**Como se hizo el Login para que entiendan como funciona.**

namespace PagoElectronico.Login

{

public partial class LogIn : Form

{

**1.Defino las variables de la clase.**

int intentosFallidos = 0;

string ultimoUserIngresado = "";

int maxIntentosFallidos = Convert.ToInt32(ConfigurationManager.AppSettings["MaxIntentosFallidosLogIn"]); (ConfigurationManager.AppSettings(atributo) Es para sacar datos del archivo de configuracion. Ver en PagoElectronico / App.Setting que esta definida la fecha, el maximo de intentos fallidos, bla bla)

Usuario user = new Usuario();

FSLogger logAuditoria; (El log auditoria hay que hacerlo en un archivo tipo log. La clase FSLogger es la que hace el log y tiene la funcion para escribir en el).

**2.LogIn\_Load realiza las siguientes acciones mientras se carga el formulario. Creo el archivo LOGAuditoria, y oculto los botones de rol. Hasta que no inicie bien la sesion, no van a aparecer los botones.**

private void LogIn\_Load(object sender, EventArgs e)

{

logAuditoria = new FSLogger("LogInAuditoria"); //Creo archivo log con nombre LogInAuditoria, en la ruta actual

btnSelecRol.Visible = false;

lblRol.Visible = false; //oculto botones de eleccion Rol. una vez que se loguea correctamente elije el rol.

cmbRol.Visible = false;

}

**3. Luego el usuario tiene que ingresar su username y password. Tengo que validar que se hayan ingresado campos en username y password. Y si se ingresaron corroborar que sea un usuario y que la contraseña ingresada sea la correcta.**

string claveIngresada = "";

private void btnIngresar\_Click(object sender, EventArgs e)

{//Ahora voy a ver que se haya logueado bien

if(ValidarCampos()){

return;} //vuelve a la app si no se ingreso nada,o datos erroneos

//este validar campos lo que hace es mirar que en username se haya ingresado algo y que ese algo sea del formato username. Lo mismo con password.

Aca tendriamos que ver que onda, porq dice en el tp que el username es admin, pero nuestro username era por le numero de dni. En la bd agregue un administrador con username 123 y aca lo que hago es que reconosca el username admin como el username 123. Habria que hacerlo de otra forma. Pero era para ir probando la aplicación.

if(txtUsername.Text == "admin"){

user.Username = "123";

}

else{

user.Username = txtUsername.Text;}

user.Password = txtPassword.Text;

**3.1 Verificar que el usuario exista y corroborar contraseña**

//si se ingresaron los campos, paso a verificar que el usuario exista y que esta sea su contraseña.

**3.1.1 encontrar el usuario en la bd**

Tengo que traer al usuario de la base de datos. Para ello utilizo la funcion user.obtenerUsuarioActivoPorUsername() que lo que hace es pedirle al usuario actual que traiga de la bd su usuario completo, con todos sus datos. Para ello hay que decirle a la bd que ejecute el storeProcedure “TraerUsuarioActivoPorIdUsuario”. Por que los activos? Porque un usuario puede estar deshabilitado (por ej, supero la cantidad max de intentos), en tal caso no podra ingresar al sistema.

**COMO HAGO PARA LLAMAR A UN STORED PROCEDURE DE LA BASE DE DATOS!??**

Para ejecutar un store procedure ejecuto la siguiente linea

DataSet ds = SQLHelper.ExecuteDataSet("traerUsuarioActivoPorUsername", CommandType.StoredProcedure, parameterList);

El primer parametro es el nombre del SP, el segundo es el tipo de cosa que le estoy mandando a la base (en este caso un stored procedure), y el tercer atributo las variables que necesita el storedProcedure.

Cuando se setea la lista de parametros se hace esto.

private void setearListaDeParametrosConUsuario(){

parameterList.Add(new SqlParameter("@Username", this.Username)); //el nombre de la variable @Username de aca tiene que ser igual a la del store procedure que defini en .sql

Una vez que ya cargue la lista con los parametros ejecuto la linea

DataSet ds = SQLHelper.ExecuteDataSet("traerUsuarioActivoPorUsername", CommandType.StoredProcedure, parameterList);

Luego el ExecuteDataSet me devuelve la consulta como se muestra en el manager. En forma de tabla. Por lo que con la funcion DataRowToObject “transformo” la tabla en variables.

public override void DataRowToObject(DataRow dr)

{

// Esto es tal cual lo devuelve el stored de la DB

// los dr["xxxx"] son los nombres de las columnas de la consulta

this.usuario\_id = Convert.ToInt32(dr["usuario\_id"]);

this.Username = dr["usuario\_username"].ToString();

this.Password = dr["usuario\_password"].ToString();

this.Estado = Convert.ToBoolean(dr["usuario\_estado"]);

this.ClaveAutoGenerada = esClaveAutoGenerada(dr);

}

Voy cargando los datos en mi objeto user que defini al principio de la clase login.

**ENTONCES EL CODIGO COMPLETO DE OBTENER USUARIOS ACTIVOS POR USERNAME QUEDA ASI**

public bool obtenerUsuarioActivoPorUsername()

{

bool @encontroUsuario = false;

setearListaDeParametrosConUsuario();

DataSet ds = SQLHelper.ExecuteDataSet("traerUsuarioActivoPorUsername", CommandType.StoredProcedure, parameterList);

parameterList.Clear();

if (ds.Tables[0].Rows.Count == 1) //esto se fija si la consulta tiene una linea. Si la tiene, encontro el usuario.

{

DataRowToObject(ds.Tables[0].Rows[0]);

encontroUsuario = true;

}

return encontroUsuario;

}

**3.1.1 Una vez encontrado corroborar que la contraseña sea la correcta. Ya con todos los datos del usuario cargado comparo que la contraseña que traje es igual a la ingresada en la app**

//verificar si el login es correcto o no

claveIngresada = Encryptor.GetSHA256(txtPassword.Text); //encripto la clave para luego compararla en la BD

if (user.obtenerUsuarioActivoPorUsername())

{//ENCONTRO EL USUARIO

if (user.Password.Trim() == claveIngresada.Trim())

{

Si la contraseña es correcta, ingreso exitosamente

logAuditoria.EscribirLog(user.Username, "Exitoso", intentosFallidos);

RealizarAccionesLogInExitoso();

}

else

{

if (txtUsername.Text != ultimoUserIngresado) //Me fijo si fue el ultimo user ingresado desde que se inicio la aplicacion

{

intentosFallidos = 0; //significa que fue su primer intento fallido

ultimoUserIngresado = txtUsername.Text;

}

intentosFallidos++;

logAuditoria.EscribirLog(user.Username, "Fallido",intentosFallidos); logueo intento fallido, con cantidad de intentos.

//Verifico si se alcanzo la max, si se alcanzo lo bloqueo. Si no, sigo.

VerificarSiSeAlcanzoLaCantidadMaxima();

}

}

Else //Aca llego si no se encontro el usuario al principio. En tal caso aviso queu el usuario es incorrecto o se encuentra bloqueado

{

intentosFallidos = 0;

MessageBox.Show("El usuario es incorrecto o se encuentra bloqueado. Por favor, ingrese los datos correctamente", "Log In fallido", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}