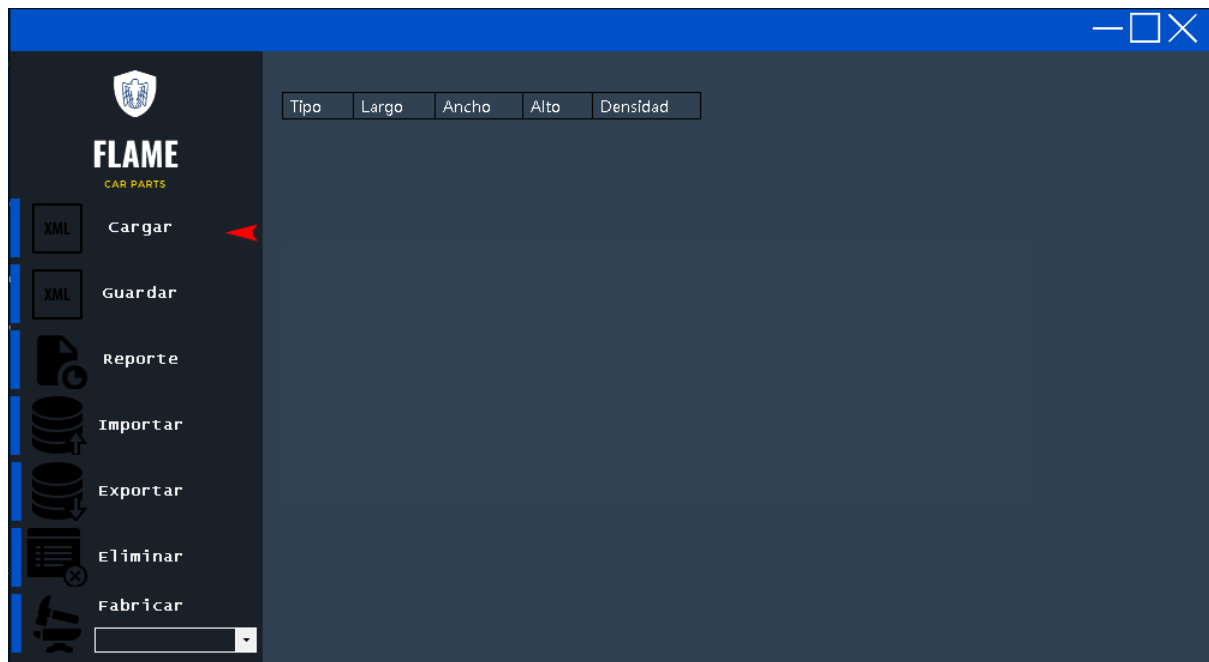


Car Parts

Para poder empezar a utilizar la aplicación, primero es necesario cargar una lista de materiales.

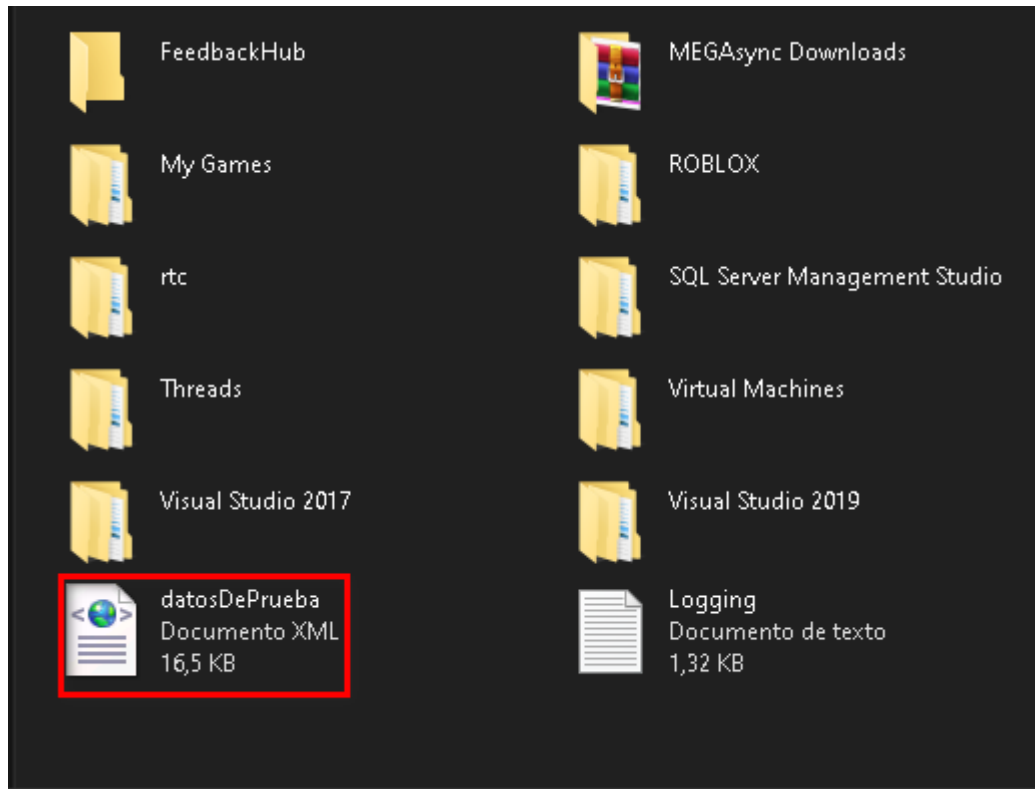
A modo de prueba el mismo le proporcionara una hasta que se sienta comodo con el sistema y usted pueda implementar sus propias listas.

1-



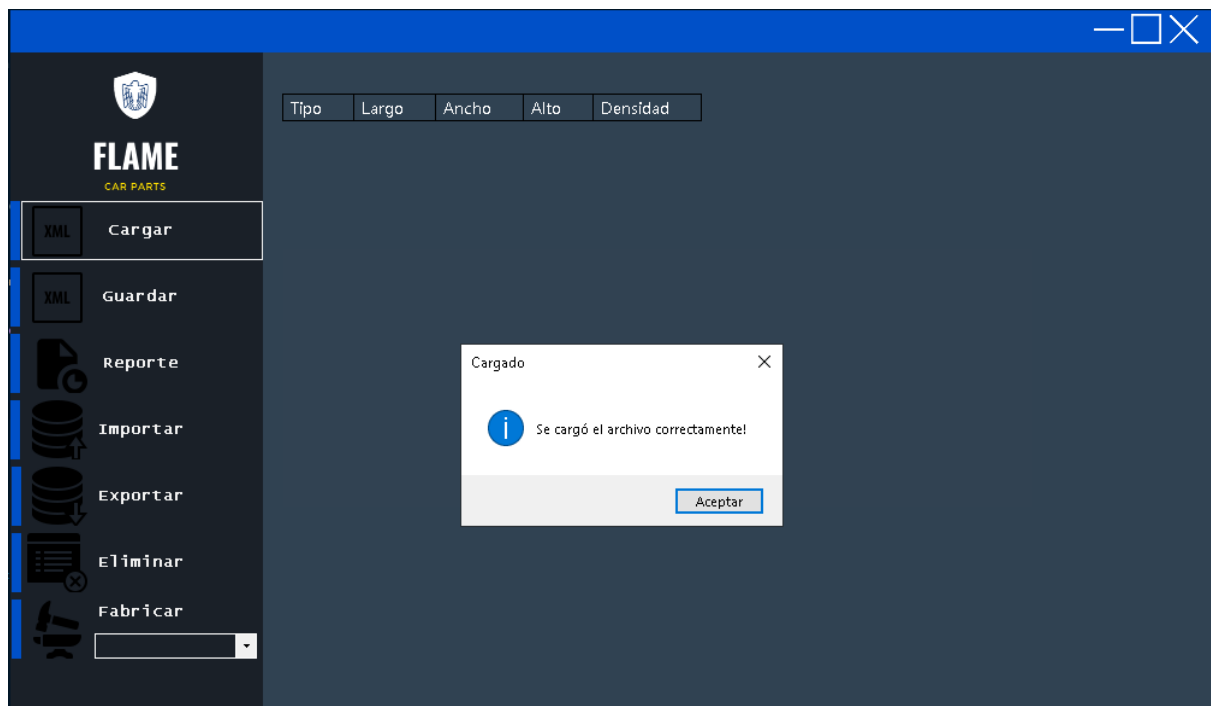
Seleccionar la opción Cargar, este abrira una ventana en su carpeta Documentos, la cual ya contara con la lista de prueba.

2-



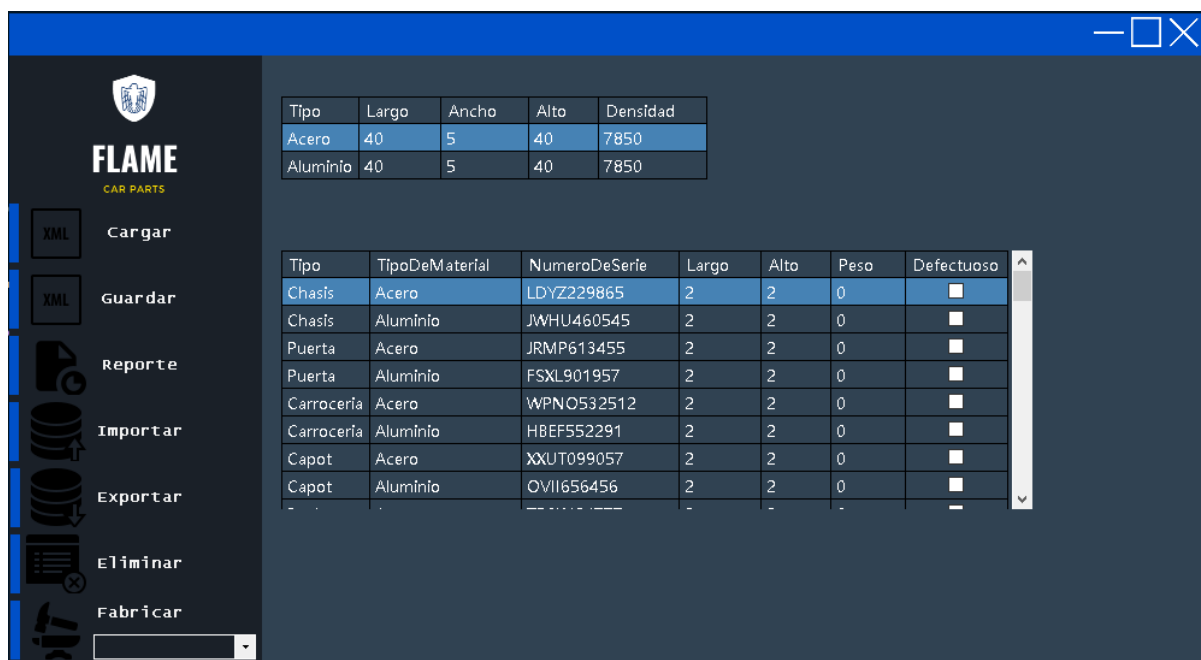
Seleccionar el archivo y presionar Aceptar.

3-



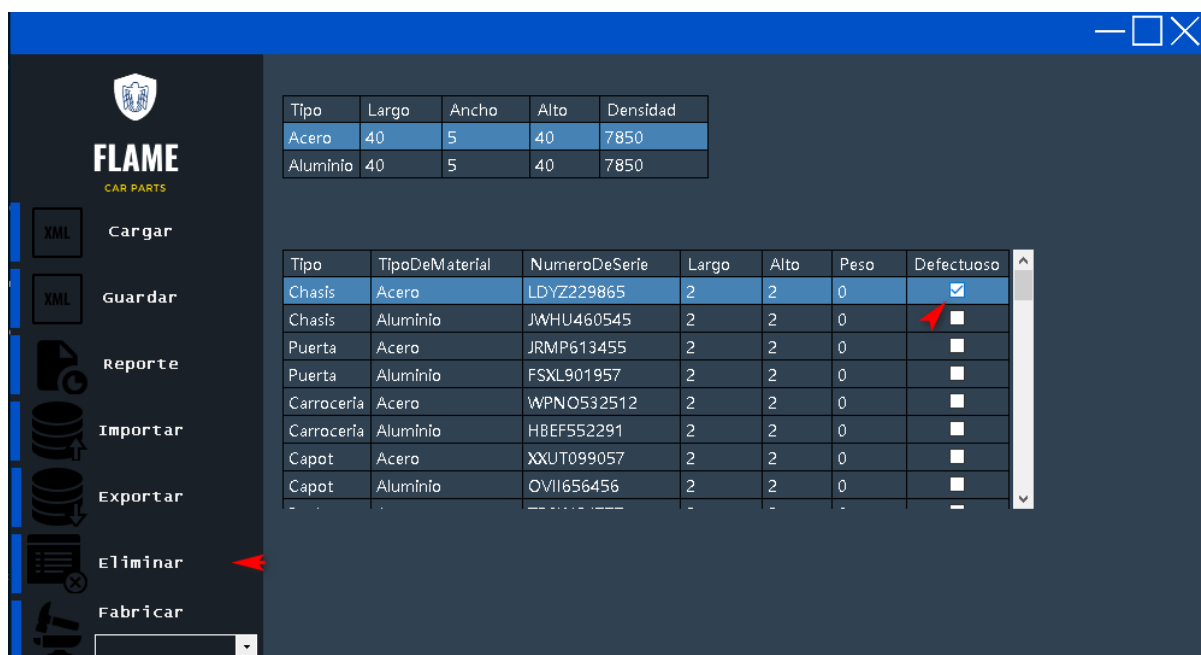
El sistema informara que el archivo se cargo correctamente.

4-



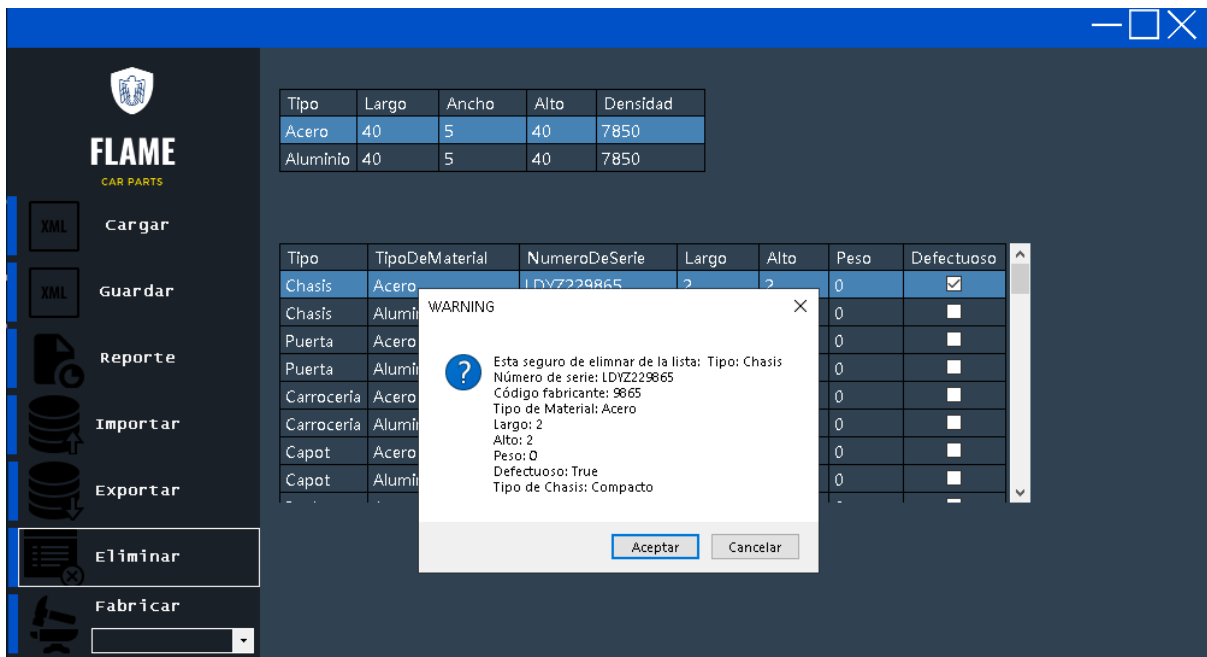
Según la lista, se actualizará las vistas con los datos cargados.

5-



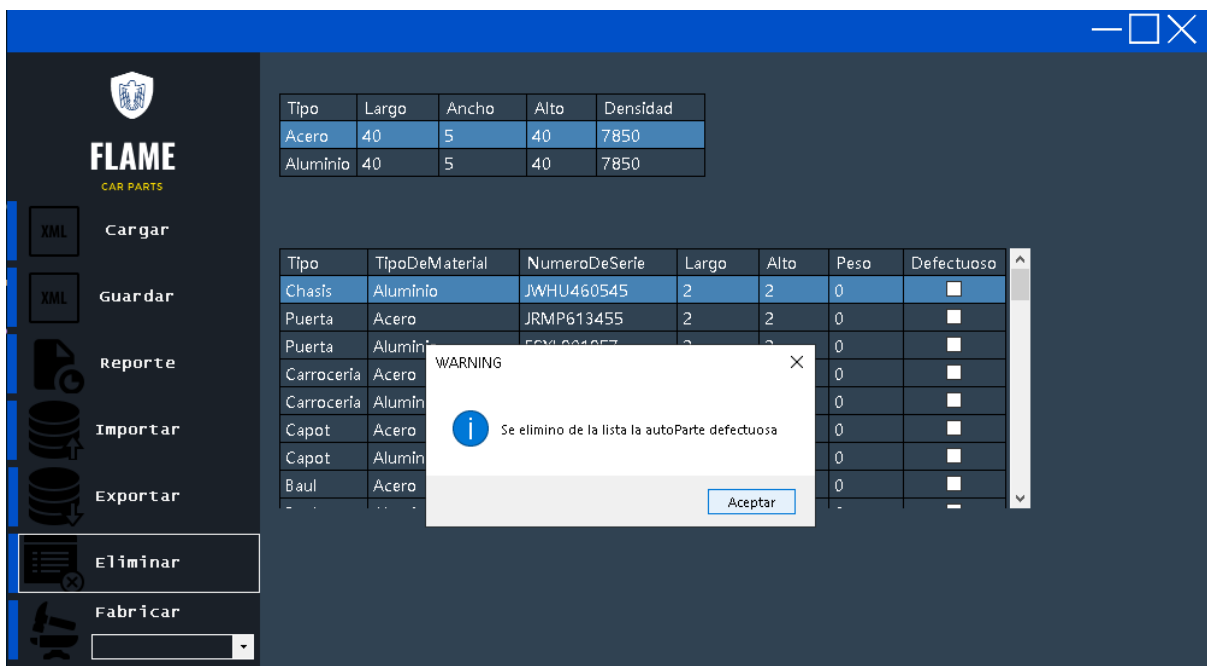
Si desea eliminar una pieza, deberá seleccionar la opción de la columna Defectuoso, y dejar marcada la pieza, luego presionar Eliminar y esto realizará la limpieza de la autoparte de la aplicación y desde la base de datos.

6-



Emitira un aviso para que usted confirme la eliminacion.

7-

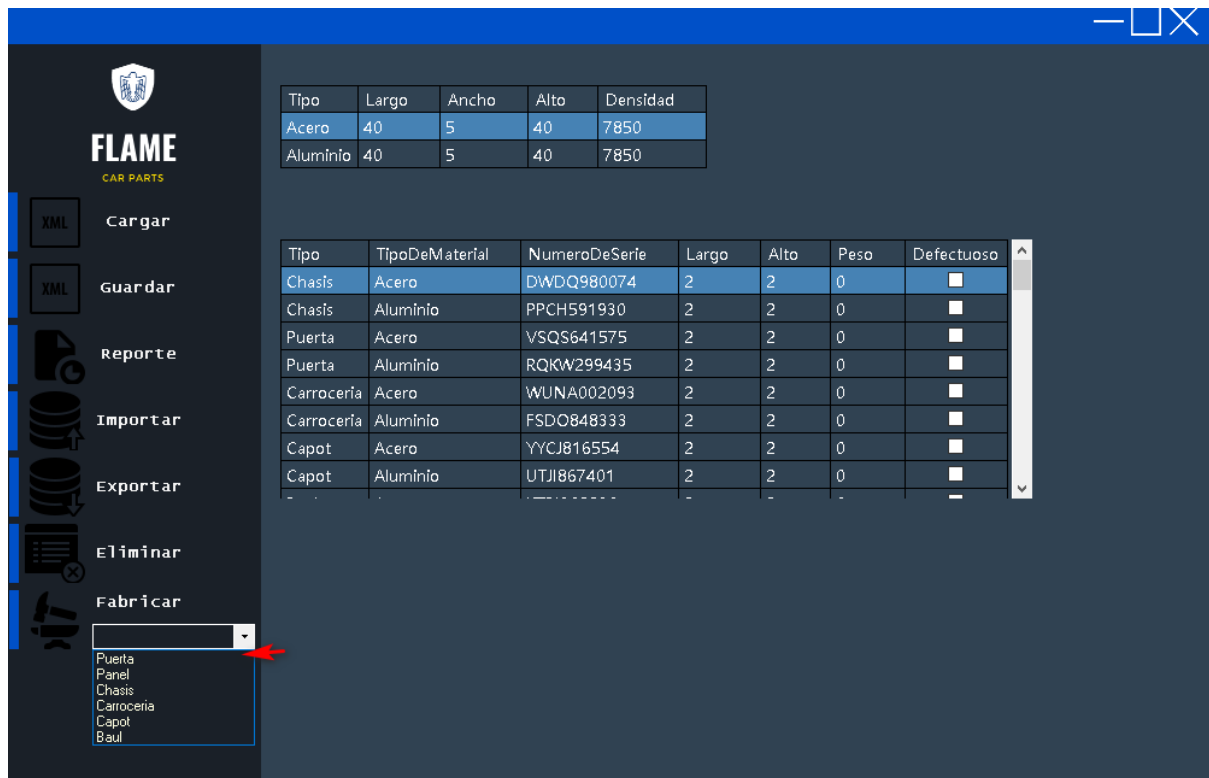


Si acepto las condiciones, la pieza se eliminara.

Para la creación de piezas.

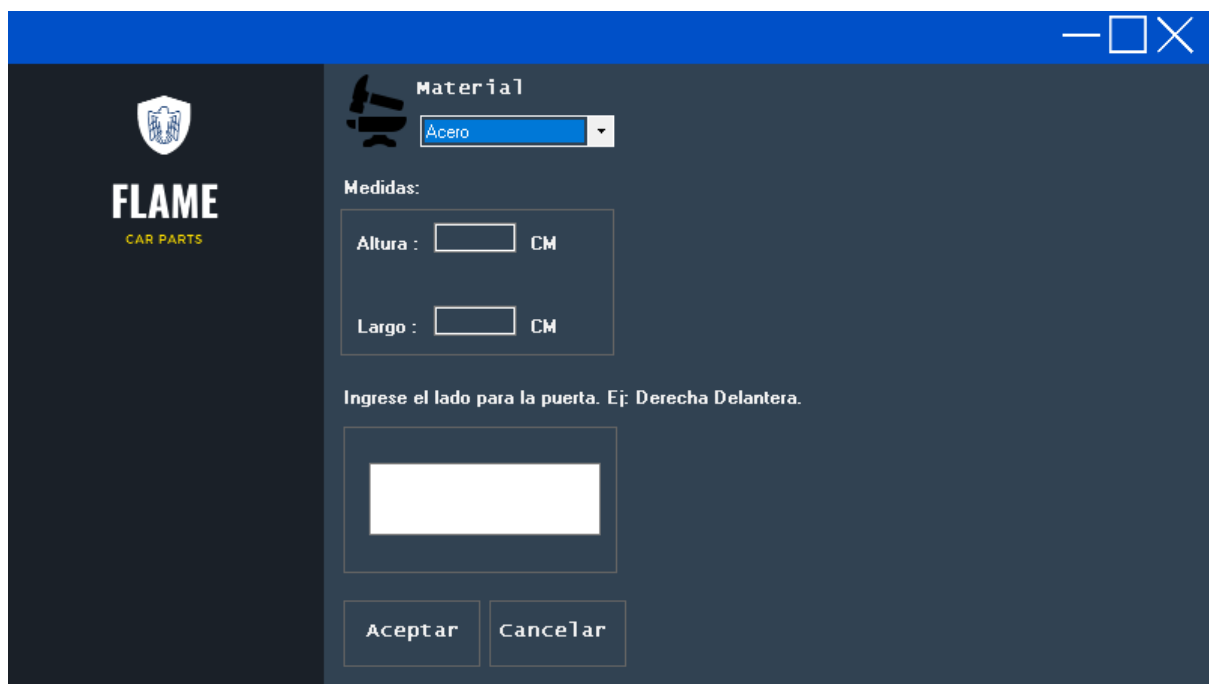
1- Cargar la lista de materiales. (Ya explicado.)

2-



Seleccionar una de las auto partes disponibles para crear.

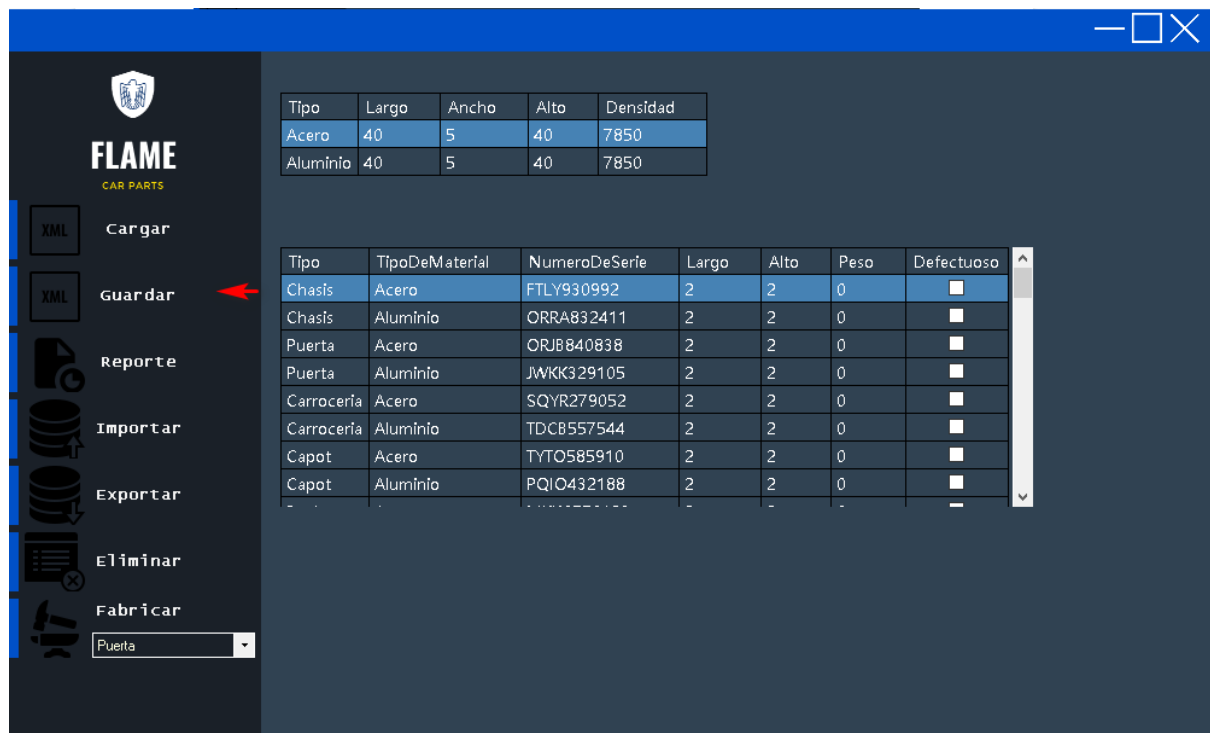
3-



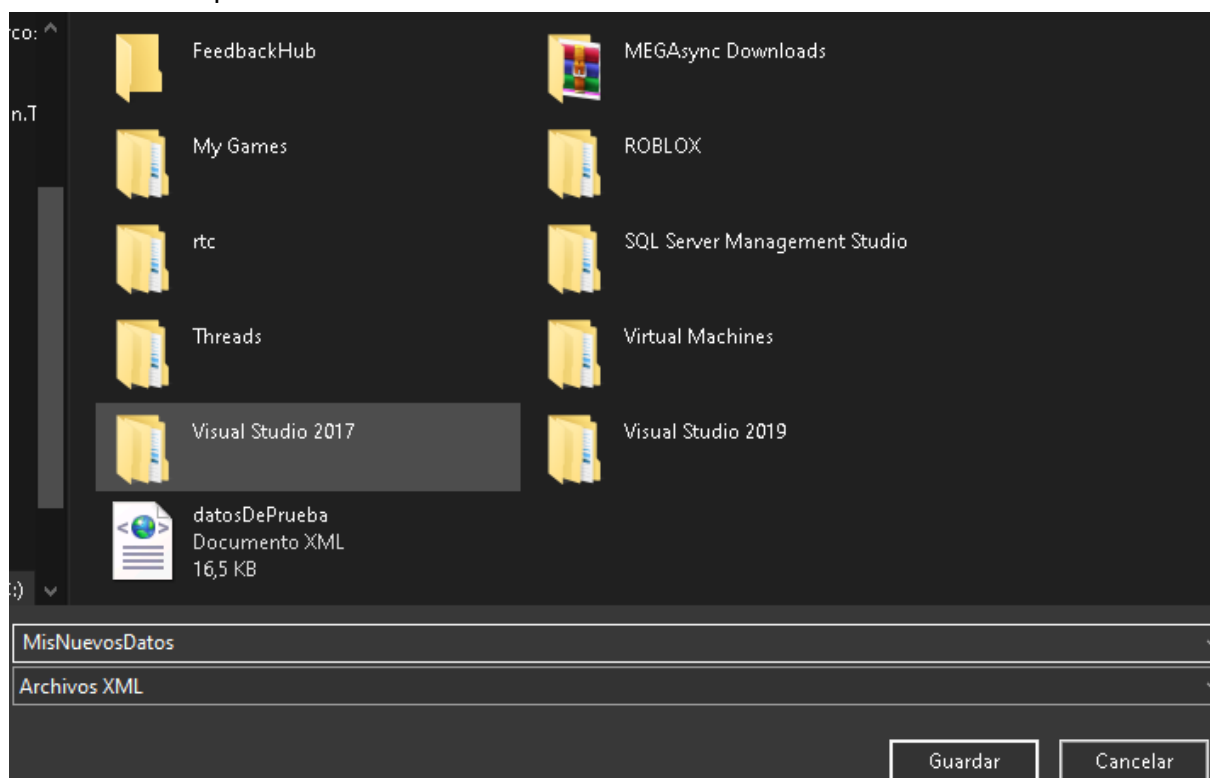
Esto habilitara una nueva ventana, donde debera ingresar las medidas para la pieza, y una descripcion. Luego presionar Aceptar y la pieza se vera reflejada en la lista.

Para Guardar los datos.

1-



Seleccionar la opcion Guardar en formato xml.



Esto le permitirá tener una copia de sus datos.

Para obtener un Reporte.

XML

Cargar

XML

Guardar

Reporte

Importar

Exportar

Eliminar

Fabricar

Puerta

Tipo	Largo	Ancho	Alto	Densidad
Acero	40	5	40	7850
Aluminio	40	5	40	7850

Tipo	TipoDeMaterial	NumeroDeSerie	Largo	Alto	Peso	Defectuoso
Chasis	Acero	FTLY930992	2	2	0	<input type="checkbox"/>
Chasis	Aluminio	ORRA832411	2	2	0	<input type="checkbox"/>
Puerta	Acero	ORJB840838	2	2	0	<input type="checkbox"/>
Puerta	Aluminio	JWKK329105	2	2	0	<input type="checkbox"/>
Carroceria	Acero	SQYR279052	2	2	0	<input type="checkbox"/>
Carroceria	Aluminio	TDCB557544	2	2	0	<input type="checkbox"/>
Capot	Acero	TYTO585910	2	2	0	<input type="checkbox"/>
Capot	Aluminio	PQIO432188	2	2	0	<input type="checkbox"/>

Esto le proporcionara un simple reporte en formato pdf.

XML

Cargar

XML

Guardar

Reporte

Importar

Exportar

Eliminar

Fabricar

Tipo	Largo	Ancho	Alto	Densidad
Acero	40	5	40	7850
Aluminio	40	5	40	7850

Tipo	TipoDeMaterial	NumeroDeSerie	Largo	Alto	Peso	Defectuoso
Chasis	Acero	FWHR432643	2	2	0	<input type="checkbox"/>
Chasis	Aluminio	BVWV316698	2	2	0	<input type="checkbox"/>
Puerta	Acero	SAHM552151	2	2	0	<input type="checkbox"/>
Puerta	Aluminio				0	<input type="checkbox"/>
Carroceria	Acero				0	<input type="checkbox"/>
Carroceria	Aluminio				0	<input type="checkbox"/>
Capot	Acero				0	<input type="checkbox"/>
Capot	Aluminio				0	<input type="checkbox"/>

WARNING.

Se genero el reporte en

C:\Users\ [redacted] \Documents\Report_12_07_2021_09_26.pdf

Aceptar

Reporte

Lista de material sobrante

Tipo	Largo	Ancho	Alto	Densidad
Acero	40	5	40	7850
Aluminio	40	5	40	7850

Se ha fabricado un total de: 60 piezas.

Lista de piezas creadas

Tipo	Material	N° de serie	Alto	Largo	Peso	Defectuoso	Descripcion
Chasis	Acero	FWHR432643	2	2	0	No	Compacto
Chasis	Aluminio	BVWV316698	2	2	0	No	Monocausico
Puerta	Acero	SAHM552151	2	2	0	No	Delantera derecha
Puerta	Aluminio	GHTZ065242	2	2	0	No	Delantera izquierda
Carroceria	Acero	ZWEL075026	2	2	0	No	SUV
Carroceria	Aluminio	XELB126589	2	2	0	No	Coupe
Capot	Acero	BBQJ466201	2	2	0	No	Apertura trasera
Capot	Aluminio	FMUS758944	2	2	0	No	Apertura delantera
Baul	Acero	ZVSE715401	2	2	0	No	101 litros
Baul	Aluminio	VVFU952288	2	2	0	No	500 litros
Panel	Acero	CKAB880688	2	2	0	No	Delantero derecho
Panel	Aluminio	MHFM141511	2	2	0	No	Delantero izquierdo
Chasis	Acero	UPJS329735	2	2	0	No	Compacto
Chasis	Aluminio	WVWK657432	2	2	0	No	Monocausico
Puerta	Acero	BVSW990740	2	2	0	No	Delantera derecha
Puerta	Aluminio	AURP650827	2	2	0	No	Delantera izquierda
Carroceria	Acero	BYRC357071	2	2	0	No	SUV
Carroceria	Aluminio	MSYA615207	2	2	0	No	Coupe
Capot	Acero	KPKB687568	2	2	0	No	Apertura trasera
Capot	Aluminio	DDWE692765	2	2	0	No	Apertura delantera
Baul	Acero	BEPD952428	2	2	0	No	101 litros
Baul	Aluminio	GCQN155807	2	2	0	No	500 litros
Panel	Acero	ZNNS179938	2	2	0	No	Delantero derecho

Hilos.

```
[señal] FormPuerta.cs FormPrincipal.cs FormPanel.cs FormChas...
```

```
    }

    /// <summary>
    /// Metodo encargado de invocar el hilo que se encarga de actualizar un datagrid.
    /// </summary>
    1 referencia
    public void Actualizar()
    {
        try
        {
            while (true)
            {
                if (this.dgvListMaterials.InvokeRequired)
                {
                    this.dgvListMaterials.BeginInvoke((MethodInvoker)delegate ()
                    {
                        dgvListMaterials.DataSource = fabrica.Materiales;
                        dgvListMaterials.Refresh();
                    });
                }
                else
                {
                    FormatoParaDataGridViewMateriales(fabrica);
                }
                Thread.Sleep(2000);
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
            logger.saveReport(ex);
        }
    }
}
```

Extension.

```
1 referencia
static class Extensora
{
    /// <summary>
    /// Devuelve el codigo de fabricante.
    /// </summary>
    /// <param name="source"></param>
    /// <returns></returns>
    1 referencia
    public static string ObtenerCodigoFabricante(this string source)
    {
        return source.Substring(source.Length - 4);
    }
}
```

Eventos y Delegados.

```
/// <summary>
/// Delegado para actualizar la lista de auto partes.
/// </summary>
public delegate void ActualizarListaDePiezas();
4 referencias
public partial class FormPrincipal : Form
{
    private Fabrica fabrica;
    private Logger logger;
    Thread actualizadorDgvMateriales;

    /// <summary>
    /// Evento encargado de actualizar los datos de la lista de autopartes.
    /// </summary>
    public event ActualizarListaDePiezas actualizarDato;

    1 referencia
    public FormPrincipal()
    {
        InitializeComponent();
        this.StartPosition = FormStartPosition.CenterScreen;
        this.fabrica = new Fabrica("Mi fábrica");
        this.Text = "Fábrica";
        actualizadorDgvMateriales = new Thread(this.Actualizar);
        logger = new Logger(Environment.GetFolderPath(Environment.SpecialFolder.MyDocuments) + "\\\" + "Logging.txt");
        FormatoParaDataGridViewMateriales(fabrica);

        CreacionAutomaticaDeMateriales(); // Cargar una lista hardcodiada.
    }

    /// <summary>
```

```
private void btnCargarXML_Click(object sender, EventArgs e)
{
    try
    {
        if (fabrica.Materiales.Count != 0)
            throw new FileNotFoundException("No puede recargar una lista nueva, aún posee material disponible.");

        OpenFileDialog openFileDialog = new OpenFileDialog();
        openFileDialog.InitialDirectory = Environment.GetFolderPath(Environment.SpecialFolder.MyDocuments);

        if (openFileDialog.ShowDialog() == DialogResult.OK)
        {
            this.fabrica = Fabrica.Read(openFileDialog.SafeFileName);
            this.Text = "Fábrica " + this.fabrica.NombreFabrica;
            MessageBox.Show("Se cargó el archivo correctamente!", "Cargado", MessageBoxButtons.OK, MessageBoxIcon.Information);
        }

        if (!(this.actualizadorDgvMateriales is null) && !actualizadorDgvMateriales.IsAlive)
        {
            actualizadorDgvMateriales.Start();
        }
        else
        {
            actualizadorDgvMateriales.Abort();
        }
        actualizarDato += ActualizarListaDeAutoPartes;
        actualizarDato.Invoke();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
        logger.saveReport(ex);
    }
}
```

Interfaces.

```

namespace Entidades.Entidades.Database
{
    public interface IDatabase<T>
    {
        List<T> GetAll();

        bool Delete(string input);

        bool Update(T entity);
    }
}

```

SQL clase MaterialesDAO

```

public override List<Material> GetAll()
{
    using (SqlConnection connection = new SqlConnection(connectionString))
    {
        using (SqlCommand cmd = new SqlCommand($"SELECT * FROM Materiales", connection))
        {
            try
            {
                List<Material> materials = new List<Material>();

                cmd.CommandType = CommandType.Text;
                connection.Open();
                using (SqlDataReader dataReader = cmd.ExecuteReader())
                {
                    while (dataReader.Read())
                    {
                        Material material = new Material();

                        switch (dataReader["tipoDeMaterial"].ToString())
                        {
                            case "Acero":
                                material = new Material { TipoDeMaterial = (ETipoDeMaterial)Enum.Parse(typeof(ETipoDeMaterial), dataReader["tipoDeMaterial"].ToString()), Largo = double.Parse(dataReader["largo"].ToString()),
                                break;
                            case "Aluminio":
                                material = new Material { TipoDeMaterial = (ETipoDeMaterial)Enum.Parse(typeof(ETipoDeMaterial), dataReader["tipoDeMaterial"].ToString()), Largo = double.Parse(dataReader["largo"].ToString()),
                                break;
                        }
                        materials.Add(material);
                    }
                }
                return materials;
            }
            catch (Exception ex)
            {
                logger.saveReport(ex);
                throw ex;
            }
            finally
            {
                if (connection.State == ConnectionState.Open)
                {
                    connection.Close();
                }
            }
        }
    }
}

```

```

public override bool Update(Material item)
{
    bool retorno = false;
    try
    {
        command = new SqlCommand();
        connection = new SqlConnection(connectionString);
        command.Connection = connection;
        command.CommandType = CommandType.Text;

        command.CommandText = "INSERT INTO Materiales(largo,alto,ancho,densidad,tipoDeMaterial) VALUES (@largo,@alto,@ancho,@densidad,@tipoDeMaterial)";

        command.Parameters.AddWithValue("@largo", item.Largo);
        command.Parameters.AddWithValue("@alto", item.Alto);
        command.Parameters.AddWithValue("@ancho", item.Ancho);
        command.Parameters.AddWithValue("@densidad", item.Densidad);
        command.Parameters.AddWithValue("@tipoDeMaterial", item.TipoDeMaterial.ToString());

        connection.Open();
        command.ExecuteNonQuery();

        retorno = true;
    }
    catch (Exception ex)
    {
        logger.saveReport(ex);
        throw ex;
    }
    finally
    {
        if (connection.State == ConnectionState.Open)
        {
            connection.Close();
        }
    }

    return retorno;
}

```

Test Unitarios

```
namespace TestUnitarios
{
    [TestClass]
    public class FabricaTest
    {
        [TestMethod]
        public void AlFabricarPartesDistintasDebeResponderOK()
        {
            Fabrica miFabrica = new Fabrica("FabricaTest");
            Chasis chasis = new Chasis(ETipoDeMaterial.Acero, 2, 2, "Compacto");
            Puerta puerta = new Puerta(ETipoDeMaterial.Aluminio, 2, 2, "Derecha delantera");

            bool output = miFabrica + chasis && miFabrica + puerta;

            Assert.IsTrue(output);
        }

        [TestMethod]
        [ExpectedException(typeof(AutoPartesException))]
        public void AlAgregarPartesIgualesDebeTirarException()
        {
            Fabrica miFabrica = new Fabrica("FabricaTest");
            Chasis chasis = new Chasis(ETipoDeMaterial.Acero, 2, 2, "Compacto");

            bool output = miFabrica + chasis && miFabrica + chasis;
        }
    }
}
```