



Universidad Tecnológica Nacional  
Facultad Regional Buenos Aires

# Gestión de Datos

Trabajo Práctico

1º Cuatrimestre 2018

FRBA-Hotel

Enunciado V1.0



## Índice

Índice.....	2
Introducción.....	3
Objetivos generales.....	3
Descripción general .....	3
Componentes del TP.....	4
<u>Base de Datos y Modelo de Datos</u> .....	4
<u>Aplicación Desktop</u> .....	4
Requerimientos.....	4
General .....	4
Base de Datos .....	4
Aplicación Desktop.....	5
General .....	14
Base de Datos .....	14
Aplicación Desktop.....	15
Condiciones de aprobación .....	16
Testing.....	16
Modelo de Datos.....	16
Consultas SQL .....	17
Respetar Guía de ABMs.....	17
Aplicación Desktop.....	17
Fecha de entrega y condiciones.....	17
Sobre los grupos .....	18
Ayuda y contacto .....	18
Donde aprender C#.....	19
Sobre la elección de C#.....	19
Obtención de herramientas.....	20
Formato de entrega .....	20
Lugar de envío .....	20
Estructura del archivo zip.....	21
Readme.txt:.....	21

## **Introducción**

### **Objetivos generales**

El presente trabajo práctico persigue los siguientes objetivos generales

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

### **Descripción general**

El siguiente trabajo práctico intenta simular la migración y remoción de un viejo sistema de gestión hotelera.

Dicho sistema deberá permitir la administración de los diversos hoteles que conforman la cadena Hotelera en cuestión. Esta administración constará de funciones como la reserva de una estancia, cancelación de la misma, registro de ingreso y salida del hotel (check-in y check-out), registro de huéspedes, entre otras funciones.

El sistema se encargará de realizar los cálculos necesarios para poder determinar que habitaciones se encuentran disponibles al momento de realizarse una reserva.

En primera instancia se desarrollará un prototipo bajo una aplicación escritorio que permita conectarse a la base de hoteles que maneja la cadena. La renovación de dicho sistema es necesaria dado que la Administración General tenía ciertas dificultades para la gestión de los diferentes recursos (Hoteles) ocurriendo en reiteradas oportunidades errores en las reservas como así también en la facturación a los diversos huéspedes que se alojaban en algún de sus hoteles.

El principal objetivo de este nuevo sistema es mejorar la relación con los huéspedes al momento de la reserva, haciendo que su estancia sea lo más confortable posible, reivindicando a la Empresa en su mejora continua de eficiencia y posicionamiento en el mercado para ajustarse a las normas de calidad internacionales.

## **Componentes del TP**

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

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

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

El alumno deberá estudiar los datos recibidos y confeccionar un modelo de datos que siga todos los standards de desarrollo de bases de datos explicados durante la cursada, como por ej. Evitar que descripciones de determinadas entidades no sean modelos por más que las mismas sean invariantes.

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

El sistema a desarrollar será utilizado por 3 tipos de usuarios distintos: administrador, recepcionista de hotel y posibles huéspedes-clientes.

Parte de la lógica del negocio a resolver deberá ser inferida por el alumno, en base a las columnas y valores presentes en los datos. De todas maneras es recomendable consultar al grupo de Google de la materia antes de tomar decisiones incorrectas, dado que ciertas cuestiones ya fueron resueltas y discutidas con anterioridad.

### **Aplicación Desktop**

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

- Creación de nuevas tablas y vistas.

- Creación de claves primarias y foráneas para relacionar estas tablas.
- Creación de constraints y triggers sobre estas tablas cuando fuese necesario.
- Creación de los índices para acceder a los datos de estas tablas de manera eficiente.
- Migración de datos: Cargar todas las tablas creadas utilizando la totalidad de los datos entregados por la cátedra en la única tabla del modelo. Para este punto se pueden utilizar Stored Procedures sobre la base de datos. No podrá realizarse la migración de datos utilizando la aplicación Desktop ni ninguna otra herramienta auxiliar.

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

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

## Aplicación Desktop

El alumno deberá crear una aplicación Desktop en C# sobre Visual Studio 2012 con Framework .NET versión 4.5. Esta aplicación deberá contar con formularios, reportes y tablas. Se recomienda que todos estos componentes deban respetar los lineamientos planteados en el documento “Guía de ABMs”. Es recomendable leerlo en este punto antes de continuar con el enunciado.

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

1. [ABM de Rol](#)
2. [Login y seguridad](#)
3. [ABM de Usuario](#)
4. [ABM de Cliente \(huéspedes\)](#)
5. [ABM de Hotel](#)
6. [ABM de Habitación](#)
7. [ABM Régimen de estadía](#)
8. [Generar o Modificar un Reserva](#)
9. [Cancelar Reserva](#)
10. [Registrar Estadía \(check-in/check-out\)](#)
11. [Registrar consumibles](#)
12. [Facturar estadía](#)
13. [Listado Estadístico](#)

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

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

1. [ABM de Rol](#)

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

Crear un rol implica cargar los siguientes datos:

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

Todos los datos mencionados anteriormente son obligatorios.

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

Debe tenerse en cuenta que actualmente existen 3 roles:

- Administrador.
- Recepcionista.
- Guest

En la modificación de un rol se pueden alterar ambos todos los campos. Se debe poder quitar funcionalidades como así también agregar otras nuevas a un determinado rol que se está modificando.

La eliminación del rol implica una baja lógica del mismo, en este caso debe pasarse a un estado no activo, por lo tanto, el rol debe poder inhabilitarse.

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

El sistema debe poder volver a habilitar un rol que fue inhabilitado, esta acción debe llevarse a cabo desde la sección de modificación.

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

## 2. [Login y Seguridad](#)

Al ejecutar la aplicación, inicio del sistema, si el usuario en cuestión se trata de un administrador o una persona asignada a la recepción el usuario no podrá acceder a ninguna funcionalidad del sistema hasta completar el proceso de Login.

Los posibles huéspedes (futuros clientes) no deberán realizar ningún proceso de login, teniendo solamente acceso a la funcionalidad de reserva hotelera y su respectiva modificación al inicio del sistema dado que no deben realizar login alguno. Bajo ninguna circunstancia un guest (huésped) puede acceder a otra funcionalidad que no sea de reservado o modificación de la misma, pero deberá contemplarse que en un futuro este rol pueda tener más funcionalidades asignadas para que el sistema sea escalable y permita su posterior modificación.

Por ende, según el tipo de usuario, el proceso de login pedirá al usuario su Username y su Password. Si el Login es correcto, el usuario podrá acceder al sistema. Si el usuario tiene un solo rol asignado se ingresará al sistema directamente, pero si dicho usuario tiene más de un rol asignado, se deberá dar la posibilidad de seleccionar uno de ellos. Luego de validar el login, la aplicación deberá generar y mostrar las entradas de menú disponibles para este usuario según sea el rol que se haya elegido además de ocultar aquellas funcionalidades a las que no tiene acceso.

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

El sistema debe llevar un registro de cantidad intentos fallidos al querer acceder. Luego de 3 intentos fallidos en cualquier momento, el usuario debe ser inhabilitado.

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, a los que les corresponda, tienen la capacidad de utilizar el Login.

### 3. [ABM de Usuario.](#)

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

Un administrador puede dar de alta a otros administradores o a personal de recepción.

Crear un Usuario implica cargar los siguientes datos:

- Username
- Password
- Rol asignado (selección especial)
- Datos identificatorios a la persona (administrador o recepcionista).
  - Nombre y Apellido.
  - Tipo y número de documento
  - Mail.
  - Teléfono.
  - Dirección.
  - Fecha de Nacimiento
  - Hotel donde se desempeña.

Todos los datos mencionados anteriormente son obligatorios.

El username debe ser único en un todo el sistema. La aplicación deberá controlar esta restricción e informar debidamente al usuario si esto no ocurre.

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

Actualmente un usuario solo puede tener asignado un solo rol en un momento dado, pero no se descarta que pueda tener más de un rol.

Debe tenerse en cuenta que el password se pueda modificar una vez que el usuario se encuentre logueado, también debe contemplarse que un administrador solo puede realizar modificaciones y dar de baja a usuarios que se desempeñen en el mismo hotel él trabaja. Esto es independiente a la inhabilitación por fallas de ingreso. **IMPORTANTE:** Toda baja debe realizarse en forma lógica.

Tanto los administradores como la gente de la recepción pueden trabajar en más de un Hotel de la cadena.

**Consideración importante:** si el usuario desempeña funciones en más de un establecimiento deberá seleccionar a cuál de ellos está ingresando en dicha sesión y solo podrá realizar acciones sobre el establecimiento elegido. Ej. Si un recepcionista trabaja en el Hotel A y B, inicia sesión en el Hotel A (se loguea), entonces no podrá realizar reservas en el Hotel B, si quiere realizar acciones sobre el Hotel B, deberá cerrar la sesión actual y abrir otra nueva sesión ingresando con el Hotel B.

### 4. [ABM de Clientes \(huéspedes\)](#)

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

Los clientes son huéspedes que van a generar una reserva o que ya poseen reservas de estadía o que ya tuvieron una reserva en el pasado independientemente si la misma se concretó o no. Los clientes almacenados se guardan independientes al hotel en donde se vayan a alojar, se alojen o se hayan alojado.

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

- Nombre.

- Apellido.
- Número de identificación y su correspondiente tipo. (tener en cuenta que pueden ser extranjeros)
- Mail.
- Teléfono.
- Dirección calle, localidad, país de origen
- Nacionalidad
- Fecha de Nacimiento.

Todos los datos mencionados anteriormente son modificables. El mail es un dato único, por ende no pueden existir 2 o más huéspedes con el mismo mail. 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 el sistema, dicha validación se realizará a través del control del tipo y número de identificación registrado en el sistema.

Un cliente inhabilitado no podrá realizar reservas en ningún hotel de la cadena bajo ninguna circunstancia. Se debe tener en cuenta que se debe poder volver a habilitar a un cliente deshabilitado desde la sección de modificación. Si alguno de los acompañantes se encuentra inhabilitado tampoco podrá registrar el ingreso al hotel para realizar check-in. La inhabilitación en la mayoría de los casos puede darse porque algún hotel de la cadena hizo uso del derecho permanencia y admisión o por pedido propio del huésped al administrador o personal responsable de la baja de datos.

Deben registrarse todos los huéspedes que ocupen la habitación, independientemente de quien haya efectuado la reserva, de esta manera se podrá contar con información de todas aquellas personas que se hayan alojado en alguno de los hoteles de la compañía en un momento determinado.

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

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

## 5. [ABM de Hotel](#)

Funcionalidad que permite a un administrador crear, modificar y dar de baja un Hotel de la cadena.

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

- Nombre.
- Mail.
- Teléfono.
- Dirección
- Cantidad de Estrellas
- Ciudad
- País
- Tipo de regímenes que posee (All inclusive, pensión completa, solo desayuno, etc.)
- Fecha de Creación

Esta funcionalidad solo puede ser atribuida a los administradores.



Solo se creará un Hotel cuando los accionistas de la cadena comunican a uno de los administradores que él se hará cargo de la gestión del nuevo establecimiento incorporando, ende, se le asignará este nuevo hotel y podrá gerenciar uno o mas hoteles.

Para la modificación se podrán alterar todos los valores, y para la baja se deberá tener en cuenta que la misma será en forma momentánea. Esto quiere decir que el hotel no estará habilitado para la recepción de pasajeros durante un período X (Fecha inicio – Fecha Fin), ya sea por motivos de mantenimiento, ampliaciones, etc. Para llevar a cabo esta baja temporal será necesario que el establecimiento se encuentre vacío, no deberá contar con reservas para dicho intervalo de fechas como así también huéspedes alojados. Durante ese período se podrán tomar reservas para fechas futuras y las mismas no podrán incluir ningún día donde el hotel se esté remodelando. Deberá llevarse un registro de todos estos períodos (hotel cerrado) a modo de poder contar con un historial, donde se describa las tareas que se realizaron durante esas fechas. Este historial permitirá en un futuro generar un reporte que informe a los accionistas los motivos por los cuales el hotel estuvo cerrado (dado de baja) por X cantidad de días.

Para elegir que Hotel se desea modificar o dar de baja se debe presentar un buscador con un listado (grid), que permita filtrar simultáneamente por alguno o todos los siguientes campos:

- Nombre.
- Cantidad de Estrellas
- Ciudad
- País

**Consideración importante:** el Hotel puede tener diferentes regímenes de estadía para ofrecer a sus huéspedes, por lo tanto deberá tenerse en cuenta que si se quita algún tipo de régimen, no deben haber reservas tomadas y tampoco huéspedes alojados bajo dicho régimen para que pueda llevarse a cabo la modificación. En caso de haber reservas o huéspedes alojados se deberá informar la imposibilidad de quitar el régimen e informarlo debidamente al usuario.

## 6. [ABM de Habitación.](#)

Funcionalidad que permite a un administrador crear, modificar y dar de baja una habitación de hotel particular perteneciente a la cadena.

Una habitación se define de la siguiente manera:

- Número de Habitación
- Piso dentro del Hotel
- Ubicación en el Hotel (Vista al exterior/Interno). Esto refiere a si es una habitación sin ventanas o si tiene algún tipo de vista, ya sea al mar, a la calle, etc.
- Tipo de habitación.
- Descripción, comodidades.

Todos los datos mencionados anteriormente son modificables, excepto el tipo de habitación a fin de simplificar el desarrollo. El número de habitación es un dato único, por ende no pueden existir 2 habitaciones con la misma numeración dentro un Hotel en particular. El sistema deberá controlar esta restricción e informar debidamente al usuario ante alguna anomalía.

La baja de una habitación debe ser de forma lógica, esto puede darse cuando se realizan tareas de mantenimiento y la misma no puede y no está en condiciones para que sea utilizada por los huéspedes, por lo tanto si la habitación está dada de baja tampoco podrá ser tenida en cuenta al momento de tomar reservas porque la misma no está disponible para su uso. Luego que las tareas de mantenimiento concluyan, el administrador habilitará nuevamente esta habitación y la misma podrá ser ocupada y reconsiderada para el cálculo de disponibles a la hora de tomar una reserva.

El tipo de habitación hace referencia a si la misma es Simple, Doble, Doble twin, Tiple, Cuádruple, etc.

## 7. [ABM de Régimen de Estadía.](#)

Esta funcionalidad que permite a un administrador crear, modificar y dar de baja un tipo de régimen de estadía. Se entiende por régimen a si el servicio que presta el hotel es All inclusive, media pensión, desayuno etc.

El modelado de esto debería incluir:

- Código.
- Descripción.
- Precio base expresado en dólares.
- Estado ( Activo/No activo)

A fin de simplificar el TP, se determina que este abm no debe desarrollarlo/codificado por el alumno, dado que debería realizar las validaciones pertinentes si el precio base del habitación se modifica para que las reservas ya tomadas no varíen su precio, o si se da de baja algún régimen habría que controlar que no existan reservas tomadas en todos los hoteles de la cadena.

El precio base es un valor expresado por persona, ósea, por huésped.

## 8. [Generar o Modificar Reserva](#)

En este caso de uso se procesan las reservas para los posibles huéspedes del hotel.

Las mismas pueden ser generadas por el recepcionista del Hotel o por los usuarios Guest.

Toda reserva hotelera deberá almacenar en principio los siguientes datos:

- Fecha en la se realizó la reserva (fecha del sistema)
- Fecha desde y hasta para la permanencia en el establecimiento.
- Tipo de habitación que necesita la persona en función de la cantidad de huéspedes.
- Tipo de régimen del hotel elegido en cuestión

Luego de que se detallen estos datos el sistema determinará la disponibilidad de dicho pedido. Puede darse el caso que el cliente no tenga en claro el tipo de régimen que desea, si no lo ingresa, se deberá mostrar en un grid las diferentes variantes de regímenes que el hotel dispone. En ambos casos (si ingresa o no el régimen) se deberá informar el costo por día de la habitación.

El valor de la habitación se obtiene a través de su precio base (ver abm de régimen) multiplicando la cantidad de personas que se alojarán en la habitación (tipo de habitación) y luego de ello aplicando un incremento en función de la categoría del Hotel (cantidad de estrellas)

Una vez que el sistema informe la existencia de disponibilidad para el pedido del cliente, se informará el costo total de la reserva, expresado en dólares. Cuando el cliente confirme su reserva será necesario que se lo registre como cliente si es que esté nunca se alojó en algún Hotel de la cadena, si se alojó alguna vez deberá buscárselo por alguno de estos datos:

- Tipo de identificación (selección acotada)
- Nro. de identificación (texto libre solo números)
- Mail

Si nunca se alojó en algún de la cadena se lo registrará y para ello, el sistema pedirá una serie de datos básicos:

- Nombre y apellido.
- Tipo de identificación (selección acotada)
- Nro. de identificación (texto libre solo números)
- Mail.
- Teléfono.
- Dirección

- Ciudad
- País

Los datos de los restantes pasajeros, si es que son más de uno, se registrarán al momento de hacer el ingreso a la habitación y también se los deberá almacenar como clientes.

En caso que la reserva sea generada en la recepción, no será necesario que se seleccione el hotel de la cadena, porque la persona de recepción ya está logueada en el hotel que trabaja y solo puede trabajar con el hotel que tiene asignado al momento del login, en cambio, el guest si deberá seleccionar en que hotel quiere alojarse.

Luego que se confirma la reserva y se almacenó (si corresponde) los datos del cliente, el sistema informará un código de reserva. El mismo será necesario para realizar modificaciones, cambios de fechas, regímenes, tipo de habitación etc. Las modificaciones podrán ser realizadas por el guest o por la recepción hasta un día antes de la reserva y no se cobrará cargo alguno. Todo proceso de generación o modificación de reserva deberá almacenar el usuario que la produce. Si es recepción, el usuario en cuestión y si es un guest deberá ser un user genérico de tipo guest.

Tener en cuenta que en una reserva se pueden requerir más de una habitación la cuales estarán a nombre del cliente que originó el pedido.

Las reservas deberán contar con estados, para poder determinar en que situación se encuentran cada una de ellas. En principio alguno de ellos pueden ser:

- Reserva correcta
- Reserva modificada (la misma sufrió algún cambio y no es la misma al momento de su creación)
- Reserva cancelada por Recepción
- Reserva cancelada por Cliente
- Reserva cancelada por No-Show
- Reserva con ingreso (efectivizada)

## 9. [Cancelar Reserva](#)

Funcionalidad que permite a la recepción o a un huésped cancelar la reserva generada en algún momento y que la misma no haya sido utilizada con anterioridad.

Para llevar a cabo este proceso el sistema pedirá que se ingrese:

- Nro de Reserva
- Motivo
- Fecha de cancelación
- Usuario que generó la cancelación.

En ningún caso se deberá cobrar cargo alguno por cancelación de reserva y la misma podrá efectuarse hasta un día antes de la fecha de inicio de la reserva.

El número de reserva que debe ingresarse es generado por el sistema y es independiente a los hoteles, esto quiere decir que si se genera una reserva en el Hotel A con un nro de reserva X, el próximo nro X+1 puede ser de una reserva en el Hotel B o A perteneciente a la cadena.

## 10. [Registrar Estadía \(ingreso/egreso del hotel\)](#)

Funcionalidad que permite a la recepción registrar el check-in y el check-out de los huéspedes alojados en el Hotel. Este proceso, es igual a los restantes en los cuales interviene la recepción, podrá ser sobre el o los hoteles sobre el cual trabaja la persona de recepción.

Para ello, se deberá ingresar el número de reserva que posee el cliente.

A fines prácticos el ingreso solo se podrá realizar el día en el que se inicia la reserva y no antes, independientemente del horario en la cual tiene que hacer el ingreso a la habitación. Si se

pretende hacer el ingreso uno o más días posteriores a lo que indica la reserva, la misma se encontrará sin efecto y no se cobrará recargo alguno, el huésped pierde la reserva. Si el cliente pretende alojarse de igual manera, deberá realizarse una nueva reserva, la cual queda sujeta a disponibilidad del establecimiento.

Al momento de realizarse una reserva, se deberá tener en cuenta si existen otras reservas con días posteriores que no fueron efectivizadas, en caso afirmativo, las mismas se cancelarán por la no presentación del huésped y liberarán las habitaciones involucradas. Esto deberá tenerse en cuenta al momento de realizar una búsqueda para reservar alojamiento.

Si se diese el caso de que un huésped se retirase con anterioridad a la fecha estipulada en la reserva, se procederá a la registración de la salida de este huésped abonado el total de la reserva, dado que las mismas no tienen devolución alguna una vez que se haya efectuado el ingreso al hotel.

En caso de que el huésped quisiese quedarse más días en el Hotel, se deberá registrar la salida de la estadía en curso, se facturará lo correspondiente (adicionando consumibles si los hubiese) y se realizará una nueva reserva, sujeta a disponibilidad, junto con un nuevo ingreso.

Al momento de facturar la estadía se deberá indicar los días que efectivamente se alojó y se detallará en forma separada (otro ítem de la factura) el cargo por los días en los cuales no se alojó que corresponder al mismo costo de la habitación x día.

Para simplificar el desarrollo del TP, se estipula que no hay cambio de habitaciones una vez realizado el check-in, por ende al momento de cerrar la estadía el administrativo ingresará el o los número de habitaciones en donde se alojó para comenzar con la facturación de los cargos.

Tanto para el ingreso como para el egreso, deberá registrarse en qué fecha ocurrieron dichos eventos y que usuario se encargó de dichas operaciones.

Cuando se está procesando el ingreso y se pudo determinar que la reserva fue válida, se deberán ingresar los datos personales de los huéspedes restantes. Si alguno de ellos fue cliente de alguno de los hoteles de la cadena deberá ser posible hallarlo como se hace en la reserva y si nunca se hospedó se deberá utilizar el proceso de alta de cliente (ver generación o modificación de reserva).

## 11. Registrar Consumibles.

Esta funcionalidad es previa a la facturación y en ella se ingresaran todos aquellos productos y/o servicios que fueron consumidos y/o utilizados durante la estadía por los huéspedes de una determinado habitación.

Deberá tenerse en cuenta que si se contrató el régimen All Inclusive todos los consumibles adquiridos no deben generar un incremento en el valor de la estadía estipulado al momento de la reserva. Para este caso, al finalizar la carga se agregará un ítem que detalle un leyenda parecida a “descuento por régimen de estadía” y neteará los cargos por consumibles ingresados anteriormente.

Antes de finalizar dicho registro, se debe permitir la modificación de los ítems ingresados. Luego de finalizar la carga, no se podrá incluir o quitar ítems dado que se tiene la conformidad del huésped. A continuación se iniciará el proceso de facturación

## 12. Facturar Estadía

En esta funcionalidad se emitirán las facturas para el pago de estadía por parte del huésped al hotel en el cual se alojó.

Será necesario que en la factura se registren en la base de datos y se detallen todos los consumibles registrados, junto con los días en los cuales se alojó verdaderamente y aquellos (si fuera el caso) que no fueron utilizados por haber retirado antes de la finalización de la estadía acordada.

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.

Siempre la estadía se abona al finalizar la misma.

### 13. [Listado Estadístico](#)

Esta funcionalidad nos debe permitir consultar el TOP 5 de:

- Hoteles con mayor cantidad de reservas canceladas.
- Hoteles con mayor cantidad de consumibles facturados.
- Hoteles con mayor cantidad de días fuera de servicio.
- Habitaciones con mayor cantidad de días y veces que fueron ocupadas, informando a demás a que hotel perteneces.
- Cliente con mayor cantidad de puntos, donde cada \$20 en estadías vale 1 punto y cada \$10 de consumibles es 1 punto, de la sumatoria de todas las facturaciones que haya tenido dentro de un periodo independientemente del Hotel. Tener en cuenta que la facturación siempre es a quien haya realizado la reserva.

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

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

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

## Implementación

### General

El alumno deberá desarrollar dos componentes: un script de base de datos SQL Server y una aplicación Desktop C#.

A continuación se detalla la implementación de cada componente:

### Base de Datos

El alumno debe instalar el motor de base de datos SQL Server 2012 con las siguientes consideraciones:

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

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

Dentro del “Management Studio” crear una nueva base de datos con los parámetros default y nombre de base “GD1C2018”.

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: “gdHotel2018”
  - Autenticación de SQL Server
  - Contraseña: “gd2018”
  - Base de Datos Predeterminada: GD1C2018.
  - El resto de los parámetros respetar sus valores default.
- Solapa “Funciones del Servidor”:
  - Seleccionar “sysadmin”
- Solapa “Asignación de Usuarios”:
  - Seleccionar asignar a “GD1C2018”
- 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 “gdHotel2018” creado. Es probable que informe que la contraseña ha caducado. Cambiar la contraseña ingresando exactamente la misma que antes: “gd2018”.

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 “gdHotel2018”.
- `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\SQLSERVER2021 -U gdHotel2018 -P gd2018 -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 “FrbaHotel”. Cuenta con un formulario principal, una barra de menú y un formulario para cada funcionalidad visual que hay que implementar en el trabajo. El alumno debe depositar su código respetando esta estructura.

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

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

- Origen de datos: Microsoft SQL Server (SqlClient)
- `localhost\SQLSERVER2012`
- Utilizar autenticación de SQL Server:
  - Nombre de Usuario: `gdHotel2018`
  - Password: `gd2018`
- Nombre de la base de datos: `GD1C2018`

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.
- La aplicación C# “FrbaHotel” 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**

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 reentregar 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 (14/05/2018) 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 reentrega del DER y la evaluación final será en base al sistema completo.

La motivación de esta entrega es la corrección de 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 retrabajo si es necesario reentregar.

### **Entrega del sistema completo**

Día: 14/06/2018 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 restandole solamente 2 instancias de presentación. Estas 2 instancias no tienen fecha asignada y serán determinadas por el equipo para entregar cuando gusten, bajo responsabilidad de los alumnos saber que han perdido una instancia de corrección.

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 llegan 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 más 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 14/07/2018

### **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 14/05/2018 se enviaran los mail correspondientes con la confirmación de los grupos y se les asignará un número de grupo además del nombre que debidamente eligieron. Luego de esa fecha y de recibido el mail con el número de grupo, la conformación de los mismos será inalterable hasta la finalización del cuatrimestre. Cualquier cambio de integrantes, sea por el motivo que fuese, deberá realizarse antes de esa fecha.

### **Ayuda y contacto**

El sitio oficial de la materia es el siguiente:

<https://sites.google.com/site/gestiondedatosutn>

También existe un grupo de Google en donde se podrán plantear dudas sobre el TP. Su dirección es la siguiente:

<http://groups.google.com/group/gestiondedatos>

Todos los mensajes referentes al trabajo práctico deberán contener la etiqueta [TP] antes del asunto. Ej: “[TP] consulta sobre base de datos”.

Es obligación del alumno revisar el grupo periódicamente y mantenerse informado sobre actualizaciones, cambios de consignas, modificaciones del programa, cambios de fecha, etc.

La cátedra no asigna ayudantes específicos a cada grupo. Todas las consultas deberán hacerse a través del grupo de Google.

A lo largo de la cursada pueden ir surgiendo dudas particulares sobre el Trabajo Práctico que sean útiles para el resto de los alumnos. Para ello la cátedra cuenta con un documento denominado “Apéndice del Enunciado” en el cuál se agregan consideraciones generales de manera online. Su dirección es la siguiente:

<https://docs.google.com/document/d/18HzFQXJzCU7gpCueKVMMop1NLS03m7sVi4HtmTm49sw/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 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 2008. No puede utilizarse la versión 2012:

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# 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 2012 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 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:

TP1C2018<b><curso><b><nombreGrupo><Nro de grupo>

<b>: espacio en blanco

Ejemplos:

TP1C2018 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:

TP1C2018 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 (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 procesimiento.
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 [gaga].[PUBLICACION] (
  DESCRIPCION, STOCK, FECHA_INICIO, FECHA_FIN, PRECIO, ID_VISIBILIDAD,
  ESTADO, TIPO, PUBLICADO_POR, PREGUNTAS, VISIBILIDAD_FACTURADA)
values (
  'cosa', 10, convert(datetime,'2015/01/01 00:00:00',121),
  convert(datetime,'08/01/2015 12:00:00',121), 100, 10006, 1, 1, 2, 1,0 )

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

Messages

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

8. Entregar el TP faltando menos de 5 minutos para la fecha 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.