

Universidad Tecnológica Nacional
Facultad Regional Buenos Aires

Gestión de Datos

Trabajo Práctico

1° Cuatrimestre 2016

Mercado-Envío

Enunciado V1.0

Introducción

Objetivos generales

El presente trabajo práctico persigue los siguientes objetivos generales

- Promover la investigación de técnicas de base de datos.
- Aplicar la teoría vista en la asignatura en una aplicación concreta.
- Desarrollar y probar distintos algoritmos sobre datos reales.
- Utilizar un lenguaje de programación de última generación.
- Fomentar la delegación y el trabajo en grupo.

Descripción general

Mediante este trabajo práctico se intenta simular la migración y remoción de un viejo sistema de compra on-line.

Dicho sistema deberá permitir la administración de clientes que venden y compran productos como así también empresas interesadas en publicar los artículos que comercializan.

El sistema se encargará de realizar cálculos de valoración a los vendedores en función de las calificaciones que otorgan los clientes, dicha información deberá estar visible al momento de seleccionar un producto de la lista de publicaciones.

En primera instancia se desarrollará un prototipo bajo una aplicación escritorio que permita conectarse en todo el país y una vez realizado el testeo de las principales funcionalidades que se presentan a continuación, se pensará pasar a un entorno web y aplicación Android.

Será necesario que las publicaciones manejen comisiones por publicación, unidades vendidas y costos de envío, dado que ahí radica la mayor parte de la ganancia para la empresa.

Debe tenerse en cuenta que el módulo de pago de artículos no estará presente en este trabajo práctico, pero si será necesario llevar registro de las diferentes facturaciones que se realicen, tanto para las comisiones de publicación, como de las comisiones por venta y envío.

Componentes del TP

El alumno recibirá dos componentes ya hechos del sistema y, en base a estos deberá crear uno nuevo e implementar nuevas funcionalidades. Los componentes a recibir son:

Base de Datos y Modelo de Datos

La cátedra provee un script que permite crear una base de datos en el motor SQL Server 2012. Esta base de datos incluye el modelo de una única tabla, llamada maestra, que es cargada con datos provistos por la cátedra. Los datos de esa tabla se encuentran desorganizados y no poseen ningún tipo de normalización.

El alumno deberá estudiar los datos recibidos y confeccionar un modelo de datos que siga todos los standards de desarrollo de bases de datos explicados durante la cursada.

Los datos de esta tabla maestra pertenecen a un dominio de compra y publicación de productos tecnológicos.

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

Parte de la lógica del negocio a resolver deberá ser inferida por el alumno, en base a las columnas y valores presentes en los datos. De todas maneras es recomendable consultar al grupo de Google de la materia antes de tomar decisiones incorrectas.

Aplicación Desktop

La cátedra provee un proyecto C# a modo de template, sobre el cual deberá desarrollarse una aplicación Desktop que interactúe con la nueva base de datos, cuyo diseño estará a cargo de los alumnos. La aplicación deberá ser del tipo Desktop desarrollada sobre C# con Visual Studio 2012 y Framework de .NET 3.5.

Esta aplicación tendrá diversas pantallas, reportes y formularios que permitirán interactuar, cargar y visualizar la información de la base de datos de SQL Server.

Requerimientos

General

El alumno deberá crear todos los componentes de base de datos e implementar todas las funcionalidades pedidas para la aplicación Desktop, cumpliendo con las siguientes pautas:

Base de Datos

El alumno deberá crear un modelo de datos que **organice y normalice** los datos de la única tabla provista por la cátedra. Este modelo de datos incluye:

- Creación de nuevas tablas y vistas.
- Creación de claves primarias y foráneas para relacionar estas tablas.
- Creación de constraints y triggers sobre estas tablas cuando fuese necesario.

- Creación de los índices para acceder a los datos de estas tablas de manera eficiente.
- Migración de datos: Cargar todas las tablas creadas utilizando la totalidad de los datos entregados por la cátedra en la única tabla del modelo. Para este punto deberán utilizarse Stored Procedures sobre la base de datos. No podrá realizarse la migración de datos utilizando la aplicación Desktop ni ninguna otra herramienta auxiliar.

El alumno deberá entregar un único archivo de Script que al ejecutar realice todos los pasos mencionados anteriormente, en el orden correcto. Todo el modelo de datos confeccionado por el alumno deberá ser creado y cargado correctamente ejecutando este Script una única vez, antes de empezar a testear la aplicación Desktop.

Todas las columnas creadas para las nuevas tablas **deberán respetar los mismos tipos de datos** de las columnas existentes en la tabla principal. A su vez el alumno podrá crear nuevas columnas, claves e identificadores para satisfacer sus necesidades. Pero nunca se podrá inventar información, por ejemplo crear un cliente que nunca existió.

Aplicación Desktop

El alumno deberá crear una aplicación Desktop en C# sobre Visual Studio 2012 con Framework .NET versión 3.5. Esta aplicación deberá contar con formularios, reportes y tablas. Todos estos componentes deberán respetar los lineamientos planteados en el documento “Guía de ABMs”. Es recomendable leerlo en este punto antes de continuar con el enunciado.

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

1. Login y seguridad
2. ABM de Rol
3. ABM de Usuarios (solo clientes y empresas)
4. ABM de Rubro
5. ABM visibilidad de publicación
6. Generar Publicación
7. Comprar/Ofertar
8. Historial del cliente.
9. Calificar al Vendedor
10. Consulta de facturas realizadas al vendedor.
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

Funcionalidad para poder crear, modificar y eliminar el acceso de un usuario a una opción del sistema.

Crear un rol implica cargar los siguientes datos:

- Nombre
- Listado de Funcionalidades (selección acotada)

Todos los datos mencionados anteriormente son obligatorios.

Un rol posee un conjunto de funcionalidades y las mismas no pueden estar repetidas dentro de un rol en particular, de más está decir que una funcionalidad puede estar en más de un rol.

Debe tenerse en cuenta que actualmente existen 3 roles:

- Empresa
- Administrativo
- Cliente.

En la modificación de un rol solo se pueden alterar ambos campos: el nombre y el listado de funcionalidades. Se deben poder quitar de a una las funcionalidades como así también agregar nuevas funcionalidades a rol que se está modificando.

La eliminación del rol implica una baja lógica del mismo. El rol debe poder inhabilitarse. No permitido la asignación de un rol inhabilitado a un usuario, por ende, se le debe quitar el rol inhabilitado a todos aquellos usuarios que lo posean.

Se debe poder volver a habilitar un rol inhabilitado desde la sección de modificación. Esto no implica recuperar las asignaciones que existían en un pasado.

Para elegir el rol que se desea modificar o eliminar se debe mostrar un listado con todos los roles existentes en el sistema.

2. Login y Seguridad

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

El proceso de Login pedirá al usuario su Username y su Password. Si Login es correcto, el usuario podrá acceder al sistema. Si el usuario tiene un solo rol asignado se ingresará al sistema directamente, pero si el mismo tiene más de uno, se deberá dar la posibilidad de seleccionar uno de ellos. Luego de validar el login, la aplicación solo deberá generar y mostrar las entradas de menú disponibles para este usuario según los roles del mismo. El usuario no debe ni siquiera ver las funcionalidades a las que no posee acceso.

Si el Login es incorrecto el usuario no podrá acceder al sistema. Se debe volver a mostrar el Login para que intente nuevamente. El sistema debe llevar un registro de cantidad intentos fallidos de login. Luego de 3 intentos fallidos en cualquier momento, el usuario debe ser inhabilitado. Al realizar un Login satisfactorio, el sistema deberá limpiar la cantidad de intentos fallidos.

El Login se considera una funcionalidad de características especiales. No se considera una funcionalidad que puede ser asignada a un rol. Todos los usuarios tienen la capacidad de utilizar el Login.

3. ABM de Usuario (sólo clientes y empresas)

Si bien el sistema cuenta con 3 tipos de usuarios, (por dicha razón solo existen 3 tipos de roles), este ABM solo se encargará de crear/registrar 2 tipos de usuarios, aquellos que sean clientes y los que sean empresas. Los usuarios administradores son fijos y serán informados por los alumnos, tener en cuenta cuales son aquellas tareas que le son propias a los administradores, por ej un administrador no podrá crear una publicación, como así tampoco realizar una compra o participar de una subasta, se encargará solamente de tareas de administración como ser, crear usuarios, roles, rubros y visibilidad, además de su respectiva modificación y eliminación.

Para simplificar la creación de los usuarios clientes y empresas, solo los usuarios administradores podrá crearlos, si bien en la realidad puede que exista otra mecánica, los tiempos para desarrollo del trabajo práctico hacen que simplifique esta operatoria.

Se deberá tener en cuenta que un cliente y una empresa no pueden ser la misma persona, siendo el primero una persona física, mientras que la segunda es una persona jurídica. Una vez que un usuario se registra en el sistema, este no podrá cambiar su tipo de usuario.

Crear un Usuario implica cargar los siguientes datos:

- Username
- Password
- Rol asignado (selección especial)
- Datos propios del tipo de usuario, que variarán según corresponda (cliente o empresa).

Todos los datos mencionados anteriormente son obligatorios.

El username debe ser único en todo el sistema. La aplicación deberá controlar esta restricción e informar debidamente al usuario.

El password deberá almacenarse encriptado de forma irreversible bajo el algoritmo de encriptación SHA256.

A un usuario se le puede asignar un único rol según corresponda a su tipo, además, se deberá tener en cuenta que se pueda modificar el password, tanto por el propio usuario como por el usuario administrativo.

Otro requisito importante es que los usuarios soporten baja lógica.

Una vez que se haya determinado el tipo de rol, esto nos habilitará a cargar los datos particulares para el tipo de usuario en cuestión, en caso de tratarse de un cliente, (usuario que puede comprar y vender productos dentro la plataforma) se deberán completar los siguientes datos:

- Nombre.
- Apellido.
- DNI.
- Tipo de documento
- Mail.
- Teléfono.
- Dirección calle, nro piso, depto y localidad
- Código Postal
- Fecha de Nacimiento.
- Fecha de Creación

Todos los datos mencionados anteriormente son modificables. Se deberá tener en cuenta, que un cliente solamente se puede registrar una vez en la plataforma de compra/venta, se debe verificar la no repetición tanto del tipo y número de documento.

Un cliente inhabilitado (baja lógica) no podrá comprar y/o vender en la plataforma bajo ninguna forma. Se debe poder volver a habilitar el cliente deshabilitado desde la sección de modificación.

En el caso de que se necesite modificar o eliminar (baja lógica) debe presentar un buscador (con datos específicos para el cliente) con un listado que permita filtrar simultáneamente por alguno o todos los siguientes campos:

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

Cuando el tipo de rol seleccionado haga referencia a una empresa, se deberán cargar los datos específicos a este tipo de usuario. Dichas empresas solo tienen la posibilidad de vender artículos electrónicos en la plataforma, denegándoseles toda posibilidad de compra.

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

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

Todos los datos mencionados anteriormente son modificables. La razón social y cuit son datos únicos, por ende, no pueden existir 2 empresas con la misma razón social y cuit. El sistema deberá controlar esta restricción e informar debidamente al administrativo ante alguna anomalía.

La eliminación de una empresa implica la baja lógica de la misma. Una empresa inhabilitada no podrá armar ofertas, realizar publicaciones ni ninguna otra operación posible en el sistema. Además de lo anteriormente expuesto, se recuerda que se debe poder volver a habilitar a un vendedor deshabilitado desde la sección de modificación.

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

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

Aclaración IMPORTANTE, que existan 2 tipos de usuarios, no significa que deban modelarse todo en una misma entidad. Tener en cuenta cuales son aquellos datos que se comparten y cuáles no, que diferencia a un usuario de otro tipo de usuario. Apelamos al uso de buenas prácticas y el hecho de no cumplir con esta premisa será razón de desaprobación del TP.

4. ABM de Rubro.

Funcionalidad que permite categorizar a una publicación y determinar dentro de que rubro/s se puede encontrar a dicha publicación.

Dichos rubros se encuentran en forma plana y no responden a una organización en forma de árbol para su simplificación.

Los rubros deben respetar y estar conformados por un código, una descripción

corta y otra descripción larga

Para reducir el tiempo de desarrollo de los alumnos, no será necesario que realicen la implementación de dicho ABM.

5. ABM visibilidad de publicación

Este tipo de funcionalidad es la que le permite al usuario determinar el grado de visibilidad para sus publicaciones.

Este grado de visibilidad se conforma principalmente por:

- Comisión por tipo de publicación.
- Comisión por producto vendido.
- Comisión por envío del producto, funcionalidad nueva que los alumnos determinarán como aplicarlo, pudiendo ser un valor fijo o variable.

Existe un tipo de visibilidad que es gratuita para el vendedor, en donde no se generan gastos de comisión por publicación y además no posee la modalidad de envío.

Para las restantes publicaciones independientemente de cual sea su visibilidad, se debe elegir si dicha publicación brindará o no el servicio de envío sobre el producto comprado.

Para promover el uso de la plataforma todo usuario nuevo en la plataforma (aquellos agregados post-migración) tendrán el beneficio de tener por única vez una publicación sin costo de comisión de publicación (aplicable para aquellas que no sean gratuitas), la cual corresponderá a la primera publicación que registren en la plataforma. Esta característica no es aplicable a los datos migrados.

6. Generar Publicación

Esta funcionalidad es la de mayor importancia en toda la plataforma dado que es núcleo de la misma como fuente de ingresos. Existen 2 tipos de publicaciones:

- Publicación para una compra inmediata
- Publicación para una subasta.

Ambas publicaciones se caracterizan por compartir una serie de estados. Los mismos se detallan

- **Borrador:**

Este estado permite al vendedor puede modificar todos los valores que sean necesarios, ya que dicha publicación todavía no está visible para la comunidad.

- **Activa:**

Este estado es aplicado por el vendedor para que dicha publicación sea vista por toda la comunidad y disponible para operar con ella. Una vez que una publicación esta activa no puede pasar a estado borrador, pero si puede ser pausada o finalizada y deberá generarse en ese mismo instante la facturación (informado en pantalla todos los datos que forman parte de una factura, maestro-detalle y aquellos que sean necesarios para su posterior identificación, fecha, monto, a quien está dirigida etc.) correspondiente de acuerdo al tipo de visibilidad que se le haya asignado. Tener en cuenta que solo deberá generarse la facturación dado que el pago de facturas no compete a este trabajo práctico

- **Pausada:**

Este estado es utilizado para que dicha publicación este visible al público pero que no puedan realizarse operaciones de compra u oferta según sea el caso, también se

deberá tener en cuenta que no se podrán realizar modificaciones en la publicación. En otras palabras, la publicación queda congelada hasta que la misma se vuelva a activar nuevamente

- **Finalizada:**

Estado utilizado por el vendedor para indicar que una publicación llegó a su fin. Una vez que una publicación está finalizada no puede cambiarse el estado de la misma a ningún otro. Cuando se trate de una publicación de compra inmediata, este estado se aplicará automáticamente cuando se venda la última unidad del producto publicado o cuando el vendedor decida dejar sin efecto su publicación aun cuando esta siga teniendo stock disponible del producto publicado. Para el caso de las subastas, el estado finalizado se aplicará al día siguiente de su vencimiento, ósea, que si el vencimiento era el 01/01/2016, se podrán realizar ofertas hasta ese mismo día (no tener en cuenta la hora), y dicha subasta caduca el 02/01/2016(no tener en cuenta la hora). Los alumnos en este caso determinarán cómo será el proceso para el cambio de estado. Lo más simple sería que una vez que se inicia la aplicación barrer la base de datos si hay alguna subasta vencida. Recordar los fines prácticos del trabajo y realizar este requerimiento de la manera más sencilla. Comentar en la estrategia cual fue el criterio utilizado.

Para el manejo de fechas es indispensable utilizar el archivo de configuración en donde se seteará la fecha del sistema, no trabajar con la fecha system de la base de datos, como así tampoco utilizar la del sistema operativo. Si no se cumple con este requisito el TP quedará automáticamente desaprobado.

El proceso para el cambio de estado por fecha de vencimiento debe ser desarrollado en la aplicación y no en la base de datos.

Las subastas tienen por características que solo se ofrecen en una sola unidad, pudiendo esta ser un lote de X cantidad de unidades pero que son ofrecidas como un único lote por ej. 20 unidades de Remeras donde la subasta es por las 20 unidades, y no por venta por separado en unidades.

En cambio, las publicaciones de compra inmediata pueden ser ofrecidas por unidad y un cliente puede comprar una o N unidades siempre y cuando exista el stock solicitado.

Al momento de generar una subasta el vendedor determinará el valor de inicio de la misma, dicho valor inicial es aquel que satisface al vendedor generando esto un compromiso de venta del producto si solamente recibe una oferta por dicha subasta. No se contempla que una subasta deba llegar a un valor X por arriba del inicial para que la venta sea concretada.

Ambas publicaciones tienen una fecha de inicio y una fecha de finalización, la cual será ingresada por el usuario. Puede que sea automática o no, o variable según sea la publicación, esta decisión quedará a cargo de los alumnos, los cuales deberá informar al momento de la entrega del TP.

Una publicación tiene que tener mínimamente los siguientes datos:

- Código de Publicación (auto-numérico y consecutivo entre publicaciones sean o no del mismo vendedor)
- Descripción
- Stock
- Fecha de Vencimiento
- Fecha de Inicio
- Precio
- Costo

- Rubro
- Visibilidad
- Usuario responsable de la publicación
- Estado de la publicación
- Tipo de Publicación
- Permitir o no la realización de preguntas

Tener en cuenta que el envío puede estar restringido según sea el tipo de publicación y aun así y todo en aquellas que lo permiten debe poder ofrecerse o no su utilización. En caso que la publicación acepte el envío y se opte por dicho servicio, al momento de hacer la compra no podrá omitir su uso, debiendo ser facturado

7. Comprar/Ofertar

Esta funcionalidad permite comprar u ofertar una publicación según sea el caso.

Se debe tener en cuenta que el envío o entrega de un producto queda a cargo del vendedor y no es motivo de evaluación. Además un vendedor no puede auto-comprarse u auto-ofertarse.

Las compras u ofertas son en pesos Argentinos o contra reembolso, no existe pago electrónico en esta etapa para esta funcionalidad en particular y la facturación del mismo queda a cargo del vendedor. La aplicación solo factura comisiones por publicaciones o cantidades vendidas.

Las publicaciones deben poder ser filtradas por:

- Uno o más rubros
- Descripción

Dichas publicaciones se deben presentar ordenadas en función de la visibilidad, primero se mostraran las que tengan más peso y de ahí hasta llegar a las gratuitas que son las ultimas en visualizarse.

Se debe tener en cuenta que las publicaciones deben ser cargadas en una grilla y las mismas deben estar paginadas (cantidad a determinar por el grupo), se debe poder adelantar y retroceder como así también volver a la primera hoja de publicaciones o ir a la última hoja.

Cuando se realiza una oferta por una subasta, solo se permiten valores que superen al valor actual y a medida que este se incrementa, el valor del producto se va actualizando en función de las ofertas recibidas. Siempre se debe ofertar en unidades enteras.

Cuando se concreta una venta sea por compra directa o por subasta, deberá generarse la facturación correspondiente al vendedor, en donde se aplican las comisiones por unidad vendida y el cobro del envío si es que este aplica en la venta. Cuando se trate de una subasta, no será necesario mostrar la factura en pantalla ya que simularemos la finalización de las mismas con el proceso que se detalló en el punto 6, estado finalizado. Cuando se trate de una compra directa, si habrá que mostrar la factura en pantalla, aplicando los mismos criterios que en la facturación de comisiones de publicación.

8. Historial de Cliente

Esta funcionalidad permite a un cliente conocer todo su historial de compras y subastas (ya sean adjudicadas o no). Debe presentarse en una grilla paginada igual a la utilizada para la visualización de las publicaciones. Se debe mostrar información acerca

de cuantas operaciones faltan por calificar, resumen de estrellas dadas y toda aquella información que se crea pertinente y que forme parte del historial de cliente.

9. Calificar Al Vendedor

Esta funcionalidad, como su nombre lo indica, permite que un cliente califique a un vendedor por el producto que este último le vendió. Este sistema de calificación solo es unidireccional, donde únicamente el cliente puede calificar.

Un vendedor va sumando calificaciones (estrellas de 1 a 5) a lo largo de toda su historia y dicha reputación se va actualizando a medida que va recibiendo calificaciones. Cada estrella obtenida tiene un peso determinado (estipulado por el grupo) la sumatoria de dichos pesos promediado con la cantidad de compras y subastas concretadas, nos da la reputación del vendedor.

Dichas calificaciones pueden ir acompañadas por el breve detalle, que puede ser dentro de un rango definido de opciones o un texto libre.

Se debe tener en cuenta que para poder calificar una subasta, la misma debe estar finalizada, dado que el vendedor se comunica directamente (por fuera del sistema) con el cliente, notificándole que se le ha adjudicado la subasta en cuestión, por ende solo tendrá la posibilidad de calificar una subasta el cliente que la haya ganado. Si un cliente ofertó en una subasta y no la ganó, este no podrá calificar. No es viable que una compra o subasta sea calificada más de una vez.

Dicha funcionalidad deberá, además, mostrar un historial de las últimas 5 compras o subastas calificadas. No se debe dejar por alto informar, cuantas compras ha realizado el cliente junto con un desglose por estrellas utilizadas, ósea, cuantas compras fueron calificadas con una estrella, cuantas con dos, etc. La misma información debe detallarse para las subastas.

Si el cliente posee más de 3 compras inmediatas o subastas sin calificar, el sistema no le permitirá realizar ninguna otra operación de compra u oferta hasta que no califique todo lo que tiene pendiente.

10. Consulta de Facturas realizadas al vendedor.

Esta funcionalidad permite listar todas las facturas realizadas a un usuario vendedor, ya sean por comisión de publicación como ventas y envíos.

Será necesario que se implementen una serie de filtros, como ser:

- intervalo de fechas,
- intervalos de importe
- buscador por contenido de detalles de factura
- A quien está dirigida.

Dichos resultados deberán listarse en una grilla, la cual deberá estar paginada.

11. Listado Estadístico

Esta funcionalidad nos debe permitir consultar el TOP 5 de:

- *Vendedores con mayor cantidad de productos no vendidos*, dicho listado debe filtrarse por grado de visibilidad de la publicación y por mes-año. Primero se deberá ordenar por fecha y luego por visibilidad.
- *Cientes con mayor cantidad de productos comprados*, por mes y por año, dentro de un rubro particular
- *Vendedores con mayor cantidad de facturas* dentro de un mes y año particular
- *Vendedores con mayor monto facturado* dentro de un mes y año particular.

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 2012 con las siguientes consideraciones:

- El nombre de la instancia del motor de base de datos a instalar debe llamarse “SQLSERVER2012”. No utilizar el nombre “Default” para la instancia. Instalar como instancia con nombre (“Named Instance”).
- La autenticación debe ser por “Modo Mixto”.
- El usuario administrador de la base de datos deberá tener la siguiente configuración:

- Username: “sa”
- Password: “gestiondedatos”

Una vez instalado el motor de base de datos se deberán instalar las herramientas cliente de trabajo: “Microsoft SQL Server Management Studio Express” para SQL Server 2012. Ejecutar esta aplicación e ingresar los datos del usuario “sa” creado anteriormente (en modo “Autenticación de SQL Server”).

Dentro del “Management Studio” crear una nueva base de datos con los parámetros default y nombre de base “GD1C2016”.

Crear un nuevo “Inicio de Sesión”, desde el item “Seguridad” perteneciente al servidor de Base de Datos general. El inicio de sesión debe poseer las siguientes características:

- Solapa “General”:
- Nombre de inicio de sesión: “gd”
- Autenticación de SQL Server
- Contraseña: “gd2016”
- Base de Datos Predeterminada: GD1C2016.
- El resto de los parámetros respetar sus valores default.
- Solapa “Funciones del Servidor”:
- Seleccionar “sysadmin”
- Solapa “Asignación de Usuarios”:
- Seleccionar asignar a “GD1C2016”
- 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: “gd2016”.

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\SQLSERVER2012 -U gd -P gd2016 -i gd_esquema.Schema.sql,gd_esquema.Maestra.Table.sql -a 32767 -o resultado_output.txt

Luego de cargados todos los datos de la tabla maestra, el alumno deberá crear su propio esquema dentro de la base de datos. El nombre del esquema deberá ser igual al nombre del grupo registrado en la materia (el proceso de registración se explica más

adelante). El nombre del esquema debe ser en mayúsculas, sin espacios y separado por guiones bajos. Ejemplo “Los mejores” debe ser “LOS_MEJORES”.

Todas las tablas, stored procedures, vistas, triggers y otros objetos de base de datos nuevos que cree el alumno deberán pertenecer a este esquema creado. Sin la solución entregada posee objetos de base de datos por fuera del esquema con el nombre del grupo, el TP será rechazado sin evaluar su funcionalidad.

Con esta configuración el alumno está listo para empezar la implementación de la parte de base de datos.

Aplicación Desktop

La cátedra provee una aplicación Desktop en C#, a modo de template, sobre la cual se debe desarrollar la aplicación del Trabajo Práctico.

Para ejecutar esta aplicación es necesario instalar Visual Studio 2012 con el Framework de .NET 3.5. La versión Express posee la funcionalidad necesaria como para desarrollar el Trabajo Práctico.

La aplicación template se denomina “MercadoEnvioDesktop”. Cuenta con un formulario principal, una barra de menú y un formulario para cada funcionalidad visual que hay que implementar en el trabajo. El alumno debe depositar su código respetando esta estructura.

Más allá de estas indicaciones, el alumno puede modificar a su criterio la aplicación template. Ante cualquier consulta sobre lo que se puede modificar consultar al grupo de Google de la materia.

La aplicación Desktop deberá conectarse a la base de datos con los siguientes parámetros:

- Origen de datos: Microsoft SQL Server (SqlClient)
- localhost\SQLSERVER2012
- Utilizar autenticación de SQL Server:
 - Nombre de Usuario: gd
 - Password: gd2016
- Nombre de la base de datos: GD1C2016

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 2012 o Microsoft Visual C# Studio Express 2012.

El motor de base de datos deberá ser Microsoft SQL Server 2012. Tanto la versión Express como la full sirven para realizar el trabajo.

No podrá utilizarse ninguna herramienta auxiliar que ayude a realizar la migración de datos. Tampoco podrá desarrollarse una aplicación personalizada para la migración de datos. La misma deberá ser efectuada en código T-SQL en el archivo de script “script_creacion_inicial.sql”.

Condiciones de aprobación

Testing

El alumno deberá entregar dos componentes:

- Un único script de base de datos (script_creacion_inicial.sql) con todo lo necesario para crear su modelo y cargarlo con datos.
- La aplicación C# “MercadoEnvioDesktop” 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

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

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

- **Entrega obligatoria del DER:**

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

Solo se aceptarán entregas dentro de la fecha límite específicamente estipulada (15/05/2016) pasada esa fecha no se aceptarán entregas bajo ninguna circunstancia.

No habrá excepciones por entrega fuera de término ni motivos que lo justifiquen, ya que la entrega del TP es grupal y esto da la posibilidad de que la responsabilidad sea de todos los integrantes del grupo para llegar en fecha.

En caso de que el DER no sea correcto, los errores serán informados en la corrección y deberán ser resueltos para la entrega del sistema completo.

La motivación de esta entrega es la corrección de los grandes errores en el modelado de la base de datos antes de la construcción del sistema final, para evitar que en la siguiente entrega se encuentren con la necesidad de modificar gran cantidad de

código por errores del modelado de la base de datos que pueden ser atacados previamente.

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

- **Entrega del sistema completo**

Día: 15/06/2016 hasta las 12:00hs del mediodía (GMT -3:00 Buenos Aires).

Los TPs entregados luego de las 12:00hs, se consideraran fuera de término perdiendo así la posibilidad de entrega y restando solamente 2 instancias de presentación. Estas 2 instancias no tienen fecha asignada y serán determinadas por el equipo.

Una vez entregado el TP, el periodo de corrección es aproximadamente de 7 días.

Este factor puede variar dependiendo de la cantidad de TPs entregados en ese momento. Por lo cual, se recomienda tenerlo en cuenta para la fecha final de entrega del trabajo.

La última fecha para recepción de TP es el día 15/07/2016

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=dG16aEltMHc1X2hPN3U2YTVoVGxfUE6MA>

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 19/05/2016 se enviarán los mails 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/1nmG-KcRGhnaPXurvYhvBw_jM0_NINadTHcq5aI85n14/edit

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

Donde aprender C#

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

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

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

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

Sobre la elección de C#

El lenguaje C, tradicionalmente usado en las cátedras de la facultad, ha demostrado tener cierta dificultad a la hora de su uso. El uso de punteros y procesamiento de cadenas muchas veces resulta complicado, sin mencionar la dificultad de encontrar un error en tiempo de ejecución. También hemos notado que se invierte mucho tiempo tratando de crear interfaces amigables mediante consola de texto.

Es por esto que creemos que C# al igual que otros lenguajes de última generación, como Java, permiten invertir más tiempo en cuestiones algorítmicas y de estructura de datos, dejando de lado las cuestiones de sintaxis propias del lenguaje C.

Por nombrar algunas ventajas de C#:

- Facilidad en la depuración en tiempo de ejecución: Nos permite inspeccionar el valor de las variables durante la ejecución del programa, incluso visualizar estructuras de datos recursivas.
- Las sintaxis está totalmente normalizada.
- El IDE permite autocompletar código.
- Provee métodos simples para el manejo de E/S.

Podemos decir que gran parte de la eficiencia de un programa depende no del lenguaje en el cual es implementado, sino de las estructuras de datos y algoritmos elegidos para resolverlo.

Por último consideramos que el paradigma orientado a objetos puede brindarnos muchas ventajas que a esta altura ya resultan evidentes y al mismo tiempo permite aplicar los mismos esquemas algorítmicos que los lenguajes estructurados.

Obtención de herramientas

El TP puede ser desarrollado con dos versiones del IDE Microsoft Visual Studio 2012. No puede utilizarse la versión 2010:

1. **Microsoft Visual Studio Professional 2012:** esta versión puede ser obtenida con licencia universitaria completa, gracias a un convenio de Microsoft con la UTN. Dirigirse al laboratorio de Microsoft ubicado en la sede Medrano (planta baja, hacia la derecha, mirando desde la entrada de la facultad hacia adentro).
La versión ofrecida es en español y contiene la ayuda completa (MSDN). Para encargarla es necesario llevar un DVD y completar un formulario. Es posible que el programa halla que encargarlo y pasarlo a buscar otro día.
2. **Microsoft Visual C# 2012 Express Edition:** Existe una versión gratuita del IDE llamada *Visual C# 2012 Express Edition*, la cual posee todas las herramientas necesarias para realizar el TP. Esta se encuentra disponible en:

<http://www.microsoft.com/express/Downloads/>

El motor de base de datos a utilizar es SQL Server 2012 Express. Puede ser descargado de la siguiente dirección:

<http://www.microsoft.com/Sqlserver/2012/en/us/express-down.aspx>

Es necesario descargar e instalar dos componentes:

- Install Microsoft SQL Server 2012 Express Edition
- SQL Server Management Studio Express

Formato de entrega

Lugar de envío

La entrega debe realizarse por mail el antes de las fechas estipuladas en el documento de enunciado

La dirección del mail es:

gestiondedatos.entregas@gmail.com

El asunto del mail debe cumplir con el siguiente formato:

TP1C2016<curso><nombreGrupo><Nro de grupo>

: espacio en blanco

Ejemplos:

TP1C2016 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:

TP1C2016 k9999 LOS_MEJORES 10.zip123

(Respetar los 2 espacios en blanco existentes)

No enviar adjuntos de más de 20 MB. La casilla de mail rechazará mails que superen esta restricción.

El cuerpo del mail debe contener lo siguiente:

Grupo:

Curso:

Integrantes: <apellido>, <nombres> - <legajo>

Nota: En caso de que haya integrantes de cursos distintos, se debe poner el curso de la persona elegida como representante

En caso de que algún alumno del grupo haya dejado de cursar o se halla cambiado de grupo, deberá ser aclarado en el mail de la entrega del TP.

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

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

Estructura del archivo zip

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

- ⇒ \
- ⇒ Readme.txt
- ⇒ Estrategia.pdf
- ⇒ \src
- ⇒ Solución entera de Visual Studio de “VentaEnvioDesktop”
- ⇒ \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:

A Archivo PDF en donde se deberá explicar en forma detallada y extensa la estrategia utilizada para desarrollar el TP. Debe incluir una descripción de todas las estructuras de datos relevantes utilizadas en el algoritmo, explicando la razón de la elección de dichas estructuras.

Cualquier consideración tomada o asumida deberá ser aclarada en este documento.

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

El archivo de estrategia deberá ser en formato PDF obligatoriamente, con carátula e índice. En caso de no cumplir esta condición, el TP será rechazado sin evaluar su funcionalidad.

Sin este archivo y un DER de la base la entrega no será tomada como válida.

\src:

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

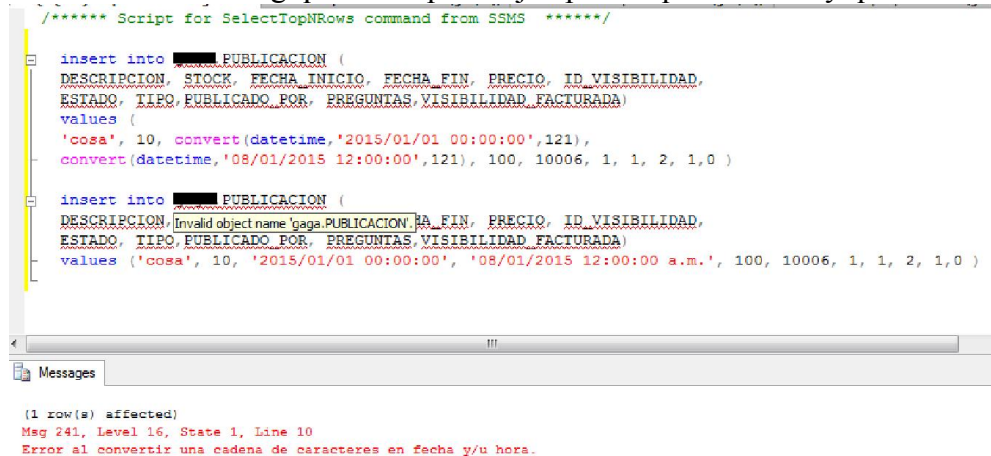
\data:

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 desaprobar Gestión de Datos

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



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

insert into [PUB] PUBLICATION (
  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 [PUB] PUBLICATION (
  DESCRIPCION, STOCK, FECHA_INICIO, 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. El tp es grupal y no individual, un integrante puede tener problemas pero los 4 al mismo tiempo no.
9. Realizar una única entrega en una fecha cercana a la fecha límite sin la posibilidad de un feedback y si aún así deciden realizar una única entrega, deberán asumir el riesgo de Aprobación o Desaprobación
10. Si hay dudas, volver a leer los puntos anteriores o consultarlos con los ayudantes.