

**UNIVERSIDAD TECNOLÓGICA DE PUEBLA**

**Gestión de inventario**

**Matesa Plastic**

**Alumnos:**

**Amaro Coyotl Irais Katherin**

**Gerónimo Martínez Vicente**

**Carrera Melchor Benjamín**

**Procopio Ramírez Rodrigo**

**Cortes Nieto Verónica**

**Grado y Grupo: 7° “B”**

**Cuatrimestre: Enero - Abril 2017**

1. **Análisis del sistema Matesa Plastic**
2. **Modelo Entidad Relación**
3. **Modelo relacional**
4. **Diagrama de casos de uso**
5. **Diseño**
6. **Codificación**

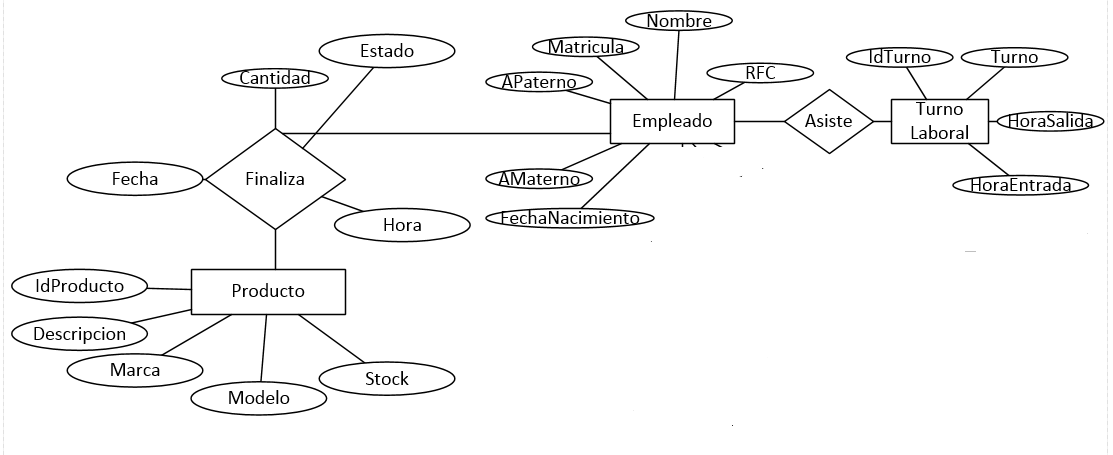
**I.- Análisis del sistema Matesa Plastic**

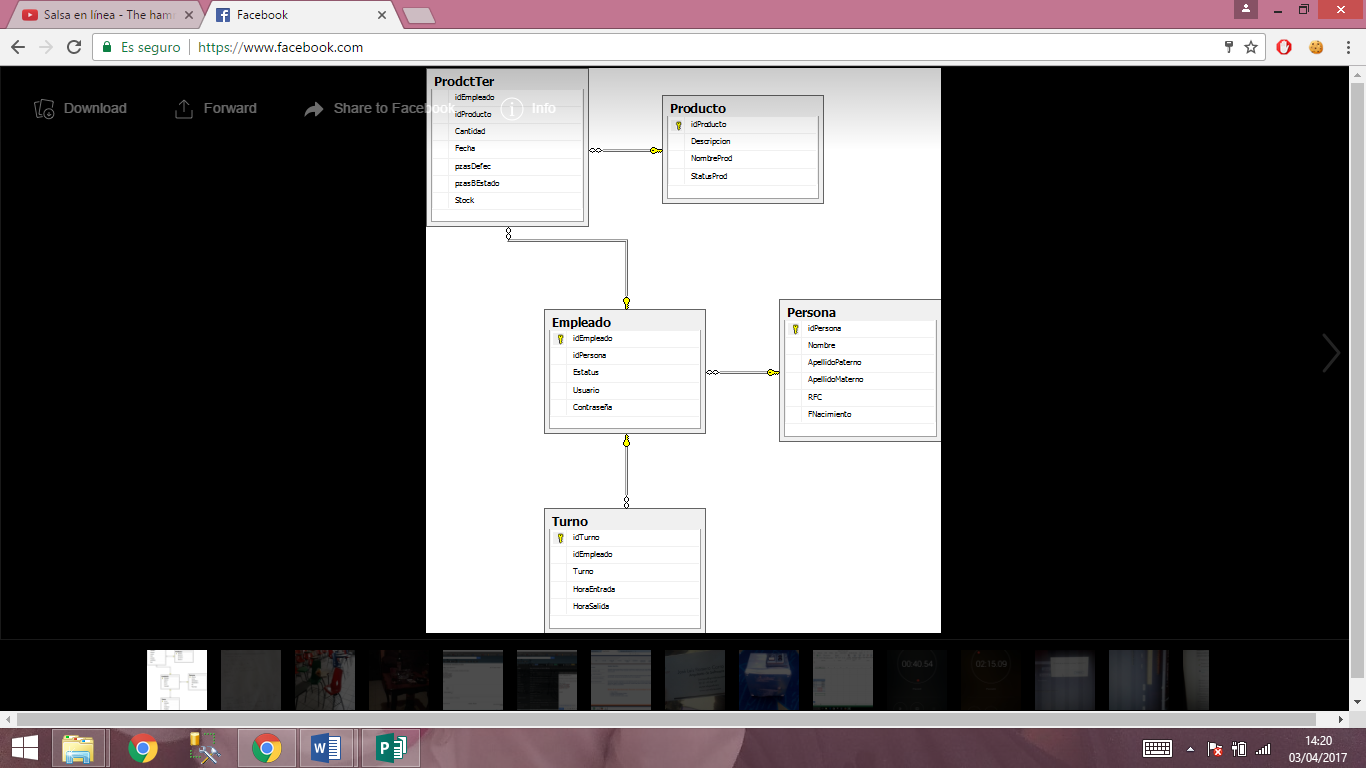
Es una empresa dedicada a la inyección de plásticos de ingeniería, que se crea en el 2015 debido a la creciente demanda en el sector automotriz.

Ofrece soluciones a las diferentes problemáticas del mercado en el sector automotriz, con experiencia de más de 15 años avalado el trabajo mediante el uso de diferentes metodologías.

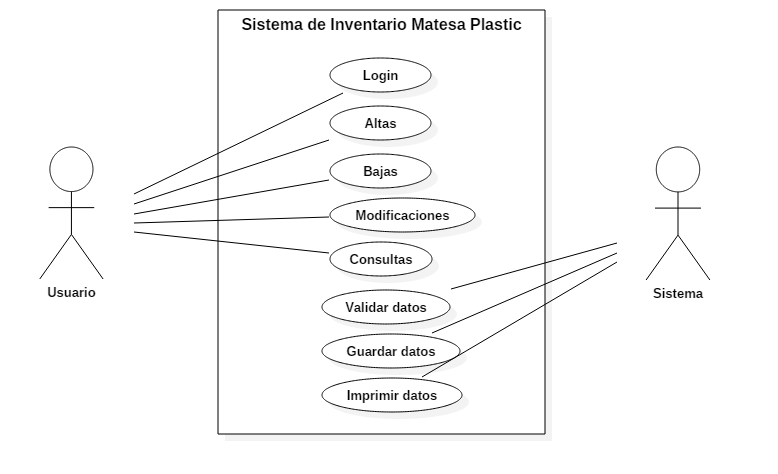
Matesa Plastic requiere un sistema de inventarios que permita:

1. Conocer la fecha, nombre del empleado y el turno en que se terminó de producir un producto.
2. Conocer el número de productos existentes en el almacén.
3. Tener un control de productos que ya se han vendido.
4. Identificar el estado en el que se encuentra un producto y la persona encargada de la producción.
5. Obtener reportes de productos terminados en un rango de fechas determinada ya sea en un día, una semana, un mes etc.
6. Exportar datos de un reporte a un archivo PDF, Word o Excel.
7. Registro de nuevos productos
8. Registro de empleados
9. Permitir que los empleados ingresen al sistema con un usuario y una contraseña y que puedan registrar productos terminados durante el turno laboral.
10. Los administradores tendrán acceso a realizar consultas de determinado empleado, consulta del inventario, modificaciones en caso de cometer errores.

**II.- Modelo Entidad Relacional**

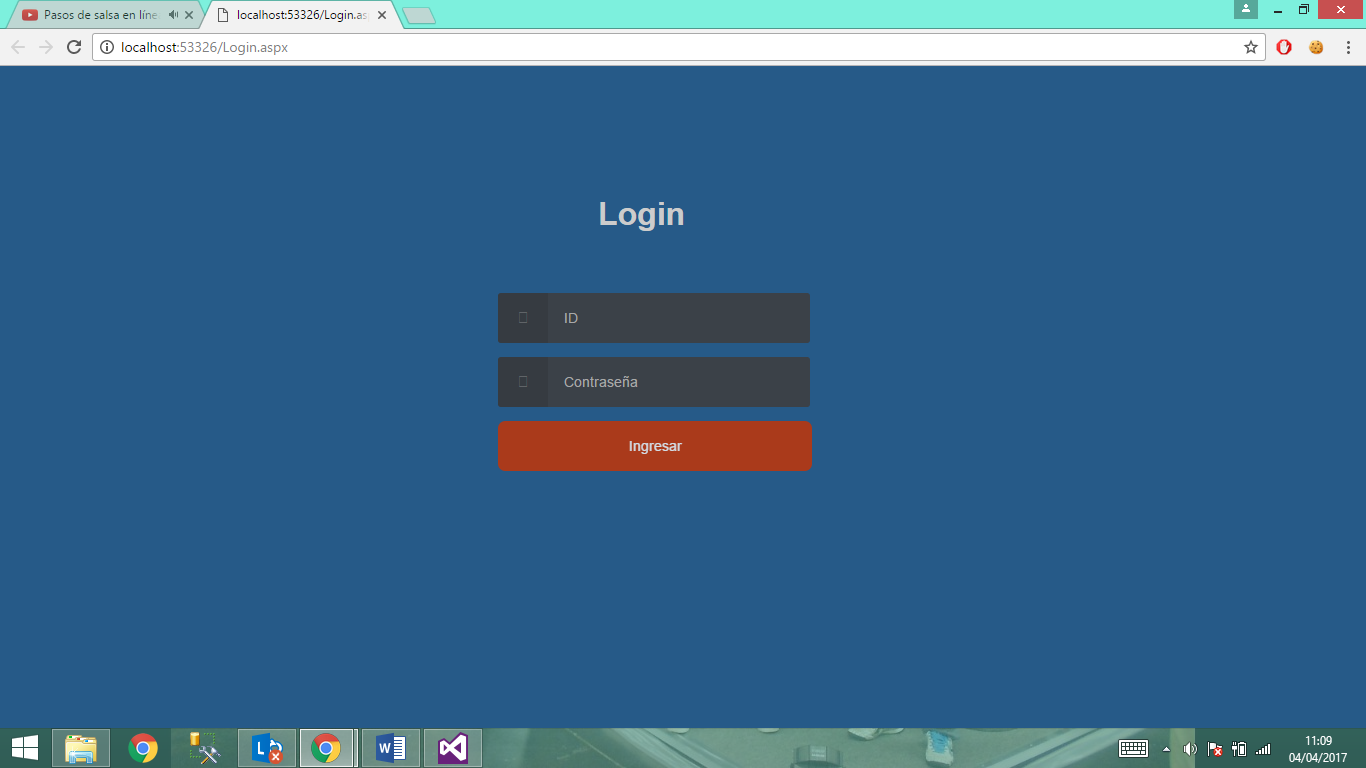
**III.- Modelo Relacional**

**IV.- Diagrama de casos de uso**

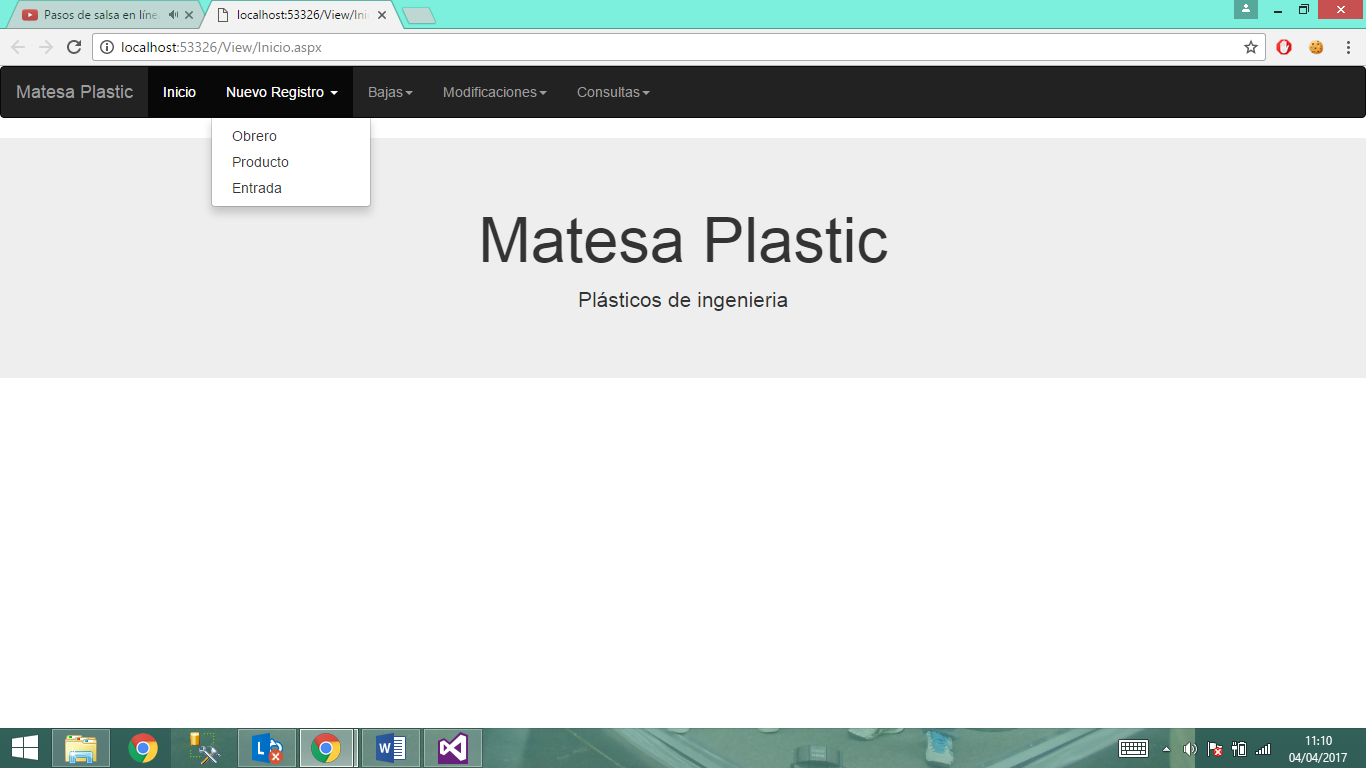


**V.- Diseño**

Login (Ingresar el nombre del usuario y su contraseña)



Barra de navegación



**VI.- Codificación**

* Conexión a la base de datos

public class Datos

{

public SqlDataReader dr { get; set; } //

public SqlCommand cadena\_sql { get; set; }

SqlDataAdapter adapt; // data set

SqlConnection conn; // cadena de conexion

public bool Conectar()

{

conn = new SqlConnection();

conn.ConnectionString = "Data source =(local); Initial catalog =MatesaP; Integrated security=true"; // nombre de servidor y nombre de la base de datos

try

{

conn.Open();

return true;

}

catch (Exception error)

{

MessageBox.Show("Error al conectarse a la base de datos" + error.Message);

return false;

}

}

public void desconectar()

{

conn.Close();

}

public void construye\_reader(string cadena)

{

cadena\_sql = new SqlCommand();

cadena\_sql.Connection = conn;

cadena\_sql.CommandText = cadena;

}

public SqlDataReader ejecuta\_reader()

{

try

{

dr = cadena\_sql.ExecuteReader();

return dr;

}

catch (Exception error)

{

MessageBox.Show("Error al ejecutar el Reader" + error.Message);

return null;

}

}

public SqlCommand construye\_command(string cadena)

{

cadena\_sql = new SqlCommand(cadena, conn);

return cadena\_sql;

}

public int ejecutanonquery()

{

int afectados;

try

{

afectados = cadena\_sql.ExecuteNonQuery();

return afectados;

}

catch (Exception error)

{

MessageBox.Show("Error al ejecutar el reader" + error.Message);

return 0;

}

}

public SqlDataAdapter construye\_adapter(string cadena)

{

adapt = new SqlDataAdapter(cadena, conn);

return adapt;

}

public DataRow extrae\_registro(SqlDataAdapter adapter, string tabla)

{

DataSet ds = new DataSet();

DataRow fila;

try

{

adapter.Fill(ds, tabla);

DataTable miTabla = ds.Tables[tabla];

fila = miTabla.Rows[0];

return fila;

}

catch (Exception error)

{

MessageBox.Show("Error al extraer el registro" + error.Message);

return null;

}

}

public DataSet contustarDataSet(string consulta)

{

DataSet ds = new DataSet();

SqlCommand comando = new SqlCommand();

SqlDataAdapter adapter = new SqlDataAdapter();

comando.CommandText = consulta;

comando.Connection = conn;

adapter.SelectCommand = comando;

try

{

adapter.Fill(ds);

return ds;

}

catch (Exception error)

{

MessageBox.Show("No se pudo crear el dataset" + error.Message);

return null;

}

}

}

* Clase usuario

public class Usuario2

{

public String idUsuario { get; set; }

public String Usuario { get; set; }

public String Contraseña { get; set; }

public Usuario2()

{

idUsuario = "";

Usuario = "";

Contraseña = "";

}

public Usuario2(String u, String c, String n)

{

idUsuario = u;

Usuario = n;

Contraseña = c;

}

public bool acceso(String usu, String contra)

{

SqlConnection miconexion = new SqlConnection("Data Source=local; Initial catalog=MatesaP; Integrated security=true");

miconexion.Open();

//Datos conecta = new Datos();

//conecta.Conectar();

SqlCommand consulta = new SqlCommand(" select \*from Empleado where Usuario = '" + usu + "'and Contraseña='" + contra + "'", miconexion);

SqlDataReader ejecuta = consulta.ExecuteReader();

if (ejecuta.Read() == true)

{

miconexion.Close();

return true;

}

else

{

miconexion.Close();

}

return false;

}

}

Clase producto

public class Producto

{

public int idProducto { get; set; }

public string Descripcion { get; set; }

public string Nombre { get; set; }

public Producto()

{

idProducto = 0;

Descripcion = "";

Nombre = "";

}

public Producto(int i, string d, string n)

{

idProducto = i;

Descripcion = d;

Nombre = n;

}

**Codigo para dar de alta un producto**

public int altaProducto()

{

SqlCommand Comando;

Datos inserta = new Datos();

int regresa = 0;

if (inserta.Conectar())

{

String comando = "Insert into Producto (Descripcion, Nombre)" +

"values ('" + Descripcion + "','"+Nombre+"')";

Comando = inserta.construye\_command(comando);

if ((inserta.ejecutanonquery()) != 0)

regresa = 1;

else

regresa = 0;

Comando.Connection.Close();

inserta.desconectar();

}

else

regresa = -1;

return regresa;

}

**Código para dar de baja un producto**

public int baja(int codigo)

{

SqlCommand comando;

Datos elimina = new Datos();

int regresa = 0;

if (elimina.Conectar())

{

comando = elimina.construye\_command("delete from Producto where idProducto=" + codigo);

if (elimina.ejecutanonquery() != 0)

regresa = 0;

else

regresa = 1;

}

else

regresa = -1;

return regresa;

}

**Código para realizar modificaciones en la entidad producto**

public int actualiza\_producto()

{

SqlCommand Comando;

Datos actualiza = new Datos();

int regresa = 0;

if (actualiza.Conectar())

{

Comando = actualiza.construye\_command("update Producto set idProducto= '" + idProducto + "',Descripcion= '" + Descripcion + "',Nombre= '" + Nombre + "' where idProducto= " + idProducto);

if (actualiza.ejecutanonquery() != 0)

regresa = 1;

else

regresa = 0;

Comando.Connection.Close();

actualiza.desconectar();

}

else

return -1;

return regresa;

}