Proyecto

Se solicita que cree una aplicación en C# que permita realizar las ventas en un punto de venta de un Restaurante.

Requerimiento de la Base de Datos

Se debe tener como mínimo las siguientes tablas:

* **tbl\_usuario**: Se almacenan los usuarios de la aplicación. Los campos mínimos son los siguientes:
  + id\_usuario: llave primaria
  + usuario: Usuario de acceso
  + contrasenna: es la contraseña del usuario.
  + nombre: es el nombre y apellido del usuario
  + cargo: es el cargo que ocupa en el restaurante
  + activo: indica si esta activo en la base de datos
* **tbl\_mesa**: almacena la información referente a la mesa:
  + numero: es el número de la mesa. Llave primaria
  + descripcion: cualquier descripción de la mesa
  + estado: indica si la mesa está ocupada o está libre. Se almacenan los valores “O” para ocupada y“L” para Libre. El valor predeterminado es “L”
  + activo: indica si esta activo en la base de datos
* **tbl\_categoria**: Almacena las categorías de productos que dispondrá el menú del restaurante
  + id\_categoria: Es el id único de cada categoría. Llave primaria.
  + nombre: es el nombre de la categoría
  + Imagen: indica la ruta de la imagen de la categoría.
  + activo: indica si esta activo en la base de datos
* **tbl\_producto**: Almacena todos los productos o ítems de venta en el restaurante. Cada producto pertenece a una categoría.
  + id\_producto: es el id único de cada producto.
  + descripcion: es una descripción breve del producto.
  + precio: es el precio del producto.
  + Cantidad: es la cantidad en existencia del producto
  + id\_categoria: indica la categoría al cual está asignado el producto.
  + activo: indica si esta activo en la base de datos
* **tbl\_orden\_encabezado**: Almacena el encabezado de la orden.
  + id\_orden: id único de la orden, debe ser autoincremental
  + fecha\_hora: es la fecha actual de la orden.
  + total: es el total de la orden.
  + id\_usuario: es el usuario que crea el pedido.
  + numero\_mesa: Es el número de la mesa en la que se realizó el pedido.
  + id\_cliente: Es el id de cliente, si lo tiene, en tal caso de no tenerlo sería cero.
  + estado: indica el estado de la orden: “P” = Pendiente, “C” = Culminada, “A” = Anulada, “E”= Elaborada.
* **tbl\_orden\_detalle**: almacena el detalle de los productos en la orden.
  + id\_orden: es el id de la orden a la que pertenece.
  + id\_producto: es el id de los productos en el pedido.
  + cantidad: es la cantidad en la orden.
  + precio: es el precio unitario del producto.
  + total: es la multiplicación del precio x cantidad
* **tbl\_cliente**: Almacena los datos del cliente
  + Id\_cliente: Número de cliente. Autoincremental
  + Nombre: Nombre de Cliente.
  + Apellido: apellido de cliente.
  + Ultima\_compra: Es la fecha de la última compra. (no la introduce el usuario)
  + Puntos: Es el total de puntos del cliente. (no la introduce el usuario)

\*Cualquier otro campo o tabla que usted considere necesario.

Requerimientos funcionales de la Aplicación

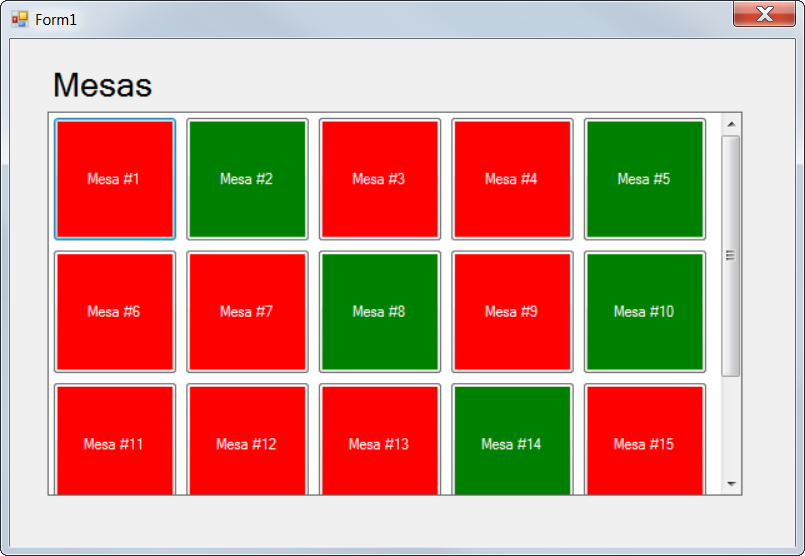
1. Pantalla de Login, se debe validar el usuario y contraseña contra la tabla tbl\_usuario.
2. Una vez logueado el usuario, debe acceder a una pantalla con las siguientes opciones:

* Administración
* Creación de Ordenes
* Pedidos Pendientes
* Listado

1. Pantallas de Administración: en esta sección se da mantenimiento a las tablas:

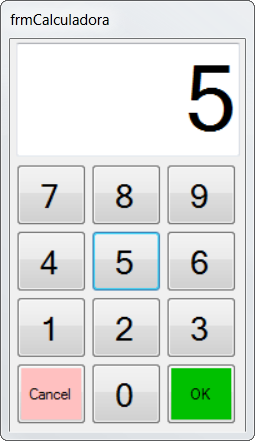
* tbl\_usuario: aquí se debe pedir al usuario que introduzca nuevos usuarios, modificarlos y eliminarlos.
* tbl\_mesa: aquí se debe pedir al usuario que introduzca nuevas mesas, modificarlas y eliminarlas.
* tbl\_categoria: aquí se debe pedir al usuario que introduzca nuevas categorías, modificarlas y eliminarlas.
* tbl\_producto: aquí se debe pedir al usuario que introduzca nuevos productos, modificarlos y eliminarlos.
* tbl\_cliente: aquí se debe pedir al usuario que introduzca nuevas categorías, modificarlas y eliminarlas.
* Cada pantalla de mantenimiento debe permitir Adicionar, modificar, eliminar los datos de cada tabla.

1. Pantalla de Creación de Ordenes: En esta pantalla se gestiona la creación de Órdenes. Al acceder a esta pantalla se deben mostrar todas las mesas (accediendo al listado de mesas en la tabla tbl\_mesa), en color verde las mesas que tengan el estado =“L” y en Rojo las mesas que tengan el estado “O”. Las mesas se deben crear de manera dinámica a partir de botones. Ver Anexo 1. Este es un diseño probable de esta pantalla:



1. Luego que selecciona la mesa (dando clic sobre el número de mesa) se debe acceder a la pantalla de captura de órdenes del cliente. La misma debe poseer un diseño parecido al siguiente:



1. Si el cliente lo tiene, se debe solicitar el código de cliente. El mismo se validará contra el webservice que devolverá el nombre, apellido, y la cantidad de puntos. Para ser mostrados en una sección de la pantalla.
2. La lista de categorías que deben crearse con botones de manera dinámica haciendo un select a la tabla tbl\_categoria (Ver Anexo 1 para creación dinámica de botones).
3. Cada categoría va a tener asignado un mínimo de 10 productos. De igual manera cada producto debe cargarse de manera dinámica en botones (Ver Anexo 1 para creación dinámica de botones) dependiendo de la categoría a la cual le fue asignado haciendo un select a la tabla tbl\_producto filtrando por el id\_categoria asignado al mismo.
4. Al dar clic sobre el botón de producto se debe preguntar la cantidad que va a vender, la misma le aparecerá en una pantalla adicional o un contenedor adicional:

1. Al dar clic en Aceptar, se adicionará a un GRID los siguientes datos (Ver Anexo 2 para ver como adicionar datos a un DataGridView de forma manual):

* Descripción del producto: Leído de la base de datos, y almacenado en algún lugar antes de enviarlo.
* Cantidad: Digitada por el usuario en el paso anterior.
* Precio Unitario: Leído de la Base de datos.
* Precio Total: Calculado multiplicando el precio Unitario por la cantidad.

Se debe tener un botón para eliminar una línea del GRID (Ver Anexo 3 de como eliminar elementos de in dataGridView).

Conforme se adicionan productos al grid se debe ir calculando el total a pagar por el cliente.

1. Al presionar el botón “Guardar” se deben realizar las siguientes tareas:

* Insertar en la tabla tbl\_orden\_encabezado los datos de la orden, como número (id\_orden), fecha, total, id\_usuario del usuario que está logueado en ese momento.
* Insertar en la tabla tbl\_orden\_detalle los datos de los productos dentro del grid: id\_producto, id\_orden, cantidad, precio, total.
* Actualizar la mesa seleccionada a que ha sido ocupada, (estado=”O”).

1. Al presionar el botón “Anular”: Este botón estará disponible solo después de haber guardado la orden. Y la función es anular un pedido ya creado. Actualizando en la tabla tbl\_orden\_encabezado el estatus a:

* “A” = Anulado y también en la tabla tbl\_mesas el estatus a “L” = Libre, para liberar la mesa.

1. Se debe adicionar un botón más que permita culminar la orden una vez el cliente paga y se retira. Este botón actualiza en la tabla tbl\_orden\_encabezado el estatus de “C” = Culminado, y también en la tabla tbl\_mesas el estatus a “L” = Libre, para liberar la mesa. Además deberá ejecutar el webservice para que actualice la cantidad de puntos, los puntos equivalen al monto de la venta, redondeando al siguiente entero. Si vendió 10.95, obtiene 11 puntos; si vendió 10.40, igual obtiene 11 puntos. El webservice también debe actualizar la fecha de última compra con la fecha actual.
2. Cabe mencionar que cuando el usuario entra a la pantalla de Mesas, si da clic en una mesa que indica que está ocupada debe mostrar el pedido que tiene esa mesa.
3. Pantalla de Listado de pedidos pendientes: en esta pantalla se muestra el listado de todos los pedidos pendientes por procesar. La misma debe listar los datos de todos las ordenes que tienen el campo estatus = “P”, y una vez elaborada la orden, el usuario puede cambiar el estado de esa orden a “E”= Elaborada, indicando que ya puede entregársela al cliente. El objetivo de esta pantalla es que en la cocina tengan acceso a los pedidos pendientes por procesar. Esta pantalla se debe refrescar de manera automática cada 10 segundos.
4. Se debe crear una sección web que no esté protegida por accesos, donde el cliente puede consultar sus puntos y última fecha de compra.
5. Pantalla MDI que incluya a todas las pantallas anteriores (menos la web), con excepción del login. Esta pantalla debe tener menús para acceder a cada una de las opciones, como también una barra de herramientas.
6. Sección de Reportes, los mismos deben presentarse en datagrids:

* Reporte de Mesas, ordenando por el campo estado.
* Reporte de los productos con el nombre de su categoría.
* Reporte de órdenes con el nombre del usuario que la creó.
* Reporte de Venta totales para una fecha específica.
* Reporte que muestre el producto más vendido en cantidad y el producto más vendido en monto para un grupo de fechas.

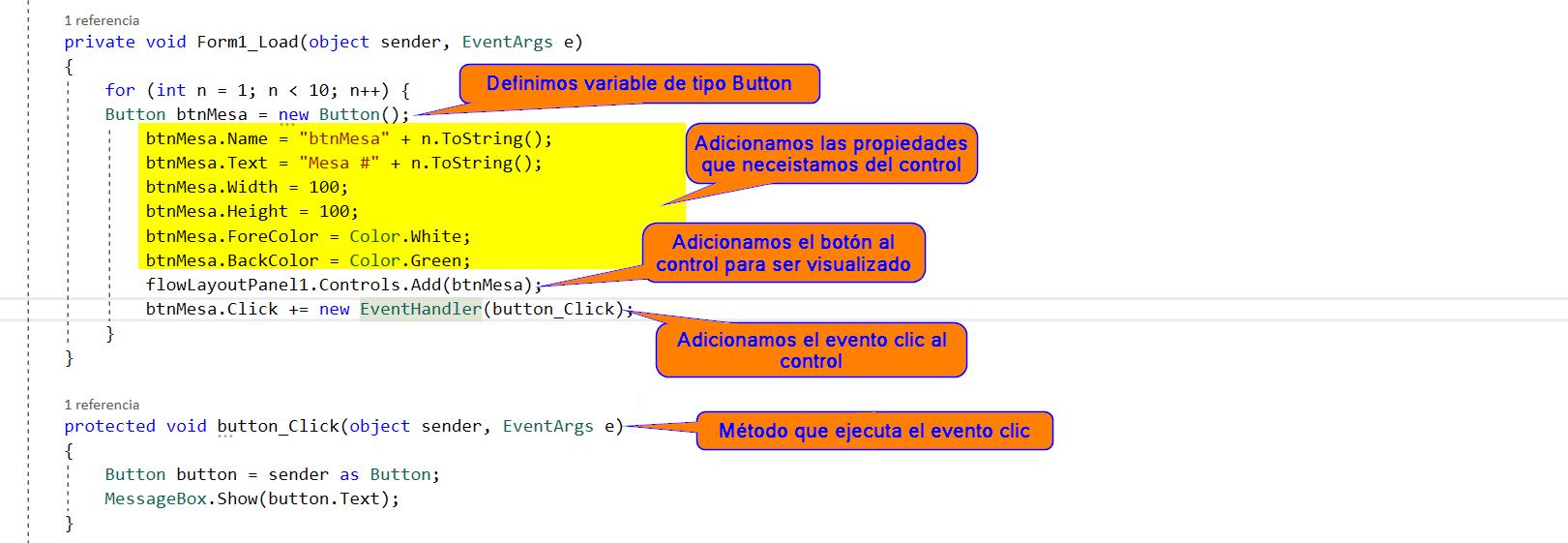
Otros Requerimientos de la Aplicación

* La base de datos sera MySQL.
* Todos los accesos a las tablas se debe realizar por procedimientos almacenados.
* Debe implementar servicios alojados remotamente, y desde los servicios se debe acceder a los objetos de la base de datos.
* La Base de datos debe estar alojada remotamente.

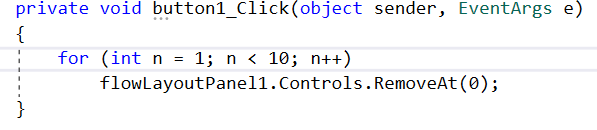
Anexo 1. Como crear botones de manera dinámica

Paso 1. Adicionar un control de tipo FlowLayoutPanel al formulario

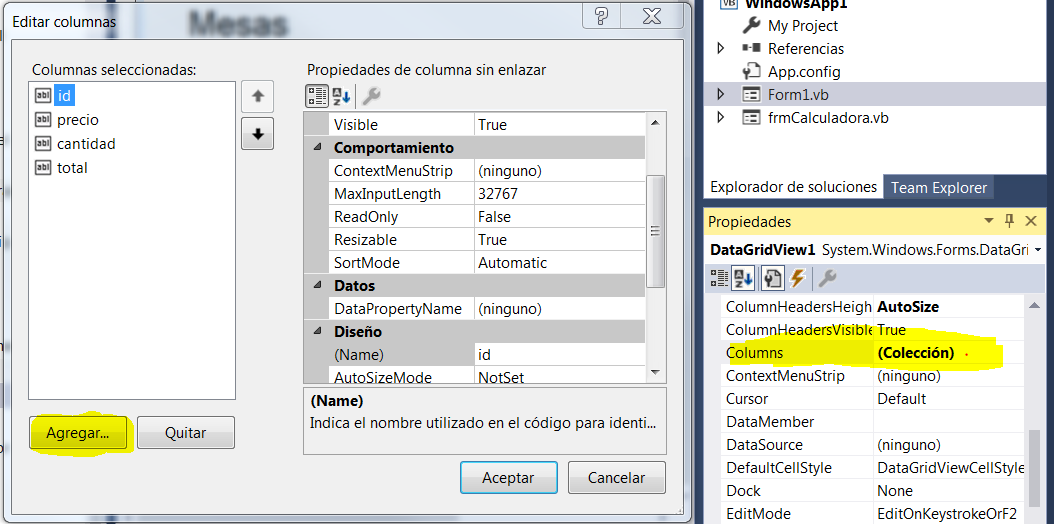
Paso 2. Escribir el siguiente código:



Para Eliminar los botones en tiempo de ejecución se debe colocar el siguiente código donde lo necesite:

 Anexo 2. Adicionar un elemento al DataGridView de forma manual.

Paso 1. Adicionar manualmente las columnas al dataGridView dándole clic en la propiedad Columns y luego en Agregar:



Paso 2. Por código se pueden adicionar registros con la siguiente línea:

DataGridView1.Rows.Add(1, 2.32, 23, 10.23)

Anexo 3. Como eliminar filas de un DataGridView

Foreach (row r In DataGridView1.SelectedRows){

DataGridView1.Rows.Remove(row);

}

Anexo 4. Evaluaciones para este proyecto

|  |  |  |
| --- | --- | --- |
|  | Descripción | Fecha |
| Parcial 2 | Diseño general de pantallas. Puntos 1, 2, 3, 4, 5, 9, 10 (usar datos de ejemplo), 17, 18 | **8 de Octubre.** |
| Parcial 3 | Creación de Base de datos y Clase de acceso a datos | **29 de Octubre** |
| Parcial 4 | Creación de Servicios WEB y conexión con clase de acceso a Datos, Validación con POSTMAN | **19 de Noviembre** |
| Proyecto Final | Pagina Web, y conexión de Windows Form con servicios Web | **Fecha de Semestral** |

Grupos de 2.