

## Trabajo práctico no. 4

### Aplicaciones de Internet

#### Problema (base de datos: Pedidos-Clientes)

1 - Sea una distribuidora de artículos escolares la cual tiene una base de datos donde guarda datos de los pedidos realizados por los clientes e información relacionada con el cumplimiento de los mismos.

De los clientes se guarda: clave, nombre y domicilio. De los artículos se guarda: clave, nombre, precio (por caja, etc.), cantidad actual en el inventario, cantidad mínima que debe haber del artículo y cantidad a pedir si la cantidad actual es menor al mínimo en existencia.

Por cada pedido se conserva: folio, fecha del pedido, fecha en la que el pedido se satisfizo completamente, monto total del pedido, saldo que debe el cliente a la distribuidora por el pedido, saldo que la distribuidora le debe al cliente por entrega de facturas y qué cliente hizo el pedido. Cada pedido tiene el detalle de los artículos solicitados incluyendo la cantidad pedida y la cantidad que ya se entregó al cliente.

La distribuidora tiene empleados que pueden levantar los pedidos; de ellos se debe guardar: clave, nombre y categoría. También se debe saber qué pedidos levantó un empleado. Cabe aclarar que los clientes también pueden ordenar pedidos por Internet por lo que no todo pedido es levantado por un empleado.

Para cada pedido también se registran los pagos que ha hecho el cliente para cubrir el monto del mismo. Por cada pago se guarda: clave, fecha y monto. Cuando se hace un pedido al menos debe hacerse un pago en ese momento, el cual puede cubrir total o parcialmente el monto del pedido.

Por cada pedido la distribuidora emite facturas para el cliente. Una factura también puede cubrir total o parcialmente el monto total del pedido. Normalmente se emitirá más de una factura por pedido cuando el cliente no cubra el monto del pedido en una sola vez, o cuando la distribuidora no satisfaga en una sola vez todas las cantidades de todos los artículos del pedido. De las facturas se debe conservar: folio, fecha de emisión y monto que cubre.

Cada que se entrega al cliente una cantidad de un artículo, hay que descontarla de sus existencias actuales. Si la cantidad actual restante es menor que la cantidad mínima permitida o si queda en cero, hay que emitir y guardar una orden de reabastecimiento con: clave, cantidad a pedir (el valor de  $\text{Artículos.CantPedir}$ ), monto a pagar (el 70% de  $\text{Artículos.CantPedir} * \text{Artículos.Precio}$ ), fecha de la solicitud ( $\text{Reabast.FechaPed}$ ), fecha de recepción de la cantidad solicitada ( $\text{Reabast.FechaEnt}$ ) y qué artículo se solicitó. Si la cantidad de unidades recibida es menor a la solicitada, se genera una nueva orden de reabastecimiento por la cantidad faltante del artículo.

Finalmente, para cada usuario del sistema (cliente, empleado de base o gerente) se debe tener una cuenta (RFC) y una contraseña para que puedan acceder al mismo; estos datos deben guardarse, junto con el tipo de usuario de que se trate.

## 2 – Elabora los siguientes triggers en la base de datos de Pedidos-Clientes:

- a. Define un trigger de actualización sobre *Artículos* de tal manera que cuando se actualice *CantAct*, se inserte una orden de reabastecimiento en *ReAbast*, si  $CantAct < CantMín$ . Para generar esta nueva orden:
  - El folio debe ser igual al folio mayor más uno.
  - La cantidad a pedir es el valor que está en  $Artículos.CantPedir$  para el artículo a reabastecer.
  - El monto es el 70% de  $Artículos.CantPedir * Artículos.Precio$ .
  - El valor en *FechaPed* será el de la fecha la actual.
- b. Define un trigger de inserción sobre *Pagos* de tal manera que cuando se agregue un nuevo pago, se inserte una nueva factura en *Facturas*. Para generar esta nueva factura:
  - El folio de la factura debe ser igual al folio mayor más uno.
  - El monto de la factura debe ser igual al monto del pago nuevo.
  - Se debe actualizar *SaldoCli* en *Pedidos* como corresponda, una vez que se ha agregado el nuevo pago.
  - Se debe actualizar *SaldoFacs* en *Pedidos* como corresponda, una vez que se ha generado la nueva factura.

## 3 – Utilizando la base de datos de Pedidos-Clientes elabora una aplicación de Internet, con Visual C#, considerando lo que se indica a continuación:

- a. La primera página de la aplicación debe ser la que pregunte la cuenta de un usuario (cliente o empleado), y su contraseña, para dar acceso a las demás páginas de la aplicación. Si hay error, se deben mostrar mensajes adecuados y no permitir el avance hasta que ambos datos se den de manera correcta. Una vez que los datos de ingreso sean correctos, en otra página se deben mostrar ligas (una especie de menú, amigable) para poder ir a cualquiera de las páginas que se describen a continuación, según el tipo de usuario registrado. Observa que no todas las páginas podrán ser accedidas por todos los tipos de usuario.
- b. Añade una página que muestre, por medio de un control Table (simple), los datos de un cliente que entró al sistema: clave, nombre y domicilio. También se debe mostrar, por medio de un DropDownList (DDL), la clave de los pedidos que tiene registrados este cliente (en orden ascendente). Al seleccionar la clave de un pedido deben mostrarse (en la misma página):
  - i. En un Table (simple): el número y la fecha del pedido, su total a pagar y el saldo que se resta por cubrir tanto de parte del cliente, como de parte de la empresa por emisión de facturas. Si no tiene pagos, mostrar un mensaje adecuado. Si el pedido ya está satisfecho totalmente, mostrar un mensaje adecuado también.
  - ii. En un GridView, por cada producto del pedido: nombre, cantidad pedida y cantidad entregada.
  - iii. En un GridView, los datos de cada pago efectuado por el cliente para ese pedido: clave, fecha y monto del pago.

Si el usuario registrado fue un empleado de base, deben verse en el DropDownList las claves de los pedidos que él levantó y debe poder verse la misma información descrita en los puntos i a iii anteriores.

- d. Si el empleado registrado fue el gerente, en la página anterior debe verse un DDL (al inicio de la misma) para ver el nombre de todos los clientes. Al elegir un cliente, debe verse todo lo comentado en los puntos i a iii del inciso *b*.
- e. Añade otra página la cual podrá ser usada tanto por los clientes como por los empleados (pero no por el gerente) para dar de alta pedidos. El dar de alta un pedido también implica dar de alta las tuplas que van en la tabla de *Detalle*. Coloca controles necesarios en la página para poder realizar las altas. Al dar de alta un pedido, se deben poder dar de alta todos los artículos que se quiera para el mismo, sin cambiar de página.

Sólo se debe teclear aquella información que no se conoce, que no puede obtenerse de otra parte de la base de datos o que no puede calcularse. Por ejemplo, la cantidad pedida de un artículo es un dato que debe teclearse, sin embargo, el folio de un pedido es un dato que puede generarse o la clave de un artículo es un dato que puede obtenerse de *Artículos*.

Muestra en un Table los datos del pedido; también usa un GridView para mostrar los datos de cada registro de detalle del pedido. Estos controles deben actualizarse cada que se pida un artículo dentro del pedido.

Actualiza los campos que corresponda en las tablas involucradas en este proceso, por ejemplo, al pedir unidades de un artículo, se debe actualizar *Artículos.CantAct* para dicho artículo. Cuida también errores básicos como: montos negativos, fechas incorrectas, etc.

- f. Añade una página la cual sólo podrá ser usada por los empleados de la empresa (pero no por el gerente). Esta página se usará para dar de alta los pagos que hace un cliente. La página contendrá los siguientes elementos:
- Un DropDownList en el cual sólo se verán los nombres de los clientes para los cuales ha levantado pedidos el empleado registrado actualmente en el sistema.
  - Otro DropDownList el cual aparecerá en el momento que el empleado seleccione el nombre de un cliente; en éste se verán sólo las claves de los pedidos del cliente que hayan sido levantados por el empleado.
  - Un Table el cual se mostrará al momento de seleccionar un pedido; en él se verán: la fecha del pedido, su total a pagar, la suma acumulada de los montos de los pagos que se han realizado para el mismo y el saldo que resta por cubrir. Si no tiene pagos, muestra un mensaje adecuado.
  - Un GridView, que también se mostrará al momento de seleccionar un pedido, con los datos de cada pago realizado por el cliente para ese pedido: clave, fecha y monto del pago (posiblemente aquí puedas reutilizar código).
  - Botones y controles (cualquier control que desees, o inclusive, directamente sobre el GridView, si es que lo quieres investigar) para dar de alta o de baja un pago, o para modificar la fecha o monto de un pago que ya existe. Cuida lo siguiente:
    - FechaPago<sub>j</sub> >= FechaPago<sub>i</sub> >= FechaPedido (para cualquier  $j > i$ ).
    - Pedidos.Monto >= SaldoCli >= 0, en cualquier momento.
    - Errores básicos como: montos negativos, fechas incorrectas, etc.
- Actualiza el GridView de los pagos (una vez que el nuevo se haya dado de alta) y el Table del punto *iii* anterior. Los controles mencionados en este punto sólo se deberán ver hasta que se dé clic en un botón: *Agregar pago*.

Recuerda que tu base de datos tiene triggers definidos que se activarán al dar las altas. Verifica que el resultado es correcto después de que un trigger se dispara.

Tu aplicación deberá ser **robusta** por lo que deben controlarse errores que ocurran durante el proceso, de manera que la aplicación no aborte y haya opciones de continuar hasta que el usuario finalice. En donde corresponda, utiliza **los controles de validación** dados por Visual C#.

Tu aplicación también deberá ser **amigable**, esto es, debe “verse” bien ante el usuario (en colores, distribución de controles, etc.), procurando guiarlo en las acciones que debe realizar, sea habilitando/deshabilitando botones o por medio de indicaciones en etiquetas.

También procura "**optimizar**" código, en el sentido de que si hay segmentos de código muy parecidos, elabora rutinas adecuadas (con parámetros) para que sólo se escriba una vez dicho código y se ejecute con los parámetros adecuados cada que se necesite.

Asimismo, agrega controles para realizar acciones "obvias" en tu aplicación aunque éstas no estén especificadas explícitamente; por ejemplo, una liga para pasar de una página a otra, etc.

Finalmente, **tu programa debe tener comentarios** al inicio de cada método explicando a grandes rasgos qué es lo que éste hace, así como comentarios distribuidos a lo largo del código de cada método.

**Entregables:** La aplicación en formato .rar, la cual se enviará por email al profesor.

**Fecha de entrega:** un día antes del segundo examen parcial, a las 22 hrs. (máximo).