



Universidad Tecnológica Nacional  
Facultad Regional Buenos Aires

# Gestión de Datos

Trabajo Práctico

1° Cuatrimestre 2014

FRBA-Commerce

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 .....	5
Aplicación Desktop .....	5
General.....	16
Base de Datos .....	16
Aplicación Desktop .....	17
Restricciones de la solución .....	19
Condiciones de aprobación.....	19
Testing .....	19
Modelo de Datos.....	20
Consultas SQL.....	20
Respetar Guía de ABMs .....	20
Aplicación Desktop .....	20
Fecha de entrega y condiciones .....	20
Sobre los grupos .....	21
Ayuda y contacto .....	21
Donde aprender C#.....	22
Sobre la elección de C# .....	22
Obtención de herramientas .....	23
Formato de entrega .....	24
Lugar de envío .....	24
Estructura del archivo zip .....	25
Readme.txt:.....	25
Estrategia.pdf:.....	25
\src: .....	25
\data .....	26

## **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 sus productos.

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 desarrollara un prototipo bajo una aplicación escritorio que permita conectarse en todo el país, y luego de testadas estas principales funcionalidades que se presentan a continuación se pensará pasar a un entorno web y aplicación Android.

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

Estas comisiones deberán ser rendidas por el vendedor, el sistema deberá generar las facturas correspondientes y registrar el pago de las mismas.

## **Componentes del TP**

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

### **Base de Datos y Modelo de Datos**

La cátedra provee un script que permite crear una base de datos en el motor SQL Server 2008. Esta base de datos incluye el modelo de una única tabla, llamada maestra, que es cargada con datos provistos por la cátedra. Los datos de esa tabla se encuentran desorganizados y no poseen ningún tipo de normalización.

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

Los datos de la tabla maestra contienen las operaciones realizadas por el sistema anterior es decir: publicaciones, compras, calificaciones, etc

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

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

### **Aplicación Desktop**

La cátedra provee un proyecto C# a modo de template, sobre el cual deberá desarrollarse una aplicación Desktop que interactúe con la nueva base de datos, cuyo diseño estará a cargo de los alumnos. La aplicación deberá ser del tipo Desktop desarrollada sobre C# con Visual Studio 2008 y Framework de .NET 3.5.

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

## **Requerimientos**

### **General**

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

## Base de Datos

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

- Creación de nuevas tablas y vistas.
- Creación de claves primarias y foráneas para relacionar estas tablas.
- Creación de constraints y triggers sobre estas tablas cuando fuese necesario.
- Creación de los índices para acceder a los datos de estas tablas de manera eficiente.
- Migración de datos: Cargar todas las tablas creadas utilizando la totalidad de los datos entregados por la cátedra en la única tabla del modelo. Para este punto 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.

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

## Aplicación Desktop

El alumno deberá crear una aplicación Desktop en C# sobre Visual Studio 2008 con Framework .NET versión 3.5. Esta aplicación deberá contar con formularios, reportes y tablas. Todos estos componentes deberán respetar los lineamientos planteados en el documento “Guía de ABMs”. Es recomendable leerlo en este punto antes de continuar con el enunciado.

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

1. Login y seguridad
2. ABM de Rol
3. Registro de Usuario
4. ABM de Cliente (comprador/vendedor)
5. ABM de Empresa (Vendedor)
6. ABM de Rubro
7. ABM visibilidad de publicación
8. Generar Publicación
9. Editar Publicación
10. Gestión de Preguntas
11. Comprar/Ofertar
12. Historial del cliente.
13. Calificar al Vendedor
14. Facturar Publicaciones
15. Listado Estadístico

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

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

## 1. ABM de Rol

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

Crear un rol implica cargar los siguientes datos:

- Nombre
- Listado de Funcionalidades (selección acotada)

Todos los datos mencionados anteriormente son obligatorios.

Un rol posee un conjunto de funcionalidades y las mismas no pueden estar repetidas dentro de un rol en particular, 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 permitiendo 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. Registro de Usuario.

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

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 la persona registra en el sistema, este no puede cambiar su tipo de usuario.

Crear un Usuario implica cargar los siguientes datos:

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

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 solo rol según corresponda a su tipo. Debe tenerse en cuenta que se pueda modificar el password. Tanto por el propio usuario como por el usuario administrativo.

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

### 4. ABM de Clientes

Funcionalidad que permite a un administrativo crear, modificar y dar de baja un cliente del sistema asignándoles un nombre de usuario por defecto.

Los clientes son usuarios que pueden comprar y vender productos dentro de la plataforma.

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

- Nombre.
- Apellido.
- DNI.
- Tipo de documento

- Mail.
- Teléfono.
- Dirección calle, nro piso, depto y localidad
- Código Postal
- Fecha de Nacimiento.

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

Se deberá tener en cuenta que cada cliente debe ser registrado una única vez, en la plataforma, dicha validación se realizará a través del control del tipo y nro de documento registrado en el sistema.

Un cliente inhabilitado 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 caso de que se cree un cliente nuevo, el sistema deberá designar automáticamente el username y el password para dicho cliente. Luego, cuando el usuario ingrese por primera vez al sistema con sus datos, proporcionados por el administrador, el sistema le solicitará inmediatamente que modifique su password autogenerada por el sistema ya que si no lo hace no podrá volver a loguearse dado que la misma tiene un tiempo de vida igual a un ingreso correcto.

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

- Nombre (texto libre)
- Apellido (texto libre)
- Tipo de documento(selección acotada)
- Nro. de documento(texto libre solo documentos)
- Email (texto libre)

## 5. ABM de Empresa

Funcionalidad que permite a un administrativo crear, modificar y dar de baja una empresa de la plataforma. Dicha empresa solo tiene la posibilidad de vender artículos y/o servicios en la plataforma, denegándoseles toda posibilidad de compra.

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

- Razón Social.
- Mail.
- Teléfono.
- Dirección calle, nro piso, depto y localidad
- Código Postal
- Ciudad
- CUIT
- Nombre de Contacto



- Fecha de Creación

Todos los datos mencionados anteriormente son modificables. La razón social y el CUIT son datos únicos, por ende no pueden existir 2 empresas con la misma razón social y cuil. Cabe aclarar que tanto el cuil como la razón social no debe repetirse en ningún momento. El sistema deberá controlar esta restricción e informar debidamente al administrativo ante alguna anomalía.

Se debe tener en cuenta que también para este tipo de usuario, se debe validar el CUIT como el CUIL para los clientes, a su vez corre la misma premisa que para los clientes, si se crea ingresa una nueva empresa al sistema, el mismo debe autogenerar su username y su password, exigiendo que se modifique esta última luego del primer ingreso correcto.

La eliminación de una empresa implica la baja lógica de la misma. Una empresa inhabilitada no podrá realizar ninguna operación en el sistema. Se debe poder volver a habilitar a un proveedor deshabilitado desde la sección de modificación.

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

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

## 6. ABM de Rubro.

Funcionalidad que permite a un administrativo crear, modificar y dar de baja un rubro del sistema. Los rubros son las categorías las cuales se podrán asignar a una publicación en el caso de crearla o modificarla.

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

- Código.
- Descripción.

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

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

## 7. ABM visibilidad de publicación

Funcionalidad que permite a un administrativo crear, modificar y dar de baja un "Visibilidad" de una publicación en el sistema. La visibilidad de una publicación es la forma de catalogar la forma en la cual se va a mostrar la publicación, por ejemplo una publicación con visibilidad "Platino" garantiza que en la mayoría de las búsquedas esta publicación aparezca en los primeros lugares en las búsquedas.

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

- Código.
- Descripción.

- Precio por publicar.
- Porcentaje de la venta (es el porcentaje que se aplica sobre el precio de cada unidad vendida de la publicación)

Existe un tipo de visibilidad que es gratuita y no genera gastos de comisión por publicación al vendedor. Se debe tener en cuenta que un vendedor no puede tener más 3 publicaciones activas simultáneamente en forma gratuita desde el momento que se pone en marcha esta nueva versión de la plataforma. Esta característica no es aplicable a los datos migrados.

Todos los tipos de publicaciones que tengan comisión de publicación tienen una gratificación en función de la cantidad que el vendedor haya generado. Al momento de la rendición de comisiones se deberá tener en cuenta que si la misma es la número 10 dentro del mismo tipo, esta no genera costo alguno y es bonificada. Una vez bonificada dicha comisión, el “contador” se resetea y debe rendir 9 comisiones más del mismo tipo para que la décima sea bonificada nuevamente.

## 8. Generar Publicación

El caso de uso "Generar Publicación" es el encargado de diseñar la publicación que le permita al vendedor/Cliente que desee vender su producto en la plataforma.

Existen dos tipos de publicaciones que se pueden generar, que son las siguientes:

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

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. Esto implica que la comisión por venta es por el total del lote.

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 determina el valor de inicio de la misma. Dicho valor inicial es aquel que satisface al vendedor en primera instancia 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.

Ambos tipos de 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, **solo**

**permite incremento en la cantidad de stock y descripción del producto ofrecido.** Una vez que una publicación esta activa no puede pasar a estado borrador.

- **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.

- **Finalizada:**

Estado utilizado por el vendedor para indicar que una publicación llegó a su fin. A fines didácticos, el vendedor se encargará de modificar dicho estado al día siguiente de finalizada la publicación (ya sea que este activa o pausada), evitando al alumno desarrollar procedimientos automáticos de cambios de estado y complicaciones con las fechas. El único cambio de estado automático permitido es cuando en una publicación del tipo compra inmediata se han vendido todas las unidades del producto. Una vez que una publicación está finalizada no puede cambiarse el estado de la misma a ningún otro.

Ambas publicaciones tienen una fecha de inicio y una fecha de finalización, esta última es determinada en forma automática por el tipo de visibilidad que se le otorga a la publicación.

Una publicación tiene que tener mínimamente los siguientes datos:

- Código de Publicación (consecutivo entre publicaciones sean o no del mismo vendedor)
- Descripción.
- Stock.
- Fecha de Vencimiento.
- Fecha de Inicio.
- Precio.
- Rubros: los rubros a los cuales pertenece la publicación.
- Visibilidad.
- Usuario que generó la publicación, registrado por el sistema.
- Estado de la publicación.
- Tipo de Publicación.
- Permitir o no la realización de preguntas.

## 9. Editar Publicación

Funcionalidad que permite la modificación de una publicación previamente registrada en la plataforma.

En función del estado de la publicación, el sistema debe permitir las siguientes operaciones:

- Borrada: en caso que la publicación se encuentre en este estado el sistema permitirá modificar la publicación en forma total, es decir todos los campos que definen la misma.

- Publicada: el sistema solo permitirá modificar el estado de la misma, a pausada, en cuyo caso no se podrá comprar/ofertar sobre la misma, en caso de modificar el estado a finalizada el sistema no permitirá ninguna posterior modificación. En caso de tratarse de una subasta el estado de la misma no podrá ser modificada luego de ingresar en el estado de publicada.

También permite la modificación del stock, únicamente cuando se trate de una publicación de compra inmediata, y esta modificación debe realizarse en forma incremental solamente.

## 10. Gestión de Preguntas

Funcionalidad que permite la gestión de las preguntas realizadas sobre las publicación que el usuario publicó con anterioridad.

Se divide en dos sub-funcionalidades:

- Responder preguntas:  
El sistema mostrará un listado con todas las preguntas de las publicación que el mismo haya publicado, y que están esperando a ser respondidas. Luego el usuario seleccionará la pregunta a responder y el sistema registrará la fecha de la respuesta y la respuesta (campo de texto) a enviar.
- Ver Respuestas:  
El sistema mostrará una lista de las respuestas (en caso de que hayan sido efectivamente respondidas) con sus respectivas respuestas y mostrará información de la publicación a la cual hace referencia la pregunta realizada.

Cada pregunta de una publicación debe tener solo y solo una respuesta.

## 11. Comprar/Ofertar

Esta funcionalidad permite comprar u ofertar una publicación según el tipo de la misma.

El sistema mostrará una lista de todas las publicaciones que se encuentren activas, todavía no vencidas, en el caso de compra inmediata que todavía tengan stock, dicha lista se podrá filtrar por los siguientes campos:

- Rubro/rubros.
- Descripción de la publicación.

Las publicaciones se mostrarán en una grilla paginada (cantidad de publicaciones por pagina a determinar por el grupo), se debe poder adelantar y retroceder por pagina como así también volver a la primera pagina de publicaciones o ir a la ultima.

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.

Luego el usuario deberá seleccionar la publicación que desee comprar/ofertar, el sistema mostrará toda la información de la publicación.

En este momento el usuario podrá comprar/ofertar según sea el tipo de publicación seleccionada. En el primer caso , publicación del tipo "Compra inmediata", el usuario

seleccionará la acción de "comprar" y el sistema mostrará toda la información del vendedor para que, por sus propios medios, el "comprador" prosiga con el proceso de compra. En el caso que se trate de una publicación "Subasta" el usuario seleccionará la acción de "Ofertar" y el sistema pedirá al usuario que ingrese el monto que desea ofertar, el cual debe ser siempre superior al actual, el sistema registrará la oferta del usuario como la fecha en la cual se realizó dicha oferta, y validar que el monto sea entero.

Cabe aclarar que la plataforma de publicación desarrollada no desarrolla ningún medio de pago sobre el producto en sí mismo, sino que solo factura/cobra las comisiones y los costos de publicación de los productos publicitados mencionada plataforma.

Antes de comprar u ofertar, el cliente, podrá realizar preguntas no más de 255 caracteres sobre la publicación, si es que esta la permite. Realizar preguntas no es compromiso de compra alguno y se pueden realizar tantas como sean necesarias sin la necesidad de comprar u ofertar.

Un cliente no puede hacer una pregunta sobre una publicación que el mismo cree, ósea, auto-preguntarse. Mismo criterio tanto para comprar u ofertar, un cliente no puede auto-comprarse u auto-ofertarse.

## 12. Calificar Al Vendedor

Esta funcionalidad, como su nombre lo indica, permite a un cliente calificar a un vendedor por el producto/servicio adquirido.

Este sistema de calificación solo es unidireccional, el cliente puede calificar solamente.

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. De acuerdo a la cantidad de estrellas obtenidas se realiza un promedio por entre la cantidad de estrellas obtenidas y las operaciones realizadas, obteniéndose así 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 oferto 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.

Si el cliente posee más de 5 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.

## 13. Historial de Cliente

Esta funcionalidad permite a un usuario (sea cliente o vendedor) conocer todo su historial de operaciones realizadas en el sistema, las cuales son las siguientes:

- Compras.

- Ofertas, todas las ofertas realizadas y si estas ganaron la subasta o no.
- Calificaciones otorgadas y recibidas.

#### 14. Facturar Publicaciones

En esta funcionalidad de emitirán las facturas para el pago de comisiones por parte del vendedor a la empresa que administra el sistema ya sea de ventas por compra directa o por subastas.

Será necesario que en la factura se registren todas aquellas publicaciones a rendir, es decir se debe registrar la facturación de la publicación en sí misma y las comisiones respectivas a la visibilidad de la publicación y en referencia la cantidad vendida. Debe elegirse la forma de pago de la misma, por ejemplo efectivo, tarjeta de crédito, la cual se debe registrar los datos de la misma.

Deberá tenerse en cuenta que primero se rinden las compras recibidas más antiguas. Bajo ningún circunstancia se podrá saltar comprar por rendir, el vendedor determinará cuantas compras quiere rendir, pero deberá tener en cuenta que si se acumulan más de 10 compras sin rendir, se procederá a inhabilitar al vendedor pausando todas sus publicaciones, el mismo no podrá loguearse hasta que regularice su situación con la administración de la plataforma.

Una vez regularizada la situación el vendedor podrá volver a ingresar al sistema y sus publicaciones estarán activas nuevamente. Para ello el vendedor se dirigirá a la oficina de administración y un administrativo deberá ingresar al sistema para corregir esta situación, generará la facturación de las compras pendientes y volverá a habilitar al vendedor desde el ABM de empresa, pero a esta instancia, el proceso de inhabilitación de vendedor no será necesario realizarlo por parte de los alumnos pero si la facturación que el mismo vendedor genera.

Una vez que el vendedor a determinado cuantas compras quiere rendir deberá ingresar los datos para el pago, el mismo deberá quedar registrado en el sistema y se realizará por medio pago que corresponde.

#### 15. 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.
- Vendedores con mayor facturación.
- Vendedores con mayores calificaciones.
- Clientes con mayor cantidad de publicaciones in calificar.

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 “GD1C2014”.

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: “gd2014”
  - Base de Datos Predeterminada: GD1C2014.
  - El resto de los parámetros respetar sus valores default.
- Solapa “Funciones del Servidor”:
  - Seleccionar “sysadmin”
- Solapa “Asignación de Usuarios”:
  - Seleccionar asignar a “GD1C2014”
- 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: “gd2014”.

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

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

La cátedra provee un archivo BATCH para ejecutar esta operación, denominado “EjecutarScriptTablaMaestra.bat”. Haciendo doble clic sobre el mismo se ejecutan ambos archivos (“gd\_esquema.Schema.sql” y “gd\_esquema.Maestra.Table.sql”) a través del modo consola. El Script necesita aproximadamente 40 minutos para finalizar su ejecución.

***sqlcmd -S <Servidor\Instancia> -U <Nombre\_de\_usuario> -P <Password> -i <Nombre\_del\_archivo1>,< Nombre\_del\_archivo2> -a 32767***

Ejemplo:

***sqlcmd -S localhost\SQLSERVER2008 -U gd -P gd2014 -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 “FrbaCommerce”. 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: gd2014
- Nombre de la base de datos: GD1C2014

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# “FrbaCommerce” 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:* 18/06/2014 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 18/07/2013.

## **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 18/05/2012 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/1jtwu\\_QXmp3\\_VFSicM0BREYY2wM\\_4wthM6\\_sJVD2BICg/edit?usp=sharing](https://docs.google.com/document/d/1jtwu_QXmp3_VFSicM0BREYY2wM_4wthM6_sJVD2BICg/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](mailto:gestiondedatos.entregas@gmail.com)

El asunto del mail debe cumplir con el siguiente formato:

TP1C2014<b><curso><b><nombreGrupo><Nro de grupo>

<b>: espacio en blanco

Ejemplos:

TP1C2014 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:

TP1C2014 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 “GrouponDesktop”  
    ⇒ \data  
        ⇒ Archivo de script de base de datos “script_creación_inicial.sql”.
```

### **Readme.txt:**

Es un archivo de texto plano con los siguientes datos:

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

### **Estrategia.pdf:**

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

Se debe incluir un DER del modelo de datos creado con una explicación detallada de cada entidad, relaciones, claves primarias y foráneas, índices, stored procedures, triggers, vistas, etc.

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

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

### **\src:**

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