



Universidad Tecnológica Nacional
Facultad Regional Buenos Aires

Gestión de Datos

Trabajo Práctico

1° Cuatrimestre 2011

Venta de Electrodomésticos

Enunciado V1.1



Índice

Índice.....	2
Introducción.....	3
Objetivos generales.....	3
Descripción general	3
Componentes del TP.....	4
Base de Datos	4
Aplicación Desktop.....	4
Modelo de datos.....	4
Requerimientos.....	6
General	6
Base de Datos	6
Aplicación Desktop.....	6
Implementación	15
General	15
Base de Datos	15
Aplicación Desktop.....	17
Restricciones de la solución	18
Condiciones de aprobación	18
Testing.....	18
Modelo de Datos.....	19
Consultas SQL.....	19
Respetar Guía de ABMs.....	19
Aplicación Desktop.....	19
Fecha de entrega y condiciones	19
Sobre los grupos	20
Ayuda y contacto	20
Donde aprender C#.....	21
Sobre la elección de C#.....	21
Obtención de herramientas.....	21
Formato de entrega	22
Lugar de envío	22
Estructura del archivo zip.....	23
Readme.txt:.....	23
Estrategia.pdf:.....	24
\src:	24
\data:	24

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

Una cadena de locales de venta de electrodomésticos solicita a la Universidad Tecnológica Nacional desarrollar un software que permite organizar, gestionar y analizar la enorme cantidad de datos de operaciones que realizan todos los días.

El software debe permitir a los empleados de las distintas sucursales de la empresa realizar las operaciones diarias de venta y pagos de productos. A su vez la aplicación debe permitir la generación de reportes y análisis de datos para su posterior estudio por parte de la gerencia.

Actualmente la empresa posee un sistema informático muy rudimentario, con todos sus datos de operaciones diarias recopilados en forma desordenada.

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

La cátedra provee un script que permite crear una base de datos en el motor SQL Server 2005. Esta base de datos incluye una única tabla 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.

Aplicación Desktop

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

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

Modelo de datos

Todos los datos provistos por la cátedra se encuentran en una única tabla de la base de datos de SQL Server, llamada “maestra”.

Los datos de esta tabla pertenecen a un dominio de negocio de ventas, con información de clientes, productos, facturas, etc.

La aplicación será utilizada por dos grupos distintos de usuarios: vendedores y analistas. Los primeros son los encargados de vender los productos a los clientes desde las distintas sucursales de la empresa. Los vendedores cargan al sistema toda la información de venta y pagos de productos.

Los analistas realizan operaciones de consultas globales sobre todas las operaciones realizadas en periodo de tiempo. En base a este análisis de datos establecen las nuevas políticas de marketing y negocio.

Los datos de la tabla maestra responden a las siguientes reglas de negocio:

- La empresa posee distintas sucursales distribuidas por todo el país. La cantidad de sucursales de la empresa es fija.
- Cada sucursal posee sus propios vendedores. Un vendedor solo trabaja en una sucursal y puede realizar operaciones en esa sucursal.

- Los analistas trabajan en la sede central de la empresa, que se encarga de tareas administrativas. Puede consultar datos de todas las sucursales en forma simultánea.
- La empresa posee diversos productos disponibles para vender. Todos los productos están disponibles para la venta en todas las sucursales.
- Todos los productos están organizados en categorías. Un producto pertenece a una única categoría.
- Existe una jerarquía de categorías de productos. Desde las categorías más generales, hasta las categorías más específicas. Las categorías existentes no pueden variar pero pueden agregarse nuevas categorías y nuevos niveles de sub-categorías.
- Cada sucursal maneja su propio stock de productos. No puede venderse un producto sin que haya stock.
- La empresa lleva un registro de los clientes que compran sus productos. La primera vez que un cliente realiza una compra, se registran todos sus datos.
- Los clientes pueden comprar productos en cualquier sucursal.
- Los clientes pueden comprar los productos en dos modalidades de pago: en efectivo o en cuotas. Al comprar en efectivo se abona el precio completo del producto en un único pago. El pago se hace en el mismo momento que en que se efectuó la compra.
- Al comprar en cuotas se hacen N pagos de igual monto hasta completar el valor total del producto. No se cobran intereses por el pago en cuota. No hay un límite de monto para pagar en cuotas. Las cuotas van desde 2 pagos hasta 12 pagos. Ninguna compra puede hacerse en más de 12 cuotas. Un cliente puede adelantar más de una cuota pendiente en un mismo pago.
- Los precios de los productos sufren modificaciones a lo largo del tiempo.
- Al realizar una venta se confecciona una factura. En una misma compra el cliente puede adquirir más de un producto. Puede adquirir incluso varios productos iguales.
- A las ventas se le puede aplicar un descuento. El descuento es un porcentaje que decide el vendedor. Este porcentaje se aplica al total del monto de toda la venta, sumando todos sus productos. El porcentaje puede variar dentro del rango de 0% a 30%.
- Cada empleado de la empresa, tanto vendedor como analista, posee un usuario para acceder al sistema. Los usuarios de los vendedores solo pueden operar sobre la sucursal en la que trabajan. Los usuarios de los analistas pueden operar en cualquier sucursal.
- A cada usuario del sistema se le puede asignar uno o varios roles, referidos a la seguridad informática. Cada rol está compuesto por un conjunto de funcionalidades. Existe una lista de funcionalidades fija en todo el sistema. Un usuario solo puede utilizar las funcionalidades que dictan sus roles.
- Los usuarios poseen un username y un password. Para poder utilizar la aplicación deben pasar por un proceso de Login. Una vez que el usuario ingreso al sistema, podrá operar con todas las funcionalidades que le permitan sus roles asignados.

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.

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

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

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

Aplicación Desktop

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

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

ABM de Empleado

Funcionalidad para poder crear, modificar y eliminar empleados del sistema.

Crear un empleado implica cargar los siguientes datos:

- Nombre
- Apellido
- DNI
- Mail
- Teléfono
- Dirección calle, número y piso/dpto.
- Provincia (selección acotada)
- Sucursal (selección acotada)
- Tipo de empleado (selección acotada)

Todos los datos mencionados anteriormente son obligatorios. Los empleados del tipo Analista también tienen Provincia, pero este valor no implica ninguna restricción sobre sus operaciones.

No pueden existir en el sistema dos empleados con el mismo DNI. La aplicación deberá controlar esta restricción e informar debidamente al usuario.

En la modificación solo se pueden cambiar los datos personales del empleado. No se pueden modificar los siguientes datos:

- DNI
- Provincia (selección acotada)
- Sucursal (selección acotada)
- Tipo de empleado (selección acotada)

La eliminación de un empleado implica una baja lógica del mismo. El empleado se inhabilita pero sus datos siguen existiendo en la base de datos. Un empleado inhabilitado no podrá acceder al sistema ni realizar operaciones. Tampoco se le puede asignar un usuario. Se debe poder volver a habilitar un empleado deshabilitado desde la sección de modificación.

Para elegir que empleado 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)
- Provincia (selección acotada)
- Sucursal (selección acotada)
- Tipo de empleado (selección acotada)

ABM de Rol

Funcionalidad para poder crear, modificar y eliminar roles 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. No puede poseer funcionalidades repetidas.

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

- ABM de Empleado
- ABM de Rol
- ABM de Usuario
- ABM de Cliente
- ABM de Producto
- Asignación de stock
- Facturación
- Efectuar Pago
- Tablero de Control
- Clientes Premium
- Mejores Categorías

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

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 que ya posee así como también permitir agregarle nuevas funcionalidades.

La eliminación del rol implica una baja lógica del mismo. El rol se inhabilita. No puede asignarse un rol inhabilitado a un usuario. 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 que rol se desea modificar o eliminar se debe mostrar un listado con todos los roles existentes en el sistema.

ABM de Usuario

Funcionalidad para poder crear, modificar y eliminar usuarios del sistema.

Crear un empleado implica cargar los siguientes datos:

- Empleado (selección por buscador)
- Username
- Password
- Listado de roles (selección especial)

Todos los datos mencionados anteriormente son obligatorios. Un empleado solo puede poseer un usuario. No se puede asignar un usuario a un empleado inhabilitado.

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

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

A un usuario se le deben poder asignar un conjunto de roles. Un usuario no puede tener asignado dos veces un mismo rol. Se debe mostrar una tabla visual con todos los roles disponibles del sistema. Por cada rol debe existir una casilla que indique si el usuario actual posee o no ese rol.

Para elegir a que empleado se le desea asignar un usuario se debe presentar un buscador con listado, que permita filtrar simultáneamente por alguno o todos los siguientes campos:

- Nombre de empleado (texto libre)

- Apellido de empleado (texto libre)
- DNI de empleado (texto libre exacto)
- Provincia (selección acotada)
- Sucursal (selección acotada)
- Tipo de empleado (selección acotada)

En la modificación de un usuario solo se puede modificar el password y el listado de roles. De la lista completa de roles del sistema, esta funcionalidad debe mostrar para cada rol, indicando si está asignado al usuario o no. Y debe permitir modificar esta asignación.

En la modificación de un usuario también podrá rehabilitarse, en caso de que éste se encontrase inhabilitado por una acción anterior.

La eliminación de un usuario implica una baja lógica del mismo. El usuario se inhabilita pero sus datos siguen existiendo en la base de datos. Un usuario inhabilitado no podrá acceder al sistema.

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

- Username (texto libre)
- Nombre del empleado (texto libre)
- Apellido del empleado (texto libre)
- DNI del empleado (texto libre exacto)
- Provincia del empleado (selección acotada)
- Sucursal del empleado (selección acotada)
- Tipo de empleado (selección acotada)

Login y Seguridad

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

El proceso de Login pedirá al usuario su Username y su Password. Si Login es correcto, el usuario podrá acceder al sistema. Al acceder al mismo, la aplicación solo deberá generar y mostrar las entradas de menú disponibles para este usuario, según los roles del mismo. El usuario no debe ni siquiera ver las funcionalidades a las que no posee acceso.

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

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

ABM de Cliente

Funcionalidad para poder crear, modificar y eliminar clientes del sistema.

Crear un cliente implica cargar los siguientes datos:

- Nombre
- Apellido

- DNI
- Mail
- Teléfono
- Dirección calle, número y piso/dpto.
- Provincia (selección acotada)

Todos los datos mencionados anteriormente son obligatorios.

No pueden existir en el sistema dos clientes con el mismo DNI. La aplicación deberá controlar esta restricción e informar debidamente al usuario. Tampoco puede repetirse el DNI con un empleado.

En la modificación de un cliente se pueden alterar todos sus datos a excepción del DNI.

La eliminación de un cliente implica una baja lógica del mismo. El cliente se inhabilita pero sus datos siguen existiendo en la base de datos. Un cliente inhabilitado no podrá realizar una compra. Se debe poder volver a habilitar un cliente inhabilitado desde la sección de modificación.

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

- Nombre (texto libre)
- Apellido (texto libre)
- DNI (texto libre exacto)
- Provincia (selección acotada)

ABM de Producto

Funcionalidad para poder crear, modificar y eliminar productos del sistema.

Crear un producto implica cargar los siguientes datos:

- Código de producto
- Nombre o Marca
- Descripción
- Categoría (selección especial)
- Precio en pesos

Todos los datos mencionados anteriormente son obligatorios.

El código de producto debe ser asignado automáticamente por el sistema. No pueden existir dos productos con igual código de producto.

La categoría de un producto debe seleccionarse visualmente, navegando una estructura de árbol en forma jerárquica.

El precio de todos los productos debe ser mayor a cero.

En la modificación de un producto se pueden alterar todos sus datos a excepción de la categoría y el código de producto.

El precio de un producto puede variar en el tiempo. Sin embargo debe quedar registrado en cada factura el precio exacto al cuál se vendió en un momento determinado.

La eliminación de un producto implica una baja lógica del mismo. El producto se inhabilita pero sus datos siguen existiendo en la base de datos. Un producto inhabilitado no podrá venderse. Se debe poder volver a habilitar un producto inhabilitado desde la sección de modificación.

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

- Código de producto (texto libre exacto)
- Nombre o Marca (texto libre)
- Categoría (selección especial)
- Precio (texto libre numérico)

El filtro de búsqueda para precio debe poder especificarse como un rango “desde-hasta” inclusive. Pueden cargarse ambos límites del rango, uno solo o ninguno.

La categoría de un producto debe seleccionarse visualmente, navegando una estructura de árbol en forma jerárquica.

Asignación de Stock

Funcionalidad para poder modificar el stock de un producto para cada sucursal.

Se debe mostrar una pantalla que permita elegir:

- Producto (selección)
- Sucursal (selección)
- Auditor (selección)

Luego de elegido estos valores la aplicación deberá mostrar el stock actual del producto especificado, para la sucursal especificada. El usuario podrá asignar una cantidad de stock. El stock debe ser un valor mayor o igual a cero.

El campo Auditor debe permitir seleccionar un usuario de tipo Analista. En cada asignación de stock debe participar un Analista para auditar la operación.

Para elegir que usuario Analista se desea cargar en el campo Auditor, se debe presentar un buscador con listado, que permita filtrar simultáneamente por alguno o todos los siguientes campos:

- Nombre de empleado (texto libre)
- Apellido del empleado (texto libre)
- DNI del empleado (texto libre exacto)

El buscador anterior solo debe mostrar usuarios del tipo Analista.

Para elegir a que producto se le desea asignar stock se debe presentar un buscador con listado, que permita filtrar simultáneamente por alguno o todos los siguientes campos:

- Código de producto (texto libre exacto)
- Nombre o Marca (texto libre)
- Categoría (selección especial)

La categoría de un producto debe seleccionarse visualmente, navegando una estructura de árbol en forma jerárquica.

Facturación

Funcionalidad que permite realizar una venta.

Se debe mostrar una pantalla que permita elegir:

- Provincia (selección acotada)
- Sucursal (selección acotada)
- Cliente (selección por buscador)

- Listado de Productos a comprar (selección por buscador)
- Porcentaje de descuento a aplicar
- Forma de pago (selección acotada)

La provincia y sucursal deberán ser campos sin posibilidad de modificación para un usuario del tipo vendedor.

El vendedor de esta operación corresponde al usuario que está utilizando la aplicación.

Al realizar la venta se debe generar una factura. La factura de registrar todos los datos propios de la venta. Debe registrar los precios originales de cada producto adquirido, así como también el descuento aplicado y el monto total resultante. La fecha de la factura corresponderá al momento en que se realiza la operación.

El porcentaje de descuento a aplicar es un número entre 0% y 30%.

La aplicación debe mostrar el monto final de toda la compra antes de aceptar la operación.

Si la forma de pago elegida es en efectivo se deberá también registrar el pago de la totalidad del monto. Si la forma de pago elegida es en cuotas, el usuario deberá cargar un nuevo campo con la cantidad de cuotas elegidas para pagar. Esta cantidad es un número entre dos y doce.

Para elegir que productos se desean agregar a la compra se debe presentar un buscador con listado, que permita filtrar simultáneamente por alguno o todos los siguientes campos:

- Código de producto (texto libre exacto)
- Nombre o Marca (texto libre)
- Categoría (selección acotada)
- Precio (texto libre numérico)

El filtro de búsqueda para precio debe poder especificarse como un rango “desde-hasta” inclusive. Pueden cargarse ambos límites del rango, uno solo o ninguno.

La categoría de un producto debe seleccionarse visualmente, navegando una estructura de árbol en forma jerárquica.

No podrá seleccionarse un producto que no posea stock suficiente para esa sucursal.

Los productos, el usuario y el cliente deben estar habilitados para poder efectuar la compra.

Para elegir Cliente realiza la compra se debe presentar un buscador con listado, que permita filtrar simultáneamente por alguno o todos los siguientes campos:

- Nombre de cliente (texto libre)
- Apellido de cliente (texto libre)
- DNI del cliente (texto libre exacto)
- Provincia (selección acotada)

Efectuar Pago

Funcionalidad que permite registrar el pago de una compra entera o una cuota realizado por un cliente.

Se debe mostrar una pantalla que permita elegir:

- Provincia (selección acotada)
- Sucursal (selección acotada)

- Cliente (selección por buscador)
- Factura (selección acotada)
- Cantidad de cuotas a pagar

La provincia y sucursal deberán ser campos sin posibilidad de modificación para un usuario del tipo vendedor.

Para elegir Cliente realiza el pago se debe presentar un buscador con listado, que permita filtrar simultáneamente por alguno o todos los siguientes campos:

- Nombre de cliente (texto libre)
- Apellido de cliente (texto libre)
- DNI del cliente (texto libre exacto)
- Provincia del cliente (selección acotada)

Una vez seleccionado el cliente, el sistema deberá detectar automáticamente todas las facturas pendientes de pago del mismo, junto con la cantidad de cuotas pendientes a pagar. El campo Factura debe permitir elegir una factura de lista de facturas pendientes de pago de ese cliente.

El campo Cantidad de cuotas a pagar debe permitir elegir la cantidad de cuotas que se desean pagar en esta operación. Esa cantidad debe ser mayor a cero y no puede superar la cantidad máxima de cuotas pendientes de pago.

El monto a abonar deberá ser calculado automáticamente por el sistema en base a la cantidad de cuotas a pagar. Este valor se debe mostrar al usuario antes de confirmar la realización del pago.

Toda la información referida al pago deberá quedar registrada en el sistema. Se deberá registrar también el empleado que ingresó el pago al sistema, junto la fecha de realización. Esta información registrada deberá ser similar cuando se realiza un pago en efectivo desde la funcionalidad de Facturación.

Tablero de Control

Funcionalidad que permite visualizar distintos valores críticos para el análisis de estado de la empresa.

Se debe mostrar una pantalla que permita seleccionar:

- Sucursal (selección acotada)
- Año a analizar (selección de fecha)

El año debe ser un valor comprendido entre 1995 y 2100.

Luego en base a esa sucursal elegida se deben calcular y mostrar una serie de valores. Cada valor deberá mostrarse uno debajo del otro, con su nombre a la derecha y su valor a la izquierda. Los valores a calcular son los siguientes:

- Total de ventas: cantidad de facturas que se generaron en esa sucursal durante el año elegido.
- Total de facturación: monto total que se facturó en esa sucursal durante el año elegido. Facturado no implica que se hayan efectuado los pagos
- Proporción de forma de pago: porcentaje de facturas realizadas en efectivo y porcentaje de facturas realizadas en cuota, respecto del total de facturas generadas en esa sucursal durante el año elegido. Mostrar un valor seguido del otro, separado por guión: X – Y.
- Mayor factura: máximo monto registrado en una factura que se haya generado en esa sucursal durante el año elegido. Mostrar el monto.

- Mayor deudor: cliente que adeuda pagar el mayor monto (sumando todas sus deudas), tomando solo las deudas contraídas en esa sucursal durante el año elegido. Solo se consideran los pagos realizados en el año elegido. Mostrar: Nombre, Apellido, DNI. Mostrar “ninguno” en caso de no existir clientes con deudas.
- Vendedor del año: vendedor que más monto total facturó en ese año y sucursal. Mostrar: Nombre, Apellido, DNI.
- Producto del año: producto que más se vendió, en cantidad, en ese año y sucursal. Mostrar: Código de producto, Nombre de producto, Categoría.
- Faltante de stock: producto al que le faltó stock por mayor cantidad de días, en ese año y sucursal. Mostrar: Código de producto y cantidad de días sin stock.

Cientes Premium

Funcionalidad que permite visualizar los mejores clientes de la empresa.

Se debe mostrar una pantalla que permita seleccionar:

- Sucursal (selección acotada)
- Año a analizar (selección de fecha)

El año debe ser un valor comprendido entre 1995 y 2100.

En base a los valores elegidos se debe mostrar un listado en pantalla con los 30 clientes que más compraron, en monto, en la sucursal elegida durante el año elegido. El listado se debe ordenar en forma descendente por el monto.

Se deben mostrar las siguientes columnas:

- Nombre del cliente
- Apellido del cliente
- DNI del cliente
- Monto total acumulado en ese año y sucursal
- Total de productos adquiridos en ese año y sucursal
- Fecha de la última compra realizada en ese año y sucursal
- DNI del último empleado que le vendió, en ese año y sucursal.

Mejores Categorías

Funcionalidad que permite visualizar las categorías de productos que más vendieron.

Se debe mostrar una pantalla que permita seleccionar:

- Sucursal (selección acotada)
- Año a analizar (selección de fecha)

El año debe ser un valor comprendido entre 1995 y 2100.

En base a los valores elegidos se debe mostrar un listado en pantalla con todas las categorías de mayor nivel jerárquico, ordenado descendientemente por monto total facturado de todos los productos de cada categoría, en el año y sucursal elegida.

Se deben mostrar las siguientes columnas:

- Nombre de la categoría
- Cantidad total de sub-categorías, acumulando todos sus niveles inferiores, de esta categoría.

- Monto total facturado de todos los productos de esa categoría, en ese año y sucursal.
- Código y nombre del producto más vendido, en cantidad, dentro de la categoría, en ese año y sucursal.
- Código, nombre y monto del producto que más monto total facturó, dentro de la categoría, en ese año y sucursal.
- Código, nombre y precio del producto que se vendió más caro, precio unitario, en esa categoría, en ese año y sucursal.
- Nombre y apellido del vendedor que más vendió, en cantidad, dentro de la categoría, en ese año y sucursal.

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 2005 con las siguientes consideraciones:

- El nombre de la instancia del motor de base de datos a instalar debe llamarse “SQLSERVER2005”. 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 2005. 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 “GD1C2011”.

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

- Solapa “General”:
 - Nombre de inicio de sesión: “gd”

- Autenticación de SQL Server
- Contraseña: “gd2011”
- Base de Datos Predeterminada: GD1C2011.
- El resto de los parámetros respetar sus valores default.
- Solapa “Funciones del Servidor”:
 - Seleccionar “sysadmin”
- Solapa “Asignación de Usuarios”:
 - Seleccionar asignar a “GD1C2011”
- 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: “gd2011”.

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

- gd_esquema.Schema.sql: Este archivo genera un esquema llamado “gd_esquema” dentro de la base de datos y lo asigna al usuario “gd”.
- gd_esquema.Maestra.Table.sql: Este archivo crea la tabla principal del trabajo práctico y la carga con los datos correspondientes. El archivo posee un volumen significativo (aprox. 3 GB) 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\SQLSERVER2005 -U gd -P gd2011 -i gd_esquema.Schema.sql,gd_esquema.Maestra.Table.sql -a 32767 -o resultado_output.txt

Luego de cargados todos los datos de la tabla maestra, el alumno deberá crear su propio esquema dentro de la base de datos. El nombre del esquema deberá ser igual al nombre del grupo registrado en la materia (el proceso de registración se explica más adelante). El nombre del esquema debe ser en mayúsculas, sin espacios y separado por guiones bajos. Ejemplo “Los mejores” debe ser “LOS_MEJORES”.

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

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

Aplicación Desktop

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

Para ejecutar esta aplicación es necesario instalar Visual Studio 2008 con el Framework de .NET 3.5. La versión Express posee la funcionalidad necesaria como para desarrollar el Trabajo Práctico.

La aplicación template se denomina “VentaElectrodomesticosDesktop”. 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\SQLSERVER2005
- Utilizar autenticación de SQL Server:
 - Nombre de Usuario: gd
 - Password: gd2011
- Nombre de la base de datos: GD1C2011

La aplicación siempre debe conectarse a localhost. En caso de que el alumno se conecte a otra dirección, deberá cambiarlo a la hora de entregar su TP para corregir.

La aplicación deberá contar con un usuario de sistema ya creado y listo para ser utilizado, con las siguientes características:

- Username: admin.
- Password: w23e
- Rol:
 - Nombre: Administrador General
 - Funcionalidades: todas las existentes

Este usuario de aplicación debe ser generado en forma automática dentro del archivo “script_creacion_inicial.sql” y quedar listo para ser utilizado por la aplicación Desktop.

Restricciones de la solución

El lenguaje de programación utilizado deberá ser únicamente C# utilizando el Framework .NET 3.5. Cualquier otra implementación que no halla sido desarrollado en éste lenguaje será rechazada, sin excepción.

El entorno de desarrollo debe ser Microsoft Visual Studio 2008 o Microsoft Visual C# Studio Express 2008. No podrá ser utilizada la reciente versión 2010.

El motor de base de datos deberá ser Microsoft SQL Server 2005. Tanto la versión Express como la full sirven para realizar el trabajo. No podrá ser utilizada la reciente versión 2008.

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# “VentaElectrodomesticosDesktop” 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

El TP puede entregarse a partir del día **02/05/2011** a las 12 del mediodía.

La fecha final de recepción de trabajos es el día **17/06/2011** hasta las 12 del mediodía. No se recibirán trabajos prácticos después de la fecha y hora final, sin excepción.

El TP se entrega una vez y si no se aprueba puede presentarse una vez más, siempre y cuando no se llegue a la fecha de entrega final. En total solo existen **dos** oportunidades para entregar el TP, siempre y cuando el alumno no se pase de la fecha de entrega final.

La cantidad de funcionalidad de cada entrega no varía.

La nota decrece con cada entrega realizada.

Los trabajos prácticos entregados deberán contar con la totalidad de las funcionalidades pedidas. Solo tendrán posibilidad de reentrega aquellos TPs desaprobados que tuvieron problemas o bugs en alguna funcionalidad particular que se consideraba terminada. Los TPs que sean entregados con funcionalidades sin terminar,

o a medio hacer, serán rechazados sin continuar su evaluación y sin posibilidad a reentrega.

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=dG16aEltMHclX2hPN3U2YTVoVGxfeUE6MA>

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”.

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/11cPz0LhFvEw9L7I9svuagvCb-VSIEsAZ3vjUGZb7B-U/edit#>

Inicialmente el Apéndice se encuentra vacío. A medida que vayan surgiendo dudas sobre el desarrollo del Trabajo Práctico, la cátedra evaluará agregar

consideraciones generales al documento. Es obligación del alumno revisar este documento periódicamente.

Donde aprender C#

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

<http://www.programacion.net/tutorial/csharp/> de José Antonio González Seco

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 2010:

1. **Microsoft Visual Studio Professional 2008:** esta versión puede ser obtenida con licencia universitaria completa, gracias a un convenio de Microsoft con la UTN. Dirigirse al laboratorio de Microsoft ubicado en la sede Medrano (planta baja, hacia la derecha, mirando desde la entrada de la facultad hacia adentro). La versión ofrecida es en español y contiene la ayuda completa (MSDN). Para encargarla es necesario llevar un DVD y completar un formulario. Es posible que el programa halla que encargarlo y pasarlo a buscar otro día.
2. **Microsoft Visual C# 2008 Express Edition:** Existe una versión gratuita del IDE llamada *Visual C# 2008 Express Edition*, la cual posee todas las herramientas necesarias para realizar el TP. Esta se encuentra disponible en:

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

El motor de base de datos a utilizar es SQL Server 2005 Express. Puede ser descargado de la siguiente dirección:

<http://www.microsoft.com/Sqlserver/2005/en/us/express-down.aspx>

Es necesario descargar e instalar dos componentes:

- Install Microsoft SQL Server 2005 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:

TP1C2011<curso><nombreGrupo>

: espacio en blanco

Ejemplos:

TP1C2011 k9999 LOS_MEJORES

(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:

TP1C2011 k9999 LOS_MEJORES.zip123
(Respetar los 2 espacios en blanco existentes)

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

El cuerpo del mail debe contener lo siguiente:

Grupo:

Curso:

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

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

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

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

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

Estructura del archivo zip

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

⇒ \
⇒ Readme.txt
⇒ Estrategia.pdf
⇒ \src
⇒ Solución entera de Visual Studio de “VentaElectrodomesticosDesktop”
⇒ \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.

\src:

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

\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.