```csharp
using MySql.Data.MySqlClient;
using sigal.Clases;
using System;
using System.Windows.Forms;

namespace sigal
{
    public partial class Det_Requisiciones : Form
    {
        int r = 0;
        Int64 ID_REQ,ID_SAL;

        Double PRECIO, TOTAL;
        string filtro2 = "";
        conexion conn = new conexion();
        conexion conn1 = new conexion();
        conexion conn2 = new conexion();
        conexion conn3 = new conexion();



        MySqlDataReader R;
        public Det_Requisiciones()
        {
            InitializeComponent();
        }

        private void bt_agregar_Click(object sender, EventArgs e)
        {
            if (validar() == 0 && validar_ent() == 0) { return; }
            panel2.Enabled = false;
            Grid.Rows.Insert(r, txtid.Text, txtnombre.Text, txtcantidad.Text, txtespecificaciones.Text);
            r = r + 1;
            btnGrabar.Enabled = true;
            limpiar();
            //cargar_det();
            txtid.Focus();
        }

        internal static void show()
        {
            throw new NotImplementedException();
        }
```

```csharp
        private void txtid_Leave(object sender, EventArgs e)
        {
            if (txtid.Text.ToString() != "")
            {
                R = conn.GetData("SELECT nombre, precio FROM cat_articulos where id=" + txtid.Text +
";");
                if (R.HasRows)
                {
                    R.Read();
                    txtnombre.Text = R[0].ToString().ToUpper();
                }
                else
                {
                    MessageBox.Show("No existe ese numero de articulo¡");
                    txtid.Focus();
                }
                R.Close();
            }
        }

        private void Det_Requisiciones_Load(object sender, EventArgs e)
        {
            //mensajes.SetToolTip(txtBusqueda, "Registro a buscar...");
            mensajes.SetToolTip(btnNuevo, "Clic para agregar un nuevo registro...");
            //mensajes.SetToolTip(btnEditar, "Mostrar Detalle del Registro...");
            mensajes.SetToolTip(btnSalir, "Regresar a Pantalla Anterior");
            mensajes.SetToolTip(btnGrabar, "Guardar Registro...");

            cargar();
            conn.AbrirBD();
            conn1.AbrirBD();
            cbclue.Focus();
        }
        void cargar_det()
        {
            return;
#pragma warning disable CS0162 // Se detectó código inaccesible
            Grid.DataSource = conn.GetTable("select d.id,d.idarticulo as Id_Articulo,a.nombre_corto as
Descripcion,cantidad as Cantidad," +
                ",d.especificaciones as Especificaciones " +
                "from det_requisiciones d, catarticulos a where d.idarticulo=a.id and d.idrequisicion=" +
                ID_REQ + " order by d.id");
#pragma warning restore CS0162 // Se detectó código inaccesible
            if ((conn.PropertyDataSet.Tables[0].Rows.Count != 0))
```

```csharp
        {
            Grid.Columns["id"].Visible = false;
            Grid.AutoResizeColumns();
        }

    }
    void cargar()
    {
        //txtfecha_factura.Value = DateTime.Now;
        txtfecha.Value = DateTime.Now;



        conn2.AbrirBD();
        cbprograma.DisplayMember = "nombre";
        cbprograma.ValueMember = "id";
        cbprograma.DataSource = conn2.GetTable("select id,nombre from cat_programas");
        conn2.cerrarBd();

        conn3.AbrirBD();
        cbclue.DisplayMember = "nombre";
        cbclue.ValueMember = "id";
        cbclue.DataSource = conn3.GetTable("select id,concat(clue,' ',descripcion) as nombre  from
cat_clues");
        conn3.cerrarBd();

    }
    private void Det_Entradas_FormClosing(object sender, FormClosingEventArgs e)
    {
        conn.cerrarBd();
        conn1.cerrarBd();
    }

    private void txtid_TextChanged(object sender, EventArgs e)
    {
        if (txtid.Text == "")
        { txtnombre.Enabled = true; }
        else { txtnombre.Enabled = false; }
    }
    private void txtcantidad_KeyPress(object sender, KeyPressEventArgs e)
    {
        if (Char.IsDigit(e.KeyChar))
        { e.Handled = false; }
```

```csharp
        else if (Char.IsControl(e.KeyChar))
        { e.Handled = false; }
        else if (Char.IsSeparator(e.KeyChar))
        { e.Handled = false; }
        else if (e.KeyChar == 46)
        { e.Handled = false; }
        else { e.Handled = true; }
    }
    void limpiar()
    {
        txtid.Text = "";
        txtnombre.Text = "";
        txtcantidad.Text = "0";
        txtespecificaciones.Text = "";


    }

    private void txtnombre_TextChanged(object sender, EventArgs e)
    {
        if (txtnombre.TextLength > 3 && txtid.Text == "")
        {
            Grid_Articulos.Visible = true;

            //grid1.DataSource = conn.GetTable("select id,nombre_corto,nombre_largo from
catarticulos");

            Grid_Articulos.DataSource = conn1.GetTable("select id as Id_Producto,nombre as
Nombre," +
                " Descripcion from cat_articulos where nombre like '%" + txtnombre.Text + "%' order
by nombre limit 20");
            if (Grid_Articulos.RowCount < 2)
            {
                Grid_Articulos.Visible = false;
                return;
            }
            Grid_Articulos.AutoResizeColumns();
        }
    }

    private void Grid_Articulos_DoubleClick(object sender, EventArgs e)
    {
        txtid.Text = Grid_Articulos.CurrentRow.Cells["Id_Producto"].Value.ToString();
        txtnombre.Text = Grid_Articulos.CurrentRow.Cells["Nombre"].Value.ToString();
        Grid_Articulos.Visible = false;
```

```csharp
    txtcantidad.Focus();
    txtnombre.Enabled = false;
}

private void txtid_KeyDown(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Enter)
    { txtcantidad.Focus(); }
    else if (e.KeyCode == Keys.Down) { txtnombre.Focus(); }
}
int validar()
{

    if (txtid.Text != "" && txtcantidad.Text != "")
    { return 1; }
    return 0;
}
int validar_ent()
{
    if (txtfecha.Text != "" && cbclue.Text != "" && cbprograma.Text != "")
    { return 1; }
    return 0;
}

private void bt_grabar_Click(object sender, EventArgs e)
{
    try
    {
        if (validar_ent() == 0) { return; }
        Boolean resp;
        conn.AbrirBD();
        conn.inicio();
        resp = conn.Executa("insert into requisiciones (fecha,fechacaptura," +
        "idclue,idprograma,estado,iduser) values('" +
        txtfecha.Value.ToString("yyyy-MM-dd") + "','" +
        DateTime.Now.ToString("yyyy-MM-dd") + "','" +
        cbclue.SelectedValue + "," +
         cbprograma.SelectedValue + ",'A'," + configuracion.IDUSER +")");
        //OBTENER ID INVENTARIO
        if (!resp) { conn.fallo(); return; }
        R = conn.GetData("SELECT id FROM requisiciones where idclue='" + cbclue.SelectedValue
              + "' and fecha='" + txtfecha.Value.ToString("yyyy-MM-dd")
              + "' and idprograma=" + cbprograma.SelectedValue +
              " and iduser=" + configuracion.IDUSER + "  order by id desc;");
```

```csharp
            if (R.HasRows)
            {
               R.Read();
               ID_REQ = Int32.Parse(R[0].ToString());
               label16.Visible = true;
               label16.Text = "R E Q U I S I C I O N E S  #" + ID_REQ.ToString();
            }
            R.Close();
            resp = grabar_grid();
            if (!resp)
            {
               conn.fallo(); return; }

            if (checkBox1.Checked)
            {
               resp = conn.Executa("insert into salidas (idclue,idrequisicion,iduser,fechareq) " +
                    " select idclue," + ID_REQ + "," + configuracion.IDUSER + ",fecha from requisiciones
where  id=" + ID_REQ);
               if (!resp) { conn.fallo(); return; }
               R = conn.GetData("SELECT id FROM salidas where idrequisicion=" + ID_REQ + ";");
               if (R.HasRows)
               {
                  R.Read();
                  ID_SAL = Int32.Parse(R[0].ToString());

               }
               R.Close();

               resp = conn.Executa("insert into det_salidas (idsalida,idarticulo,cantidad) " +
                    " select " + ID_SAL + ",idarticulo,cantidad from det_requisiciones where estado=1
and idrequisicion=" + ID_REQ);

               conn.Executa("update requisiciones set estado='V' where id=" + ID_REQ);

            }


            resp = conn.Executa("insert into bitacora (fechasis,usuario,motivo,tabla,idtabla) values('"
+
               DateTime.Now.ToString("yyyy-MM-dd hh:mm:ss") + "','" + configuracion.USER +
               "','ALTA','SALIDA'," + ID_REQ + ")");
            if (!resp) { conn.fallo(); return; }

            conn.exito();
```

```csharp
                conn.cerrarBd();
                MessageBox.Show("Alta de Requisicion " + ID_REQ + " Realizada con Exito¡");
                this.Close();
            }
            catch (Exception ex)
            {
                MessageBox.Show("Fallo al guardar en la BD:" + ex.Message, Application.ProductName,
MessageBoxButtons.YesNo, MessageBoxIcon.Question);
            }

        }

        private void Grid_Det_Ent_DoubleClick(object sender, EventArgs e)
        {
            if (Grid.CurrentRow.Index != -1)
            {
                Grid.Rows.RemoveAt(Grid.CurrentRow.Index);
            }
        }

        private void btsalir_Click(object sender, EventArgs e)
        {
            this.Close();
        }

        private void txtcantidad_KeyDown(object sender, KeyEventArgs e)
        {
            if (e.KeyCode == Keys.Enter || e.KeyCode == Keys.Down)
            {
                txtespecificaciones.Focus();
            }
            else if (e.KeyCode == Keys.Up)
            {
                txtid.Focus();
            }
        }

        private void bt_agregar_KeyDown(object sender, KeyEventArgs e)
        {
            if (e.KeyCode == Keys.Enter)
            { txtid.Focus(); }
        }

        private void txtid_KeyPress(object sender, KeyPressEventArgs e)
```

```csharp
        {
            if (Char.IsDigit(e.KeyChar))
            { e.Handled = false; }
            else if (Char.IsControl(e.KeyChar))
            { e.Handled = false; }
            else if (Char.IsSeparator(e.KeyChar))
            { e.Handled = false; }
            else { e.Handled = true; }
        }


        private void txtcantidad_Leave(object sender, EventArgs e)
        {
            /* if (modulo1.IsNumeric(txtcantidad.Text.ToString()) &&
modulo1.IsNumeric(txtprecio.Text.ToString()))
            {
                costo();
            }*/

        }

        private void checkBox1_CheckedChanged(object sender, EventArgs e)
        {

        }

        private void Grid_Articulos_CellContentClick(object sender, DataGridViewCellEventArgs e)
        {

        }

        private void txtespecificaciones_KeyDown(object sender, KeyEventArgs e)
        {
            if (e.KeyCode == Keys.Enter || e.KeyCode == Keys.Down)
            {
                btnNuevo.Focus();
            }
            else if (e.KeyCode == Keys.Up)
            {
                txtcantidad.Focus();
            }
        }

        Boolean grabar_grid()
```

```csharp
    {
        for (int fila = 0; fila < Grid.Rows.Count - 1; fila++)
        {
            string str = "insert into det_requisiciones (idrequisicion,idarticulo," +
                "cantidad,precio,costo,especificaciones) " +
                "values(" + ID_REQ + "," +
                Grid.Rows[fila].Cells[0].Value.ToString() + "," +
                Grid.Rows[fila].Cells[2].Value.ToString() + "," +
                "0,0,'" +
                txtespecificaciones.Text + "')";
            if (!conn.Executa(str))
            {
                conn.fallo();
                return false;
            }
        }
        return true;
    }

  }

}
```