

Universidad Tecnológica Nacional  
Facultad Regional Buenos Aires

# Gestión de Datos

Trabajo Práctico

2° Cuatrimestre 2017

**PAGO ÁGIL**

Enunciado V1.0



## Índice

Introducción	
Objetivos generales	
Descripción general	
Componentes del TP	
<u>Base de Datos y Modelo de Datos</u>	
<u>Aplicación Desktop</u>	
Requerimientos	
General	
Base de Datos	4
Aplicación Desktop	4
General	4
Base de Datos	4
Aplicación Desktop	4
Restricciones de la solución	16
Condiciones de aprobación	16
Testing	6
Modelo de Datos	6
Consultas SQL	7
Respetar Guía de ABMs	17
Aplicación Desktop	7
Fecha de entrega y condiciones	17
Sobre los grupos	18
Ayuda y contacto	18
Donde aprender C#	
Sobre la elección de C#	19
Obtención de herramientas	20
Formato de entrega	21
Lugar de envío	21
Estructura del archivo zip	2
Readme.txt:	22
Estrategia.pdf:	22
\src:	22
\data	23

## **Introducción**

### **Objetivos generales**

El presente trabajo práctico persigue los siguientes objetivos generales

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

### **Descripción general**

El presente TP describe la situación en donde una empresa de cobranza de impuestos de rubros generales (Pago Ágil) se encarga de recepcionar el pago de diferentes facturas por parte de diversas personas que se acercan una determinada sucursal para que se les cobre las mismas.

Uno de los tipos de usuarios del sistema, en este caso el personal de ventanilla, recibe de una persona las diversas facturas de servicios que quieran pagar, en donde dicho empleado de ventilla registra el pago de ellas. Luego de recibido el pago, los administradores del sistema (el segundo tipo de usuario del sistema) se encargan de realizar las rendiciones pertinentes a las empresas de servicios por los servicios que se han cobrado en ventanilla por los cobradores.

Esta empresa de cobro dispone de varias sucursales a lo largo de todo el país y la idea de este TP es reemplazar el viejo sistema de registración que existía.

Tener en cuenta que por fines académicos esto solo es una aproximación o simulación de dicha situación.

### **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 estándares de desarrollo de bases de datos explicados durante la cursada.

Los datos de esta tabla maestra pertenecen a un dominio de la gestión de transporte mediante una aplicación de celular.

El sistema a desarrollar será utilizado por 2 tipos de usuarios con funciones bien determinadas: administradores y cobradores. Los cobradores son los empleados que registran el pago de las diversas facturas.

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 entregados en la tabla maestra. 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 4.5.

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, siguiendo siempre reglas de buenas prácticas. Este modelo de datos incluye:

- Creación de esquema correspondiente.
- 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.

- Además de la migración, se debe incluir la inserción de datos no incluidos en la tabla maestra que sean necesarios para el funcionamiento de la aplicación desktop, como por ejemplo, los roles.
- Se debe tener en cuenta que tal vez no sea necesario utilizar todos los conocimientos de SQL aprendidos durante la cursada. Los alumnos deberán decidir si es necesario usarlos en su totalidad y cuáles de ellos le resultan adecuados para la resolución de la problemática.

El alumno deberá entregar OBLIGATORIAMENTE 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. No será necesario que vuelva a entregar la tabla maestra o que esta sea incluida en el nuevo modelo, del mismo modo, el script entregado deberá contener la creación del esquema propio del grupo.

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 factura que nunca existió.

## Aplicación Desktop

El alumno deberá crear una aplicación Desktop en C# sobre Visual Studio 2012 con Framework .NET versión 4.5. Esta aplicación deberá contar con formularios, reportes y tablas.

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

- ABM de Rol
- Login y Seguridad
- Registro de Usuario
- ABM de Cliente
- ABM de Empresa
- ABM de Sucursal
- ABM Facturas
- Registro de Pago de Facturas
- Rendición de Facturas cobradas
- Listado Estadístico

## 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, además, una funcionalidad puede estar en más de un rol al mismo tiempo. Debe tenerse en cuenta, que actualmente existen 2 roles.

- Administradores
- Cobradores

En la modificación de un rol solo se pueden alterar los campos: 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, ósea, el rol debe poder inhabilitarse. No está permitida la asignación de un rol inhabilitado a un usuario, por ende, se debe quitar el rol inhabilitado a todos aquellos usuarios que lo posean. Un usuario puede tener más de un rol asignado.

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, EN CASO DE QUE EL USUARIO TENGA ASIGNADO MAS DE UN ROL, SE DEBERÁ SELECCIONAR CON CUAL DE ELLOS SE QUIERE LOGUEAR.

Al acceder al sistema, la aplicación solo deberá generar y mostrar las entradas de menú disponibles para este usuario, según el rol seleccionado. El usuario no debe poder ver las funcionalidades a las que no posee acceso.

Si el Login es incorrecto el usuario no podrá acceder al sistema, para lo cual se debe volver a mostrar nuevamente el Login para que intente otra vez. 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

Para reducir el tiempo de confección del TP y así beneficiar al alumno se determinó que no será necesario que se codifique un ABM que gestione los usuarios, pero deben tener en cuenta que su estructura en la base de datos debe existir, la no codificación de algún punto descripto en el presente TP no exonera la realización de sus respectivas tablas en la base de datos.

Deberán tener en cuenta que al momento de realizar la entrega de dicho TP, deben entregar un usuario con perfil Administrador, username **admin** y la password a asignar será **w23e**. Para el usuario cobradores, tanto aquellos que sean migrados (si existiese), los alumnos determinarán cuáles serán los username y password, los cuales deberán ser especificados en archivo de estrategia pedido para la entrega. Deben tener en cuenta que los cobradores pertenecen a una sucursal en particular.

No cumplir con este último requerimiento será motivo suficiente de no corrección del TP, pasando a re-entrega directamente, de la misma manera, que no sea necesario codificar un ABM completo de esta funcionalidad no implica no modelar una entidad de usuario.

Otros lineamientos a tener en cuenta son que el username debe ser único en un todo el sistema, el password deberá almacenarse encriptado de forma irreversible bajo el algoritmo de encriptación SHA256 (se deberá utilizar este algoritmo de encriptación de manera exclusiva, usar otro será motivo de re-entrega). Téngase en cuenta con que herramienta se generara la encriptación porque puede que la base de datos no tenga dicha solución.

Por cuestiones de practicidad a la hora de probar el TP, se recomienda que el username y el password sean iguales, a excepción del admin.

### 4. ABM de Clientes

Funcionalidad que permite crear, modificar y dar de baja un nuevo cliente al sistema. Estos clientes son aquellos que se acercan a la ventanilla de sucursal de cobro para efectuar el pago de uno o más servicios y del cual nuestra empresa de cobro de servicios quiere llevar un registro para funcionalidades futuras o hacerle llegar información pertinente.

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

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

Todos los datos mencionados anteriormente son modificables, en cuanto a la obligatoriedad de los mismos, es para todos los datos. El mail es un dato único, por ende no pueden existir 2 clientes con el mismo mail. El sistema deberá controlar esta restricción e informar debidamente al usuario ante alguna anomalía.

La eliminación de un cliente implica la baja lógica del mismo. Un cliente inhabilitado no podrá efectuar pago de servicios. Se debe poder volver a habilitar el cliente deshabilitado desde la sección de modificación.

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

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

## 5. ABM de Empresa

Esta funcionalidad permite poder crear, modificar y eliminar las empresas que tienen convenio con nuestra entidad que registra pago de servicios

Dar de alta una nueva empresa implica cargar los siguientes datos:

- Nombre
- Cuit
- Dirección
- Rubro (selección acotada)

Todos los datos mencionados anteriormente son obligatorios.

No pueden existir en el sistema más de una empresa con el mismo cuit.

El dato rubro hace referencia al tipo de servicio que brinda la empresa a sus respectivos clientes.

En la modificación de una empresa se pueden alterar todos sus datos.

Cuando se trate de una baja, la misma será lógica. Para una empresa inactiva sus datos seguirán existiendo en la base de datos. Se debe poder volver a habilitar dicha empresa inhabilitada desde la sección de modificación, si la empresa se encuentra inactiva no se podrán registrar cobros de facturas a su nombre. Dada dicha situación, el cliente deberá buscar otro medio de pago para sus facturas. Además tampoco está permitido realizar rendiciones a empresas que se encuentren inactivas, se deberá controlar al momento de cambiar el estado de una empresa si hay facturas pendientes de rendición. Para por poner una empresa en estado inactivo debe estar todas sus facturas cobradas de manera rendidas.

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

- Nombre
- Cuit
- Rubro (selección acotada)



## 6. ABM de Sucursal

Esta funcionalidad permite crear, modificar y eliminar una sucursal del sistema de cobro de facturas. Las sucursales poseen usuarios cobradores, si se quiere dar de baja una sucursal se deberá tener en cuenta que ninguno de los usuarios que la componen podrán loguearse en el sistema.

Dar de alta una nueva sucursal implica cargar los siguientes datos:

- Nombre
- Dirección.
- Código Postal.

Todos los datos mencionados anteriormente son obligatorios. Bajo ninguna circunstancia puede haber más de una sucursal por código postal existente.

La eliminación de una sucursal implica su baja lógica. Se debe poder volver a habilitar una sucursal inhabilitada desde la sección de modificación y la misma debe cumplir con las restricciones del párrafo anterior.

Para elegir una sucursal que se desea modificar o eliminar se debe presentar un buscador con listado, que permita filtrar por cualquiera de sus campos.

## 7. ABM de Facturas.

Esta funcionalidad permite procesar las facturas que luego podrán ser cobradas en las sucursales.

Para llevar a cabo esta acción el operador deberá ingresar los siguientes dato de las facturas y sus items:

- Cliente
- Empresa (selección acotada)
- Número de factura
- Fecha de alta de la factura.
- Fecha de vencimiento.
- Total
  - Items:
  - Monto
  - Cantidad

Todo los datos mencionados anteriormente son obligatorios. Las facturas no se podrán dar de baja ni modificarlas si estas fueron pagadas, y/o rendidas.

## 8. Registro de Pago de Facturas

Esta funcionalidad permite registrar el pago de un cliente por parte del operador de cobro de una o más facturas que dicho cliente presente.

Para llevar a cabo esta acción el operador procederá a ingresar los datos de la/s factura/s:

- Número de factura
- Fecha de cobro (automática del sistema)
- Empresa (selección acotada)
- Cliente (selección acotada)
- Fecha de vencimiento de la factura.
- Importe
- Sucursal

Todos los datos mencionados anteriormente son obligatorios.

Para la carga de estos datos, el cobrador hará una inspección visual del soporte papel (factura) entregada por el cliente y cargará los datos que correspondan. Se debe controlar que la fecha de vencimiento a ingresar sea menor o igual a la fecha del sistema, del mismo modo no puede haber facturas con importe menor o igual a cero.

Otro punto a corroborar es que la empresa a elegir debe estar activa, no se pueden registrar pagos a empresas que no estén activas.

El operador ingresará tantas facturas como sea las que entregue el cliente en cuestión, luego de finalizada la carga total de facturas entregadas, se registrará el pago por el total de las mismas. Los medios de pago pueden ser en efectivo u otros medios de pagos que los alumnos creen convenientes, según su creatividad, debiendo ser más de uno para dar más de una posibilidad de pago.

Del mismo modo se deberá registrar la sucursal en la cual se efectuó el pago, dicho dato se obtendrá del usuario operador. En caso de que el operador trabaje en más de una sucursal, se deberá tomar aquella que se seleccionó al momento del login cuando el cobrador ingresó al sistema.

## 9. Rendición de Facturas cobradas


Funcionalidad que permite a los administradores de nuestra entidad de cobro realizar la rendición de las facturas cobradas a los clientes que se acercan a nuestras sucursales. Dichas rendiciones se realizan una vez al mes y por única vez dentro de dicho mes para una empresa en cuestión. Las fechas de rendiciones son pactadas con cada empresa en forma individual, ende, no se rinde el mismo día para todas las empresas asociadas a nuestra forma de cobro.

Estas rendiciones conforman el total cobrado para una empresa de servicios X. No está permitido rendir 2 veces una misma factura. Dichas rendiciones incluyen a todas las facturas del mes en cuestión.

Nuestra empresa cobra un pequeño porcentaje (ganancia por el cobro) el cual es descontado del total rendido. Dicho importe cobrado a la empresa de servicios debe

quedar registrado el momento de la rendición como así también todas aquellas facturas que forman parte de la rendición, se debe seguir un patrón maestro detalle para su modelado.

Para hacer frente a esta funcionalidad es necesario que se registre los siguientes campos:

- Fecha de rendición.
- Cantidad de facturas rendidas
- Conjunto de facturas rendidas
-  Importe – comisión por el cobro de facturas.
- Empresa
- Porcentaje de comisión (que puede variar en el tiempo, por lo cual no puede ser fijo)
- Importe total de la rendición.

Todos los datos mencionados anteriormente son obligatorios.

## 10. Devoluciones

Funcionalidad que permite a los administradores realizar devoluciones sobre las facturas cobradas a los clientes, siempre y cuando estas no hayan sido rendidas.

Las facturas seleccionadas para realizar la devolución puede ser cobradas nuevamente. La razón por la cual se desarrolla esta funcionalidad puede ser por errores de cobro o simplemente que el cliente decida retrotraer el pago efectuado.

Para ambos caso debe quedar registrado el motivo de la devolución.

Los datos a ingresar dependerán si la devolución es del tipo rendición o factura.

Solo los administradores pueden devolver rendiciones y solo los cobradores pueden devolver facturas pagadas.

## 11. Listado Estadístico

Esta funcionalidad nos debe permitir consultar el TOP 5 de:

- Porcentaje de facturas cobradas por empresa.
- Empresas con mayor monto rendido.
- Clientes con mas pagos
- Clientes con mayor porcentaje de facturas pagadas (clientes cumplidores).

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

El listado se debe ordenar en forma descendente según sea el campo principal de la consulta.

Además de ingresar el año a consultar, el sistema nos debe permitir seleccionar que tipo de listado se quiere visualizar.

Cabe aclarar que los campos a visualizar en la tabla del listado para las 3 consultas no son los mismo, y al momento de seleccionar un tipo solo deben visualizarse las columnas pertinentes al tipo de listado elegido.

Las columnas del listado para cada una de las consultas quedan a cargo del alumno y dichas columnas deben ser lo suficientemente descriptivas para poder brindar un informe detallado a la gerencia de la tropa.

## 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 “GD2C2017”.

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

- Solapa “General”:
- Nombre de inicio de sesión: “gd”
- Autenticación de SQL Server
- Contraseña: “gd2017”
- Base de Datos Predeterminada: GD2C2017.
- El resto de los parámetros respetar sus valores default.
- Solapa “Funciones del Servidor”:
- Seleccionar “sysadmin”
- Solapa “Asignación de Usuarios”:
- Seleccionar asignar a “GD2C2017”

- 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: “gd2017”.

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 gd2017 -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 4.5. La versión Express posee la funcionalidad necesaria como para desarrollar el Trabajo Práctico.

La aplicación template se denomina “UberFrba”. 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:
  - o Nombre de Usuario: gd
  - o Password: gd2017
- Nombre de la base de datos: GD2C2017

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 4.5. 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. No se admite otro formato que no sea este, cualquier otro formato será considerado como reentrega.
- La aplicación C# “PagoAgilFrba” 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.

## 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 (5/10/2017) y 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. Esto quiere

decir que no hay re-entrega de esta instancia y la evaluación final será en base al 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: 5/11/2017 hasta las 12:00hs del mediodía (GMT -3:00 Buenos Aires).**

Los TPs entregados luego de las 12:00hs, se considerarán 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 10 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.

Si llegaron a realizar una sola entrega del TP en la última fecha de entrega queda a responsabilidad del grupo, ya que se les da la posibilidad de tener mas de una entrega, en caso de llegar a esa situación, solo contarán con una sola entrega habiendo perdido la chances anteriores de entrega.

**La última fecha para recepción de TP es el día 7/12/2017**

## **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 5/10/2017 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://drive.google.com/file/d/0B4kHJe7wwAn2NU15cjhCMmxWMEk/view?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 lo hora de su uso. El uso de punteros y procesamiento de cadenas muchas veces resulta complicado, sin mencionar la dificultad de encontrar un error en tiempo de ejecución. También hemos notado que se invierte mucho tiempo tratando de crear interfaces amigables mediante consola de texto.

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

Por nombrar algunas ventajas de C#:

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

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

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

## 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](mailto:gestiondedatos.entregas@gmail.com)

El asunto del mail debe cumplir con el siguiente formato:

TP2C2017<b><curso><b><nombreGrupo><Nro de grupo>

<b>: espacio en blanco

Ejemplos:

TP2C2017 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:

TP2C2017 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 “UberFrba”
- ⇒ \data
- ⇒ Archivo de script de base de datos “script\_creación\_inicial.sql”.

### **Readme.txt:**

Es un archivo de texto plano con los siguientes datos:

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

### **Estrategia.pdf:**

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

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

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

## 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, [Invalid object name 'gaga.PUBLICACION'].FEA_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 limite 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 aun 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.