingreso de los siguientes datos:

- Número de cuenta.
- País en la cual se registra la cuenta.
- Moneda (En este caso siempre será dólares, pero se deberá prever que en un futuro se pueda seleccionar el tipo de moneda)
- Fecha de apertura
- Tipo de cuenta (Oro/Plata/Bronce/Gratuita)

En cuanto a las características de las cuentas se define que un cliente puede tener registradas más de una cuenta, no se prevé que existen cuenta compartidas entre diferentes clientes, por lo que una cuenta tiene asignado un solo cliente.

Si un cliente tiene más de una cuenta registrada, las transferencias que realice entre las mismas no tienen costo alguno y otra característica importante es que ninguna cuenta (sea del tipo que sea) permite giro en descubierto.

A estas cuentas se les tiene que asignar un país, el cual no necesariamente tiene que ser el mismo país de origen o donde resida de cliente.

Para esta versión y a modo de simplificar la elaboración del TP, solo se aceptará como moneda dólares estadounidenses, pero se espera que se puedan manejar más de una moneda a la vez en un futuro cercano y ya se establezcan las estructuras, a nivel diseño de base de datos, para que el modelo a entregar cumpla con esta condición.

Las cuentas electrónicas que administra este sistema se encuentran tipificadas, dichas categorías pueden ser:

- Oro
- Plata
- Bronce
- Gratuitas

Estos tipos de cuenta tienen una duración en el tiempo determinada en días y una vez que se vence su suscripción la cuenta queda inhabilitada. Recordamos que cada una de estas categorías tiene asociado un costo (debe poder modificarse) que varía entre los diferentes tipos y dicho costo debe poder ser facturado (ver caso de uso facturación), salvo la categoría gratuita que no genera costo alguno. Si bien no es necesario que se realice la codificación del **ABM Tipo de Cuentas**, se les pide a los alumnos que planteen una solución que permita la modificación de los tipos sin tener que realizar cambios en el código fuente y de producirse un cambio no afecte a otros registros.

En cualquier momento, un cliente puede decidir modificar la categoría de su cuenta ya sea para tener una cuenta con mejor categoría superior o inferior.

Por último, se les informa que una cuenta tiene distintos tipos de estados:

- **Pendiente de activación:** Se crea la cuenta pero para que la misma pase a activa es necesario que pague el costo de apertura por medio de la generación de factura (ver caso de uso correspondiente a facturación)
- **Cerrada:** La cuenta fue cerrada y no puede volver a activarse, previo a ello, el cliente deberá tener pago todos los costos de las transacciones pendientes. Si la cuenta se cierra, debe registrarse la fecha en que la se realizó esta acción.
- **Inhabilitada:** La cuenta puede recibir dinero de otras cuentas, pero el cliente no puede realizar depósitos y/o retiro de dinero hasta que se le hayan facturado los costos por las transacciones realizadas y que aún no fueron rendidas.
- **Habilitada:** La cuenta está disponible para su uso.

6. Asociar/Desasociar tarjetas de crédito.

Funcionalidad que permite a un cliente asociar una tarjeta para la realización de los depósitos como también desasociar las mismas que se dieron de bajar.

Los datos de la tarjeta de crédito a registrar, son los siguientes:

- Numero.
- Emisor (selección acotada).
- Fecha de emisión.
- Fecha de vencimiento.
- Código de seguridad (encriptado).

Cabe destacar que la funcionalidad debe proveer un buscador mediante el cual el cliente pueda seleccionar la tarjeta de crédito la cual va a modificar los datos de la misma o desasociarla.

7. Depósitos.

Funcionalidad que permite a un cliente crear depósitos de dinero en alguna de las cuentas que tenga registrada.

Para registrar un depósito será necesario que se determinen ciertos datos a persistir:

- Cuenta del cliente (tendrá que ser seleccionable, no se permite que se ingrese dígito a dígito)
- Importe (mayor o igual a 1).
- Tipo moneda(selección simple, por el momento dólares).
- Selección de la tarjeta de crédito (TC) de las cuales tenga asociada por medio de la cual quiere realizar el depósito. Se debe validar que la dicha TC no se encuentre vencida.
- Fecha del depósito

En esta primera versión solo está previsto que los ingresos se realicen por medio de TC. Los depósitos a realizarse solo podrán ser efectuados por el mismo titular de la TC, la cuenta receptora de este depósito debe existir y estar habilitada.

Estos depósitos no tienen comisiones ni tampoco generar un cargo que tenga que ser facturado dentro del caso de uso de facturación, así mismo es necesario que se genere un comprobante de depósito en donde se registrará la fecha de dicho ingreso y el número de operación. El número de ingreso no será consecutivo a la numeración utilizada para los retiros de dinero y los números de transacciones realizadas.

A modo de simplificación, los ingresos siempre serán efectivos y no habrá rechazo alguno por falta de disponibles en las TC de los clientes, impactando de manera instantánea en la cuenta electrónica, por esta razón la fecha del depósito deberá ser la fecha actual del sistema, obtenida del archivo de configuración, no se permiten depósitos con fecha pasadas o futuras a la que se encuentra en dicho archivo.

8. Retiro de efectivo.

Esta funcionalidad que permite a un cliente retirar dinero de una cuenta que posea a su nombre por medio de un cheque que generará el sistema. Para que este retiro sea efectivo la cuenta debe:

- Estar habilitada, cualquier otro estado no permite el retiro de dinero, por más que la cuenta tuviese saldo.
- Tener saldo.
- El importe ingresado debe ser menor o igual al saldo de la cuenta.
- El importe debe estar expresado en dólares.

Para poder generar el cheque, el sistema pedirá que se seleccione la cuenta de la cual se quiere extraer dinero y a su vez ingresar el número de documento que para poder validar la identidad de dicho cliente, que deberá ser igual al usuario que se encuentra logueado.

Una característica de los retiro de dinero es que no se cobra comisión por la generación de cheques, por ende esta acción no deberá facturarse.

Además de registrar el retiro de efectivo se deberán registrar los datos propios al cheque, el numero de chque, la fecha de hoy (obtendrán la fecha de la misma manera que para los depósitos y con las mismas validaciones), registrándose el importe, nombre a quien se le libra, (del cliente) y banco al cual pertenece el cheque. También será necesario que se genere un código o número de egreso que no será consecutivo con los depósitos y número de transacciones.

9. Transferencia entre cuentas

Funcionalidad que permite a un cliente realizar un pago a otro usuario del sistema que posea una cuenta en la plataforma. Para poder realizar un pago se debe poder seleccionar una cuenta Origen, la cual debe ser alguna de las cuentas que tenga asignada el cliente y luego seleccionar una cuenta Destino, esta puede ser otra cuenta propia o alguna otra cuenta que se encuentra habilitada o inhabilitada. Las cuentas que se encuentren cerradas o pendientes de activación no podrán recibir dinero.

Estas operaciones de transferencias de fondos deben ser por un importe mayor a cero, y a su vez, este importe debe ser menor o igual al saldo de la cuenta que origina la operación.

Está permitido que las transferencias puedan ser entre cuentas de distintos países.

Las operaciones realizadas son con fecha del día (de la misma forma que se detalla para los depósitos y retiro de efectivo), no hay transferencias programadas con fechas futuras o pasadas y debe generarse un número de operación independiente de los depósitos y retiros.

Una característica importante de las transferencias es que las mismas tienen un costo fijo que se cobra a la cuenta origen, dicho costo fijo dependerá del tipo de cuenta, pero si la cuenta origen y destino son del mismo usuario, estas transferencias no tienen costo.

10. Facturación de costos.

Esta opción puede ser utilizada tanto por los clientes como por los administradores del sistema.

En ella se tendrán que facturar tanto las comisiones pertinentes a las transferencias realizadas como así también los costos por apertura de cuenta o modificación del tipo de cuenta. Como la suscripción al tipo de cuenta tiene N cantidad de días de vigencia (período que no es fraccionable), se puede optar por pagar el costo de X suscripciones todas del mismo tipo.

Si llegan a tener más de 5 transacciones, ya sean comisión por transferencia de fondos o apertura o por cambio de tipo de cuenta, el sistema procederá a inhabilitar la cuenta en cuestión.

Una factura puede contener transacciones de diferentes cuentas que el cliente posee, no es necesario que tenga que realizarse una factura por cuenta.

La factura deberá tener una serie de datos propios los cuales definirán los alumnos de manera criteriosa sin olvidar la generación de un número de factura, del mismo modo la factura debe tener la fecha del día siguiendo los lineamientos tanto para los depósitos, retiros y transferencias que anteriormente se les detalló.

Luego de los datos propios de la factura deberán detallarse los datos propios a los movimientos o transacciones a facturar, se aclara que no se podrá facturar 2 veces el mismo movimiento.

Al momento de generar la factura se deben facturar el total de transacciones pendientes, teniendo en cuenta el costo de transacción puede variar en el tiempo.

11. Consulta de Saldo de Cuentas

Funcionalidad básica que permite a un cliente o administrador saber el saldo de una cuenta junto con un listado independiente entre sí de los últimos 5 depósitos, los últimos 5 retiros y las últimas 10 transferencias de fondos.

12. Listado Estadístico

Esta funcionalidad debe permitir a un administrador consultar el TOP 5 de:

1. Clientes que alguna de sus cuentas fueron inhabilitadas por no pagar los costos de transacción
2. Cliente con mayor cantidad de comisiones facturadas en todas sus cuentas
3. Clientes con mayor cantidad de transacciones realizadas entre cuentas propias
4. Países con mayor cantidad de movimientos tanto ingresos como egresos
5. Total facturado para los distintos tipos de cuentas.

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

Además el sistema nos pedirá que ingresemos obligatoriamente el año por el cual queremos consultar, luego nos pedirá el trimestre y finalmente debe permitir seleccionar que tipo de listado se quiere visualizar.

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

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 “GD1C2015”.

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: “gd2015”
 - Base de Datos Predeterminada: GD1C2015.
 - El resto de los parámetros respetar sus valores default.
- Solapa “Funciones del Servidor”:
 - Seleccionar “sysadmin”
- Solapa “Asignación de Usuarios”:
 - Seleccionar asignar a “GD1C2015”
- Para el resto de los parámetros respetar sus valores default.

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

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

- `gd_esquema.Schema.sql`: Este archivo genera un esquema llamado “gd_esquema” dentro de la base de datos y lo asigna al usuario “gd”.
- `gd_esquema.Maestra.Table.sql`: Este archivo crea la tabla principal del trabajo práctico y la carga con los datos correspondientes. El archivo posee un volumen significativo y no puede ser ejecutado desde el “Managment 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 gd2015 -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 “PagoElectronico”. 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: gd2015
- Nombre de la base de datos: GD1C2015

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# “PagoElectronico” 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:* 17/06/2015 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. Se considera que la fecha última de entrega sea el día 17/07/2015.

Sobre los grupos

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

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

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

Luego, el 18/05/2015 se enviarán los mail correspondientes con la confirmación de los grupos y se les asignará un número de grupo además del nombre que debidamente eligieron. Luego de esa fecha 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/1jtwu_QXmp3_VFSicM0BREYY2wM_4wtbM6_sJV_D2BICg/edit?usp=sharing

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

Donde aprender C#

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

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

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

Documentación de MSDN en español

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

Sobre la elección de C#

El lenguaje C, tradicionalmente usado en las cátedras de la facultad, ha demostrado tener cierta dificultad a 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:

TP1C2015<curso><nombreGrupo><Nro de grupo>

: espacio en blanco

Ejemplos:

TP1C2015 k9999 LOS_MEJORES 10

(Respetar los 2 espacios en blanco existentes)

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

Por cuestiones de seguridad Gmail rechaza todos los adjuntos que contengan archivos zip con .exe y .dll en su interior, por lo que es necesario renombrar la extensión .zip a .zip123.

Por ejemplo:

TP1C2015 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 haya cambiado de grupo, deberá ser aclarado en el mail de la entrega del TP.

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

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

Estructura del archivo zip

El archivo zip (.zip123) debe contener la siguiente estructura de directorios:

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

Readme.txt:

Es un archivo de texto plano con los siguientes datos:

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

Estrategia.pdf:

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

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

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 “XXXXXX”. 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:

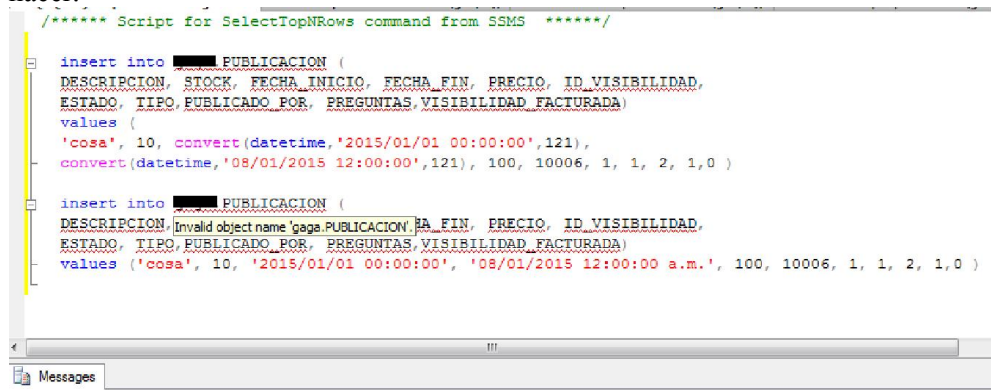
Archivo “script_creación_inicial.sql” con toda la creación del modelo de datos. El archivo debe poder ejecutar perfectamente de una sola vez, sin ningún tipo de error. Todas las

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.

10 Tips para desaprobatión de Gestión de Datos

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



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

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

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

Messages

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

8. Entregar el TP faltando menos de 5 minutos para la fecha límite saturando la casilla de entrega o pasada la fecha límite para la recepción para su corrección
9. Realizar una única entrega en una fecha cercana a la fecha límite sin la posibilidad de un feedback y si aún así deciden realizar una única entrega, deberán asumir el riesgo de Aprobación o Desaprobatión
10. Si hay dudas, volver a leer los puntos anteriores o consultarlos con los ayudantes.