Universidad Tecnológica Nacional
Facultad Regional Buenos Aires

Gestión de Datos

Trabajo Práctico

2° Cuatrimestre 2018

Frba-PalcoNet

Enunciado V1.0

Última actualización: 27 de Septiembre de 2018

Índice

Introducción

Objetivos generales

El presente trabajo práctico persigue los siguientes objetivos generales

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

Descripción general

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

Dicho sistema deberá permitir la administración de clientes que compran localidades para diversos eventos a realizarse.

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

Componentes del TP

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

Base de Datos y Modelo de Datos

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

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

Los datos de esta tabla maestra pertenecen a un dominio de compra y publicación de cupones

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

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

Aplicación Desktop

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

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

Aplicación Desktop

El alumno deberá crear una aplicación Desktop en C# sobre Visual Studio 2008 con Framework .NET versión 4.5. Esta aplicación deberá contar con formularios, reportes y tablas. Todos estos componentes deberán respetar los lineamientos planteados en el documento "Guía de ABMs". Es recomendable leerlo en este punto antes de continuar con el enunciado.

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

1. Login y seguridad
2. ABM de Rol
3. Registro de Usuario
4. ABM de Cliente
5. ABM de Empresa de espectáculos
6. ABM de Categoría
7. ABM grado de publicación
8. Generar Publicación
9. Editar Publicación
10. Comprar
11. Historial del cliente.
12. Canje y administración de puntos
13. Generar Pago de comisiones
14. Listado Estadístico

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

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

1. ABM de Rol

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

Crear un rol implica cargar los siguientes datos:

- Nombre
- Listado de Funcionalidades (selección acotada)

Todos los datos mencionados anteriormente son obligatorios.

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

Debe tenerse en cuenta que actualmente existen 3 roles:

- Empresa
- Administrativo
- Cliente.

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

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

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

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

2. Login y Seguridad

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

El proceso de Login pedirá al usuario que ingrese su Username y su Password. Si Login es correcto, el usuario podrá acceder al sistema. En caso de que el usuario tenga un solo rol asignado, ingresará al sistema directamente luego de haberse logueado. En cambio si dicho usuario tiene más de un rol, se deberá dar la posibilidad de seleccionar uno de ellos.

Luego de validar el login, la aplicación solo deberá generar y mostrar las entradas de menú disponibles para este usuario según los roles del mismo. El usuario no debe ni siquiera ver las funcionalidades a las que no posee acceso.

Si el Login es incorrecto el usuario no podrá acceder al sistema y se debe volver a mostrar el Login para que intente nuevamente. El sistema debe llevar un registro de cantidad intentos fallidos de login. Luego de 3 intentos fallidos, independientemente del momento en que se efectue, 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 y no se considera una funcionalidad que puede ser asignada a un rol, ya que todos los usuarios tienen la capacidad de utilizar el Login.

3. Registro de Usuario.

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

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

Crear un Usuario implica cargar los siguientes datos:

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

Todos los datos mencionados anteriormente son obligatorios.

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

El password deberá almacenarse encriptado de forma irreversible bajo el algoritmo de encriptación SHA256. (Tener en cuenta que estrategia van a usar para cumplir con este requerimiento)

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

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

4. ABM de Clientes

Funcionalidad que permite a un administrativo crear, modificar y dar de baja un cliente del sistema. Para el caso que sea una alta, también se podrá asignarles un usuario en ese mismo momento reutilizando la funcionalidad de registro de usuario.

Los clientes son usuarios que pueden comprar localidades dentro la plataforma.

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

- Nombre.
- Apellido.
- Tipo de documento
- Numero documento
- CUIL
- Mail.
- Teléfono.
- Dirección calle, nro piso, depto y localidad
- Código Postal
- Fecha de Nacimiento.
- Fecha de Creación
- Datos de tarjeta de crédito asociada

Todos los datos mencionados anteriormente son modificables.

Se deberá tener en cuenta que un cliente solo puede registrarse una vez en la plataforma de compra, se deberá verificar la no repetición tanto del tipo y número de documento como la del cuil. Para este último dato es sumamente importante y necesario que el sistema indique si el número ingresado es un cuil valido o no.

Un cliente inhabilitado no podrá comprar en la plataforma bajo ninguna forma.

Se debe poder habilitar un cliente que fue deshabilitado desde la sección de modificación.

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

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

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

5. ABM de Empresa de Espectáculos

Funcionalidad que permite a un administrativo crear, modificar y dar de baja una empresa de espectáculos. Dichas empresas solo tiene la posibilidad de venta en la plataforma, denegándoseles toda posibilidad de compra.

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

- Razón Social.
- Mail.
- Teléfono.

- Dirección calle, nro piso, depto y localidad
- Código Postal
- Ciudad
- CIUT

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

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

La eliminación de una empresa implica la baja lógica de la misma. Se debe poder volver a habilitar a una empresa deshabilitada desde la sección de modificación.

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

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

6. ABM de Rubro.

Funcionalidad que permite categorizar a un espectáculo y determinar dentro de que género se puede encontrar a dicha publicación.

Las categorías deben estar conformadas por un código y una descripción.

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

7. ABM Grado de Publicación

Este tipo de funcionalidad es la que le permite al usuario determinar la prioridad de visualización con que sus publicaciones van a mostrarse, las cuales son Alta, Media y Baja

Todos los grados de publicaciones tienen una comisión, la cual es transparente al cliente que compra una localidad. Si la empresa de espectáculos publica una entrada a $100, el cliente paga $100 por la entrada, a la empresa solo se le rendirá el valor de la entrada menos el porcentaje correspondiente al grado de publicación.

8. Generar Publicación

Esta funcionalidad es la de mayor importancia en toda la plataforma porque es núcleo de la misma.

Una publicación se caracteriza por tener una serie de estados. Los mismos se detallan

- **Borrador**:
  Este estado permite al vendedor modificar todos los datos que sean necesarios, ya que dicha publicación todavía no está visible para la comunidad.

- **Activa o publicada**:
  Este estado es aplicado por el vendedor para que dicha publicación sea vista por toda la comunidad y esté disponible para operar con ella. Una vez que una publicación esta activa no puede pasar a estado borrador, se deberá chequear siempre que la fecha de compra sea menor o igual a la fecha del evento, así evitando que se realicen compras sobre eventos ya transcurridos.

- **Finalizada**:
  Estado utilizado por el vendedor para indicar que una publicación llegó a su fin y decide no ofrecer más localidades a la venta. A fines didácticos, el vendedor se encargará de modificar dicho estado al día siguiente de finalizada la publicación, evitando al alumno desarrollar procedimientos automáticos de cambios de estado y complicaciones con las fechas. El único cambió de estado automático permitido es cuando en una publicación se han vendido todas las localidades del evento. Una vez que una publicación está finalizada no puede cambiarse el estado de la misma a ningún otro.

Una publicación tiene una fecha de inicio y una fecha del evento a concretarse, además de ello, deberá determinar cuántas localidades se ofrecen, lo cual determinaría nuestro stock.

Una publicación tiene que tener mínimamente los siguientes datos:

- Código de Publicación (auto-numérico y consecutivo entre publicaciones sean o no del mismo vendedor)
- Descripción
- Ubicaciones (filas, asientos, precio y tipo de ubicaciones)
- Fecha de publicación
- Fecha y hora del espectáculo
- Rubro
- Dirección del espectáculo.
- Grado publicación
- Usuario responsable de la publicación
- Estado de la publicación

En esta funcionalidad también se deberá permitir la generación de publicaciones por lotes (batch - varias publicaciones al mismo tiempo). Dado que un espectáculo puede repetirse en el tiempo, como por ejemplo funciones de cine u obras teatrales. Se tendrá que permitir el ingreso de varias fecha-hora del espectáculo, de este modo, si contamos con una lista de 10 elementos fecha-hora de espectáculo, se deberán generar 10 publicaciones diferentes con su correspondiente fecha-hora. El rango de fecha-hora ingresado debe ser incremental y el próximo elemento a ingresar tiene que ser mayor al siguiente, si ingresamos 01/01/2019 10:00 el próximo valor tiene que ser mayor a este último, el sistema debe validar estas condiciones e informar posible errores.

9. Editar Publicación

Dicha funcionalidad permite la modificación de una publicación contenida en la plataforma.

Para realizar modificaciones se deberá tener en cuenta el estado de la publicación, solo en estado borrador. El estado de la publicación es la que nos determinará si se puede o no modificar.

Deberá tenerse en cuenta que no podrán haber 2 espectáculos iguales a la misma fecha y hora, y en caso de modificarse la fecha, la misma no podrá tener una fecha anterior a la del día de hoy (archivo config)

10. Comprar

Esta funcionalidad permite comprar una entrada(ubicación) para un espectáculo publicado.

Las compras son en pesos Argentinos y la facturación del mismo queda a cargo del vendedor, es por ello que se le solicita a los clientes que registren un mail para que la empresa de espectáculos les envié la factura correspondiente.

La aplicación solo factura comisiones a las empresas de espectáculos por aquellas localidades vendidas.

Las compras podrán realizarse para publicaciones activas, la fecha de compra se encuentre dentro del rango de fecha de publicación y la fecha del evento y las publicaciones no estén finalizadas ni pausadas ni en borrador.

Tener en cuenta al momento de validar las fechas, utilizar la fecha del archivo config para acelerar el proceso de corrección y prueba del TP. DATO IMPORTANTE: Tener en cuenta el ejemplo que se muestra al final de este documento para no cometer errores en el manejo fechas. Siendo que se les da un ejemplo con lo que SI deben hacer y con lo que NO deben hacer, si se producen errores de manejo de fechas al momento de la corrección queda a responsabilidad del grupo el método utilizado.

Las publicaciones deben poder ser filtradas por:

- Una o más categorías
- Descripción
- Rango de fecha para el espectáculo.

Dichas publicaciones se deben presentar ordenadas en función del grado de visibilidad, mostrándose primero las que tengan más peso.

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

Luego se deberá seleccionar el tipo y las ubicaciones elegidas para ese espectáculo, aclarando que en cada compra pueden adquirirse varias ubicaciones (entradas) de distintos tipos y precio.

El método de pago será únicamente electrónico con los datos asociados al usuario al momento de la creación o modificación del mismo. Como puede haber posibles cambios en los medios de pago se deberá dejar registrado con cual ellos se efectuó la compra. Si por algún motivo, el usuario no posee datos registrados para la compra, se le pedirá que asocie una tarjeta de crédito en dicho momento, sin tener salir de la funcionalidad de compra para asociar dicha la información.

Bajo ninguna circunstancia se deberá permitir la generación de compras con importes negativos.

Cada compra realizada por el cliente generará puntos para el sistema de cliente frecuente. Los alumnos determinarán cuantos puntos implican una compra y el vencimiento de los mismos.

11. Historial de Cliente

Esta funcionalidad permite a un cliente conocer todo su historial de compras. Debe presentarse en una grilla paginada igual a la utilizada para la visualización de las publicaciones. Se debe mostrar toda aquella información que se crea pertinente y el medio de pago utilizado por el cliente al momento de efectuarse la compra.

12. Canje y Administración de puntos

Esta funcionalidad, como su nombre lo indica, permite a un cliente consultar sus puntos de usuario frecuente como así también realizar canjes con los puntos que ya tienen asociados.

Los premios y productos ofrecidos por la plataforma serán determinados por los alumnos, como así también las políticas para el vencimiento de los mismos.

13. Generar rendición de comisiones

Funcionalidad utilizada que registra facturas por el cobro de comisiones de ventas de publicaciones a la empresa de espectáculos.

Será necesario que en la factura se registren todas aquellas ubicaciones(entradas) a rendir (compras), detallando el importe de cada venta y en forma separada que importe corresponde a cada comisión.

Deberá tenerse en cuenta que primero se deben rendir las compras más antiguas. Bajo ninguna circunstancia se podrá saltear compras por rendir, los administradores del sistema determinarán cuantas compras quieren rendir.

Una vez generadas las rendiciones de comisión, se pagará a la empresa de espectáculos los importes recaudados por las ventas, descontando las comisiones generadas anteriormente.

14. Listado Estadistico

Esta funcionalidad nos debe permitir consultar el TOP 5 de:

- Empresas con mayor cantidad de localidades no vendidas, dicho listado debe filtrarse por grado de visibilidad de la publicación y por mes-año. Primero se deberá ordenar por fecha y luego por visibilidad.

- Clientes con mayores puntos vencidos.

- Clientes con mayor cantidad de compras, agrupando las publicaciones por empresa.

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 "GD2C2018".

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

- Solapa "General":
  - Nombre de inicio de sesión: "gdEspectaculos2018"
  - Autenticación de SQL Server
  - Contraseña: "gd2018"
  - Base de Datos Predeterminada: GD2C2018.
  - El resto de los parámetros respetar sus valores default.
- Solapa "Funciones del Servidor":
  - Seleccionar "sysadmin"
- Solapa "Asignación de Usuarios":
  - Seleccionar asignar a "GD2C2018"
- 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: "gdEspectaculos2018".

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 significante 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 gdEspectaculos2018 -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 “PalcoNet”. 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: gdEspectaculos2018
  - Password: gd2018
- Nombre de la base de datos: GD2C2018

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 ha sido desarrollada en éste lenguaje será rechazada, sin excepción.

El entorno de desarrollo debe ser Microsoft Visual Studio 2012 o Microsoft Visual C# Studio Express 2012. No podrá ser utilizada otra versión.

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 ser utilizada la reciente versión 2010.

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

Condiciones de aprobación

Testing

El alumno deberá entregar dos componentes:

- Un único script de base de datos (script_creacion_inicial.sql) con todo lo necesario para crear su modelo y cargarlo con datos.
- La aplicación C# "PalcoNet" 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 respeta las buenas prácticas de programación y diseño de bases de datos explicados durante la cursada de la materia.

También deberán ser considerados criterios de performance a la hora de crear relaciones e índices en las tablas.

Consultas SQL

Todas las consultas SQL que haga la aplicación serán evaluadas de acuerdo al standard de programación SQL explicados en clase. La performance de las mismas será tenida en cuenta a la hora de fijar la nota.

Respetar Guía de ABMs

Todo el código y las pantallas creadas en la aplicación Desktop deberá respetar a la perfección los lineamientos especificados en el documento "Guía de ABMs". Aquellos TPs que no respeten las indicaciones en forma total serán rechazados, por más que cumplan la funcionalidad pedida.

Aplicación Desktop

La calidad y orden del código fuente será tenida en cuenta a la hora de fijar la nota. Es obligatorio que existan comentarios de código en todas las secciones principales de implementación.

Deberán crearse componentes de código reutilizable para aquellas porciones de código ejecutadas en muchas secciones de la aplicación. Todo tipo de configuración o parametrización de la aplicación deberá estar centralizado en un solo punto. Aquellos TPs que no respeten estos puntos mencionados serán rechazados, sin continuar su evaluación (ej: los parámetros de conexión a la base de datos).

Fecha de entrega y condiciones

Existe una sola fecha de entrega posible para el TP, pudiendo presentarse hasta 2 veces más (entregas que no tienen fecha fija).

La cantidad de funcionalidad de cada entrega no varía.

- *Entrega unica:*
    - *Día:* 25/11/2018 hasta las 12:00 del mediodía (GMT -3:00 Buenos Aires).
      Los TPs entregados ese día, después de las 12:00 del mediodía se consideran que no son entregados en fecha y le restan 2 oportunidades más de entrega, como se menciona anteriormente. Se considera que la fecha última de entrega sea el día 20/12/2018.

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 25/10/2018 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/1PfeN_WNhfxaO9ksNuDzrV0jag3FadyoZhoJ9egR2jko/edit

Inicialmente el Apéndice se encuentra vacío. A medida que vayan surgiendo dudas sobre el desarrollo del Trabajo Práctico, la cátedra evaluara 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 que no sea la indicada:

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 laboratório 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 2008 Express. Puede ser descargado de la siguiente dirección:

   http://www.microsoft.com/Sqlserver/2008/en/us/express-down.aspx

   Es necesario descargar e instalar dos componentes:
   - Install Microsoft SQL Server 2008 Express Edition
   - SQL Server Management Studio Express

Formato de entrega

Lugar de envío

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

La dirección del mail es:

gestiondedatos.entregas@gmail.com

El asunto del mail debe cumplir con el siguiente formato:

TP2C2018<b><curso><b><nombreGrupo><Nro de grupo>

<b>: espacio en blanco

Ejemplos:
TP2C2018 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:

TP2C2018 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 "PalcoNet"
- ⇒ \data
  - ⇒ Archivo de script de base de datos "script_creación_inicial.sql".

Readme.txt:

Es un archivo de texto plano con los siguientes datos:

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

Estrategia.pdf:

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

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

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

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

\src:

Dentro de este directorio se encuentra la solución entera de Visual Studio del proyecto "PalcoNet". Evitar enviar archivos de SVN, CVS u otros. No enviar archivos ejecutables ni DLLs. Limpiar la solución desde Visual Studio (Proyecto => Limpiar Solución) antes de enviarla.

Además dentro del source deberá existir un archivo de configuración en donde se encuentren los parámetros de conexión a la base de datos, **la fecha que tomará el sistema para funcionar** (se utiliza este criterio para simplificar al alumno el manejo de las fechas y así evitar el cambio de fechas del sistema operativo).

Sin ese archivo de configuración la entrega no será tomada como válida.

\data:

Archivo "script_creación_inicial.sql" con toda la creación del modelo de datos. El archivo debe poder ejecutar perfectamente de una sola vez, sin ningún tipo de error. Todas las sentencias deben estar perfectamente ordenadas para ejecutar correctamente. Cada sentencia debe estar comentada explicando su intención.

Cualquier TP entregado que no cumpla con alguno de los requisitos mencionados en este documento, será rechazado sin ser evaluado, perdiendo una oportunidad de reentrega.