



Manual Técnico

Módulo de Inventarios

Christian Ernesto López Dávila 0901-13-9857
Maria Gabriela Zimeri Suady 0901-13-942

CONTENEDOR

En el navegador debemos instanciar todos los forms que aparecerán en el.

```
namespace Inventario
{
    public partial class contenedor : Form
    {
        public contenedor()
        {
            InitializeComponent();

            //instanciando a los forms en el contenedor
            FormInventarioInicio inv_ini = new FormInventarioInicio();
            FormNuevoProducto np = new FormNuevoProducto();
            Categoria cat = new Categoria();
            FormMedidas fm = new FormMedidas();
            marca mar = new marca();
            FormRecibirStock rstck = new FormRecibirStock();
            FormBodega bod = new FormBodega();
            FormLinea fl = new FormLinea();
            FormRequisicion gr = new FormRequisicion();
            Detalle_muestreo dm = new Detalle_muestreo();
            Detalle_bodega_producto dbp = new Detalle_bodega_producto();
            FormMuestreo fmu = new FormMuestreo();
            Modificar_encabezado_muestreo mem = new Modificar_encabezado_muestreo();
            FormMovimientos mov = new FormMovimientos();
            FormDevolucionCompra devc = new FormDevolucionCompra();
            FormDocumento doc = new FormDocumento();
            FormExistenciasPorBodega exbod = new FormExistenciasPorBodega();
            FormFacturasPendientes fac = new FormFacturasPendientes();
            Formkardex kar = new Formkardex();
        }
    }
}
```

PARAMETRO DNS A LA DLL SEGURIDAD

En esta área debemos mandar el parámetro DSN para la conexión.

```
private void contenedor_Load(object sender, EventArgs e)
{
    // mandando parametro DSN a la dll de seguridad
    Conexion.DSN = "prueba2";
    MdiClient Chld;
```

COLOR DE FONDO EN EL CONTENEDOR

Debemos escribir el siguiente código para ponerle color de fondo al navegador, y en propiedades seleccionar el número de color de nuestro módulo en este caso verde.

```
// para poderle poner color de fondo al contenedor
foreach (Control ctrl in this.Controls)
{
    try
    {
        Chld = (MdiClient)ctrl;
        Chld.BackColor = this.BackColor;
    }
    catch (InvalidCastException exe)
    { }
}
```

CODIGO PARA QUE APAREZCA FORMULARIO EN CONTENEDOR

Debemos realizar lo siguiente para cada formulario de la aplicación para que aparezca en el Contenedor

```
// codigo para que el formulario de nuevoProducto aparezca en el conetenedor al dar clic sobre el nombre
private void productoToolStripMenuItem_Click(object sender, EventArgs e)
{
    if (np == null)
    {
        np = new FormNuevoProducto();
        np.MdiParent = this;

        np.FormClosed += new FormClosedEventHandler(FormNuevoProducto_formclosed);
        np.Show();
    }
}
```

AREA DE MANTENIMIENTOS

BOTÓN NUEVO

```
,  
//botón nuevo para que todos los textbox aparezcan para poder escribir en ellos  
private void btn_nuevo_Click(object sender, EventArgs e)  
{  
    Editar = false;  
    txt_nombre.ReadOnly = false;  
    txt_ident.ReadOnly = false;  
    txt_ident.Text = "";  
    txt_nombre.Text = "";  
}
```

BOTÓN ELIMINAR

```
,  
// botón para actualizar a inactivo en la BD un id seleccionado en el datagridview  
private void btn_eliminar_Click(object sender, EventArgs e)  
{  
    SistemaInventarioDatos sid = new SistemaInventarioDatos();  
    string id_linea = Convert.ToString(dgw_categorias.CurrentRow.Cells[0].Value);  
    sid.Eliminar("Update categoria set estado = 'inactivo' where id_categoria_pk= '" + id_linea + "'");  
    dgw_categorias.DataSource = sid.ObtenerCategorias();  
}
```

BOTÓN ACTUALIZAR

```
//botón para actualizar el Datagridview correspondiente  
private void btn_actualizar_Click(object sender, EventArgs e)  
{  
    SistemaInventarioDatos sd = new SistemaInventarioDatos();  
    dgw_categorias.DataSource = sd.ObtenerCategorias();  
}
```

BOTÓN EDITAR

```
//botón de editar en esta parte ingresando en variables los resultados del datagridview a los cuales se les dieron
//anteriormente click
private void btn_editar_Click(object sender, EventArgs e)
{
    try
    {
        Editar = true;
        string id_categoria = Convert.ToString(dgw_categorias.CurrentRow.Cells[0].Value);
        string tipo_categoria = Convert.ToString(dgw_categorias.CurrentRow.Cells[1].Value);

        txt_ident.ReadOnly = true;
        txt_ident.Text = id_categoria;
        txt_nombre.Text = tipo_categoria;

        txt_ident.ReadOnly = false;
        txt_nombre.ReadOnly = false;
        cate_anterior = id_categoria;
    }
    catch
    {
        MessageBox.Show("No existen registros que modificar");
    }
}
```

BOTÓN CANCELAR

```
//Botón cancelar paraa que aparezcan todos los textbox del formulario en limpio.
private void btn_cancelar_Click(object sender, EventArgs e)
{
    Editar = false;
    txt_ident.Text = "";
    txt_nombre.Text = "";
}
}
```

FORMULARIO DEVOLUCION SOBRE COMPRA

EN EL LOAD

Para llenar los combobox al inicializar el form de devolución sobre compra.

```
// load del formulario Devolución sobre compra , para llenar los combobox
private void FormDevolucionCompra_Load(object sender, EventArgs e)
{
    SistemaInventarioDatos sd = new SistemaInventarioDatos();
    cbo_bodega.DataSource = sd.ObtenerBodega();
    cbo_bodega.DisplayMember = "nombre_bodega";
    cbo_bodega.ValueMember = "id_bodega_pk";
    //*****
    DataGridViewComboBoxColumn cbobienes = dgw_requisicion.Columns[1] as DataGridViewComboBoxColumn;
    cbobienes.DataSource = sd.ObtenerBien();
    cbobienes.DisplayMember = "descripcion";
    cbobienes.ValueMember = "CODIGO";

    dgw_requisicion.AutoGenerateColumns = false;

    cbo_compra.DataSource = sd.ObtenerOrdendev();
    cbo_compra.ValueMember = "id_factura_compra_pk";
    cbo_compra.DisplayMember = "id_factura_compra_pk";

    cbo_prov.DataSource = sd.ObtenerProveedor();
    cbo_prov.ValueMember = "id_proveedor_pk";
    cbo_prov.DisplayMember = "nombre_proveedor";
}
```

BOTON PARA LA TRANSACCION DE DEVOLUCIÓN DE COMPRA

```
// BOTON DE PARA LLEVAR A CABO LA TRANSACCIÓN DE LA DEVOLUCION DE COMPRA |
private void btn_crear_Click(object sender, EventArgs e)
{
    SistemaInventarioDatos sd = new SistemaInventarioDatos();
    char delimitador = '-';
    string bodega = cbo_bodega.SelectedValue.ToString();

    bool y = ValidarExistenciaProd();

    if (y == true)
    {
        bool x = ValidarCantidadDelProducto();

        if (x == true)
        {
            foreach (DataGridViewRow fila in dgw_requisicion.Rows)
            {
                if (fila.Cells["Codigo"].Value != null)
                {
                    //----------------OBTENER CANTIDAD Y BIEN
                    int cantidad = Convert.ToInt32(fila.Cells["Cant_devuelta"].Value);

                    DataGridViewComboBoxCell CboBienes = fila.Cells["Descripcion"] as DataGridViewComboBoxCell;
                    string codigo = CboBienes.Value.ToString();

                    string[] codigo_separado = codigo.Split(delimitador);
                    string categoria = codigo_separado[0].Trim();
                    string id_bien = codigo_separado[1].Trim();

                    int existencias = sd.ObtenerExistenciasEnBodega(id_bien, categoria, bodega);
                    int nueva_existencia = existencias - cantidad;
                    sd.ActualizarExistenciasEnBodega(id_bien, categoria, bodega, nueva_existencia);
                }
            }
        }
    }
}
```

Si la transacción fue procesada con éxito entonces ingresa en documento, detalle documento y registra un movimiento en la BD.

```
MessageBox.Show("devolucion procesada con exito");
//---INSERTAR ENCABEZADO
string encargado = txt_encargado.Text.Trim();
DateTime f = Convert.ToDateTime(dtp_fecha_dev.Value);
string fecha = f.ToString("yyyy-MM-dd");
string compra = cbo_compra.SelectedValue.ToString();
string proveedor = cbo_prov.SelectedValue.ToString();

sd.InsertarDevCompra(encargado, fecha, bodega, compra, proveedor);
//--DETALLE
int id_dev = sd.ObtenerUltimaDevC();
InsertarDetalleDevolucion(id_dev);

//--INSERTAR DOC INV
sd.InsertarDocInvDin(id_dev.ToString(), "-", "Devolucion sobre compra", fecha, "-", "procesado", proveedor);
//--DETALLE DOC INV
InsertarDetalleDocInv(id_dev);

//--REGISTRAR MOV
RegistrarMovimiento(id_dev);
```

FORM FACTURAS PENDIENTES

LOAD DE FACTURAS PENDIENTES

```
//LOAD DEL FORMULARIO FACTURAS PENDIENTES Y LISTADO DE LOS TITULOS DEL DATAGRIDVIEW
int cont = 0;
private void FormFacturasPendientes_Load(object sender, EventArgs e)
{
    SistemaInventarioDatos sd = new SistemaInventarioDatos();
    cbo_facturas.DataSource = sd.ObtenerFacturas();
    cbo_facturas.DisplayMember = "COD_FAC";
    cbo_facturas.ValueMember = "COD_FAC";
}
```


FUNCION PARA LLENAR DATAGRIDVIEW

```
// SELECCIONA EL VALOR DEL COMBOBOX Y LO MUESTRA SEPARADO
private void cbo_facturas_SelectedIndexChanged(object sender, EventArgs e)
{
    SistemaInventarioDatos sd = new SistemaInventarioDatos();
    cont++;
    if (cont >= 3)
    {
        string factura = cbo_facturas.SelectedValue.ToString().Trim();
        string[] factura_separada = factura.Split('-');

        string no = factura_separada[0].Trim();
        string serie = factura_separada[1].Trim();
        string empresa = factura_separada[2].Trim();

        dgw_detfac.DataSource = sd.ObtenerDetalleDoc(no,serie,"Factura",empresa);
        dgw_detfac.Columns[0].HeaderText = "Codigo";
        dgw_detfac.Columns[1].HeaderText = "Descripción";
        dgw_detfac.Columns[2].HeaderText = "Cantidad";

        dgw_detfac.Columns[0].Width = 55;
        dgw_detfac.Columns[1].Width = 267;
        dgw_detfac.Columns[2].Width = 55;

        lbl_no.Text = no;
        lbl_serie.Text = serie;
        lbl_no.Visible = true;
        lbl_serie.Visible = true;
    }
}
```

INSERTANDO A MOVIMIENTOS DE INVENTARIO Y EN DOCUMENTO

```
//-----INSERTAR
sd.RegistrarMovimientoInvFac("Venta","2" , id_bien, categoria, cantidad, no, serie, "Factura", empresa);
//-----desaparecer fac
sd.FacturaProcesada(no,serie,"Factura", empresa);
```

MOSTRAR COLUMNAS DATAGRIDVIEW

En esta área llamamos a MostrarInventario el cual es una unión de varios campos de interés para que los muestre en el datagridview y se definieron el nombre de los campos y el tamaño de las columnas.

```
// LOAD PARA QUE APAREZCAN LOS ENCABEZADOS EN EL DATAGRIDVIEW
private void FormInventarioInicio_Load(object sender, EventArgs e)
{
    SistemaInventarioDatos sd = new SistemaInventarioDatos();
    dgw_bienes.DataSource = sd.MostrarInventario();
    dgw_bienes.Columns[0].HeaderText = "Codigo";
    dgw_bienes.Columns[1].HeaderText = "Descripción";
    dgw_bienes.Columns[2].HeaderText = "Categoría";
    dgw_bienes.Columns[3].HeaderText = "Existencias";
    dgw_bienes.Columns[4].HeaderText = "Precio";
    dgw_bienes.Columns[5].HeaderText = "Costo";
    dgw_bienes.Columns[6].HeaderText = "Medida";
    dgw_bienes.Columns[7].HeaderText = "Línea";
    dgw_bienes.Columns[8].HeaderText = "Marca";

    dgw_bienes.Columns[0].Width = 67;
    dgw_bienes.Columns[1].Width = 282;
    dgw_bienes.Columns[2].Width = 78;
    dgw_bienes.Columns[3].Width = 62;
    dgw_bienes.Columns[4].Width = 50;
    dgw_bienes.Columns[5].Width = 50;
    dgw_bienes.Columns[6].Width = 69;
    dgw_bienes.Columns[7].Width = 88;
    dgw_bienes.Columns[8].Width = 89;

}
```

FORMULARIO DE MOVIMIENTOS

```
//load para que aparezcan los campos en el datagridview y llamando a todos los movimientos en caso de no seleccionar ninguna
//para mostrarse en el load
private void FormMovimientos_Load(object sender, EventArgs e)
{
    SistemaInventarioDatos sd = new SistemaInventarioDatos();
    dgw_movimientos.DataSource = sd.TodosLosMovimientos();

    dgw_movimientos.Columns[0].HeaderText = "Mov.";
    dgw_movimientos.Columns[1].HeaderText = "Fecha";
    dgw_movimientos.Columns[2].HeaderText = "Codigo prod.";
    dgw_movimientos.Columns[3].HeaderText = "Descripción";
    dgw_movimientos.Columns[4].HeaderText = "Cant.";
    dgw_movimientos.Columns[5].HeaderText = "Transacción";
    dgw_movimientos.Columns[6].HeaderText = "Bodega";
    dgw_movimientos.Columns[7].HeaderText = "Doc.";
    dgw_movimientos.Columns[8].HeaderText = "Tipo doc.";

    dgw_movimientos.Columns[0].Width = 40;
    dgw_movimientos.Columns[1].Width = 70;
    dgw_movimientos.Columns[2].Width = 55;
    dgw_movimientos.Columns[3].Width = 290;
    dgw_movimientos.Columns[4].Width = 35;
    dgw_movimientos.Columns[5].Width = 70;
    dgw_movimientos.Columns[6].Width = 95;
    dgw_movimientos.Columns[7].Width = 42;
    dgw_movimientos.Columns[8].Width = 103;
}
```

BOTONES DE FILTRADO EN MOVIMIENTOS

```
// PARA MOSTRAR RESULTADOS SOBRE UNA COMPRA
private void btn_compras_Click(object sender, EventArgs e)
{
    SistemaInventarioDatos sd = new SistemaInventarioDatos();
    dgw_movimientos.DataSource = sd.MovimientosCompras();
}

//PARA MOSTRAR LOS RESULTADOS SOBRE UNA DEVOLCUIN
private void btn_devv_Click(object sender, EventArgs e)
{
    SistemaInventarioDatos sd = new SistemaInventarioDatos();
    dgw_movimientos.DataSource = sd.MovimientosDevV();
}

//PARA MOSTRAR TODAS LAS VENTAS
private void btn_ventas_Click(object sender, EventArgs e)
{
    SistemaInventarioDatos sd = new SistemaInventarioDatos();
    dgw_movimientos.DataSource = sd.MovimientosVentas();
}

// PARA MOSTRAR LOS REUSLTADOS DE LAS DEVOLCUIONES
private void btn_devc_Click(object sender, EventArgs e)
{
    SistemaInventarioDatos sd = new SistemaInventarioDatos();
    dgw_movimientos.DataSource = sd.MovimientosDevC();
}

//PARA MOSTRAR TODOS LOS MOVIMIENTOS EN EL DATAGRIDVIEW
private void btn_todos_Click(object sender, EventArgs e)
{
    SistemaInventarioDatos sd = new SistemaInventarioDatos();
    dgw_movimientos.DataSource = sd.TodosLosMovimientos();
}
```

BOTONES PARA MOVERSE EN EL NAVEGADOR

```
//NAVEGADOR BOTON ANTERIOR
private void btn_anterior_Click(object sender, EventArgs e)
{
    CapaNegocio fn = new CapaNegocio();
    fn.Anterior(dgw_movimientos);
    //TextBox[] textbox = { txt_abrev, txt_nombre };
    //fn.llenartextbox(textbox, dgw_movimientos);
}
//NAVEGADOR BOTON SIGUIENTE
private void btn_siguiete_Click(object sender, EventArgs e)
{
    CapaNegocio fn = new CapaNegocio();
    fn.Siguiente(dgw_movimientos);
}
//NAVEGADOR BOTON PRIMERO
private void btn_primero_Click(object sender, EventArgs e)
{
    CapaNegocio fn = new CapaNegocio();
    fn.Primerio(dgw_movimientos);
}
//NAVEGADOR BOTON ULTIMO
private void btn_ultimo_Click(object sender, EventArgs e)
{
    CapaNegocio fn = new CapaNegocio();
    fn.Ultimo(dgw_movimientos);
}
```

LOAD FORMULARIO DE RECIBIR STOCK

```
//Load de Recibir Stock para mostrar campos en datagridview |
private void FormRecibirStock_Load(object sender, EventArgs e)
{
    SistemaInventarioDatos sd = new SistemaInventarioDatos();

    DataGridViewComboBoxColumn cbobienes = dgw_entradas.Columns[1] as DataGridViewComboBoxColumn;
    cbobienes.DataSource = sd.ObtenerBien();
    cbobienes.DisplayMember = "descripcion";
    cbobienes.ValueMember = "CODIGO";

    DataGridViewComboBoxColumn cbobodega = dgw_entradas.Columns[2] as DataGridViewComboBoxColumn;
    cbobodega.DataSource = sd.ObtenerBodega();
    cbobodega.DisplayMember = "nombre_bodega";
    cbobodega.ValueMember = "id_bodega_pk";

    dgw_entradas.AutoGenerateColumns = false;

    //*****
    rdbt.Checked = true;
    cbo_proveedor.DataSource = sd.ObtenerProveedor();
    cbo_proveedor.ValueMember = "id_proveedor_pk";
    cbo_proveedor.DisplayMember = "nombre_proveedor";
    //*****
    //cbo_orden.DataSource = sd.ObtenerReq(); /*REQ*/
    //cbo_orden.ValueMember = "id_requisicion_pk";
    //cbo_orden.DisplayMember = "id_requisicion_pk";

    cbo_orden.DataSource = sd.ObtenerOrden(); /*OC*/
    cbo_orden.ValueMember = "id_factura_compra_pk";
    cbo_orden.DisplayMember = "id_factura_compra_pk";
}
```

