

MANUAL TÉCNICO

Marvin Yoque - Yony Calito
GEEK SOFTWARE - PRODUCCIÓN

Formulario crear Proceso

Load del formulario en donde se cargará el DSN de seguridad:

```
private void NuevoProceso_Load(object sender, EventArgs e)
{
    try
    {
        seguridad.Conexion.DSN = "hotelsancarlos";
    } catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}
```

Botón de cancelar, va a ocultar el formulario y nos va a mostrar el formulario anterior.

```
private void btn_cancelar_Click(object sender, EventArgs e)
{
    frm_nuevaformula frm = new frm_nuevaformula();
    this.Hide();
}
```

Botón Guardar, la primer tarea que deberá realizar el botón de aceptar será la de detectar si los datos ingresados son del tipo correcto.

Si, los datos ingresados son válidos, entonces va a proceder a enviar los datos de los combo box y los textbox hacia la capa de datos para proceder a ser enviados a la base de datos.

Cuando los datos han sido ingresados nos aparecerá un mensaje notificándonos que el registro a sido guardado con éxito.

```
private void btn_guardar_Click(object sender, EventArgs e)
{
    try
    {
        ValidacionNumerica val = new ValidacionNumerica();
        if (!val.funnumero(txt_tiempo_elaboracion.Text))
        {
            MessageBox.Show("debe ingresar solo valores numericos");
        }
        else
        {
            CapaObjetos.nuevoproceso objetoinsertar = new CapaObjetos.nuevoproceso();
            CapaNegocio insertar = new CapaNegocio();
            objetoinsertar.nom_proceso = txt_nombre_proceso.Text.Trim();
            objetoinsertar.tiempo_elabora = horas;
            objetoinsertar.medida_tiempo = cmb_medida_tiempo.Text;
            objetoinsertar.descripcion = txt_descripcion.Text.Trim();
            objetoinsertar.observacion = txt_observaciones.Text.Trim();
            insertar.InsertarNuevoProceso(objetoinsertar);

            MessageBox.Show("Proceso ingresado con exito");
            limpiar();
        }
    }
    catch (Exception ex)
```

Combo box de medida de tiempo, hará una operación convirtiendo a horas el dato que le ingresemos:

```
private void cmb_medida_tiempo_SelectedIndexChanged(object sender, EventArgs e)
{
    cantidad = Convert.ToDouble(txt_tiempo_elaboracion.Text);

    if (cmb_medida_tiempo.SelectedItem == "Minutos")
    {
        horas = cantidad / 60;
    }
    else if (cmb_medida_tiempo.SelectedItem == "Segundos")
    {
        horas = cantidad / 3600;
    }
    else if (cmb_medida_tiempo.SelectedItem == "Horas")
    {
        horas = cantidad;
    }
    //MessageBox.Show(horas.ToString("N3"));
}
}
```

Formulario de nueva formula



Load del formulario: Cargaremos el combo box de clasificación creando un objeto de la capa de datos, de esta forma vamos a realizar una consulta que nos devuelva la clasificación de materia prima y su id, mostrando en el combo box el nombre de la clasificación. De la misma forma haremos para llenar nuestro combo box de medida y el de procesos.

```
private void frm_nuevaformula_Load(object sender, EventArgs e)
{
    try
    {
        // Carga de combobox Clasificacion
        CapaDatos cdcat = new CapaDatos();
        DataTable dtcat = cdcat.CargarDatosBienClasificacion();

        cmb_categoria.DataSource = dtcat;
        cmb_categoria.DisplayMember = "nombre_linea";
        cmb_categoria.ValueMember = "id_linea_pk";
        cmb_categoria.SelectedItem = -1;

        // Carga de combobox procesos

        CapaDatos cd = new CapaDatos();
        DataTable recolector = cd.CargaDatosProceso();

        cmb_proceso.DataSource = recolector;
        cmb_proceso.DisplayMember = "nombre_proceso";
        cmb_proceso.ValueMember = "id_proceso_pk";
    }
    catch { }
}
```

```
//Cargar combobox de medida

CapaDatos cdmedida = new CapaDatos();
DataTable medida = cdmedida.CargarDatosMedida();

cmb_medida.DataSource = medida;
cmb_medida.DisplayMember = "nombre_medida";
cmb_medida.ValueMember = "id_medida_pk";
} catch (Exception ex)
```

Combo box de categoría: lo que haremos ahora es seleccionar una categoría y según la categoría seleccionada nos va a mostrar la materia prima que contiene esta categoría:

```
private void cmb_categoria_SelectedIndexChanged(object sender, EventArgs e)
{
    // carga de materia prima segun categoria
    CapaDatos cd = new CapaDatos();
    opcioncategoria = cmb_categoria.SelectedValue.ToString();

    DataTable dt = cd.CargaDatosBien(opcioncategoria);

    cmb_materia_prima.DataSource=dt;
    cmb_materia_prima.DisplayMember = "descripcion";
    cmb_materia_prima.ValueMember = "id_bien_pk";
}
```

Botón agregar: lo que hará el botón agregar es ir insertando al detalle toda la materia prima que requiriremos para hacer dicha receta.

Este detalle se irá incrementando en el datagridview, e ira sumando las horas hombre, así como también, ira sumando el costo.

```
private void btn_agregar_Click(object sender, EventArgs e)
{
    try
    {
        if (cmb_categoria.Text == "" || cmb_materia_prima.Text == "" || cmb_medida.Text == "" || cmb_proceso.Text == "")
        {
            MessageBox.Show("uno o mas campos estan vacios");
        }
        else
        {
            // consulta para tomar las horas hombre del proceso
            if (cmb_proceso.SelectedValue.ToString() != "System.Data.DataRowView")
            {
                //MessageBox.Show(cmb_proceso.SelectedValue.ToString());
                CapaDatos cd = new CapaDatos();
                DataTable dt = cd.SeleccionarHorasHombre(cmb_proceso.SelectedValue.ToString());

                DataRow casilla = dt.Rows[0];
                string valor = casilla["tiempo_proceso"].ToString();

                // Tomar dato de costo de cada materia prima seleccionada
                CapaDatos cdcosto = new CapaDatos();
                DataTable tabla = cdcosto.SeleccionCostoMateriaPrima(cmb_materia_prima.SelectedValue.ToString());
                DataRow row = tabla.Rows[0];
                String costo = row["Costo"].ToString();
            }
        }
    }
    catch { }
}
```

```
// Datagrid a nivel de usuario
dgv_cuerpo_formula.Rows.Add(cmb_categoria.Text, cmb_materia_prima.Text,
txt_cantidad.Text, cmb_medida.Text, cmb_proceso.Text, valor, costo);

// Datagrid a nivel base de datos
dgv_cuerpo_formula_ID.Rows.Add("mp", cmb_materia_prima.SelectedValue.ToString(), txt_cantidad.Text, cmb_medida.SelectedValue.ToString(),
cmb_proceso.SelectedValue.ToString(), valor, costo);
}

// sumar las filas de las horas hombre
double resultado = 0;

foreach (DataGridViewRow columna in dgv_cuerpo_formula.Rows)
{
    resultado += Convert.ToDouble(columna.Cells["horas_hombre"].Value);
}

lbl_precio.Text = resultado.ToString();

// sumar filas de costo de materia prima
double suma = 0;
foreach (DataGridViewRow celda in dgv_cuerpo_formula.Rows)
{
    suma += Convert.ToDouble(celda.Cells["costo"].Value);
}
lbl_costo.Text = suma.ToString();
```

Botón aceptar: una vez ingresada toda la materia prima a utilizarse la vamos a insertar en la base de datos, esto lo haremos insertando primero el encabezado de la receta, una vez generado el id del encabezado haremos una consulta llamando ese id de la receta encabezado, para poder usarlo para insertar dentro del detalle.

```
// boton que envia a la base de datos
private void btn_aceptar_Click(object sender, EventArgs e)
{
    OdbcConnection con = seguridad.Conexion.ObtenerConexionODBC();
    // insertar detalle receta

    String cadena = "insert into receta (nombre_receta, horas_hombre, costo_receta) values ('" + lbl_nombre.Text + "
    OdbcCommand cmd = new OdbcCommand(cadena, con);
    cmd.ExecuteNonQuery();
    con.Close();
    CapaDatos cd = new CapaDatos();
    DataTable dtuvalor = cd.SeleccionUltimoDatoFormula();
    DataRow fila = dtuvalor.Rows[0];
    String ultimovalor = fila["max(id_receta_pk)"].ToString();
    lbl_encabezado.Text = ultimovalor;
```



El siguiente paso será insertar en el detalle del pedido, esto lo haremos recorriendo el datagrid que tenemos y llenando un datatable, una vez lleno el datatable vamos a proceder a recorrerlo para poder insertarlo dentro de la base de datos.

```
// insertar en detalle formula

DataTable dt = new DataTable();
dt.Columns.Add("categoria", typeof(string));
dt.Columns.Add("materia_prima", typeof(string));
dt.Columns.Add("cantidad", typeof(string));
dt.Columns.Add("medida", typeof(string));
dt.Columns.Add("tipo_proceso", typeof(string));

foreach (DataGridViewRow dgv in dgv_cuerpo_formula_ID.Rows)
{
    dt.Rows.Add(dgv.Cells[0].Value, dgv.Cells[1].Value, dgv.Cells[2].Value, dgv.Cells[3].Value, dgv.Cells[4].Value);
}

foreach (DataRow row in dt.Rows)
{
    string cf = ":";
    if (row[0].ToString() != "" || row[1].ToString() != "" || row[2].ToString() != "" || row[3].ToString() != "" || row[4].ToString() != "")
    {
        OleDbConnection con2 = seguridad.Conexion.ObtenerConexionODBC();
        string query = "insert into detalle_receta_mp(id_receta_pk,id_categoria_pk,id_bien_pk,cantidad,id_medida_pk,id_proceso_pk)values(' + ultimovalor + " +
        OleDbCommand comando = new OleDbCommand(query, con2);
        comando.ExecuteNonQuery();
        con2.Close();
    }
}

MessageBox.Show("Agregado con exito", "Notificacion", MessageBoxButtons.OK, MessageBoxIcon.Information);
this.Close();
```

Botón quitar, si queremos quitar una fila no deseada del detalle, vamos a proceder a utilizar el botón de quitar, esto lo que hará es eliminar uno de los registros del data gridview y volver a recalcular las horas hombre como el costo de la receta:

```
// boton para quitar una fila del datagridview
private void btn_quitar_Click(object sender, EventArgs e)
{
    try
    {
        int x = dgv_cuerpo_formula.CurrentRow.Index;
        //DataGridViewRow row = dgv_cuerpo_formula_ID.Rows[x];

        dgv_cuerpo_formula.Rows.RemoveAt(dgv_cuerpo_formula.CurrentRow.Index);
        dgv_cuerpo_formula_ID.Rows.RemoveAt(x);

        // sumar las filas de las horas hombre

        double resultado = 0;

        foreach (DataGridViewRow columna in dgv_cuerpo_formula.Rows)
        {
            resultado += Convert.ToDouble(columna.Cells["horas_hombre"].Value);
        }

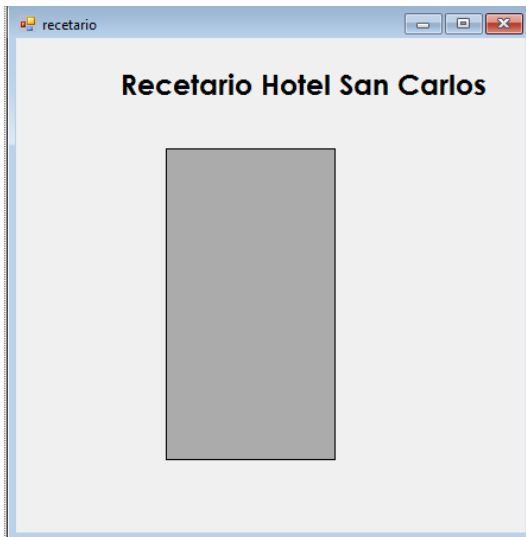
        lbl_precio.Text = resultado.ToString();

        // sumar columna de costo de materia prima
        double costo = 0;

        foreach (DataGridViewRow columna in dgv_cuerpo_formula.Rows)
        {
            costo += Convert.ToDouble(columna.Cells["Costo"].Value);
        }

        lbl_costo.Text = costo.ToString();
    }
}
```


Formulario recetario



Form Load: esta parte lo que contendrá será una llamada a la base de datos con todas las recetas del restaurante:

```
private void frm_recetario_Load(object sender, EventArgs e)
{
    try
    {
        // cargar datagridview con recetas
        CapaDatos cd = new CapaDatos();
        DataTable dt = cd.ConsultarReceta();

        dgv_recetario.DataSource = dt;

        dgv_recetario.Columns["id_receta_pk"].Visible = false;
        dgv_recetario.Columns["horas_hombre"].Visible = false;
        dgv_recetario.Columns["costo_receta"].Visible = false;
        dgv_recetario.Columns["nombre_receta"].HeaderText = "Receta";
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}
```

Si se desea modificar una receta haremos clic en la casilla de la receta y nos aparecerá un formulario para poder realizar nuestro cambio:



```
// enviar los datos del datagridview hacia formulario de cambiar receta
private void dgv_recetario_CellContentClick(object sender, DataGridViewCellEventArgs e)
{
    try
    {
        frm_modificar_recetario = new frm_modificar_recetario();

        foreach (Form form in Application.OpenForms)
        {
            if (form.Name == "frm_modificar_recetario")
            {
                frm_modificar_recetario = (frm_modificar_recetario)form;
                frm_modificar_recetario.txt_nombre_receta.Text = dgv_recetario.CurrentRow.Cells["nombre_receta"].Value.ToString();
                frm_modificar_recetario.lbl_id_receta_enc.Text = dgv_recetario.CurrentRow.Cells["id_receta_pk"].Value.ToString();
                frm_modificar_recetario.lbl_hrs_hombre.Text = dgv_recetario.CurrentRow.Cells["horas_hombre"].Value.ToString();
                frm_modificar_recetario.lbl_costo.Text = dgv_recetario.CurrentRow.Cells["costo_receta"].Value.ToString();
                break;
            }
        }
        frm_modificar_recetario.StartPosition = FormStartPosition.CenterParent;
        frm_modificar_recetario.Show();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}
```

frm_modificar_recetario

Modificar Receta

Nombre Categoría Materia Prima

Cantidad Medida Proceso

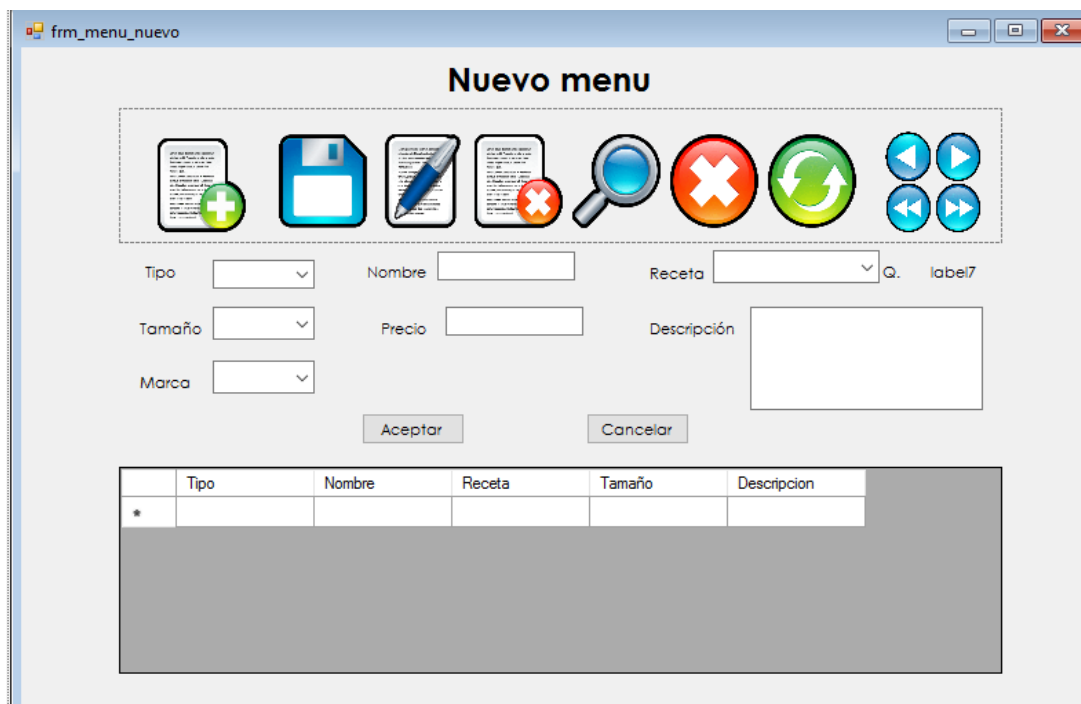
Descripción

Formula:

Hrs Hombre:

Costo:

Formulario de nuevo menú



	Tipo	Nombre	Receta	Tamaño	Descripción
*					

Form load: cargaremos los combo box de tipo de platillo que vamos a crear, también cargaremos las recetas, el combo box de tamaño y la marca del producto que vamos a crear:

```
private void frm_menu_nuevo_Load(object sender, EventArgs e)
{
    try
    {
        // carga de combo box de recetas
        CapaDatos cd = new CapaDatos();
        DataTable dt = cd.ConsultarReceta();
        cmb_receta_seleccion.DataSource = dt;
        cmb_receta_seleccion.DisplayMember = "nombre_receta";
        cmb_receta_seleccion.ValueMember = "id_receta_pk";

        // cargar combobox de tamaño de porciones
        CapaDatos datos = new CapaDatos();
        DataTable dat = datos.ConsultarTamanioPorcion();
        cmb_tamanio_porcion.DataSource = dat;
        cmb_tamanio_porcion.DisplayMember = "tamanio";
        cmb_tamanio_porcion.ValueMember = "id_tamanio_pk";
    }
    catch (Exception ex)
    {
        //MessageBox.Show(ex.Message);
    }
    try
    {
        CapaDatos datos = new CapaDatos();
        DataTable data2 = new DataTable();
        data2 = datos.ObtenerMarcastodo();
        cmb_marca.DataSource = data2;
        cmb_marca.DisplayMember = "nombre_marca";
        cmb_marca.ValueMember = "id_marca_pk";
    }
}
```

```

try
{
    CapaDatos datos2 = new CapaDatos();
    DataTable data3 = new DataTable();
    data3 = datos2.Consulta_linea();
    cmb_linea.DataSource = data3;
    cmb_linea.DisplayMember = "nombre_linea";
    cmb_linea.ValueMember = "id_linea_pk";
}
catch
{
}
}

```

Combo box de tamaño porción: para calcular un precio estimado al valor del producto vamos a multiplicar el valor obtenido del combo box con el costo de la receta:

```

private void cmb_tamano_porcion_SelectedIndexChanged(object sender, EventArgs e)
{
    // consulta de valor de la porcion
    try
    {
        if (cmb_tamano_porcion.ValueMember != "")
        {
            CapaDatos capa = new CapaDatos();
            DataTable tabla = capa.ConsultarValorPorcion(cmb_tamano_porcion.SelectedValue.ToString());
            DataRow tupla = tabla.Rows[0];
            String valor = tupla["valor"].ToString();

            // calcular precio segun tamaño de porcion y costo de receta seleccionado
            decimal precio = Convert.ToDecimal(lbl_costo_receta.Text) * Convert.ToDecimal(valor);
            txt_precio.Text = precio.ToString();
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

```

Selección de costo de receta:

```
private void cmb_receta_seleccion_SelectedIndexChanged(object sender, EventArgs e)
{
    // extraer costo de receta, previamente seleccionada
    try
    {
        if (cmb_receta_seleccion.SelectedValue.ToString() != "System.Data.DataRowView")
        {
            CapaDatos cd = new CapaDatos();
            DataTable dt = cd.ConsultarPrecioReceta(cmb_receta_seleccion.SelectedValue.ToString());
            DataRow fila = dt.Rows[0];
            decimal valor = Convert.ToDecimal(fila["costo_receta"].ToString());
            lbl_costo_receta.Text = valor.ToString();
        }
    } catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}
```

Botón guardar: lo que hará el botón guardar será ir a almacenar a la tabla de precios la descripción de la receta, una vez realizado esto irá a almacenar el detalle del menú, a bien:

```
private void btn_guardar_Click(object sender, EventArgs e)
{
    try
    {
        CapaDatos datos = new CapaDatos();
        int medida = 1;

        int result = datos.AgregarBien(Convert.ToDecimal(txt_precio.Text), Convert.ToDecimal(lbl_costo_receta.Text));
        if (result == 1)
        {
            MessageBox.Show("Ingresado exitosamente");
        }
        else
        {
            MessageBox.Show("Ingresado Error al insertar bien");
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
        MessageBox.Show("Error al crear bien", "Warning!", MessageBoxButtons.OK, MessageBoxIcon.Error); ;
    }
    //try
    //{
    //    // envio de parametros para insertar nuevo precio
    //    CapaDatos capad = new CapaDatos();
    //    capad.InsertarNuevoPrecio(txt_precio.Text.ToString(), "PT", cmb_tamano_porcion.SelectedValue.ToString());

    //    //----- tomar valor ingresado de precio
    //    CapaDatos cd = new CapaDatos();
    //    DataTable dat = cd.ConsultarUltimoValorPrecio();
    //    DataRow fila = dat.Rows[0];
    //    String id_precio = fila["max(id_precio)"].ToString();
    //}
```



```
cd.InsertarNuevoMenu(cmb_linea.SelectedValue.ToString(), txt_nombre.Text.Trim(), txt_descripcion.Text.Trim(), cmb

dgv_nuevo_menu.Rows.Add(cmb_linea.SelectedValue, txt_nombre.Text.Trim(), cmb_receta_seleccion.Text, cmb_tamano;
txt_nombre.Text = "";
txt_descripcion.Text = "";
txt_precio.Text = "";
lbl_costo_receta.Text = "";
lbl_tamano.Text = "";

//accion.Insertar("tipo"+cmb_tipo.SelectedValue.ToString()+"nombre"+txt_nombre.Text.Trim()+"receta"+cmb_receta_selec
//accion.Insertar(cmb_tipo.SelectedValue.ToString() + txt_nombre.Text.Trim() + cmb_receta_seleccion.SelectedValue.To
MessageBox.Show("Menu agregado con exito");
dgv_nuevo_menu.Rows.Clear();
//}
```

Formulario de asignación de materia prima

Form load: cargaremos el combo box de pedido:

```
pedido pdo = new pedido();
//CapaDatos cd = new CapaDatos();
DataTable dt2 = pdo.carga_pedidos();
cmb_pedido.DataSource = dt2;
cmb_pedido.DisplayMember = "id_encabezado_pedido_pk";
cmb_pedido.ValueMember = "id_encabezado_pedido_pk";
comboBox1.DataSource = dt2;
comboBox1.ValueMember = "id_encabezado_pedido_pk";
```

Combo box pedido: Ahora llenaremos el datagridview de pedido con todo su cuerpo de detalle, una vez seleccionado un valor este se desplegará en el grid:

```
// gridview con todo el cuerpo del pedido
private void cmb_pedido_SelectedIndexChanged(object sender, EventArgs e)
{
    CapaDatos cd = new CapaDatos();
    DataTable dt = cd.ConsultarPedidoDetalle(cmb_pedido.SelectedValue.ToString());
    //MessageBox.Show(cmb_pedido.SelectedValue.ToString());

    dgv_pedido.DataSource = dt;
    dgv_pedido.Columns["id_menu_pk"].Visible = false;
    dgv_pedido.Columns["orden"].Visible = false;
    dgv_pedido.Columns["correlativo"].Visible = false;
}
```

Botón aceptar: Ahora vamos a determinar cuanta materia prima se necesita para realizar el pedido seleccionado, lo primero que haremos es mandar a llamar toda la materia prima y desplegarla, para realizar esto debemos recorrer el datatable y mandar a llamar la receta de cada producto solicitado.

Una vez realizado lo anterior, vamos a multiplicar cada cantidad solicitada para determinar cuanta materia prima utilizaremos.

Dentro del segundo datagridview que es el de detalle de materia prima, vamos a recorrer, el campo de cantidad de cada columna e ir haciendo una comparación con la bodega de restaurante (en la base de datos), para determinar si existe suficiente materia prima o no, en caso que no halla materia prima suficiente vamos a solicitar una requisición de materia prima. En caso de que si exista materia prima suficiente vamos a cambiar de estado al pedido, para que vaya a producción:

```
private void btn_aceptar_Click(object sender, EventArgs e)
{
    int contador2 = 0;
    while (contador2 < dgv_pedido.RowCount - 1)
    {
        llave = dgv_pedido.Rows[contador2].Cells["id_menu_pk"].Value.ToString();
        correlativo = dgv_pedido.Rows[contador2].Cells["correlativo"].Value.ToString();
        CapaDatos cd = new CapaDatos();
        DataTable dt = cd.ConsultarMateriaPrimaSegunPedido(llave, correlativo);

        // recorre el datatable para ingresar detalle de materia prima a utilizarse segun detalle de pedido
        foreach (DataRow row in dt.Rows)
        {
            cantidad_total = 0;
            int contador = 0;
            string cantidad_pedido = dgv_pedido.Rows[contador].Cells["cantidad"].Value.ToString();
            //string cantidad_pedido = row["cantidad"].ToString();

            string cantidad_mp = row[3].ToString();

            cantidad_total = Convert.ToInt32(cantidad_pedido) * Convert.ToInt32(cantidad_mp);

            // nombre de menu, receta, cantidad, medida, id bien, descripcion,
            //dgv_detalle_mp.Rows.Add(row[0].ToString(), row[1].ToString(), cantidad_total.ToString(), row[4].ToString(), row[2].ToString(), row[5].ToString());
            dgv_detalle_mp.Rows.Add(row["nombre"].ToString(), row["nombre_receta"].ToString(), cantidad_total.ToString(), row["nombre_medida"].ToString(), row
            contador += 1;
        }
    }
}
```

```

        contador2 += 1;
    }
    // ----- COSTO Y HORAS HOMBRE SUMATORIA
    double costo = 0;
    double hrs_hombre = 0;

    foreach (DataGridViewRow columna in dgv_pedido.Rows)
    {
        costo += Convert.ToDouble(columna.Cells["costo_receta"].Value);
        hrs_hombre += Convert.ToDouble(columna.Cells["horas_hombre"].Value) * Convert.ToDouble(columna.Cells["cantidad"].Value)
    }

    lbl_costo_total.Text = costo.ToString();
    lbl_hrs_hombre_total.Text = hrs_hombre.ToString();

    // ----- INSERTAR EN TABLA ASIGNACION_MP
    DataTable dtmanual = new DataTable();
    //dtmanual.Columns.Add("id_encabezado", typeof(string));
    dtmanual.Columns.Add("cant_mp", typeof(string));
    dtmanual.Columns.Add("id_menu", typeof(string));
    dtmanual.Columns.Add("correlativo", typeof(string));
    dtmanual.Columns.Add("bien", typeof(string));
    //dtmanual.Columns.Add("cant_hh", typeof(string));

    // inserción de datagridview en datatable
    foreach (DataGridViewRow dgv in dgv_detalle_mp.Rows)
    {
        //
        dtmanual.Rows.Add(dgv.Cells[2].Value, dgv.Cells[5].Value, dgv.Cells[6].Value, dgv.Cells[7].Value);

        // inserción de datagridview en datatable
    }

    // inserción de datagridview en datatable
    foreach (DataGridViewRow dgv in dgv_detalle_mp.Rows)
    {
        //
        dtmanual.Rows.Add(dgv.Cells[2].Value, dgv.Cells[5].Value, dgv.Cells[6].Value, dgv.Cells[7].Value);

        // DETERMINAR SI HAY SUFICIENTE MATERIA PRIMA EN BODEGA:
        string descripcion = "";
        total_materia_prima = Convert.ToDouble(dgv.Cells["existencia"].Value) - Convert.ToDouble(dgv.Cells["cantidad"].Value);

        if (total_materia_prima > 0)
        {
            descripcion = dgv.Cells["descripcion"].Value.ToString();
            //MessageBox.Show(descripcion + " " + total_materia_prima);
        }
        else if (total_materia_prima < 0)
        {
            //MessageBox.Show("no hay materia prima");
            controlador = "requisicion";
        }
    }
}

```



```

if (controlador != "requisicion")
{
    foreach (DataRow row in dtmanual.Rows)
    {
        valor_combo = cmb_pedido.SelectedValue.ToString();
        horas_total = lbl_hrs_hombre_total.Text;

        if (row[0].ToString() != "" || row[1].ToString() != "" || row[2].ToString() != "" || row[3].ToString() != "")
        {
            OdbcConnection con2 = seguridad.Conexion.ObtenerConexionODBC();
            string query = "insert into asignacion_mp (id_encabezado_pedido_pk,cant_mp,id_menu_pk,correlativo,id_bien_pk,cant_hh)values('" +
            valor_combo + "','" + row["cant_mp"].ToString() + "','" + row["id_menu"].ToString() + "','" + row["correlativo"].ToString() + "','" +
            //total_materia_prima
            OdbcCommand comando = new OdbcCommand(query, con2);
            comando.ExecuteNonQuery();
            con2.Close();
        }
    }
    CapaDatos cd = new CapaDatos();
    cd.ActualizarEstadoEncabezadoPedido(cmb_pedido.SelectedValue.ToString());
    MessageBox.Show(cmb_pedido.SelectedValue.ToString());
    lbl_tot_mat_prima.Text = total_materia_prima.ToString();
    MessageBox.Show("Materia prima asignada exitosamente");
    // aqui va el cambio de estado del pedido

    cargar_mo_det();
}
else if (controlador == "requisicion")
{
    MessageBox.Show("Debe realizar una requisicion de materia prima");
}

```

Formulario de Requisición



Producción

Materia prima Mano obra Resumen Requisición

Requisición

ID encabezado: label12

button14

Categoría: Materia Prima:

Cantidad: Medida:

Categoría	Materia Prima	Cantidad	Medida
*			

Load form: dentro del load form de requisición agregaremos el llenado de combo box de categoría o línea y el combo box de medida.

```
// ----- AREA DE TRABAJO PARA REQUISICION: -----
// Carga de combobox Clasificacion
CapaDatos cdcat = new CapaDatos();
DataTable dtcat = cdcat.CargarDatosBienClasificacion();

cmb_categoria.DataSource = dtcat;
cmb_categoria.DisplayMember = "nombre_linea";
cmb_categoria.ValueMember = "id_linea_pk";

// Carga de combobox Medida
DataTable medida = cdcat.CargarDatosMedida();

cmb_medida.DataSource = medida;
cmb_medida.DisplayMember = "nombre_medida";
cmb_medida.ValueMember = "id_medida_pk";

//-----cargando pedidos en proceso
try
{
```

Botón agregar: este botón ira agregando materia prima que solicitamos al detalle:

```
private void btn_agregar_Click(object sender, EventArgs e)
{
    try
    {
        if (cmb_categoria.Text == "" || cmb_materia_prima.Text == "" || cmb_medida.Text == "" || txt_cantidad.Text == "")
        {
            MessageBox.Show("uno o mas campos estan vacios");
        }
        else
        {
            // Datagrid a nivel de usuario
            dgv_cuerpo_formula.Rows.Add(cmb_categoria.Text, cmb_materia_prima.Text, txt_cantidad.Text, cmb_medida.Text);

            // Datagrid a nivel base de datos
            dgv_cuerpo_formula_ID.Rows.Add("mp", cmb_materia_prima.SelectedValue.ToString(), txt_cantidad.Text, cmb_medida.Text);

            // limpiar campos
            cmb_categoria.Text = "";
            cmb_materia_prima.Text = "";
            cmb_medida.Text = "";
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}
```

Botón quitar: este botón ira quitando filas que no deseemos en nuestro detalle:

```
private void btn_quitar_Click(object sender, EventArgs e)
{
    try
    {
        int x = dgv_cuerpo_formula.CurrentRow.Index;
        //DataGridViewRow row = (field) DataGridView frm_prod.dgv_cuerpo_formula
        dgv_cuerpo_formula.Rows.RemoveAt(dgv_cuerpo_formula.CurrentRow.Index);
        dgv_cuerpo_formula_ID.Rows.RemoveAt(x);
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}
```

Combo box categoría: será el encargado de filtrar la materia prima que vamos a seleccionar:

```
private void cmb_categoria_SelectedIndexChanged(object sender, EventArgs e)
{
    try
    {
        // carga de materia prima segun categoria
        CapaDatos cd = new CapaDatos();
        opcioncategoria = cmb_categoria.SelectedValue.ToString();

        DataTable dt = cd.CargaDatosBien(opcioncategoria);

        cmb_materia_prima.DataSource = dt;
        cmb_materia_prima.DisplayMember = "descripcion";
        cmb_materia_prima.ValueMember = "id_bien_pk";
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}
```

Botón guardar: será el encargado de enviar los datos a la base de datos, realizando como primera operación la de agregar al encabezado la respectiva descripción y generando un id, que servirá para ir asignando materia prima al detalle.

```
private void btn_guardar_Click(object sender, EventArgs e)
{
    try
    {
        // ----- insertar encabezado -----
        CapaDatos cd = new CapaDatos();
        string usuario = seguridad.Conexion.User;
        cd.InsertarEncabezadoRequisicion(usuario);

        //----- llamar correlativo de encabezado
        DataTable dtuvalor = cd.ConsultarUltimoValorRequisicion();
        DataRow fila = dtuvalor.Rows[0];
        String ultimovalor = fila["max(id_requisicion_pk)"].ToString();
        //MessageBox.Show(ultimovalor);

        // ----- insertar detalle
        DataTable dt = new DataTable();
        dt.Columns.Add("categoria", typeof(string));
        dt.Columns.Add("materia_prima", typeof(string));
        dt.Columns.Add("cantidad", typeof(string));
        dt.Columns.Add("medida", typeof(string));

        foreach (DataGridViewRow dgv in dgv_cuerpo_formula_ID.Rows)
        {
            dt.Rows.Add(dgv.Cells[0].Value, dgv.Cells[1].Value, dgv.Cells[2].Value, dgv.Cells[3].Value);
        }

        foreach (DataRow row in dt.Rows)
        {
            if (row[0].ToString() != "" || row[1].ToString() != "" || row[2].ToString() != "" || row[3].ToString() != "")
            {
                OleDbConnection con2 = seguridad.Conexion.ObtenerConexionODBC();
                string query = "insert into detalle_requisicion (cantidad,id_bien_pk,id_categoria_pk,id_requisicion_pk,medida) values (" + row[2].ToString() + "," + row[0].ToString() + "," + row[1].ToString() + "," + row[3].ToString() + "," + ultimovalor + ")";
                OleDbCommand comando = new OleDbCommand(query, con2);
                comando.ExecuteNonQuery();
                con2.Close();
            }
        }

        MessageBox.Show("Requisición agregada exitosamente");
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}
```

Formulario Pedido



Utilizando las funciones creadas en nuestra capa, cargamos en el Load la información importante sobre los datos del usuario que está ingresando el pedido, para lo cual utilizamos comboBox, donde el *DISPLAY MEMBER* muestra únicamente el nombre y la empresa a la que está ligado el usuario, y el *VALUE MEMBER* guarda los ID's correspondientes.

```

info_cte info_cte = new info_cte();
private void frm_pedido_Load(object sender, EventArgs e)
{
    //txt_nopedido.Text = pedido.generador_correlativo();
    timer1.Start();
    try
    {
        DataTable dt = new DataTable();
        dt = pedido.Carga_datos(seguridad.Conexion.User);
        if (dt.Rows.Count != 0)
        {
            cmb_emp.DataSource = dt;
            cmb_colab.DataSource = dt;
            cmb_colab.DisplayMember = "nombre_colab";
            cmb_colab.ValueMember = "id_empleado";
            cmb_emp.ValueMember = "id_empresa";
            cmb_emp.DisplayMember = "nombre_emp";
        }
        else
        {
            cmb_colab.SelectedIndex = -1;
            cmb_emp.SelectedIndex = -1;
        }
    }
    catch
    {
    }
}

```

Se cargan los menús disponibles para ser seleccionados por el usuario

```

//Cargar los menús disponibles
try
{
    DataTable dt = new DataTable();
    dt = pedido.Carga_tipo();
    if (dt.Rows.Count != 0)
    {
        cmb_tipo.DataSource = dt;
        cmb_tipo.ValueMember = "id_menu_pk";
    }
    else
    {
        cmb_tipo.SelectedIndex = -1;
        //MessageBox.Show("Sin datos");
    }
}
catch
{
    MessageBox.Show("Error en carga de Menú", "Warning!", MessageBoxButtons.OK, MessageBoxIcon.Error); ;
}

```

Nos ayuda a controlar la hora y fecha exacta en la que se está ingresando el pedido

```
private void timer1_Tick(object sender, EventArgs e)
{
    timer1.Start();
    txt_fechaing.Text = DateTime.Now.ToString("yyyy-MM-dd");
    txt_horaing.Text = DateTime.Now.ToString("HH:mm:ss");
}
```

Capturamos el código del cliente al que se le está asignando el pedido

```
private void btn_buscar_Click(object sender, EventArgs e)
{
    frm_buscacliente buscte = new frm_buscacliente();
    buscte.ShowDialog();
    if (buscte.cte_seleccionado.id_cliente != null)
    {
        txt_cte.Text = buscte.cte_seleccionado.id_cliente;
    }
}
```

Para mostrar el precio del menú seleccionado utilizamos esta función

```
}
public void cargar_precio()
{
    txt_precio.Clear();
    try
    {
        string precio = "";
        DataTable dt = new DataTable();
        dt = pedido.carga_precio(cmb_tipo.SelectedValue.ToString(), cmb_prod.SelectedValue.ToString());
        if (dt.Rows.Count != 0)
        {
            foreach (DataRow dr in dt.Rows)
            {
                precio = Convert.ToString(dr["precio"]);
            }
        }
        else
        {
            txt_precio.Text = precio;
        }
    }
    catch
    {
        //MessageBox.Show("Error en carga de precios", "Warning!", MessageBoxButtons.OK, MessageBoxIcon.Error); ;
    }
    btn_agregar.Enabled = true;
}
```

Para agregar ítems al pedido utilizamos un datagrid, al cual vamos a validar primero que si el textbox de cantidad está vacío, que este se convierta automáticamente a 1, y si el textbox de descuento está vacío que lo convierta a 0.00, para evitar los errores. Luego procedemos a agregar los datos validados al control datagridview, lo cual nos permitirá seguir agregando y quitando ítems al pedido cuantos queramos.

```
private void btn_agregar_Click(object sender, EventArgs e)
{
    double subtotal = 0.00;
    double total = 0.00;
    double descuento = 0.00;
    if (string.IsNullOrEmpty(txt_cant.Text))
    {
        txt_cant.Text = "1";
    }
    if (string.IsNullOrEmpty(txt_desc.Text))
    {
        txt_desc.Text = "0.00";
    }
    double importe;
    double cantidad = Convert.ToDouble(txt_cant.Text);
    double precio = Convert.ToDouble(txt_precio.Text);
    importe = precio * cantidad;

    dgv_det_ped.Rows.Add(new string[] { cmb_tipo.SelectedValue.ToString(), cmb_prod.SelectedValue.ToString(), txt_cant.Text.Trim(), importe.ToString(), txt_desc.Text.Trim() });
    txt_cant.Clear();
    txt_desc.Clear();
    foreach (DataGridViewRow row in dgv_det_ped.Rows)
    {
        subtotal += Convert.ToDouble(row.Cells[3].Value);
        descuento += Convert.ToDouble(row.Cells[4].Value);
        total = subtotal - descuento;
    }
    txt_sub.Text = Convert.ToString(subtotal);
    txt_totdesc.Text = Convert.ToString(descuento);
    txt_total.Text = Convert.ToString(total);
    btn_aceptar.Enabled = true;
}
```

Función para eliminar ítems del Datagridview, el cual permite quitar del pedido la fila seleccionada

```
private void dgv_det_ped_CellContentClick(object sender, DataGridViewCellEventArgs e)
{
    double subtotal = 0.00;
    double total = 0.00;
    double descuento = 0.00;
    if (dgv_det_ped.Rows.Count > 1)
    {
        if (dgv_det_ped.CurrentCell.ColumnIndex == 5)
        {
            var row = dgv_det_ped.CurrentRow;
            dgv_det_ped.Rows.Remove(row);
            foreach (DataGridViewRow row1 in dgv_det_ped.Rows)
            {
                subtotal += Convert.ToDouble(row1.Cells[3].Value);
                descuento += Convert.ToDouble(row1.Cells[4].Value);
                total = subtotal - descuento;
            }
            txt_sub.Text = Convert.ToString(subtotal);
            txt_totdesc.Text = Convert.ToString(descuento);
            txt_total.Text = Convert.ToString(total);
        }
    }
    else
    {
        MessageBox.Show("No hay datos para eliminar");
    }
}
```

Para guardar el pedido primero hacemos una función, la que nos devuelve el número de pedido correspondiente, con lo cual podremos visualizar en el textbox No. Pedido éste número, esto permitirá una inserción del detalle más fácil al encabezado del pedido. Ya que cada detalle del Datagridview será relacionado a este número de pedido con un número de orden.

```
private void btn_aceptar_Click(object sender, EventArgs e)
{
    int conteo = 1;
    if (dgv_det_ped.Rows.Count <= 1)
    {
        MessageBox.Show("Debe ingresar datos al pedido", "Warning!", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    else
    {
        try
        {
            txt_nopedido.Text = pedido.generador_correlativo();
            DateTime fecha_e = Convert.ToDateTime(dateTimePicker1.Value);
            int resultado = pedido.inserta_encabezado(txt_nopedido.Text.Trim(), txt_horaing.Text.Trim(), txt_fechaing.Text.Trim(), txt_cte.Text.Trim(), fecha_e);
            if (resultado == 1)
            {
                MessageBox.Show("Guardado con éxito");
            }
        }
        catch
        {
            MessageBox.Show("Error en almacenamiento de pedido", "Warning!", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
}
```

```
}
try
{
    foreach (DataGridViewRow row1 in dgv_det_ped.Rows)
    {
        if (conteo < dgv_det_ped.Rows.Count)
        {
            string id_menu = Convert.ToString(row1.Cells[0].Value);
            int correlativo = Convert.ToInt16(row1.Cells[1].Value);
            double cantidad = Convert.ToDouble(row1.Cells[2].Value);
            double descuento = Convert.ToDouble(row1.Cells[4].Value);
            int resultado = pedido.inserta_detalle(txt_nopedido.Text.Trim(), id_menu, correlativo, cantidad, descuento);
            conteo = conteo + 1;
        }
        else
        {
            break;
        }
    }
}
catch
{
    //MessageBox.Show(ex.Message);
    MessageBox.Show("Error en el pedido", "Warning!", MessageBoxButtons.OK, MessageBoxIcon.Error);
}
string codigo = txt_nopedido.Text;

MessageBox.Show("Su número de pedido es: " + codigo, "Datos de Pedido", MessageBoxButtons.OK, MessageBoxIcon.Information);
limpiar();
}
```