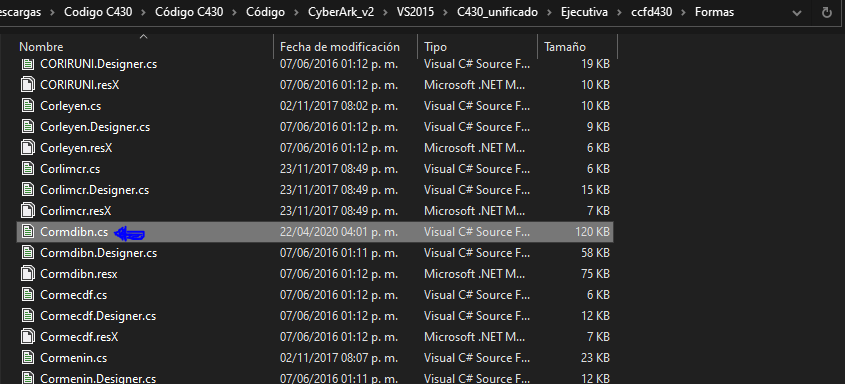
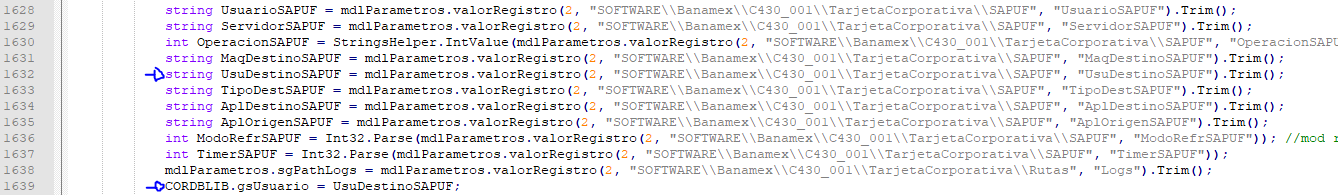
INTEGRACIÓN DE CÓDIGO .NET

1.- Dentro del proyecto de **TCd430** en la carpeta de **Formas**, tomamos la clase Cormdibn.cs. Como se muestra en la siguiente imagen.

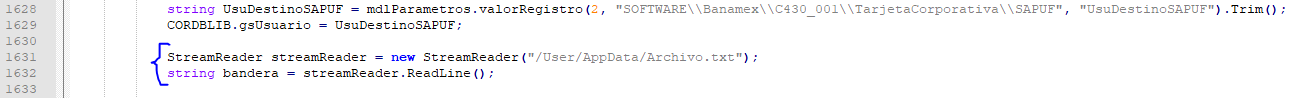
  
Ilustración 1: Explorador de archivos. Windows

2.- Moverse a la línea de código ***1628*** y seleccionar el código hasta la línea ***1766*** cortar y pegar en un archivo externo como *respaldo* ya que esté código se va a ingresar en la condición correspondiente

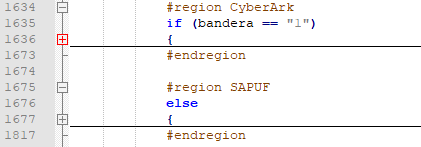
3.- Las líneas que se señalan a continuación se quedan (línea 1632 y 1639), el resto de las líneas de código se copian al *respaldo*.

  
Ilustración 2: Archivo Cormdibn.cs. Notepad++

4.- Debajo del resultado del código presentado anteriormente ingresamos el proceso de lectura del archivo para asignarle un valor a la bandera, la cual realizará la condición de selección del proceso entre SAPUF y CyberArk, el código quedaría de la siguiente manera.

  
Ilustración 3: Archivo Cormdibn.cs. Notepad++

5.- Debajo del código insertado de la lectura del archivo para asignarle un valor a la bandera, implementaremos una condición, la cual quedaría de la siguiente manera:

  
Ilustración 4: Archivo Cormdibn.cs. Notepad++

Si la bandera es igual a 1, implementamos el proceso de CyberArk y si la bandera es diferente de 1, insertaremos el código respaldado, el cual contiene el proceso normal de SAPUF.

6.- Si la bandera es igual a 1, incluir el siguiente código:

// Obtener contraseña con CyberArk

string sPassword = "";

string sCertificateName = "";

string sCertificatePwd = "";

try

{

//ServicePointManager.ServerCertificateValidationCallback = new RemoteCertificateValidationCallback(CertificateValidationCallBack); //This was original

System.Net.ServicePointManager.ServerCertificateValidationCallback = new System.Net.Security.RemoteCertificateValidationCallback(delegate { return true; });

System.Net.ServicePointManager.SecurityProtocol = (System.Net.SecurityProtocolType)3072; // required for establishing TLS1.2 protocol connection

BasicHttpBinding binding = new BasicHttpBinding(BasicHttpSecurityMode.Transport);

binding.Security.Transport.ClientCredentialType = HttpClientCredentialType.Certificate;

var ea = new EndpointAddress("https://cyberarkccpcbdev.wlb3.nam.nsroot.net/AIMWebservice/v1.1/AIM.asmx");

var client = new ServiceReference2.AIMServiceSoapClient(binding, ea);

//sCertificateName = "C:\\TFS\_WORKSPACE\_MAE\_SAF\_FAL\\App.SafariGUI\\Ver222\_Cyberark\\SAFARI\\bin\\Debug\\Certificatekey.p12";

sCertificateName = "C:\\Users\\AP09377\\Desktop\\ReallyNewPKCS.p12";

sCertificatePwd = "xxxxxxx";

client.ClientCredentials.ClientCertificate.Certificate = new System.Security.Cryptography.X509Certificates.X509Certificate2(sCertificateName, sCertificatePwd);

ServiceReference2.PasswordRequest pr = new ServiceReference2.PasswordRequest();

pr.AppID = "CCSS\_D\_AIM\_145791\_User";

pr.Safe = "CCSS\_D\_AIM\_145791\_Safe";

pr.Folder = "Root";

pr.Object = "SybasASEISO\_1-vm-2d9b-6102-C430\_DBO\_CA";

pr.Reason = "Test";

ServiceReference2.PasswordResponse Prs1 = new ServiceReference2.PasswordResponse();

Prs1 = client.GetPassword(pr);

sPassword = Prs1.Content;

CORDBLIB.gsPassword = sPassword;

//MessageBox.Show("DB Details retrived Successfully from cyberark");

}

catch (Exception ex)

{

sPassword = "CYBERARK:ERROR";

MessageBox.Show(sPassword);

Environment.Exit(0);

}

El código se vería de la siguiente manera:

  
Ilustración 5: Archivo Cormdibn.cs. Notepad++

7.- Si la bandera es diferente de 1, incluir el código del respaldo realizado en un archivo externo, contemplando las líneas de código que conservamos en las primeras líneas (1628 y 1629).

//Obtener la contraseña por SAPUF

string UsuarioSAPUF = mdlParametros.valorRegistro(2, "SOFTWARE\\Banamex\\C430\_001\\TarjetaCorporativa\\SAPUF", "UsuarioSAPUF").Trim();

string ServidorSAPUF = mdlParametros.valorRegistro(2, "SOFTWARE\\Banamex\\C430\_001\\TarjetaCorporativa\\SAPUF", "ServidorSAPUF").Trim();

int OperacionSAPUF = StringsHelper.IntValue(mdlParametros.valorRegistro(2, "SOFTWARE\\Banamex\\C430\_001\\TarjetaCorporativa\\SAPUF", "OperacionSAPUF").Trim());

string MaqDestinoSAPUF = mdlParametros.valorRegistro(2, "SOFTWARE\\Banamex\\C430\_001\\TarjetaCorporativa\\SAPUF", "MaqDestinoSAPUF").Trim();

string TipoDestSAPUF = mdlParametros.valorRegistro(2, "SOFTWARE\\Banamex\\C430\_001\\TarjetaCorporativa\\SAPUF", "TipoDestSAPUF").Trim();

string AplDestinoSAPUF = mdlParametros.valorRegistro(2, "SOFTWARE\\Banamex\\C430\_001\\TarjetaCorporativa\\SAPUF", "AplDestinoSAPUF").Trim();

string AplOrigenSAPUF = mdlParametros.valorRegistro(2, "SOFTWARE\\Banamex\\C430\_001\\TarjetaCorporativa\\SAPUF", "AplOrigenSAPUF").Trim();

int ModoRefrSAPUF = Int32.Parse(mdlParametros.valorRegistro(2, "SOFTWARE\\Banamex\\C430\_001\\TarjetaCorporativa\\SAPUF", "ModoRefrSAPUF")); //mod refresh inicia con cero

int TimerSAPUF = Int32.Parse(mdlParametros.valorRegistro(2, "SOFTWARE\\Banamex\\C430\_001\\TarjetaCorporativa\\SAPUF", "TimerSAPUF"));

mdlParametros.sgPathLogs = mdlParametros.valorRegistro(2, "SOFTWARE\\Banamex\\C430\_001\\TarjetaCorporativa\\Rutas", "Logs").Trim();

string cve\_operacion = OperacionSAPUF.ToString(); // "11"

//AIS-1182 NGONZALEZ

//int resultado = API.Encryption.Solicitar\_Password(cve\_operacion, MaqDestinoSAPUF, UsuDestinoSAPUF, TipoDestSAPUF, AplDestinoSAPUF, AplOrigenSAPUF, ModoRefrSAPUF.ToString(), TimerSAPUF.ToString());

int resultado = SapufEnlace.Solicitar\_Password(cve\_operacion, MaqDestinoSAPUF, UsuDestinoSAPUF, TipoDestSAPUF, AplDestinoSAPUF, AplOrigenSAPUF, ModoRefrSAPUF.ToString(), TimerSAPUF.ToString());

if (resultado == 0)

{

//AIS-1182 NGONZALEZ

//tam = API.Encryption.Recuperar\_msg(s, sError, sSubError);

tam = SapufEnlace.Recuperar\_msg(out s, out sError, out sSubError);

CORDBLIB.gsPassword = s.ToString();

if (sError.ToString() == "0000" && sSubError.ToString() == "0000")

{

//Tiempo del ConComDrive

//mdlParametros.igTimeComDrive = Int32.Parse(mdlParametros.fncReadIniS("Corporativa", "Tiempo"));

mdlParametros.igTimeComDrive = Int32.Parse(mdlParametros.valorRegistro(2, "SOFTWARE\\Banamex\\C430\_001\\TarjetaCorporativa\\Comdrv", "comdrvTime"));

//Realiza la conexión al servidor configurado

if (CORDBLIB.Conexion\_Servidor() == VBSQL.FAIL)

{

ModoRefrSAPUF = 1; //primera sol en modo refresco

//AIS-1182 NGONZALEZ

//resultado = API.Encryption.Solicitar\_Password(cve\_operacion, MaqDestinoSAPUF, UsuDestinoSAPUF, TipoDestSAPUF, AplDestinoSAPUF, AplOrigenSAPUF, ModoRefrSAPUF.ToString(), TimerSAPUF.ToString());

resultado = SapufEnlace.Solicitar\_Password(cve\_operacion, MaqDestinoSAPUF,

UsuDestinoSAPUF, TipoDestSAPUF, AplDestinoSAPUF, AplOrigenSAPUF, ModoRefrSAPUF.ToString(), TimerSAPUF.ToString());

if (resultado == 0)

{

//AIS-1182 NGONZALEZ

//tam = API.Encryption.Recuperar\_msg(s, sError, sSubError);

tam = SapufEnlace.Recuperar\_msg(out s, out sError, out sSubError);

CORDBLIB.gsPassword = s.ToString();

if (sError.ToString() == "0000" && sSubError.ToString() == "0000")

{

MessageBox.Show("eror " + sError.ToString() + "suberror" + sSubError.ToString(), Application.ProductName);

//Tiempo del ConComDrive

//mdlParametros.igTimeComDrive = Int32.Parse(mdlParametros.fncReadIniS("Corporativa", "Tiempo"));

mdlParametros.igTimeComDrive = Int32.Parse(mdlParametros.valorRegistro(2, "SOFTWARE\\Banamex\\C430\_001\\TarjetaCorporativa\\Comdrv", "comdrvTime"));

if (CORDBLIB.Conexion\_Servidor() == VBSQL.FAIL)

{

ModoRefrSAPUF = 2; //segunda sol en modo refresco

//AIS-1182 NGONZALEZ

//resultado = API.Encryption.Solicitar\_Password(cve\_operacion, MaqDestinoSAPUF, UsuDestinoSAPUF, TipoDestSAPUF, AplDestinoSAPUF, AplOrigenSAPUF, ModoRefrSAPUF.ToString(), TimerSAPUF.ToString());

resultado = SapufEnlace.Solicitar\_Password(cve\_operacion, MaqDestinoSAPUF,

UsuDestinoSAPUF, TipoDestSAPUF, AplDestinoSAPUF, AplOrigenSAPUF, ModoRefrSAPUF.ToString(), TimerSAPUF.ToString());

if (resultado == 0)

{

//AIS-1182 NGONZALEZ

//tam = API.Encryption.Recuperar\_msg(s, sError, sSubError);

tam = SapufEnlace.Recuperar\_msg(out s, out sError, out sSubError);

CORDBLIB.gsPassword = s.ToString();

if (sError.ToString() == "0000" && sSubError.ToString() == "0000")

{

//Tiempo del ConComDrive

//mdlParametros.igTimeComDrive = Int32.Parse(mdlParametros.fncReadIniS("Corporativa", "Tiempo"));

mdlParametros.igTimeComDrive = Int32.Parse(mdlParametros.valorRegistro(2, "SOFTWARE\\Banamex\\C430\_001\\TarjetaCorporativa\\Comdrv", "comdrvTime"));

if (CORDBLIB.Conexion\_Servidor() == VBSQL.FAIL)

{

CORDBLIB.gsMsg = "DESINCRONIZACIÓN CON SAPUF";

//UPGRADE\_WARNING: (6021) Casting 'int' to Enum may cause different behaviour.

//UPGRADE\_WARNING: (1046) MsgBox Parameter 'context' is not supported, and was removed.

//UPGRADE\_WARNING: (1046) MsgBox Parameter 'helpfile' is not supported, and was removed.

Interaction.MsgBox(CORDBLIB.gsMsg, (MsgBoxStyle)(((int)CORVB.MB\_ICONSTOP) + ((int)CORVB.MB\_OK)), CORSTR.STR\_APP\_TIT);

this.Cursor = Cursors.Default;

ModoRefrSAPUF = 3; //se notifica a SAPUF que existe desincronización

Tipo\_problema = "01";

Codigo\_error = "01";

cve\_operacion = Tipo\_problema + Codigo\_error;

//AIS-1182 NGONZALEZ

//resultado = API.Encryption.Solicitar\_Password(cve\_operacion, MaqDestinoSAPUF, UsuDestinoSAPUF, TipoDestSAPUF, AplDestinoSAPUF, AplOrigenSAPUF, ModoRefrSAPUF.ToString(), TimerSAPUF.ToString());

resultado = SapufEnlace.Solicitar\_Password(cve\_operacion, MaqDestinoSAPUF,

UsuDestinoSAPUF, TipoDestSAPUF, AplDestinoSAPUF, AplOrigenSAPUF, ModoRefrSAPUF.ToString(), TimerSAPUF.ToString());

if (resultado != 0)

{

CORDBLIB.gsMsg = "NO SE PUDO REALIZAR EL ANUNCIO DE DESINCRONIZACIÓN CON SAPUF";

//UPGRADE\_WARNING: (6021) Casting 'int' to Enum may cause different behaviour.

//UPGRADE\_WARNING: (1046) MsgBox Parameter 'context' is not supported, and was removed.

//UPGRADE\_WARNING: (1046) MsgBox Parameter 'helpfile' is not supported, and was removed.

Interaction.MsgBox(CORDBLIB.gsMsg, (MsgBoxStyle)(((int)CORVB.MB\_ICONSTOP) + ((int)CORVB.MB\_OK)), CORSTR.STR\_APP\_TIT);

this.Cursor = Cursors.Default;

Environment.Exit(0);

}

Environment.Exit(0);

}

}

else

{

MessageBox.Show("TERCER ERROR SAPUF " + " Error " + sError.ToString() + " SubError " + sSubError.ToString(), Application.ProductName);

Environment.Exit(0);

}

}

else

{

MessageBox.Show("Error de SAPUF con segundo regreso (Atributo=2)" + Conversion.Str(resultado), "Error en Operación", MessageBoxButtons.OK, MessageBoxIcon.Error); //FSWB NR 20070315

this.Cursor = Cursors.Default;

Environment.Exit(0);

} //fin del resultado=0 del pwd solicitado con atributo =2

} //fin de la segunda conexión solicitada con Atributo=1

}

else

{

MessageBox.Show("SEGUNDO ERROR SAPUF " + " Error " + sError.ToString() + " SubError " + sSubError.ToString(), Application.ProductName); //FSWB NR 20070315

Environment.Exit(0);

}

}

else

{

MessageBox.Show("Error de SAPUF con primer refresco (Atributo=1) " + Conversion.Str(resultado), "Error en Operación", MessageBoxButtons.OK, MessageBoxIcon.Error); //FSWB NR 20070315

this.Cursor = Cursors.Default;

Environment.Exit(0);

} //fin del if resultado =0 del envío con 1

} //fin de la primera conexión

}

else

{

MessageBox.Show("PRIMER ERROR SAPUF " + " Error " + sError.ToString() + " SubError " + sSubError.ToString(), Application.ProductName); //FSWB NR 20070315

Environment.Exit(0);

}

}

else

{

MessageBox.Show("Error de SAPUF " + Conversion.Str(resultado), "Error en Operación", MessageBoxButtons.OK, MessageBoxIcon.Error); //FSWB NR 20070315

this.Cursor = Cursors.Default;

Environment.Exit(0);

}

8.- Una vez integrado se agrega la librería de **ServiceModel** para el consumo del Web Service.

9.- Se agrega como Referencia web el archivo .wsdl para agregar la librería para las variables que se utilizan en el Web Service.