

Proyecto: Integración de Timbrado usando .NET

Documento: Manual de implementación

1. TimbraCFDI

```
//Ubicación del servicio de pruebas
//http://www.timbracfdipruebas.mx/ServicioIntegracionPruebas/Timbrado.asmx

//Parámetro usuarioIntegrador
string usuarioIntegrador = "mvpNUXmQfK8=";
//Parámetro comprobante a timbrar en base64
string comprobanteBase64 =
Convert.ToBase64String(System.IO.File.ReadAllBytes(@"C:\comprobanteSinTimbrar.xml"));
//Parámetro Id del comprobante, deberá ser un identificador único, para efecto de pruebas se utilizará un
número aleatorio
Random miRandom = new Random();
string idComprobante = miRandom.Next(0, 999999).ToString();

//Llamamos al web method TimbraCFDI
ServicioIntegracionReference.TimbradoSoapClient soapClient = new
ServicioIntegracionReference.TimbradoSoapClient();
ServicioIntegracionReference.ArrayOfAnyType resultados = soapClient.TimbraCFDI(usuarioIntegrador,
comprobanteBase64, idComprobante);

//Obtenemos datos del timbrado

//Tipo de excepcion
string tipoExcepcion = resultados[0].ToString();
//Numero de error
string numeroError = resultados[1].ToString();
//Descripcion del resultado
string descripcion = resultados[2].ToString();
//Xml timbrado
string xml = resultados[3].ToString();
//Codigo bidimensional
byte[] codigoBidimensional = (byte[])(resultados[4]);
//Cadena timbre
string cadenaTimbre = resultados[5].ToString();

//Verificamos que el resultado haya sido exitoso
if (numeroError == "0")
{
    MessageBox.Show("El comprobante fue timbrado exitosamente");
}
else
{
    //Ocurrió un error en el timbrado
    MessageBox.Show(descripcion);
}
```

2. ObtieneCFDI

```
//Ubicación del servicio de pruebas
//http://www.timbracfdipruebas.mx/ServicioIntegracionPruebas/Timbrado.asmx

//Parámetro usuarioIntegrador
string usuarioIntegrador = "mvpNUXmQfK8=";
//Parámetro rfcEmisor
string rfcEmisor = "AAA010101AAA";
//Parámetro folio fiscal (Folio UUID)
string folioFiscal = txtFolioFiscal.Text.Trim().ToUpper();

//Llamamos al web method ObtieneCFDI
ServicioIntegracionReference.TimbradoSoapClient soapClient = new
ServicioIntegracionReference.TimbradoSoapClient();
ServicioIntegracionReference.ArrayOfAnyType resultados = soapClient.ObtieneCFDI(usuarioIntegrador,
rfcEmisor, folioFiscal);

//Obtenemos datos del resultado

//Tipo de excepción
string tipoExcepcion = resultados[0].ToString();
//Numero de error
string numeroError = resultados[1].ToString();
//Descripcion del resultado
string descripcion = resultados[2].ToString();
//Xml
string xml = resultados[3].ToString();
//Codigo bidimensional no aplica en obtiene cfdi
byte[] codigoBidimensional = (byte[])(resultados[4]);
//Cadena timbre no aplica en obtiene cfdi
string cadenaTimbre = resultados[5].ToString();

//Verificamos que el resultado haya sido exitoso
if (numeroError == "0")
{
    MessageBox.Show("El comprobante fue consultado exitosamente");
}
else
{
    //Ocurrió un error en la obtención del CFDI
    MessageBox.Show(descripcion);
}
```

3. CancelaCFDI

```
//Ubicación del servicio de pruebas
//http://www.timbracfdipruebas.mx/ServicioIntegracionPruebas/Timbrado.asmx

//Parámetro usuarioIntegrador
string usuarioIntegrador = "mvpNUXmQfK8=";
//Parámetro rfcEmisor
string rfcEmisor = "AAA010101AAA";
//Parámetro folio fiscal (Folio UUID)
string folioFiscal = txtFolioFiscalCancelacion.Text.Trim().ToUpper();

//Llamamos al web method CancelaCFDI
ServicioIntegracionReference.TimbradoSoapClient soapClient = new
ServicioIntegracionReference.TimbradoSoapClient();
ServicioIntegracionReference.ArrayOfAnyType resultados = soapClient.CancelaCFDI(usuarioIntegrador,
rfcEmisor, folioFiscal);

//Obtenemos datos de la cancelación

//Tipo de excepcion
string tipoExcepcion = resultados[0].ToString();
//Numero de error
string numeroError = resultados[1].ToString();
//Descripcion del resultado
string descripcion = resultados[2].ToString();
//Xml
string xml = resultados[3].ToString();
//Codigo bidimensional no aplica en obtiene cfdi
byte[] codigoBidimensional = (byte[])(resultados[4]);
//Cadena timbre no aplica en obtiene cfdi
string cadenaTimbre = resultados[5].ToString();

//Verificamos que el resultado haya sido exitoso
if (numeroError == "0")
{
    MessageBox.Show("El comprobante fue cancelado exitosamente");
}
else
{
    //Ocurrió un error en la cancelación del CFDI
    MessageBox.Show(descripcion);
}
```

4. RegistraEmisor

```
//Ubicación del servicio de pruebas
//http://www.timbracfdipruebas.mx/ServicioIntegracionPruebas/Timbrado.asmx

//Parámetro usuarioIntegrador
string usuarioIntegrador = "mvpNUXmQfK8=";
//Parámetro rfcEmisor
string rfcEmisor = "AAA010101AAA";
//Parámetro base64Cer
string archivoCerBase64
=Convert.ToBase64String(System.IO.File.ReadAllBytes(@"C:\aaa010101aaa__csd_01.cer"));
//Parámetro base64Key
string archivoKeyBase64
=Convert.ToBase64String(System.IO.File.ReadAllBytes(@"C:\aaa010101aaa__csd_01.key"));
//Parámetro contraseña sello digital
string contrasena = "12345678a";

//Llamamos al web method Registra Emisor
ServicioIntegracionReference.TimbradoSoapClient soapClient = new
ServicioIntegracionReference.TimbradoSoapClient();
ServicioIntegracionReference.ArrayOfAnyType resultados = soapClient.RegistraEmisor(usuarioIntegrador,
rfcEmisor, archivoCerBase64, archivoKeyBase64, contrasena);

//Obtenemos datos del registro emisor

//Tipo de excepcion
string tipoExcepcion = resultados[0].ToString();
//Numero de error
string numeroError = resultados[1].ToString();
//Descripcion del resultado
string descripcion = resultados[2].ToString();
//Xml no aplica en registro emisor
string xml = resultados[3].ToString();
//Codigo bidimensional no aplica en registro emisor
Bitmap codigoBidimensional = resultados[4] as Bitmap;
//Cadena timbre no aplica en registro emisor
string cadenaTimbre = resultados[5].ToString();

//Verificamos que el resultado haya sido exitoso
if (numeroError == "0")
{
    MessageBox.Show("El emisor fue registrado correctamente");
}
else
{
    //Ocurrió un error en el registro del emisor
    MessageBox.Show(descripcion);
}
```