

Manual Técnico de SIFEI Signature

Mayo 2023

CFDI v4.0

Control de Versiones

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre | No. Versión | Modificaciones | Fecha |
| Daniel Jesús Hernández Francisco | 1.0 | Generación del documento | 19/oct/2017 |
| Daniel Jesús Hernández Francisco | 1.1 | Modificacion del documento | 25/nov/2017 |
| Daniel Jesús Hernández Francisco | 1.2 | Modificacion del documento | 09/abril/2019 |

Control de Revisiones

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre | No. Versión | Modificaciones | Fecha |
| Mary Quiahua Iztacua | 1.0 | Revisión técnica del documento. | 19/oct/2017 |
|  |  |  |  |
|  |  |  |  |

Control de Autorizaciones

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre | No. Versión | Modificaciones | Fecha |
| Asley Alberto Cristales Pavón | 1.0 | Revisión, correcciones menores técnicas y de redacción y estilo, y autorización de la publicación. | 30/oct/2017 |
|  |  |  |  |
|  |  |  |  |

Contenido

[Listado de Imágenes x](#_Toc140772004)

[1. Generalidades 1](#_Toc140772005)

[1.1 Introducción 1](#_Toc140772006)

[1.2 Objetivo 1](#_Toc140772007)

[1.3 Alcance 1](#_Toc140772008)

[1.4 Requerimientos Mínimos 1](#_Toc140772009)

[1.5 Términos y Definiciones 1](#_Toc140772010)

[1.6 Términos y definiciones técnicas 2](#_Toc140772011)

[1.7 Referencias 2](#_Toc140772012)

[1.8 Recursos 2](#_Toc140772013)

[1.9 Versiones de CFDI que se timbran 2](#_Toc140772014)

[1.9.1 Complementos para CFDI 4.0 2](#_Toc140772015)

[1.9.2 Complementos para Comprobante de Retención e Información de Pagos 3](#_Toc140772016)

[1.9.3 Métodos de Timbrado CFDI 4.0 3](#_Toc140772017)

[1.9.4 Métodos de Timbrado Comprobante de Retención e Información de Pagos 3](#_Toc140772018)

[1.9.5 Metodos de proceso de cancelación CFDI 4.0 3](#_Toc140772019)

[2. Convenciones de documentación 3](#_Toc140772020)

[2.1 Atributos opcionales 4](#_Toc140772021)

[3. Primeros pasos (*getting started*) 5](#_Toc140772022)

[3.1 Agregar recursos 6](#_Toc140772023)

[3.1.1 Copiar la carpeta SAT y las librerías 6](#_Toc140772024)

[3.1.2 Incluir la carpeta SAT y dll en el proyecto 6](#_Toc140772025)

[3.1.3 Configurar los archivos para ser copiados en el directorio de salida 7](#_Toc140772026)

[3.2 Agregar la biblioteca al proyecto 8](#_Toc140772027)

[3.3 Agregar información de los servicios web (configurar WS) 10](#_Toc140772028)

[3.4 Crear base de datos 13](#_Toc140772029)

[3.4.1 Comandos usados para crear la base de datos 13](#_Toc140772030)

[3.4.2 Cambiar max\_allowed\_packet (Opcional) 14](#_Toc140772031)

[3.4.3 Ejecutar el *script* 16](#_Toc140772032)

[3.4.4 Crear la base de datos en Mysql 17](#_Toc140772033)

[3.5 Mysql y MariaDb en Servidores Linux, Windows y Mac OS X 21](#_Toc140772034)

[3.5.1 *Case sensitive* (sensibilidad a minúsculas y mayúsculas) 21](#_Toc140772035)

[3.5.2 Hacer insensible Mysql y MariaDb en sistemas basados en Unix. 23](#_Toc140772036)

[4. Ejemplos CFDI 26](#_Toc140772037)

[4.1 Instanciar un Objeto tipo CFDI40 26](#_Toc140772038)

[4.2 Establecer información del nodo InformacionGlobal 27](#_Toc140772039)

[4.3 Agregar CfdiRelacionados 27](#_Toc140772040)

[4.4 Establecer información del Emisor 27](#_Toc140772041)

[4.5 Establecer información del Receptor 28](#_Toc140772042)

[4.6 Agregar un Concepto 28](#_Toc140772043)

[4.7 Agregar Información aduanera a un Concepto 29](#_Toc140772044)

[4.8 Agregar un impuesto traslado a un concepto 29](#_Toc140772045)

[4.9 Agregar un impuesto retenido a un concepto 29](#_Toc140772046)

[4.10 Establecer información de A Cuenta de Terceros 30](#_Toc140772047)

[4.11 Agregar Cuenta Predial al concepto 30](#_Toc140772048)

[4.12 Establecer información del total impuestos de comprobante 30](#_Toc140772049)

[4.13 Agregar un impuesto traslado a impuestos de comprobante 31](#_Toc140772050)

[4.14 Autorización para el uso de Sifei Signature 31](#_Toc140772051)

[4.15 Generar el XML 32](#_Toc140772052)

[4.16 Sellar el comprobante 34](#_Toc140772053)

[4.17 Timbrar el comprobante 35](#_Toc140772054)

[4.18 Recuperar timbre 36](#_Toc140772055)

[5. Crear Complementos 36](#_Toc140772056)

[5.1 Complemento Comercio Exterior 36](#_Toc140772057)

[5.1.1 Definir complemento Comercio Exterior 36](#_Toc140772058)

[5.1.2 Establecer Emisor de Comercio Exterior 37](#_Toc140772059)

[5.1.3 Definir Receptor de Comercio Exterior 37](#_Toc140772060)

[5.1.4 Agregar mercancía de Comercio Exterior 37](#_Toc140772061)

[5.1.5 Generar el Complemento Comercio Exterior 37](#_Toc140772062)

[5.2 Complemento Nómina 38](#_Toc140772063)

[5.2.1 Definir complemento Nómina 38](#_Toc140772064)

[5.2.2 Agregar un emisor a Nómina 38](#_Toc140772065)

[5.2.3 Agregar un receptor a Nómina 38](#_Toc140772066)

[5.2.4 Establecer que Nómina tiene percerpciones 38](#_Toc140772067)

[5.2.5 Agregar una percepción a percepciones de Nómina 39](#_Toc140772068)

[5.2.6 Agregar una Separación indemnización a percepciones de Nómina 39](#_Toc140772069)

[5.2.7 Establer deducciones a Nómina 39](#_Toc140772070)

[5.2.8 Agregar una deducción a deducciones de Nómina 39](#_Toc140772071)

[5.2.9 Generar el Complemento Nómina 39](#_Toc140772072)

[5.3 Complemento Detallista 39](#_Toc140772073)

[5.3.1 Definir complemento Detallista 39](#_Toc140772074)

[5.3.2 Establecer el requestForPaymentIdentification a Detallista 40](#_Toc140772075)

[5.3.3 Agregar un specialInstruction a Detallista 40](#_Toc140772076)

[5.3.4 Establecer el orderIdentification a Detallista 40](#_Toc140772077)

[5.3.5 Agregar un referenceIdentification al AdditionalInformation de Detallista 40](#_Toc140772078)

[5.3.6 Establecer un DeliveryNote a Detallista 40](#_Toc140772079)

[5.3.7 Definir el *Buyer* (comprador) de Detallista 40](#_Toc140772080)

[5.3.8 Definir el *Seller* (vendedor) a Detallista 40](#_Toc140772081)

[5.3.9 Establer el shipTo (lugar de envío) de Detallista 41](#_Toc140772082)

[5.3.10 Establecer el InvoiceCreator a Detallista 41](#_Toc140772083)

[5.3.11 Agregar un Customs a Detallista 41](#_Toc140772084)

[5.3.12 Agregar un currency a Detallista 41](#_Toc140772085)

[5.3.13 Establecer el paymentTerms a Detallista 41](#_Toc140772086)

[5.3.14 Establecer un allowanceCharge a Detallista para definir cargos o descuentos 41](#_Toc140772087)

[5.3.15 Definir el totalAmount a Detallista 42](#_Toc140772088)

[5.3.16 Definir el TotalAllowanceCharge a Detallista para definir el monto total de cargos o descuentos 42](#_Toc140772089)

[5.3.17 Generar el complemento Detallista 42](#_Toc140772090)

[5.4 Complemento Recepción de Pagos 2.0 42](#_Toc140772091)

[5.4.1 Definir el nodo Totales 42](#_Toc140772092)

[5.4.2 Agregar un Pago 42](#_Toc140772093)

[5.4.3 Agregar un DoctoRelacionado al último Pago agregado 43](#_Toc140772094)

[5.4.4 Agregar un Impuesto TrasladoDR / RetencionDR al último DoctoRelacionado agregado 43](#_Toc140772095)

[5.4.5 Agregar un Impuesto TrasladoP al ultimo Pago agregado. 43](#_Toc140772096)

[5.4.6 Agregar un Impuesto RetencionP al ultimo Pago agregado. 44](#_Toc140772097)

[5.4.7 Generar el Complemento Pago 44](#_Toc140772098)

[5.5 Definir y generar el complemento IEDU (a un concepto) 44](#_Toc140772099)

[5.6 Definir y generar el complemento Donatarias 1.1 44](#_Toc140772100)

[5.7 Complemento Carta Porte 2.0 45](#_Toc140772101)

[5.7.1 Definir el complemento Carta Porte 45](#_Toc140772102)

[5.7.2 Agregar una Ubicación 45](#_Toc140772103)

[5.7.3 Establecer el Domicilio para una ubicación 45](#_Toc140772104)

[5.7.4 Establecer información de Mercancias 45](#_Toc140772105)

[5.7.5 Agregar una Mercancía 46](#_Toc140772106)

[5.7.6 Agregar CantidadTransporta a la última Mercancía 46](#_Toc140772107)

[5.7.7 Establecer elemento Autotransporte para Mercancias 46](#_Toc140772108)

[5.7.8 Establecer información de Seguros para el Autotransporte 46](#_Toc140772109)

[5.7.9 Agregar un Tipo de Figura para FiguraTransporte 47](#_Toc140772110)

[5.7.10 Generar el Complemento Carta Porte 2.0 47](#_Toc140772111)

[5.8 Complemento Impuestos Locales 47](#_Toc140772112)

[5.8.1 Definir el complemento Impuestos Locales 47](#_Toc140772113)

[5.8.2 Agregar Impuesto Local Trasladado al complemento 47](#_Toc140772114)

[5.8.3 Generar el Complemento Impuestos Locales 48](#_Toc140772115)

[6. Proceso de Cancelación 48](#_Toc140772116)

[6.1 Cancelar un CFDI 2018 (actualización) 48](#_Toc140772117)

[6.2 Cancelar CFDIs Nuevo Proceso 49](#_Toc140772118)

[6.3 Consulta CFDI relacionados 50](#_Toc140772119)

[6.4 Consulta CFDI Estatus 51](#_Toc140772120)

[6.5 Consulta peticiones de Cancelación pendientes (que requieren autorización). 52](#_Toc140772121)

[6.6 Procesar Respuesta de Peticion de Cancelacion (Aceptacion o Rechazo de cancelación) 53](#_Toc140772122)

[7. Comprobante de Retención de Información de Pagos 54](#_Toc140772123)

[7.1 Instanciar un objeto de Comprobante de Retención e Información de Pagos 54](#_Toc140772124)

[7.2 Definir la retención 54](#_Toc140772125)

[7.3 Definir CfdiRetenRelacionados 55](#_Toc140772126)

[7.4 Definir el Emisor 55](#_Toc140772127)

[7.5 Definir Receptor 55](#_Toc140772128)

[7.6 Definir el Periodo 56](#_Toc140772129)

[7.7 Definir Totales 56](#_Toc140772130)

[7.8 Agregar Impuesto Retenido 56](#_Toc140772131)

[7.9 Autorización para el uso de Sifei Signature 57](#_Toc140772132)

[7.10 Generar el XML 57](#_Toc140772133)

[7.11 Sellar el Comprobante de Retención e Información de Pagos 58](#_Toc140772134)

[7.12 Timbrar el Comprobante de Retenciones e Información de Pagos 59](#_Toc140772135)

[8. Complementos de Comprobante de Retenciones e Información de Pagos 60](#_Toc140772136)

[8.1 Complemento Servicios Plataformas Tecnológicas 60](#_Toc140772137)

[8.1.1 Definir Complemento Servicios Plataformas Tecnológicas 60](#_Toc140772138)

[8.1.2 Agregar DetallesDelServicio 60](#_Toc140772139)

[8.1.3 Generar Complemento Servicios Plataformas Tecnológicas 61](#_Toc140772140)

[9. Referencia 62](#_Toc140772141)

[9.1 *Namespaces* 62](#_Toc140772142)

[9.2 Clases principales 62](#_Toc140772143)

[9.3 *Namespace* SifeiSignature3\_3.Version3\_3 62](#_Toc140772144)

[9.3.1 Clase CFDI33 63](#_Toc140772145)

[9.3.2 Métodos 63](#_Toc140772146)

[9.3.3 Enumeraciones: 68](#_Toc140772147)

[9.4 *Namespace* SifeiSignature3\_3.Timbre11 68](#_Toc140772148)

[9.4.1 Clases 68](#_Toc140772149)

[9.4.2 Clase TimbreBE 68](#_Toc140772150)

[9.5 *Namespace* SifeiSignature3\_3.Complementos 68](#_Toc140772151)

[9.5.1 Clase Complemento 68](#_Toc140772152)

[9.6 *Namespace* SifeiSignature3\_3.Returns 88](#_Toc140772153)

[9.6.1 Clase ReturnXML 88](#_Toc140772154)

[9.6.2 Clase ReturnTimbre 89](#_Toc140772155)

[9.6.3 Clase ReturnXmlComplemento 89](#_Toc140772156)

[9.6.4 Clase ReturnCancelacion2018 90](#_Toc140772157)

[9.6.5 Clase ReturnDocumentosRelacionados 90](#_Toc140772158)

[9.6.6 Clase ReturnQrAcuse 91](#_Toc140772159)

[9.6.7 Clase ReturnPeticionesPendientes 91](#_Toc140772160)

[9.6.8 Clase ReturnAceptacionRechazoResult 92](#_Toc140772161)

[9.6.9 Clase DetalleFolios 92](#_Toc140772162)

[9.7 *Namespace* SifeiSignature3\_3.Complementos.Detallista131 93](#_Toc140772163)

[9.7.1 Enumeraciones de complemento Detallista versión 1.3.1 93](#_Toc140772164)

[9.8 *Namespace* SifeiSignature3\_3.Complementos.Iedu10 93](#_Toc140772165)

[9.8.1 Enumeraciones de complemento Instituciones Educativas versión 1.0 93](#_Toc140772166)

[9.8.2 Enumeraciones de complemento Nómina versión 1.2 93](#_Toc140772167)

[9.9 Referencia del *Namespace* SifeiSignature3\_3.Cancelacion 94](#_Toc140772168)

[9.9.1 Clase Acuse 94](#_Toc140772169)

[9.9.2 Clase AcuseDetalle 94](#_Toc140772170)

[9.9.3 Clase Cancelacion (deprecada, obsoleta) 94](#_Toc140772171)

[9.9.4 Clase Cancelacion 2018 95](#_Toc140772172)

[9.10 Referencia del *Namespace* SifeiSignature3\_3.Complementos.Cce11 96](#_Toc140772173)

[9.10.1 Clases 96](#_Toc140772174)

[9.10.2 Clase ComercioExterior 96](#_Toc140772175)

[9.10.3 Clase ComercioExteriorDestinatario 97](#_Toc140772176)

[9.10.4 Clase ComercioExteriorDestinatarioDomicilio 97](#_Toc140772177)

[9.10.5 Clase ComercioExteriorEmisor 98](#_Toc140772178)

[9.10.6 Clase ComercioExteriorEmisor 98](#_Toc140772179)

[9.10.7 Clase ComercioExteriorMercancia 98](#_Toc140772180)

[9.10.8 Clase ComercioExteriorMercanciaDescripcionesEspecificas 99](#_Toc140772181)

[9.10.9 Clase ComercioExteriorPropietario 99](#_Toc140772182)

[9.10.10 Clase ComercioExteriorReceptor 99](#_Toc140772183)

[9.10.11 Clase ComercioExteriorReceptorDomicilio 100](#_Toc140772184)

[9.11 Referencia del *Namespace* SifeiSignature3\_3.Complementos.ImpLocal10 100](#_Toc140772185)

[9.11.1 Clase ImpuestosLocales 100](#_Toc140772186)

[9.11.2 Clase ImpuestosLocalesRetencionesLocales 100](#_Toc140772187)

[9.11.3 Clase ImpuestosLocalesTrasladosLocales 101](#_Toc140772188)

[9.12 Referencia del *Namespace* SifeiSignature3\_3.Complementos.Nomina12 101](#_Toc140772189)

[9.12.1 Clases 101](#_Toc140772190)

[9.12.2 Enumeración NominaReceptorSindicalizado 101](#_Toc140772191)

[9.12.3 Clase Nomina 101](#_Toc140772192)

[9.12.4 Clase NominaDeducciones 102](#_Toc140772193)

[9.12.5 Clase NominaDeduccionesDeduccion 102](#_Toc140772194)

[9.12.6 Clase NominaEmisor 102](#_Toc140772195)

[9.12.7 Clase NominaEmisorEntidadSNCF 102](#_Toc140772196)

[9.12.8 Propiedades 102](#_Toc140772197)

[9.12.9 Clase NominaIncapacidad 103](#_Toc140772198)

[9.12.10 Clase NominaOtroPago 103](#_Toc140772199)

[9.12.11 Clase NominaOtroPagoCompensacionSaldosAFavor 103](#_Toc140772200)

[9.12.12 Clase NominaOtroPagoSubsidioAlEmpleo 103](#_Toc140772201)

[9.12.13 Clase NominaPercepciones 103](#_Toc140772202)

[9.12.14 Clase NominaPercepcionesJubilacionPensionRetiro 104](#_Toc140772203)

[9.12.15 Clase NominaPercepcionesPercepcion 104](#_Toc140772204)

[9.12.16 Clase NominaPercepcionesPercepcionAccionesOTitulos 104](#_Toc140772205)

[9.12.17 Clase NominaPercepcionesPercepcionHorasExtra 104](#_Toc140772206)

[9.12.18 Clase NominaPercepcionesSeparacionIndemnizacion 104](#_Toc140772207)

[9.12.19 Clase NominaReceptor 105](#_Toc140772208)

[9.12.20 Clase NominaReceptorSubContratacion 105](#_Toc140772209)

[9.13 Referencia del *Namespace* SifeiSignature3\_3.Complementos.Pagos10 105](#_Toc140772210)

[9.13.1 Clases 105](#_Toc140772211)

[9.13.2 Clase Pagos 105](#_Toc140772212)

[9.13.3 Clase PagosPago 106](#_Toc140772213)

[9.13.4 Clase PagosPagoDoctoRelacionado 106](#_Toc140772214)

[9.13.5 Clase PagosPagoImpuestos 106](#_Toc140772215)

[9.13.6 Clase PagosPagoImpuestosRetencion 107](#_Toc140772216)

[9.13.7 Clase PagosPagoImpuestosTraslado 107](#_Toc140772217)

[9.14 Referencia del *Namespace* SifeiSignature3\_3 .Complementos.cartaporte20 107](#_Toc140772218)

[9.14.1 Clases 107](#_Toc140772219)

[9.14.2 Clase CartaPorte 108](#_Toc140772220)

[9.14.3 Clase CartaPorteMercancias 108](#_Toc140772221)

[9.14.4 Clase CartaPorteMercanciasAutotransporte 108](#_Toc140772222)

[9.14.5 Clase CartaPorteMercanciasAutotransporteIdentificacionVehicular 109](#_Toc140772223)

[9.14.6 Clase CartaPorteMercanciasAutotransporteRemolque 109](#_Toc140772224)

[9.14.7 Clase CartaPorteMercanciasAutotransporteSeguros 109](#_Toc140772225)

[9.14.8 Clase CartaPorteMercanciasMercancia 109](#_Toc140772226)

[9.15 Referencia del *Namespace* SifeiSignature3\_3.srvSifeiCancelacion2018 110](#_Toc140772227)

[9.15.1 Clases 110](#_Toc140772228)

[9.15.2 Enumeraciones 110](#_Toc140772229)

[9.15.3 Clase folios 110](#_Toc140772230)

[9.16 Códigos de Respuesta de Cancelación 110](#_Toc140772231)

[9.17 Códigos de Error de la Matriz de validaciones para CFDI versión 4.0 111](#_Toc140772232)

[9.18 Codigos de Error del SAT para el metodo Consulta CFDI Relacionado. 111](#_Toc140772233)

[9.19 Codigos de error de SAT para el metodo obtener Peticiones de Cancelacion Pendientes. 111](#_Toc140772234)

[9.20 Codigos de error del SAT para el metodo Aceptacion/Rechazo de cancelacion (procesa Respuesta). 111](#_Toc140772235)

[9.21 Codigos de respuesta del servicio de consulta CFDI. 112](#_Toc140772236)

[10. Información de Contacto con SIFEI 113](#_Toc140772237)

# Listado de Imágenes

[Imagen 1 Creación de proyecto .Net Framework 3.5 5](#_Toc140772259)

[Imagen 2 Proyecto recién creado 5](#_Toc140772260)

[Imagen 3 Copiando los recursos al proyecto 6](#_Toc140772261)

[Imagen 4 Inclusión de carpeta SAT en el proyecto 7](#_Toc140772262)

[Imagen 5 Inclusión del resto de bibliotecas 7](#_Toc140772263)

[Imagen 6 Modificando la propiedad "Copiar al directorio de Salida" 8](#_Toc140772264)

[Imagen 7 Modificando la propiedad "Copiar al directorio de salida" a "Copiar siempre” 8](#_Toc140772265)

[Imagen 8 Agregar referencia al proyecto 9](#_Toc140772266)

[Imagen 9 Carpeta donde está SifeiSignature 9](#_Toc140772267)

[Imagen 10 Referencia de biblioteca agregada 10](#_Toc140772268)

[Imagen 11 Configuración de aplicación requerida para *SIFEI Signature* 11](#_Toc140772269)

[Imagen 12 Seleccionado Archivo de Configuración de Aplicación 12](#_Toc140772270)

[Imagen 13 Archivo de configuración limpio 12](#_Toc140772271)

[Imagen 14 Archivo de configuración con la configuración necesaria para *SIFEI Signature* 13](#_Toc140772272)

[Imagen 15 Ejecuta el *script* para la creación de la base de datos 13](#_Toc140772273)

[Imagen 16 Consulta para mostrar el valor de max\_allowed\_packet 13](#_Toc140772274)

[Imagen 17 Error 2006 MySQL server has gone away 14](#_Toc140772275)

[Imagen 18 Variable max\_allowed\_packet = 1MB: causará “Error 2006: MySQL server has gone away” 14](#_Toc140772276)

[Imagen 19 Comando para cambiar de forma global el límite que define max\_allowed\_packet a 32 MB 14](#_Toc140772277)

[Imagen 20 Cambiando variable Max\_allowed\_packet temporalmente (este efecto no cambia la sesión actual y dura solo mientras el servicio esté activo -hasta que el servicio sea reiniciado-) 15](#_Toc140772278)

[Imagen 21 Archivo de configuración del servidor de MariaDB, cambio de 1mb a 16 MB (se debe reiniciar el servicio) 15](#_Toc140772279)

[Imagen 22 Procedimiento para la ejecución del *script*, comenzando por el listado de la base de datos, la ejecución del *script* y finalizando por la comprobación de su creación 16](#_Toc140772280)

[Imagen 23 Error de sintaxis en Mysql al importar el script de la base de datos catalogossat 17](#_Toc140772281)

[Imagen 24 Vista del catalogossat que muestra PAGE\_CHECSUM y el numero de ocurrencias 17](#_Toc140772282)

[Imagen 25 Catalogossat sin PAGE\_CHECSUM 18](#_Toc140772283)

[Imagen 26 Captura que muestra de forma parcial las tablas creadas para la base de datos catalogossat en Mysql 5.5 y superior 19](#_Toc140772284)

[Imagen 27 Captura de remplazar todas las ocurrencias de "ENGINE=Aria" por "ENGINE=InnoDB" 20](#_Toc140772285)

[Imagen 28 Captura de la importacion del script catalogossat con motor InnoDb 21](#_Toc140772286)

[Imagen 29 Consulta de la variable lowe\_case\_table\_names donde su valor es 0, y por **tanto debe ser cambiado a 1** 22](#_Toc140772287)

[Imagen 30 Error al emitir un Cfdi cuando Mysql reside un servidor Unix y la variable de sistema ***lower\_case\_table\_names*** tiene valor 0. 23](#_Toc140772288)

[Imagen 31 Configurando el valor de lowe\_case\_table\_names= 1 24](#_Toc140772289)

[Imagen 32 Consulta de variable del sistema lower\_case\_table\_names , donde se muestra el cambio del valor de 0 a 1 (en un Centos) tras reiniciar Mysql Server, notar que mysql client pierde la conexión y se conecta de nuevo debido al reinicio de Mysql Server. 25](#_Toc140772290)

[Imagen 33 Atributos de CFDI agregados 26](#_Toc140772291)

[Imagen 34 Nodo InformacionGlobal agregado 27](#_Toc140772292)

[Imagen 35 Nodo CfdiRelacionados agregado 27](#_Toc140772293)

[Imagen 36 Nodo Emisor agregado 28](#_Toc140772294)

[Imagen 37 Nodo Receptor agregado 28](#_Toc140772295)

[Imagen 38 Nodo Concepto agregado 28](#_Toc140772296)

[Imagen 39 InformacionAduanera agregada a conceptos 29](#_Toc140772297)

[Imagen 40 Resultado de agregar un impuesto traslado al concepto 29](#_Toc140772298)

[Imagen 41 Resultado de agregar un impuesto traslado al concepto 30](#_Toc140772299)

[Imagen 42 Resultado de agregar un impuesto traslado al concepto 30](#_Toc140772300)

[Imagen 42 Resultado de agregar un impuesto traslado al concepto 30](#_Toc140772301)

[Imagen 41 Nodo Impuestos agregado a comprobante 31](#_Toc140772302)

[Imagen 42 Impuesto traslado agregado 31](#_Toc140772303)

[Imagen 43 Autenticandose para el uso de Sifei Signature 31](#_Toc140772304)

[Imagen 44 Error de autenticación 32](#_Toc140772305)

[Imagen 45 Comprobante Sellado 35](#_Toc140772306)

[Imagen 46 Comprobante timbrado 36](#_Toc140772307)

[Imagen 47 Atributos de Retenciones agregados 55](#_Toc140772308)

[Imagen 48 Nodo CfdiRetenRelacionados agregado 55](#_Toc140772309)

[Imagen 48 Nodo Emisor agregado 55](#_Toc140772310)

[Imagen 49 Nodo Receptor agregado 56](#_Toc140772311)

[Imagen 50 Nodo Periodo agregado 56](#_Toc140772312)

[Imagen 51 Nodo Totales agregado 56](#_Toc140772313)

[Imagen 52 Nodo ImpRetenidos agregado 56](#_Toc140772314)

[Imagen 53 Comprobante Sellado 59](#_Toc140772315)

[Imagen 54 Comprobante timbrado 59](#_Toc140772316)

# Generalidades

## Introducción

El presente documento es el *Manual Técnico* para el uso e integración de la biblioteca *SIFEI Signature* versión 4.0 a proyectos bajo plataforma .Net, además de ofrecer la referencia de la biblioteca y ejemplos para emitir, sellar, timbrar y cancelar CFDIs.

Solución Integral de Facturación Electrónica e Informática SIFEI S. A. de C. V. es un Proveedor Autorizado de Certificación de CFDI (PCCFDI antes PAC), con número de autorización 58355 y cumplimos con los requerimientos funcionales, servicios generales y niveles de servicio (SLA) exigidos por el SAT.

## Objetivo

Orientar a usuario de la biblioteca *SIFEI Signature* para emitir y certificar CFDIs válidos siguiendo las normas establecidas para tipo de documento fiscal.

## Alcance

* Emitir CFDIs en versión 4.0.
* La certificación se realiza con base al Anexo 20 y a la *Matriz de Validaciones* de los CFDI y complementos.
* Entrega respuestas sobre los códigos de error de la *Matriz de Validaciones* vigente.

## Requerimientos Mínimos

* Visual Studio 2012 o superior
* Tener instalado como mínimo .Net Framework 3.5
* Tener instalado una versión MariaDB (versión estable) o tener acceso remoto a este gestor
  + Tener las credenciales de un usuario con privilegios para crear base de datos
  + Contar con un cliente para el gestor (para la creación de la base de datos)
  + El servicio MariaDB debe tener la variable “max\_allowed\_packet” establecida para paquetes mayores a 8 MB.
  + Si se usara Mysql esta debe ser una version estable (5.1,5.5,5.6,5.7) y seguir los pasos de la sección [crear base de datos en Mysql](#_Crear_la_base)

## Términos y Definiciones

| Acrónimo | Definición |
| --- | --- |
| CFDI | Comprobante Fiscal Digital por Internet |
| NDA | *Non-Disclosure Agreement*¸ documento de Acuerdo de No Divulgación |
| PAC | Proveedor Autorizado de Certificación |
| PCCFDI | Proveedor de Certificación de Comprobantes Fiscales Digitales por Internet (antes PAC) |
| PCECFDI | Proveedor de Certificación de Expedición de Comprobantes Fiscales Digitales por Internet, que aplica a los contribuyentes que expiden CFDIs al ser Adquirientes de Bienes y Servicios, y Sector Primario (simplificado “Sector Primario”) |
| SAT | Servicio de Administración Tributaria |
| SIFEI | Solución Integral de Facturación Electrónica e Informática SIFEI S. A. de C. V. |
| SLA | *Service Level Agreement*, Acuerdo de Niveles de Servicios |

## Términos y definiciones técnicas

| Acrónimo | Definición |
| --- | --- |
| Biblioteca (Library) | Es una colección de recursos e implementaciones funcionales codificadas que son utilizadas por programas, en ellas se incluyen datos, configuraciones y código (generalmente pre compilado). Una biblioteca ofrece interfaces por las cuales un programa puede invocar funcionalidades. |
| DLL | Es una biblioteca (de enlace dinámico) que contiene código y datos que pueden utilizarse por varios programas al mismo tiempo. Las DLL son ligadas dinámicamente al programa que los usa durante su ejecución en vez de ser compiladas con el programa principal. |
| Microsoft Visual Studio (VS) | Es un entorno de desarrollo integrado que soporta múltiples lenguajes de programación, tales como C++, C#, Visual Basic .NET, F#, Java, Python, Ruby y PHP, al igual que entornos de desarrollo web, como ASP.NET MVC. |
| MariaDB Server | Es un potente servidor de base de datos derivado de MySQL y por tanto comparte muchas características de interfaces, comandos, sentencias y nomenclatura de programas. |

## Referencias

| Información | Fuente |
| --- | --- |
| Formato de factura (Anexo 20) | <http://omawww.sat.gob.mx/tramitesyservicios/Paginas/anexo_20.htm> |
| Manual técnico de Cancelacion Sifei. | https://www.sifei.com.mx/slides/slide/sifei-web-service-manual-tecnico-del-nuevo-servicio-de-cancelacion-99 |

## Recursos

Los recursos listados a continuación son necesarios para poder emitir y timbrar comprobantes.

| Archivo/Carpeta | Descripción |
| --- | --- |
| SifeiSignature3\_3.dll | Biblioteca principal la cual debe ser referenciada en el proyecto para su uso e integración. |
| SAT | Carpeta que contiene los XSLT y XSD del SAT, además de incluir openssl. |
| bd\_catálogossat.sql | Base de datos que contiene los catálogos del SAT. |
| MySQL.Data.dll | Biblioteca para el acceso a datos a MySQL y MariaDB. |
| zxing.dll | Biblioteca para el procesamiento de imágenes. |
| ICSharpCode.SharpZipLib | Biblioteca que permite varios formatos de compresión como ZIP y TAR entre otros. |

## Versiones de CFDI que se timbran

*SIFEI Signature 3.3* permite:

* Comprobante Fiscal Digital por Internet versión 4.0
* Comprobante de Retención e Información de Pagos

### Complementos para CFDI 4.0

* Comercio Exterior 1.1
* Nómina 1.2 Rev. C
* Detallista
* Pagos 2.0
* Impuesto Local
* IEDU
* Carta Porte 2.0
* Donatarias 1.1

### Complementos para Comprobante de Retención e Información de Pagos

* Servicios de Plataformas Tecnológicas

### Métodos de Timbrado CFDI 4.0

* Recupera Timbre
* Deposita Factura Global
* Recupera Timbre Factura Global

### Métodos de Timbrado Comprobante de Retención e Información de Pagos

* Recupera Timbre

### Metodos de proceso de cancelación CFDI 4.0

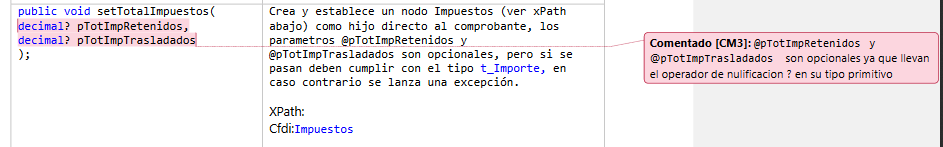
* Cancela 2018 Nuevo Proceso
* Consultar CFDI (Estado,Estado Cancelacion, etc)
* Consultar CFDI Relacionados
* Obtener Peticiones Pendientes
* Aceptacion/ Rechazo de Peticiones de cancelación.

# Convenciones de documentación

* Los tipos de clase, clase, y enumeraciones están en color azul claro.
* Los tipos de datos primitivos están en azul marino.
* Cuando en la descripción de un método se haga referencia a un parámetro se le antepondrá el símbolo ‘@’ al nombre del parámetro para su fácil reconocimiento.
* Los comentarios en las firmas de métodos y constructores van en color verde.
* Los métodos que alteren un nodo, en la sección *descripción* existirá el apartado *XPath* con la ruta del o los nodos modificados.
* Un nodo de tipo any, es un nodo de cualquier tipo, es decir, no hay restricciones como Addenda.
* A los parámetros que son opcionales (definidos así por el SAT), se les coloca el operador de nulificación ‘?’ para pasar null a tipos de datos primitivos como decimal. Ejemplo: “decimal? descuento“ donde descuento es un atributo opcional y el comprobante no tiene un descuento se debe pasar null.
* Cada ejemplo estará encerrado en un rectángulo.
* Para representar los comandos, se usará el nombre de la herramienta de línea de comandos seguido del simbolo mayor que “>”, por ejemplo, si se está en la línea de comandos de Windows: ”shell>”, si se está dentro de MariaDB “MariaDB>”, esto para dar un contexto explícito de donde debe ejecutarse el comando.

## Atributos opcionales

Los atributos opcionales están representados con el operador “?”, para el caso de las cadenas si se pasa como argumento null, o "" –cadena vacía- no se incluirá en el comprobante dicho atributo. Para el caso de los decimales, si se pasa null, tampoco se incluirá dicho atributo.



# Primeros pasos (*getting started*)

El primer paso es crear nuestro proyecto, el cual para este ejemplo será de tipo *WinForms*:

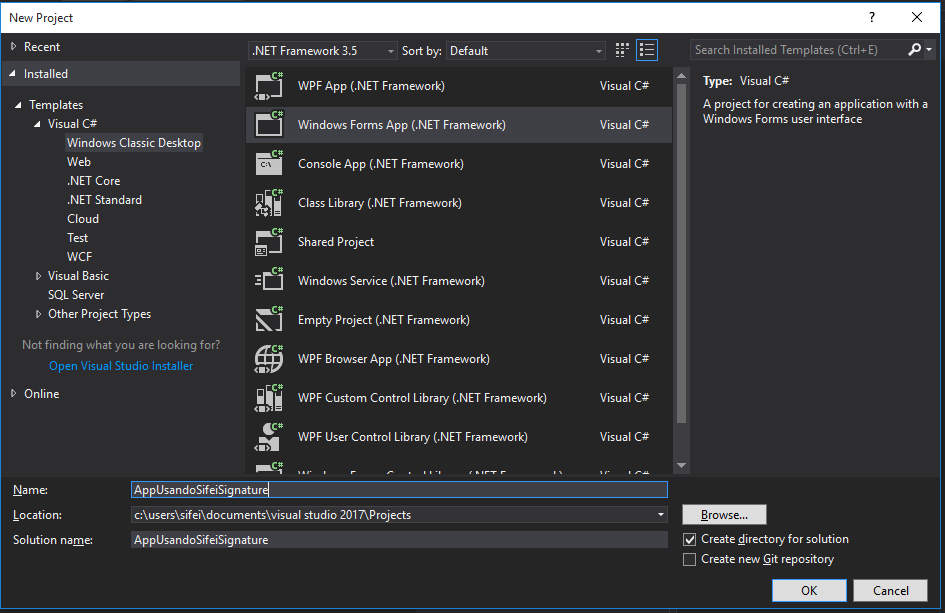


Imagen Creación de proyecto .Net Framework 3.5

Lo cual debe generar algo similar a la siguiente imagen:

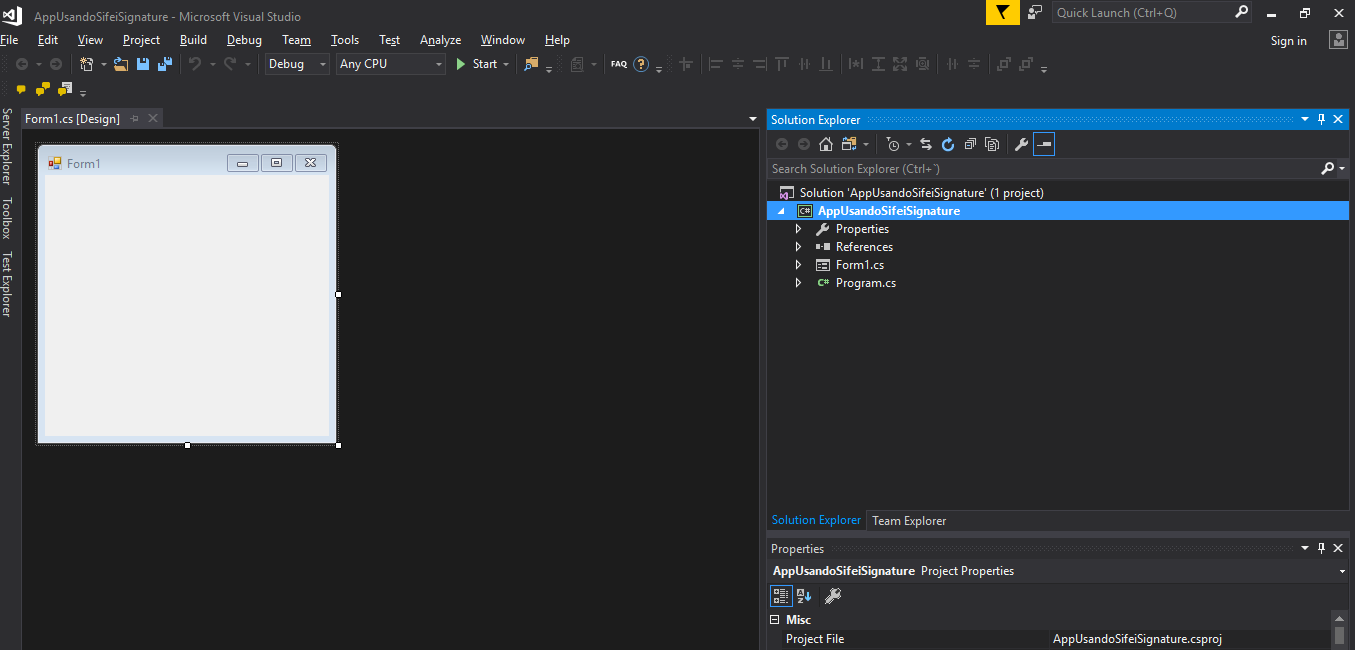


Imagen Proyecto recién creado

## Agregar recursos

Es necesario que los recursos estén en la misma carpeta que el programa (ejecutable del proyecto), para hacerlo debemos seguir los siguientes pasos:

1. Copiar la carpeta SAT y dll en la misma carpeta que el proyecto
2. Incluir la carpeta SAT y dll en el proyecto
3. Seleccionar todos los archivos y configurar para que sean copiados en el directorio de salida.

### Copiar la carpeta SAT y las librerías

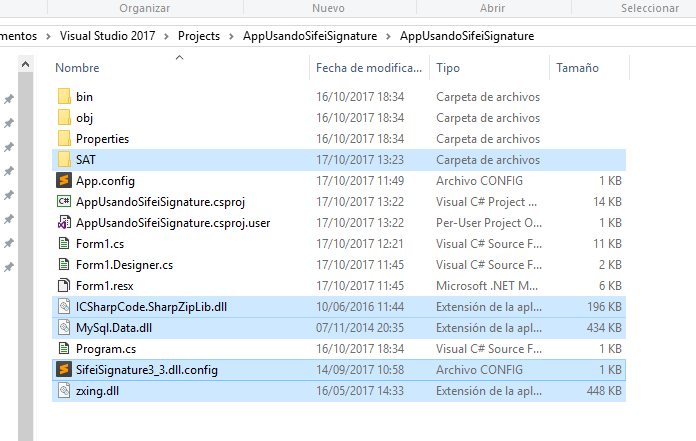


Imagen Copiando los recursos al proyecto

### Incluir la carpeta SAT y dll en el proyecto

Por defecto los archivos no serán visibles, así que se debe presionar el icono “Mostrar todos los archivos” y después seleccionar Imagen 1la carpeta SAT (Imagen 4) y los dll para ser incluidos en el proyecto (Imagen 5).

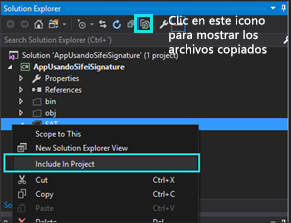


Imagen Inclusión de carpeta SAT en el proyecto

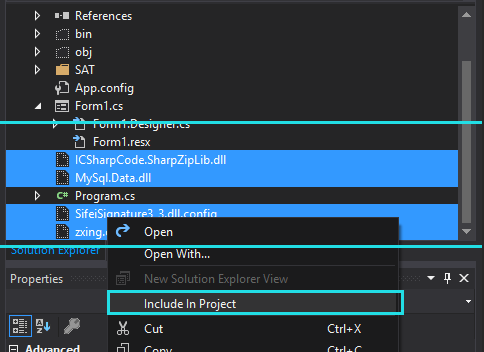


Imagen Inclusión del resto de bibliotecas

### Configurar los archivos para ser copiados en el directorio de salida

A pesar de haber incluido los recursos al proyecto, estos no son copiados al momento de compilación, por ello se deben de seleccionar e indicar la propiedad “Copiar al directorio de Salida” el valor “Copiar siempre”, esto debe hacerse para los XSLT, XSD y openssl.

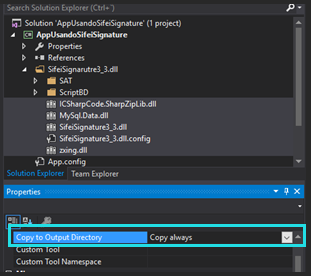


Imagen Modificando la propiedad "Copiar al directorio de Salida"

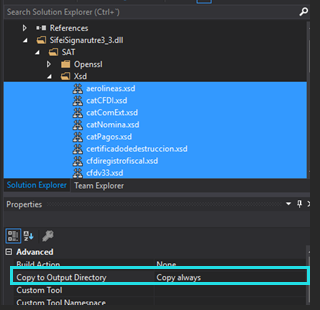


Imagen Modificando la propiedad "Copiar al directorio de salida" a "Copiar siempre”

## Agregar la biblioteca al proyecto

El método para agregar una DLL a un proyecto es a través de Referencias, para agregar *SIFEI Signature* primero se debe desplegar el menú secundario en el proyecto y seleccionar en “agregar referencia” o “*add reference*” (si Visual Studio está en idioma inglés).

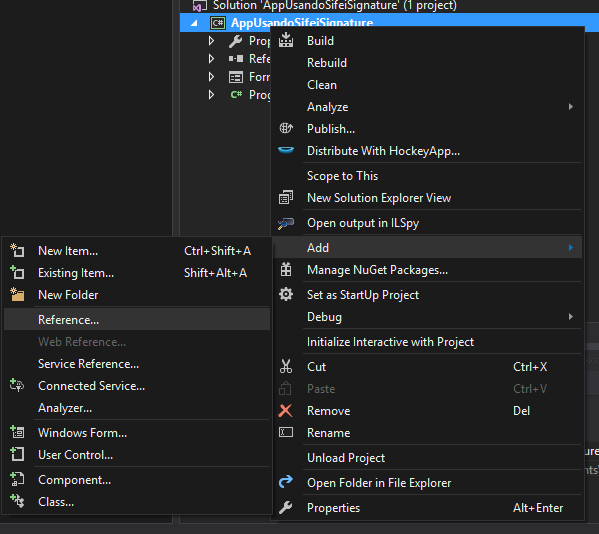


Imagen Agregar referencia al proyecto

Se debe seleccionar la biblioteca “SifeiSignature3\_3.dll” y agregar.

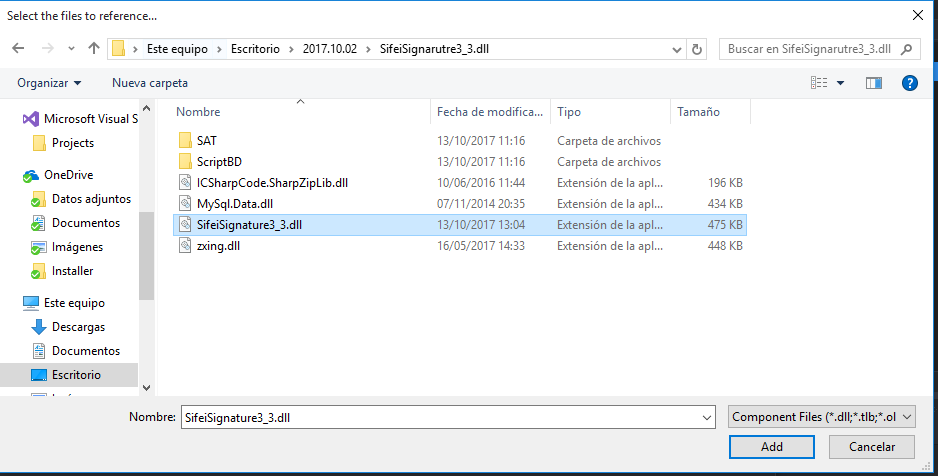


Imagen Carpeta donde está SifeiSignature

Se recomienda que la ruta sea relativa al proyecto, es decir, que esté dentro de la carpeta del proyecto, esto en caso de que copie el área de trabajo.

SifeiSignature3\_3.dll tiene dependencias sobre otras librerías y recursos para funcionar, es por ello que los demás archivos dll y carpeta SAT deben estar localizados en la misma carpeta y nivel que SifeiSignature3\_3.dll al momento de ejecución y como se muestra en la imagen

Ya marcada la biblioteca, se debe dar clic en *Aceptar* –o en *Ok* si VS está en idioma inglés-.

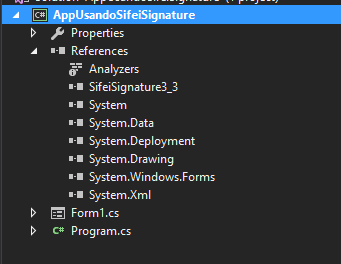


Imagen Referencia de biblioteca agregada

## Agregar información de los servicios web (configurar WS)

Para poder consumir los servicios a través de la DLL, es **necesario agregar los endpoint de los servicios**, para ello se debe agregar (si no existe) el siguiente nodo “system.serviceModel” al archivo de configuración App.config (como hijo de configuration). Donde deberá tener 5 nodos “endpoint” dentro del nodo “client” con sus respectivos “binding” en el nodo “basicHttpBinding”.

<system.serviceModel>

<bindings>

<basicHttpBinding>

<binding name="SIFEIPortBinding" />

<!-- Binding para nueva cancelacion -->

<binding name="CancelacionPortBinding" />

<binding name="BasicHttpBinding\_IConsultaCFDIService">

<security mode="Transport" />

</binding>

<binding name="SIFEIPortBinding1" />

<binding name="WSseries-certificadosBinding" />

</basicHttpBinding>

</bindings>

<client>

<!--End point de timbrado, nota se controla con TipoTimbrado(en codigo) la URL. -->

<endpoint address="http://192.168.16.251:8080/DESKTOP33/SIFEI" binding="basicHttpBinding" bindingConfiguration="SIFEIPortBinding" contract="srvSifeiTimbrador.SIFEI" name="SIFEIPort" />

<!--End point de cancelacion, nota se controla con TipoTimbrado(en codigo) la URL. -->

<endpoint address="http://devcfdi.sifei.com.mx:8888/CancelacionSIFEI/Cancelacion" binding="basicHttpBinding" bindingConfiguration="CancelacionPortBinding" contract="srvSifeiCancelacion2018.Cancelacion" name="CancelacionPort" />

<!--End point de consultaQr, nota se controla con TipoTimbrado(en codigo) la URL. -->

<endpoint address="https://consultaqr.facturaelectronica.sat.gob.mx/ConsultaCFDIService.svc" binding="basicHttpBinding" bindingConfiguration="BasicHttpBinding\_IConsultaCFDIService" contract="srvSatConsultaQR.IConsultaCFDIService" name="BasicHttpBinding\_IConsultaCFDIService" />

<!--End point de Timbrado de Retenciones, nota se controla con TipoTimbrado(en codigo) la URL. -->

<endpoint address=<http://192.168.16.251:8080/RETENCIONES/SIFEI> binding="basicHttpBinding" bindingConfiguration="SIFEIPortBinding1" contract="srvSifeiTimbradorRetenciones.SIFEI" name="SIFEIPort1" />

<!--End point de Autenticación de Sifei Signature, nota se controla con TipoTimbrado(en codigo) la URL. -->

<endpoint address=<http://192.168.16.101/cfdipac/server_WS/WSSerieCertif.php> binding="basicHttpBinding" bindingConfiguration="WSseries-certificadosBinding" contract="srvSifeiAuth.WSseriescertificadosPortType" name="WSseries-certificadosPort" />

</client>

</system.serviceModel>

Imagen Configuración de aplicación requerida para *SIFEI Signature*

Nota. Si se omite la configuración de algún servicio, una excepción de tipo InvalidOperationException será arrojada, ya que no encontrara el elemento endpoint correspondiente en el nodo ServiceModel, es necesario contar con estos 5 endpoint configurados.

Para hacerlo se debe crear el archivo de configuración (si no existe):

* Seleccionar el proyecto (no a la solución)
* Dar clic secundario y selecciona “*Agregar*” -> “*nuevo elemento*”
* Seleccionar en *Archivo de configuración de aplicación* y dar clic en “*Agregar*”.

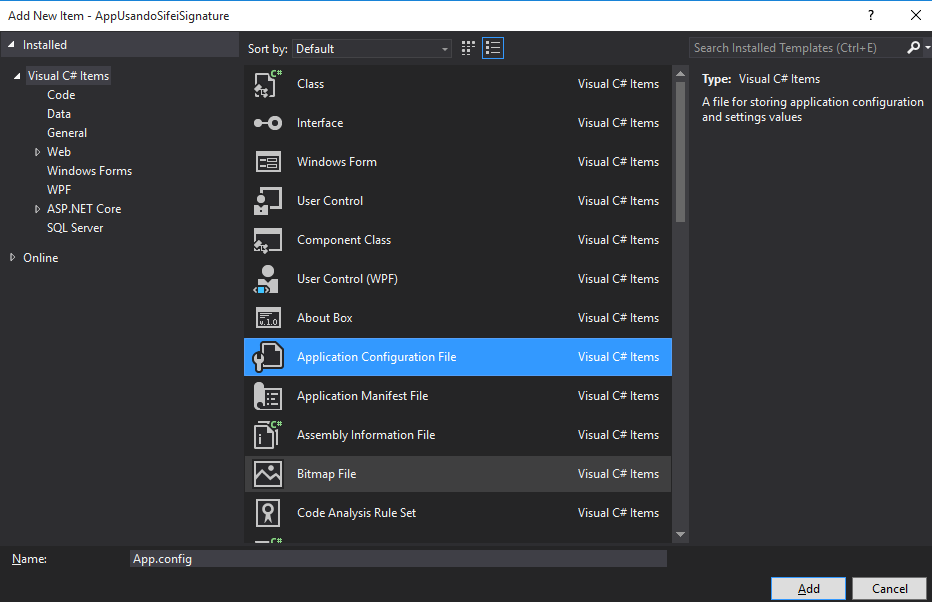


Imagen Seleccionado Archivo de Configuración de Aplicación

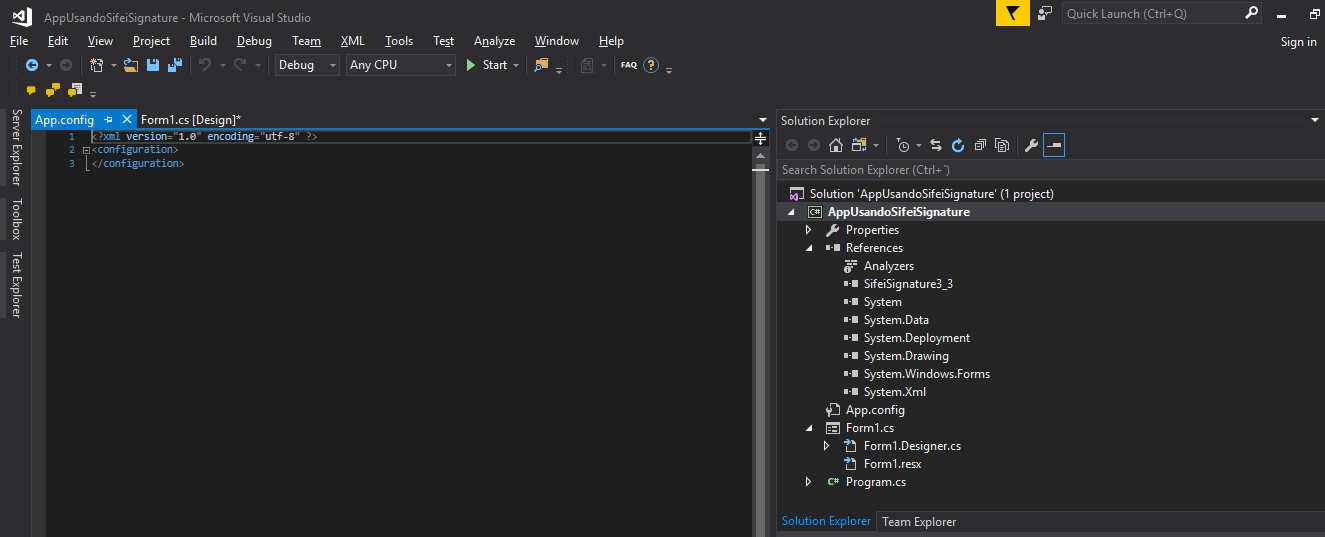


Imagen Archivo de configuración limpio

Copiar el system.servicemodel.

Para resultar en algo similar a como se muestra en la Imagen 14.

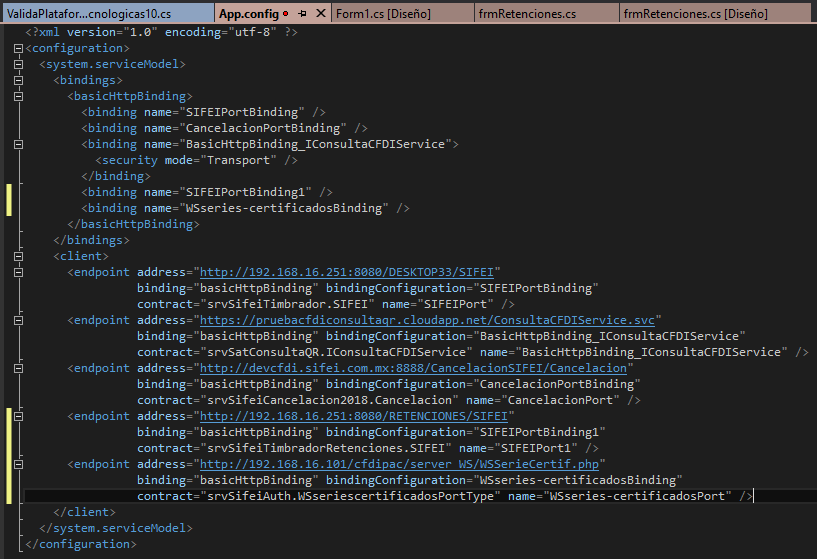


Imagen Archivo de configuración con la configuración necesaria para *SIFEI Signature*

## Crear base de datos

Dentro de los recursos ofrecidos por *SIFEI Signature* se incluye un script con los catálogos del SAT listo para ser usado en MySQL o MariaDB, el nombre del script es “bd\_catálogossat.sql“.

Existen varias formas de ejecutar el script, la forma que se cubrirá en este apartado es a través de la *herramienta de línea de comando de MySQL* suponiendo que el gestor está instalado de forma local y que la carpeta bin de MariaDB está definida en las variables de entorno. Además, la cuenta de MariaDB debe tener privilegios para crear base de datos y tablas.

**Nota:** Para ver los requisitos ir al apartado requerimientos.

### Comandos usados para crear la base de datos

Shell> mysql –U root –p catalogosat< bd\_catalogossat.sql

Imagen Ejecuta el *script* para la creación de la base de datos

MariaDb> show variables where variable\_name='max\_allowed\_packet';

Imagen Consulta para mostrar el valor de max\_allowed\_packet

### Cambiar max\_allowed\_packet (Opcional)

Este paso es opcional y debe realizarse cuando la variable max\_allowed\_packet sea menor a 8mb y aparezca el siguiente error “Error: 2006”, ya que el *script* cuenta con sentencias que superan este límite de tamaño de paquete.



Imagen 17 Error 2006 MySQL server has gone away

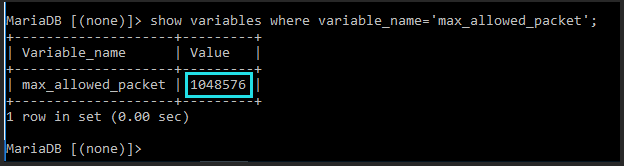


Imagen 18 Variable max\_allowed\_packet = 1MB: causará “Error 2006: MySQL server has gone away”

La forma más fácil y **temporal** para hacer el cambio es ejecutar el comando inferior, mismo que tendrá efecto a partir de la **siguiente conexión** (no la actual) y solo hasta que el **servicio sea reiniciado**:

MariaDb>set global max\_allowed\_packet=32\*1024\*1024;

Imagen Comando para cambiar de forma global el límite que define max\_allowed\_packet a 32 MB

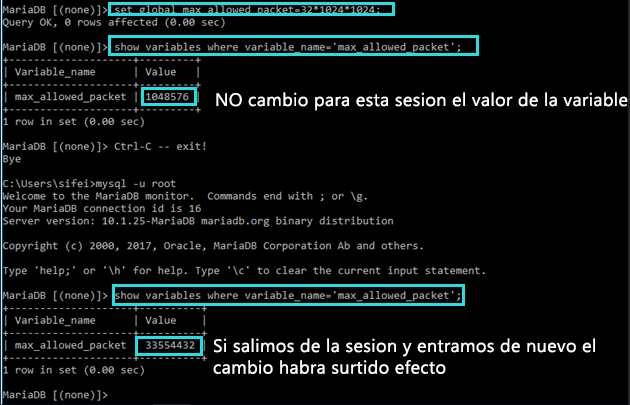


Imagen Cambiando variable Max\_allowed\_packet temporalmente (este efecto no cambia la sesión actual y dura solo mientras el servicio esté activo -hasta que el servicio sea reiniciado-)

Para hacerlo de forma permanente se debe cambiar el archivo de configuración de MariaDB:

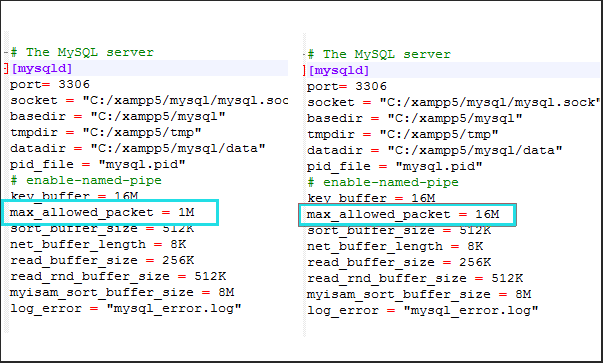


Imagen Archivo de configuración del servidor de MariaDB, cambio de 1mb a 16 MB (se debe reiniciar el servicio)

### Ejecutar el *script*

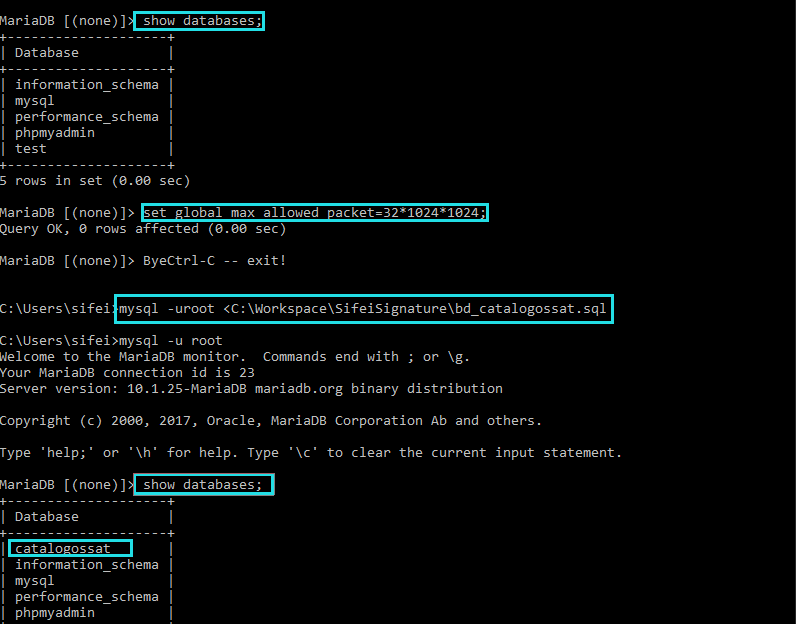


Imagen Procedimiento para la ejecución del *script*, comenzando por el listado de la base de datos, la ejecución del *script* y finalizando por la comprobación de su creación

### Crear la base de datos en Mysql

**Nota:** Antes de seguir este apartado se debe leer el apartado [Crear base de datos](#_Crear_base_de), ya que en este se explica como importar la base de datos, los comandos y las configuraciones que pudiera requerir (validas para MariaDb y Mysql) , seguir los siguientes pasos tras tener configurado correctemte la variable [max\_allowed\_packet](#_Cambiar_max_allowed_packet_(Opciona)

#### Remover PAGE\_CHECSUM=1

En el caso particular de Mysql, **se deben de hacer ligeras modificaciones al script** del catalogossat que se incluye, de lo contrario, un error como el que muestra en la imagen inferior ocurrirá:

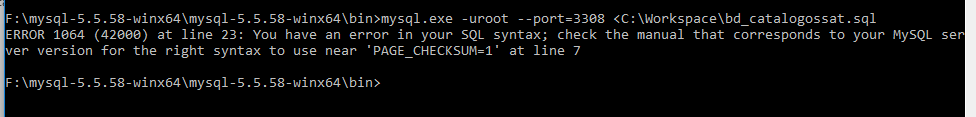


Imagen Error de sintaxis en Mysql al importar el script de la base de datos catalogossat

Esto ocurre porque Mysql no reconoce PAGE\_CHECKSUM dentro de su sintaxis (es propia de MariaDB), así que para poder continuar debemos **remover** **todas** las ocurrencias de ‘**PAGE\_CHECSUM=1**’ que se encuentran dentro del **script**, para hacerlo rápido podemos apoyarnos de un editor través de la herramienta “**buscar y reemplazar**”, donde buscaremos **PAGE\_CHECSUM=1** y remplazaremos por vacío, es decir, **que no exista un carácter** en el cuadro de texto “Reemplar con” (Replace with).

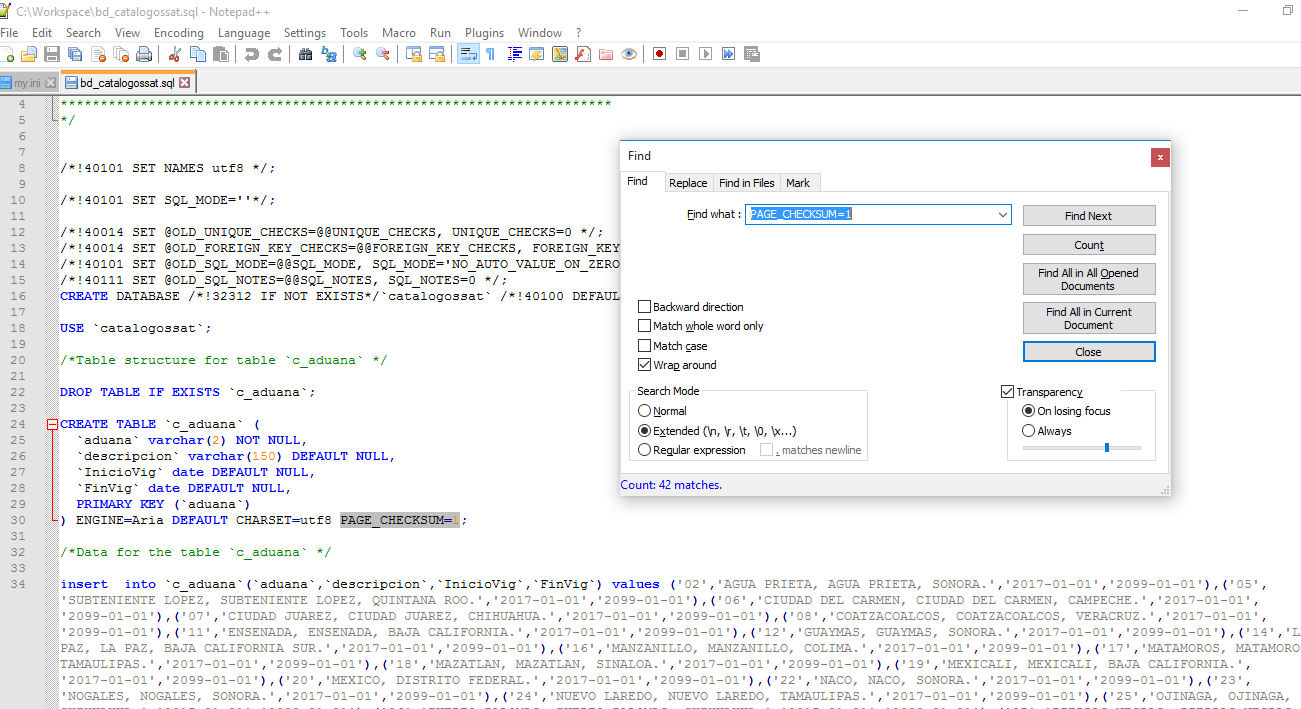


Imagen Vista del catalogossat que muestra PAGE\_CHECSUM y el numero de ocurrencias

Ahora basta con seleccionar remplazar a todo “*Replace all*” y guardaremos el catalogo.

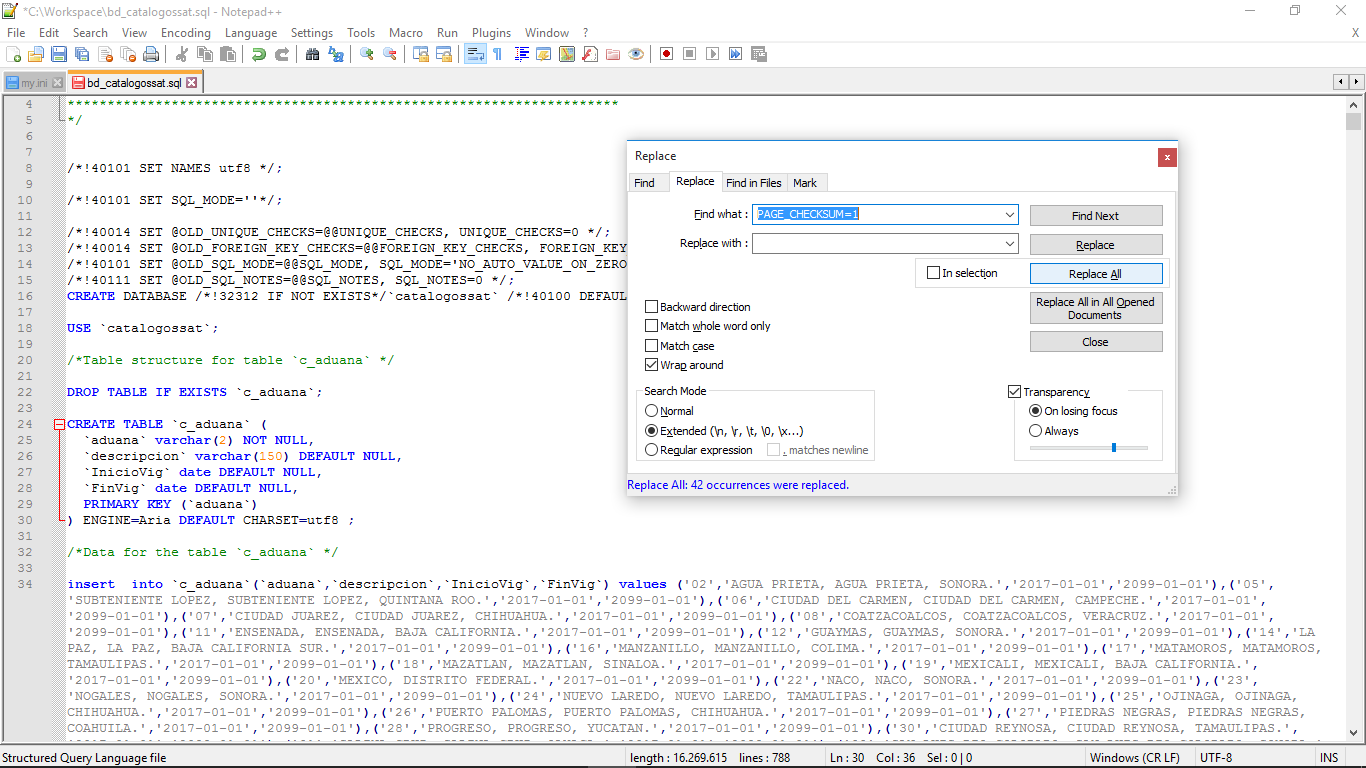
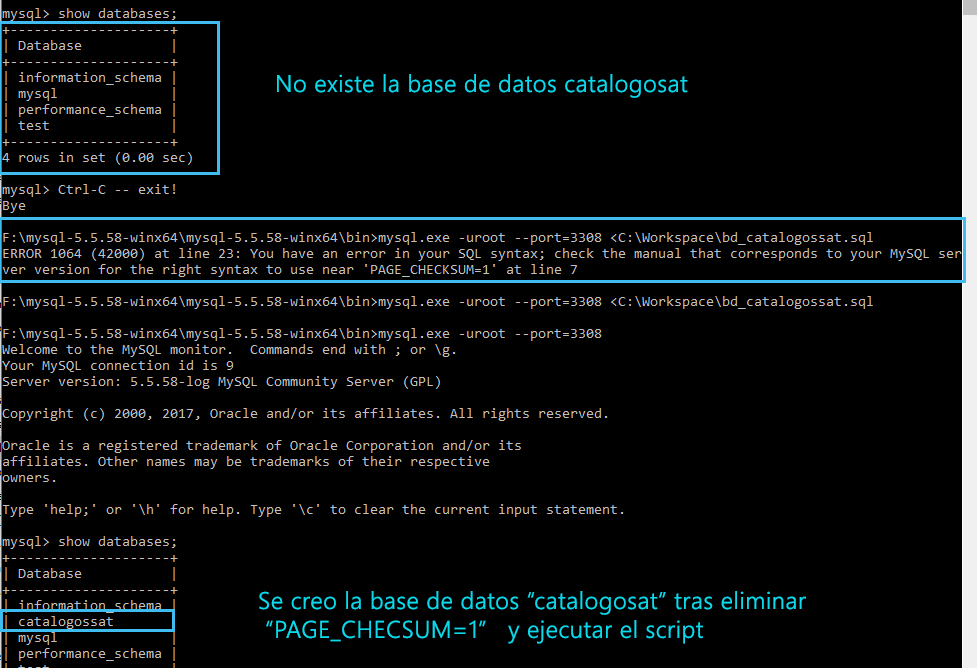


Imagen Catalogossat sin PAGE\_CHECSUM

Ahora podemos regresar al cliente Mysql e intentarlo de nuevo.



Podemos consultar la información sobre las tablas de catalogosat si ejecutamos la siguiente instrucción:

Mysql> select table\_schema,table\_name,engine from information\_schema.tables where table\_schema='catalogossat';

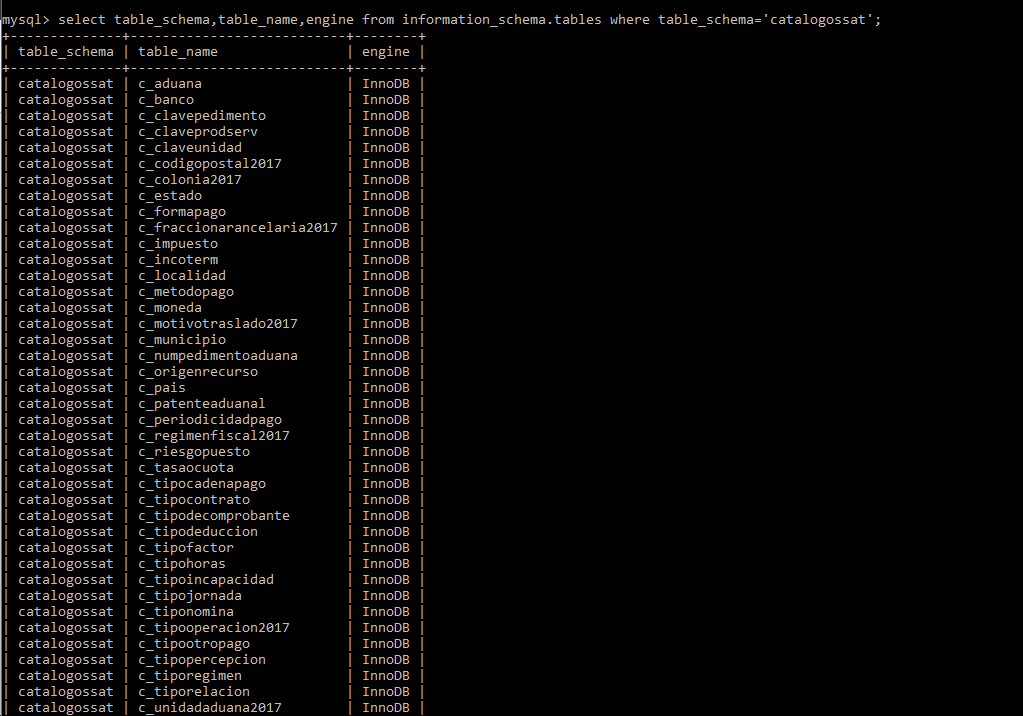


Imagen Captura que muestra de forma parcial las tablas creadas para la base de datos catalogossat en Mysql 5.5 y superior

En este caso podemos notar que además de crear tablas, les asigno como motor InnoDB, esto ocurre porque en Mysql cuando un motor no esta compilado o desactivado (no esta disponible) Mysql tratara de usar el motor por defecto (InnoDb en Mysql 5.5 y superior, MYISAM como motor en 5.1) esto se explica en el siguiente párrafo de la documentación de Mysql:

*“If you try to use a storage engine that is not compiled in or that is compiled in but deactivated, MySQL instead creates a table using the default storage engine*. This behavior is convenient when you want to copy tables between MySQL servers that support different storage engines.”

Ya que a partir de Mysql 5.5 el motor de base de datos por **defecto** es InnoDB en versiones iguales y superiores a esta el cambio a InnoDB sucede de forma transparente para el cliente(al menos que el motor por defecto sea diferente). Sin embargo en la misma documentación se indica que si en Mysql esta habilitado el modo sql [NO\_ENGINE\_SUBSTITUTION](https://dev.mysql.com/doc/refman/5.5/en/sql-mode.html#sqlmode_no_engine_substitution), un error ocurrirá y la tabla no será creada o alterada si el motor no está disponible:

“*This automatic substitution of the default storage engine for unavailable engines can be confusing for new MySQL users. A warning is generated whenever a storage engine is automatically changed. To prevent this from happening if the desired engine is unavailable, enable the* [*NO\_ENGINE\_SUBSTITUTION*](https://dev.mysql.com/doc/refman/5.5/en/sql-mode.html#sqlmode_no_engine_substitution) *SQL mode.* ***In this case, an error occurs*** *instead of a warning* ***and the table is not created or altered if the desired engine is unavailable****”*

Por este motivo si ocurre un error al generar el catalogossat se deberá de deshabilitar el modo [NO\_ENGINE\_SUBSTITUTION](https://dev.mysql.com/doc/refman/5.5/en/sql-mode.html#sqlmode_no_engine_substitution) en el servidor Mysql, o bien como se propone a continuación **cambiar** en el script todas las ocurrencias de **Aria** por **InnoDB.**

#### Cambiar de motor de Aria por InnoDb en el script

Este método se propone como la solución más rápida y fácil, basta con reemplazar todas las ocurrencias de **ENGINE=Aria** por **ENGINE= InnoDB**, como se muestra a continuación:

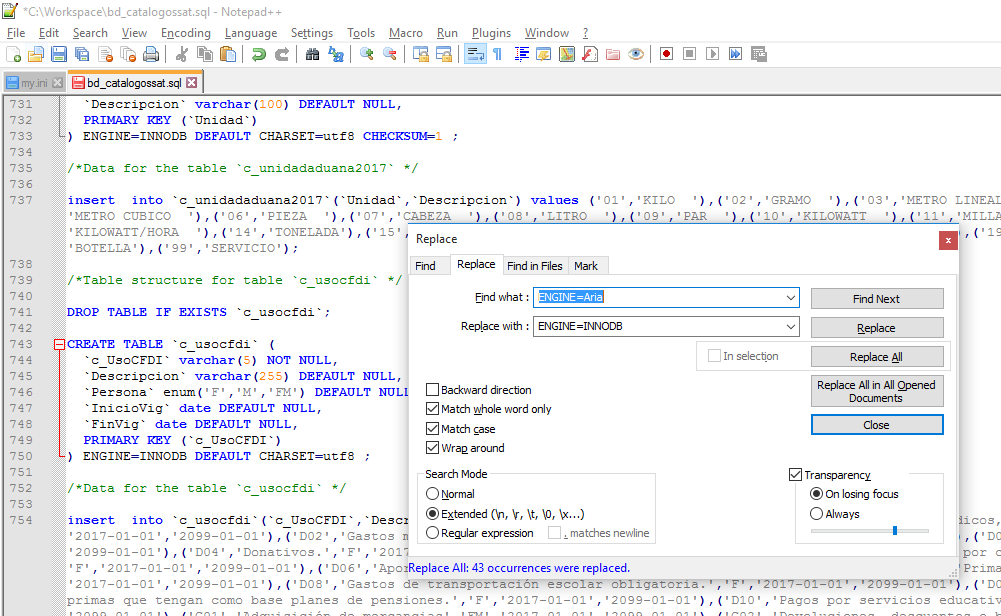


Imagen Captura de remplazar todas las ocurrencias de "ENGINE=Aria" por "ENGINE=InnoDB"

Se debe asegurar el cambio en el script antes de importarlo, se importa la base de datos y comprobamos que existe.

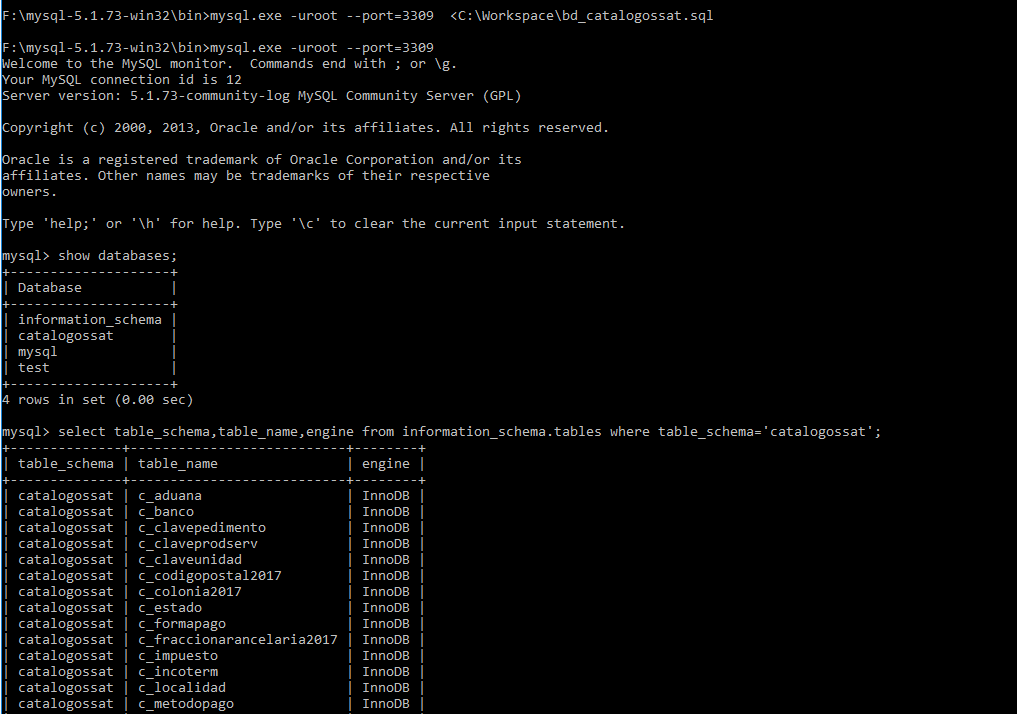


Imagen Captura de la importacion del script catalogossat con motor InnoDb

## Mysql y MariaDb en Servidores Linux, Windows y Mac OS X

Ya habiendo ejectado el script de la base de datos catalogossat, esto es, importar la base de datos, es importante tomar en cuenta el sistema operativo (SO) donde reside el Servidor Mysql o MariaDB, ya que afecta las configuracion del servidor a través de las **variables de sistema** que pueden ser modificadas 1) en el archivos de configuración, ó 2) ser pasadas como parámetros a mysqld al iniciar el servidor.

Como nota los valores por defecto son aquellos que no se han definido en el archivo de configuración de Mysql —nombrado comunmente como ***my.cnf***.

### *Case sensitive* (sensibilidad a minúsculas y mayúsculas)

La <<sensibilidad a mininusculas y mayúsculas>> (***case sensitive***) es una de las características que varia dependiendo del SO donde se instale Mysql/MariaDB, regida por la varible de sistema [**lower\_case\_table\_names**](https://dev.mysql.com/doc/refman/5.5/en/server-system-variables.html#sysvar_lower_case_table_names).En sistemas basados en unix el valor por defecto es 0(sensible), en Windows el valor por defecto es 1(insensible) y en OS X, el valor por defecto es 2. Es posible hacer que sistemas basados en unix se comporten como Windows y sean insenbles (como se ve en este apartado).

La siguiente tabla —extraída de la documentación de Mysql— muestra los valores posibles.

|  |  |
| --- | --- |
| Valor | Descripcion |
| 0 | Los nombre de tablas y base de datos son almacenados respetando el lettercase (tal cual estan escritas) de las sentencias the [CREATE TABLE](https://dev.mysql.com/doc/refman/5.7/en/create-table.html) or [CREATE DATABASE](https://dev.mysql.com/doc/refman/5.7/en/create-database.html). La compacion de nombres es **sensible** a minisculas y mayusculas. |
| 1 | Los nombre de tablas son almacenadas en minisculas en el disco y las comparaciones de nombres son **insensibles**. Este comportamiento se aplica también a nombres de base de datos y alias de tablas. Mysql convirte todos los nombres de tabla a miniscula cuando se almacenan y realizan búsquedas, |
| 2 | Los nombres de tabla y base de datos son almacenados en disco usando el lettercase especificado en las sententencias [CREATE TABLE](https://dev.mysql.com/doc/refman/5.7/en/create-table.html) o [CREATE DATABASE](https://dev.mysql.com/doc/refman/5.7/en/create-database.html) pero Mysql convierte a minisculas cuando se hace una búsqueda. Los nombres de comparación son insensibles. Esto solo funciona en sistemas de archivos que son insensibles. |

Nota: nunca cambiar el valor de [***lower\_case\_table\_names***](https://dev.mysql.com/doc/refman/5.5/en/server-system-variables.html#sysvar_lower_case_table_names) a 0 en sistemas insensibles como Windows(cuyo valor es 1).

Podemos consultar la variable con el siguiente comando:

Mysql> show variables where variable\_name='lower\_case\_table\_names';

Ejemplo de Servidor Mysql en un Centos 7

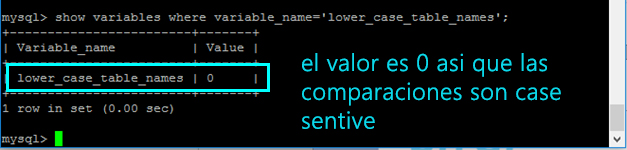


Imagen Consulta de la variable lowe\_case\_table\_names donde su valor es 0, y por **tanto debe ser cambiado a 1**

Por lo cual si no cambiamos el valor a 1 (comparaciones insensibles a Mayusculas, minúsculas) al usar la dll mostrara la siguiente excepción, ya que no encuentra la tabla **por que la comparación es sensible a minúsculas y mayusculas.**

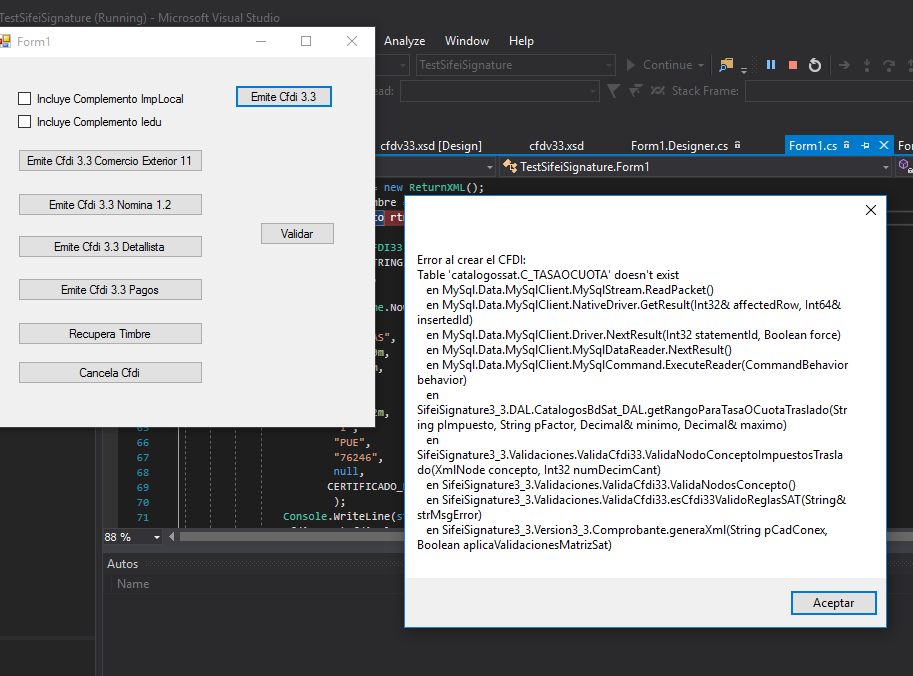


Imagen Error al emitir un Cfdi cuando Mysql reside un servidor Unix y la variable de sistema [***lower\_case\_table\_names***](https://dev.mysql.com/doc/refman/5.5/en/server-system-variables.html#sysvar_lower_case_table_names) tiene valor 0.

### Hacer insensible Mysql y MariaDb en sistemas basados en Unix.

Para hacer insensible la comparación de nombre de tablas se debe modificar el valor de lower\_case\_table\_names a **1** en el archivo de configuración de Mysql/MariaDb tal y como se muestra a continuación (nota: es posible que la variable no este definida en el archivo, por lo cual debe agregarse).

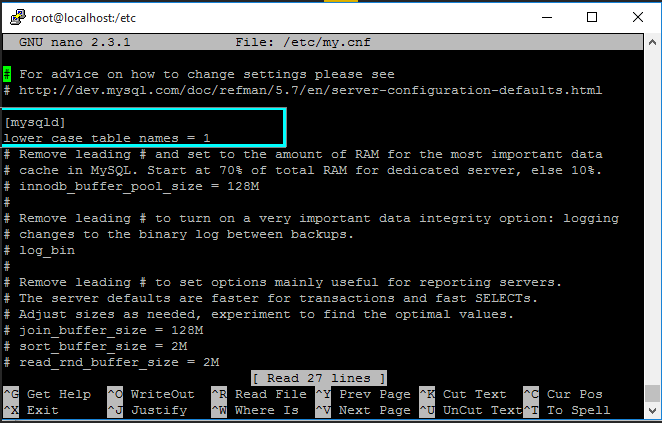


Imagen Configurando el valor de lowe\_case\_table\_names= 1

Ahora solo basta reiniciar Mysql (el comando varia entre distribuciones) y consultar que el valor de la variable haya cambiado a 1.

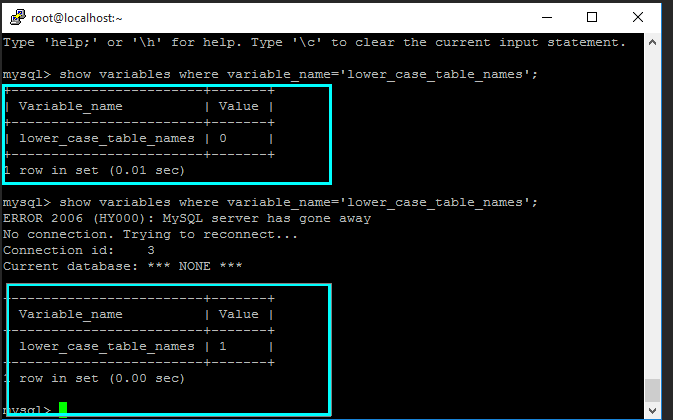


Imagen Consulta de variable del sistema lower\_case\_table\_names , donde se muestra el cambio del valor de 0 a 1 (en un Centos) tras reiniciar Mysql Server, notar que mysql client pierde la conexión y se conecta de nuevo debido al reinicio de Mysql Server.

Con esto ya es posible usar la base de datos en Servidores basados en Unix por la Dll.

# Ejemplos CFDI

A continuación, se presentan los ejemplos para emitir, sellar, timbrar y cancelar CFDIs y Comprobantes de Retención e Información de Pagos, además de recuperar timbres. Para mayor detalle sobre las clases, métodos y enumeraciones ver la sección 9 Referencia.

## Instanciar un Objeto tipo CFDI40

La clase [CFDI40](#_Clase_CFDI33) es la clase principal para comenzar a emitir CFDIs, en este ejemplo se le pasa como primer argumento la cadena de conexión a la base de datos Catálogossat, el resto de los argumentos son los atributos que poseerá el CFDI.

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07  08  09  10  11  12  13  14  15  16  17  18  19  20 | cfdi40 = new CFDI40("Server=localhost;Port=3306;Database=catalogossat;Uid=root;Pwd=xxx;",//connString  "BZNPLUS", //pSerie  "196", //pFolio  fechaCfdi, //pFecha  "99", //pFormaPago  "", //pCondPago  2750.00m, //pSubtotal  0.00m, //pDescuento  "MXN", //pMoneda  1m, //pTipoCambio  3190.00m, //pTotal  "I", //pTipoCompr  "01", //pExportacion  "PPD", //pMetodoPago  "72010", //pLugarExped  null, //pConfirmacion  this.rutaCert //rutaCert  ); |

Lo cual produce en el XML lo siguiente:



Imagen Atributos de CFDI agregados

## Establecer información del nodo InformacionGlobal

Utilizando la instancia de CFDI40 anterior invocamos el método “setInformacionGlobal()” y le pasamos la información correspondiente al nodo InformacionGlobal.

|  |  |
| --- | --- |
| 01  02  03  04 | //Estableciendo el nodo InformacionGlobal  cfdi40.setInformacionGlobal("04", //pPeriodicidad  "05", //pMeses  2021); //pAño |

Lo cual produce el nodo InformacionGlobal:



Imagen Nodo InformacionGlobal agregado

## Agregar CfdiRelacionados

Utilizando la instancia de [CFDI40](#_Clase_CFDI33) anterior invocamos el método “addCfdirelacionados()“ y le pasamos el tipo de relación y los UUIDs de los CFDIs Relacionados, lo cual genera un primer nodo CfdiRelacionados, con los UUIDs indicados.

|  |  |
| --- | --- |
| 01  02  03  04 | //Estableciendo el nodo CfdiRelacionados  cfdi40.addCfdirelacionados("04", //pTipoRelacion  "DB4A59FB-5BD5-4744-87B6-0F50E7DE2951", //pUUID  "806EA594-A850-48C3-B56E-1EB0C4D570C2"); //pUUID |

Lo cual produce el nodo CfdiRelacionados y los nodos CfdiRelacionado correspondientes a cada uno de los UUIDs indicados en el método anterior (ya que si existe CfdiRelacionados debe existir al menos un CfdiRelacionado):

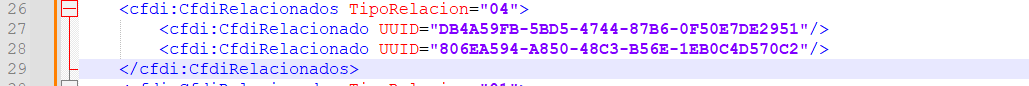


Imagen Nodo CfdiRelacionados agregado

En caso de requerir agregar más CFDIs relacionados con distinto valor para TipoRelacion, se deberá utilizar el mismo método tantas veces sea necesario.

## Establecer información del Emisor

Para agregar el emisor del CFDI se debe invocar al método setEmisor().

|  |  |
| --- | --- |
| 01  02  03  04  05 | // Estableciendo el Emisor  cfdi40.setEmisor(this.rfcEmisor, //pRfc  "HERRERIA & ELECTRICOS SA DE CV", //pNombre  "626"); //pRegFiscal |

El nodo agregado es Emisor:



Imagen Nodo Emisor agregado

## Establecer información del Receptor

Para agregar el Receptor del CFDI se debe invocar al método setReceptor().

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07 | // Estableciendo el Receptor  cfdi40.setReceptor(this.rfcReceptor, //pRfc  "TRANSPORTADORA CER-MA DE AHUAZOTEPEC", //pNombre  "73180", //pDomicilioFiscalReceptor  null, //pResidenciaFiscal  null, //pNumRegIdTrib  "626", //pRegimenFiscalReceptor  "S01"); //pUsoCfdi |

El nodo agregado es Receptor, el cual tiene los atributos con los valores que se les paso al método setReceptor():

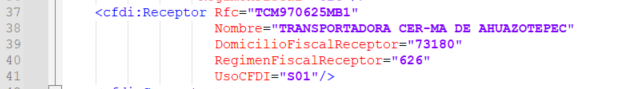


Imagen Nodo Receptor agregado

## Agregar un Concepto

Para agregar un concepto se debe invocar al método addConcepto() el cual devuelve un entero con el índice del concepto que se creó, **mismo que servirá en posteriores métodos para agregar información adicional al concepto (como impuestos, información aduanera, etc.)**.

El parámetro @indexConc es de tipo entero y recibe el índice del nodo concepto.

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07  08  09  10 | indexConc = cfdi40.addConcepto("78101800", //pCveProdServ  "", //pNoIdentif  1m, //pCantidad  "E48", //pCveUnidad  "F", //pUnidad  "VIAJE DE PARAFINA DEL 26 DE NOVIEMBRE DE 2021 CON 17,000 KG DE PESO", //pDescripcion  2750.00m, //pValorUnit  2750.00m, //pImporte  0.00m, //pDescuento  "02"); //pObjetoImp |

El nodo agregado es concepto con los atributos que se le paso al método:

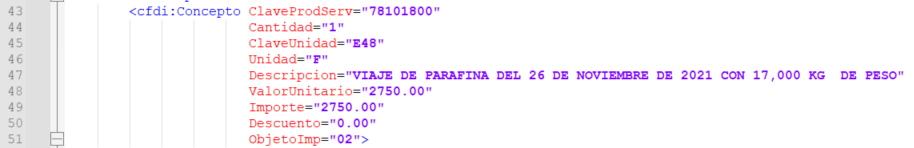


Imagen Nodo Concepto agregado

## Agregar Información aduanera a un Concepto

Cuando se ha definido un concepto al CFDI, es posible agregar “información aduanera”, para ello es necesario indicar el índice (posición) del concepto y pasar los parámetros al método addInformacionAduaneraToConcepto();

|  |  |
| --- | --- |
| 01  02  03  04  05  06 | // Agregando Información aduanera al concepto con posición indexConc,  Cfdi40.addInformacionAduaneraToConcepto(indexConc, //pIndexConc seria= 1  "15 43 3740 6000564" //pNumPedimento  ); |

Modifica el nodo concepto apuntado por @indexConc agregándole como hijo el nodo información Aduanera:



Imagen InformacionAduanera agregada a conceptos

## Agregar un impuesto traslado a un concepto

Cuando se ha definido un concepto al CFDi, es posible indicar los impuestos traslados por medio del método addImpTrasladoToConcepto(). El concepto se indica con el parámetro @indexConc.

|  |  |
| --- | --- |
| 01  02  03  04  05  06 | // agregando impuesto traslado al concepto  cfdi40.addImpTrasladoToConcepto(indexConc, //pIndexConc  2750.00m, //pBase  "002", //pImpuesto  "Tasa", //pTipoFactor  0.160000m, //pTasaOCuota  440.00m); //pImporte |

Agrega el traslado al concepto:

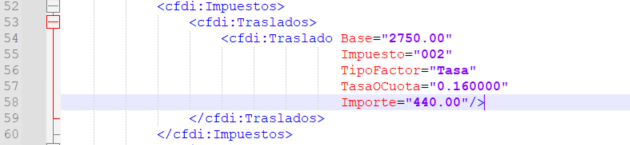


Imagen Resultado de agregar un impuesto traslado al concepto

## Agregar un impuesto retenido a un concepto

Cuando se ha definido un concepto al CFDI, es posible indicar los impuestos retenidos por medio del método addImpRetencionToConcepto(). El concepto se indica con el parámetro @indexConc.

|  |  |
| --- | --- |
| 01  02  03  04  05  06 | // agregando impuesto retenido al concepto  cfdi40.addImpRetencionToConcepto(indexConc, //pIndexConc  2750.00m, //pBase  "001", //pImpuesto  "Tasa", //pTipoFactor  0.106666m, //pTasaOCuota  293.3315m); //pImporte |

Agrega la retención al concepto:

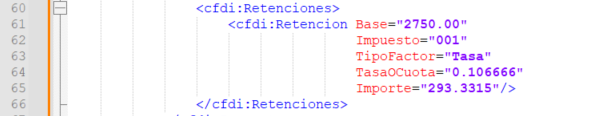


Imagen Resultado de agregar un impuesto traslado al concepto

## Establecer información de A Cuenta de Terceros

Cuando se ha definido un concepto al CFDI, es posible indicar la información correspondiente al nodo A Cuenta de Terceros por medio del método setACuentaTercerosToConcepto(). El concepto se indica con el parámetro @indexConc.

|  |  |
| --- | --- |
| 01  02  03  04  05  06 | // agregando información ACuentaTerceros al concepto  cfdi40.setACuentaTercerosToConcepto(indexConc, //pIndexConc  "EMS810717R34", //pRfcACuentaTerceros  "EMPAQUES MODERNOS SAN PABLO", //pNombreACuentaTerceros  "601", //pRegimenFiscalACuentaTerceros  "54090"); //pDomicilioFiscalACuentaTerceros |

Agrega el nodo ACuentaTerceros al concepto:

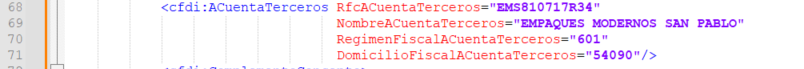


Imagen Resultado de agregar un impuesto traslado al concepto

## Agregar Cuenta Predial al concepto

Cuando se ha definido un concepto al CFDI, es posible indicar la información correspondiente a la Cuenta Predial por medio del método addCuentaPredialToConcepto(). El concepto se indica con el parámetro @indexConc.

|  |  |
| --- | --- |
| 01  02  03  04  05  06 | // agregando información Cuenta Predial al concepto  cfdi40.addCuentaPredialToConcepto(indexConc, //pIndexConc  "Predial1", //pLsNumeros  "Predial2",  "Predial3"); |

Agrega los nodos CuentaPredial al concepto:



Imagen Resultado de agregar un impuesto traslado al concepto

## Establecer información del total impuestos de comprobante

Definiendo el total impuestos con el método setTotalImpuestos().

|  |  |
| --- | --- |
| 01  02  03  04  05 | // estableciendo el total de impuestos  cfdi40.setTotalImpuestos(null, //pTotImpRetenidos  440.00m); //pTotImpTrasladados |

Lo cual agrega el nodo Impuestos:



Imagen Nodo Impuestos agregado a comprobante

## Agregar un impuesto traslado a impuestos de comprobante

Solo si se ha definido el nodo Impuestos al comprobante (Xpath: cfdi:Comprobante/cfdi:Impuestos), es posible indicar el impuesto Traslado por medio del método addImpTraslado().

|  |  |
| --- | --- |
| 01  02  03  04  05 | //estableciendo un impuesto traslado  cfdi40.addImpTraslado(2750.00m, //pBase  "002", //pImpuesto  "Tasa", //pTipoFactor  0.160000m, //pTasaCuota  440.00m); //pImporte |

Este agrega los nodos traslados y traslado al nodo “impuestos”

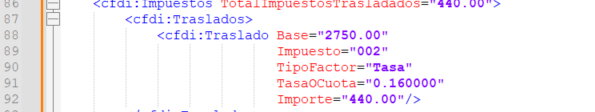


Imagen Impuesto traslado agregado

## Autorización para el uso de Sifei Signature

Antes de utilizar los métodos para generar los complementos, generar el XML del comprobante CFDI 4.0 o de Retenciones e Información de Pagos, o sellar cualquiera de los tipos de comprobantes que emite Sifei Signature debe identificarse con los datos de acceso que le fueron asignados, ya sea en ambiente de pruebas o ambiente de producción y que son los mismos que se utilizan para timbrar. Esto se hace mediante el método setIdentificacionSifei():

|  |  |
| --- | --- |
| 01  02  03  04  05 | //identificándose para el uso de Sifei Signature  ret.setIdentificacionSifei(this.usuarioSifei, //Usuario Sifei  this.pwdSifei, //Contraseña  this.tipoTimb); //Tipo de timbrado |

En caso de no haber invocado el método especificado, se generará la siguiente excepción:

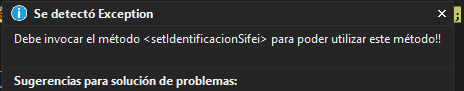


Imagen Autenticandose para el uso de Sifei Signature

En caso de que los datos de usuario no estén correctos, se generará la siguiente excepción:

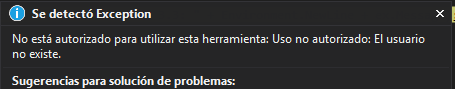


Imagen Error de autenticación

## Generar el XML

Para generar el XmlDocument debemos invocar al método generaXml() y pasarle el *string* de conexión, el objeto XmlDocument no es retornado de forma directa, sino que es devuelto en un objeto *Wrapper* de tipo [ReturnXML](#_Clase_ReturnXML), este objeto nos permite averiguar si se generó el XML o no. (Para más información ver la referencia [ReturnXML](#_Clase_ReturnXML)).

Para utilizar este método antes debió invocar al método setIdentificacionSifei().

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07  08  09  10  11  12 | //generamos el XML, el objeto de tipo ReturnXML, contiene propiedades que indican si se creó el XML, y si es así en su propiedad XML, el objeto XmlDocument  ReturnXML rtnCfdi = cfdi40.generaXml(ConnString);  //comprobamos que se ha creado con la propiedad IsXMLCreated  if (rtnCfdi.IsXMLCreated) {  // si se creó, obtenemos de la propiedad XML para guardarlo rtnCfdi.XML.Save(@"C:\Workspace\SifeiSignature\CfdiGeneradosTest\cfdi\_creado.xml");  if (!File.Exists(@"C:\Workspace\SifeiSignature\CfdiGeneradosTest\cfdi\_creado.xml")) {  MessageBox.Show("No se escribió el archivo");  }  } |

Tras seguir los ejemplos de código anteriores el XML producido es el siguiente:

|  |
| --- |
| <?xml version=**"1.0"** encoding=**"utf-8"**?>  <cfdi:Comprobante xmlns:cfdi=**"http://www.sat.gob.mx/cfd/4"**  xmlns:xsi=**"http://www.w3.org/2001/XMLSchema-instance"**  xmlns:iedu=**"http://www.sat.gob.mx/iedu"**  xsi:schemaLocation=**"http://www.sat.gob.mx/cfd/4 http://www.sat.gob.mx/sitio\_internet/cfd/4/cfdv40.xsd http://www.sat.gob.mx/iedu http://www.sat.gob.mx/sitio\_internet/cfd/iedu/iedu.xsd"**  Version=**"4.0"**  Serie=**"BZNPLUS"**  Folio=**"196"**  Fecha=**"2023-05-15T11:28:26"**  Sello=**"@"**  FormaPago=**"99"**  NoCertificado=**"30001000000400002463"**  Certificado=**"MIIFwTCCA6mgAwIBAgIUMzA..."**  SubTotal=**"2750.00"**  Descuento=**"0.00"**  Moneda=**"MXN"**  TipoCambio=**"1"**  Total=**"3190.00"**  TipoDeComprobante=**"I"**  Exportacion=**"01"**  MetodoPago=**"PPD"**  LugarExpedicion=**"72010"**>  <cfdi:InformacionGlobal Periodicidad=**"04"**  Meses=**"05"**  Año=**"2021"**/>  <cfdi:CfdiRelacionados TipoRelacion=**"04"**>  <cfdi:CfdiRelacionado UUID=**"DB4A59FB-5BD5-4744-87B6-0F50E7DE2951"**/>  <cfdi:CfdiRelacionado UUID=**"806EA594-A850-48C3-B56E-1EB0C4D570C2"**/>  </cfdi:CfdiRelacionados>  <cfdi:CfdiRelacionados TipoRelacion=**"01"**>  <cfdi:CfdiRelacionado UUID=**"DB4A59FB-5BD5-4744-87B6-0F50E7DE2953"**/>  <cfdi:CfdiRelacionado UUID=**"806EA594-A850-48C3-B56E-1EB0C4D570C4"**/>  </cfdi:CfdiRelacionados>  <cfdi:Emisor Rfc=**"H&amp;E951128469"**  Nombre=**"HERRERIA &amp; ELECTRICOS SA DE CV"**  RegimenFiscal=**"626"**/>  <cfdi:Receptor Rfc=**"TCM970625MB1"**  Nombre=**"TRANSPORTADORA CER-MA DE AHUAZOTEPEC"**  DomicilioFiscalReceptor=**"73180"**  RegimenFiscalReceptor=**"626"**  UsoCFDI=**"S01"**/>  <cfdi:Conceptos>  <cfdi:Concepto ClaveProdServ=**"78101800"**  Cantidad=**"1"**  ClaveUnidad=**"E48"**  Unidad=**"F"**  Descripcion=**"VIAJE DE PARAFINA DEL 26 DE NOVIEMBRE DE 2021 CON 17,000 KG DE PESO"**  ValorUnitario=**"2750.00"**  Importe=**"2750.00"**  Descuento=**"0.00"**  ObjetoImp=**"02"**>  <cfdi:Impuestos>  <cfdi:Traslados>  <cfdi:Traslado Base=**"2750.00"**  Impuesto=**"002"**  TipoFactor=**"Tasa"**  TasaOCuota=**"0.160000"**  Importe=**"440.00"**/>  </cfdi:Traslados>  <cfdi:Retenciones>  <cfdi:Retencion Base=**"2750.00"**  Impuesto=**"001"**  TipoFactor=**"Tasa"**  TasaOCuota=**"0.106666"**  Importe=**"293.3315"**/>  </cfdi:Retenciones>  </cfdi:Impuestos>  <cfdi:ACuentaTerceros RfcACuentaTerceros=**"EMS810717R34"**  NombreACuentaTerceros=**"EMPAQUES MODERNOS SAN PABLO"**  RegimenFiscalACuentaTerceros=**"601"**  DomicilioFiscalACuentaTerceros=**"54090"**/>  <cfdi:InformacionAduanera NumeroPedimento=**"15 43 3740 6000564"**/>  <cfdi:CuentaPredial Numero=**"Predial1"**/>  <cfdi:CuentaPredial Numero=**"Predial2"**/>  <cfdi:CuentaPredial Numero=**"Predial3"**/>  <cfdi:ComplementoConcepto>  <iedu:instEducativas version=**"1.0"**  nombreAlumno=**"SOLIS HERRERA VALERIA"**  CURP=**"SOHV031025MHGLRLA4"**  nivelEducativo=**"Primaria"**  autRVOE=**"13PPR0034X"**  rfcPago=**"AALE5706229F6"**/>  </cfdi:ComplementoConcepto>  </cfdi:Concepto>  </cfdi:Conceptos>  <cfdi:Impuestos TotalImpuestosTrasladados=**"440.00"**>  <cfdi:Traslados>  <cfdi:Traslado Base=**"2750.00"**  Impuesto=**"002"**  TipoFactor=**"Tasa"**  TasaOCuota=**"0.160000"**  Importe=**"440.00"**/>  </cfdi:Traslados>  </cfdi:Impuestos>  </cfdi:Comprobante> |

En caso de error se debe verificar la propiedad *ErrorDetalle* del objeto rtnCfdi de tipo ReturnXML (linea 07)

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07  08 | //.  //.codigo  //.  if (rtnCfdi.IsXMLCreated) {  }  else {  MessageBox.Show("Error al sellar el CFDI: " +rtnCfdi.ErrorDetalle);  } |

Para más saber sobre los valores que contiene la propiedad rtnCfdi.ErrorDetalle ver [Códigos de Error de la Matriz de validaciones para CFDI versión 4.0](#_Códigos_de_Error).

## Sellar el comprobante

Es posible sellar un comprobante a través del método *sellaComprobante()* el *cual retorna un objeto tipo* [ReturnXML](#_Clase_ReturnXML), mismo que nos sirvepara saber si el comprobante fue sellado y recuperarlo, el código siguiente muestra como sellar el comprobante.

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07  08  09  10  11  12  13  14  15 | //sellamos el comprobante pasándole el path del certificado y de la llave  ReturnXML rtnCfdi = cfdi40.sellaComprobante(rtnCfdi.XML, @"C:\Workspace\DatosBuzonMariaDB\Cert\DEMO20170725123153.cer", @"C:\Workspace\DatosBuzonMariaDB\Cert\DEMO20170725123153.key",  "12345678a",  false,  true);  //comprobamos que se halla sellado  if (rtnCfdi.EstaSellado) {  //escribimos el Comprobante sellado:  rtnCfdi.XML.Save(@"C:\Workspace\SifeiSignature\CfdiGeneradosTest\cfdi\_sellado.xml");  } |

Comprobante sellado atributo sello (línea 10, marcado en púrpura):



Imagen Comprobante Sellado

Para utilizar este método antes debió invocar al método setIdentificacionSifei().

## Timbrar el comprobante

Es posible timbrar un comprobante a través del método timbraComprobante()el cual retorna un objeto tipo[ReturnTimbre](#_Clase_ReturnTimbre), mismo que sirvepara saber si el comprobante fue timbrado y recuperarlo, tal y como se ve en el siguiente ejemplo:

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07  08  09  10  11 | //timbramos el comprobante  ReturnTimbre rtnTimbre = cfdi40.timbraComprobante("A",  "VPjftCMG1beV+/wkNWc8lkaFEtxTrC/rt20oZgbnl30S6SD6kPAmLozt7CMkqzZK3L7+mlxZtXYDvu9oauJ1WhEWqSieNh96KCdNJdIXezM=",  rtnCfdi.XML);  //verificamos que haya sido timbrado:  if (rtnTimbre.IsTimbrado) {  //escribimos el cfdi timbrado rtnTimbre.Xml.Save(@"C:\Workspace\SifeiSignature\cfdi\_timbrado.xml"); |

Para utilizar este método antes debió invocar al método setIdentificacionSifei().Es importante mencionar que este método ya no solicita el usuario y contraseña de timbrado, ni el ambiente en el que efectuará el timbrado (Pruebas o Producción), ya que estos son tomados de los valores que se le asignaron al invocar el método setIdentificacionSifei().

Al guardar el CFDI contendrá el nodo TimbreFiscalDigital (líneas 78 a 86):



Imagen Comprobante timbrado

## Recuperar timbre

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07  08  09  10  11  1213  14  15  16  17  18  19 | //creamos el XmlDocument y cargamos el comprobante  System.Xml.XmlDocument cfdi = new System.Xml.XmlDocument();  cfdi.Load(@"C:\Workspace\comprobante.xml");  ReturnTimbre recuperado = CFDI40.recuperaTimbre(  "JGE080430LN8", //usuario  "0e84ec92", //password  cfdi, //objeto XmlDocument  CFDI33.tipoTimbrado.Pruebas  );  //verificamos si esta timbrado (si existe un timbre)  if (recuperado.IsTimbrado) {  //escribimos el comprobante  recuperado.Xml.Save(@"C:\Workspace\hash\_cfdi\_timbrado.xml");    // escribimos el código de barra bidimensional en un archivo jpg  ByteArrayToFile(@"C:\Workspace\hash\_cbb.jpg", recuperado.Cbb);  Console.WriteLine(string.Format("{0} - Proceso finalizado", DateTime.Now));  } |

# Crear Complementos

Con la clase Complemento podemos crear varios tipos de complemento.

|  |  |
| --- | --- |
| 01  02  03  04  05 | //Instanciamos la clase complemento, pasándole el string de conexión.  Complemento complCfdi = new Complemento("Server=localhost;Port=3306;Database=catálogossat;Uid=root;Pwd=123456;", fechaCfdi); |

## Complemento Comercio Exterior

### Definir complemento Comercio Exterior

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07  08  09  10  11 | //=================>>>>>>>>>>> Inclusión de complemento CCE1.1  complCfdi.Cce11\_Define\_ComercioExterior(  "", //pMotivoTrasl  "2", //pTipoOperacion  "A1", //pClavePedim  0, //pCertifOrigen  null, //pNumCertifOrigen  null, //pNumExportadorConf  "CIF", //pIncoterm  0, //pSubdivision  null, //pObservaciones  17.473200m, //pTipoCambioUSD  242.11m); //pTotalUSD |

### Establecer Emisor de Comercio Exterior

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07  08  09  10  11  12 | complCfdi.Cce11\_setEmisor(  "", //pCurp  "1a. Privada de Oriente 17", //pDomCalle  "32", //pDomNumExt  null, //pDomNumInt  "2805", //pDomColonia  "12", //pDomLocalidad  "Entre Norte 2 y 4", //pDomReferencia  "118", //pDomMunicipio  "VER", //pDomEstado  "MEX", //pDomPais  "94300"); //pDomCP |

### Definir Receptor de Comercio Exterior

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07  08  09  10  11  12 | complCfdi.Cce11\_setReceptor(  "", //pCurp  "Domicilio Conocido", //pDomCalle  "", //pDomNumExt  null, //pDomNumInt  "", //pDomColonia  "", //pDomLocalidad  "", //pDomReferencia  "MPIO", //pDomMunicipio  "CA", //pDomEstado  "USA", //pDomPais  "99999"); //pDomCP |

### Agregar mercancía de Comercio Exterior

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07 | indexMerc = complCfdi.Cce11\_addMercancia(  "A-RTGT-56", //pNoIdentificacion  "7103910100", //pFraccionAranc  null, //pCantidadAduana  null, //pUnidadAduana  null, //pValorUnitAduana  242.11m); //pValorDolares |

### Generar el Complemento Comercio Exterior

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07  08  09  10 | /// generamos el complemento  ReturnXmlComplemento rtnXmlCompl = complCfdi.Cce11\_generaComplemento();  //verificamos que se haya creado.  if (rtnXmlCompl.IsXmlComplementoCreated) {  // le pasamos a al objeto CFDI33 el complemento y el namespace de comercio exterior  cfdi40.addComplementoComprobante(rtnXmlCompl.XMLComplemento);  cfdi40.addNamespace(complCfdi.NsComercioExterior1\_1);  }  // ahora para generar el comprobante, sellar y timbrar ver la clase CFDI40 |

Para generar el comprobante, sellar y timbrar ver la clase [CFDI40](#_Clase_CFDI33)

## Complemento Nómina

### Definir complemento Nómina

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07  08  09  10  11 | //complCfdi es de tipo Complemento  complCfdi.Nomina12\_DefineNomina("E", //pTipoNomina  new DateTime(2022, 12, 23), //pFechaPago  new DateTime(2022, 01, 26), //fechaInicPago  new DateTime(2022, 02, 20), //fechaFinPago  7, //numDiasPag  16744.39m, //totPercepciones  296.79m, //totDeducciones  10m); //totOtrosPagos |

### Agregar un emisor a Nómina

|  |  |
| --- | --- |
| 01  02  03  04  05 | complCfdi.Nomina12\_setEmisor(  null, //pCurp  "E52-19201-10-3", //pRegPatronal  null); //pRfcPatronOrigen |

### Agregar un receptor a Nómina

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07  08  09  10  11  12  13  14  15  16  17  18  19  20  21 | //establecemos el receptor  complCfdi.Nomina12\_setReceptor(  "BAPJ570702HSLLLN02", //pCurp  "23755735414", //pNumSegSocial  new DateTime(2013, 12, 09), //pFechaInicRelLab  "P164W", //pAntiguedad  "01", //pTipoContrato  Complemento.Nomina12\_pSiNoNoEspecificado.NoEspecificado, //esSindicalizado  "01", //pTipoJornada  "02", //pTipoRegimen  "49", //pNumEmpleado  "1", //pDepartamento  "ENC ALMACEN DE MAQUI", //pPuesto  "5", //pRiesgo  "99", //pPeriodiciadPago  "072", //pBanco  null, //pCuentaBanco  210.00m, //pSalarioBaseCotApor  231.52m, //pSalarioDiario  "SIN"); //pEstado |

### Establecer que Nómina tiene percerpciones

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07 | complCfdi.Nomina12\_setPercepciones(  5054.74m, //pTotalSueldos  70160.94m, //pTotalSeparacionIndemn  null, //pTotalJubilacionPensionRetiro  50212.88m, //pTotalGravado  25002.80m); //pTotalExento |

### Agregar una percepción a percepciones de Nómina

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07 | complCfdi.Nomina12\_addPercepcion(  "029", //pTipoPercepcion  "500100010004", //pClave  "Vales de Despensa", //pConcepto  0.00m, //pImporteGravado  462.00m); //pImporteExento |

### Agregar una Separación indemnización a percepciones de Nómina

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07 | complCfdi.Nomina12\_setSeparacionIndemnizacion(  70160.94m, //pTotalPagado  3, //pNumAñosServicio  6384m, //pUltimoSueldoMensOrd  6384.00m, //pIngresoAcumulable  63776.94m); //pIngresoNoAcumulable |

### Establer deducciones a Nómina

|  |  |
| --- | --- |
| 01  02  03 | complCfdi.Nomina12\_setDeducciones(  null, //pTotalOtrasDeducc  3781.05m); //pTotalImpuestosReten |

### Agregar una deducción a deducciones de Nómina

|  |  |
| --- | --- |
| 01  02  03  04  05  06 | complCfdi.Nomina12\_addDeduccion(  "002", //pTipoDeducc  "20010004", //pClave  "ISR", //pConcepto  3781.05m); //pImporte |

### Generar el Complemento Nómina

Una vez agregado al objeto CFDI40 ya se puede generar el XML, sellar y timbrar con los respectivos métodos de esta clase (CFDI40).

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07 | ReturnXmlComplemento rtnXmlCompl = complCfdi.Nomina12\_generaComplemento();  //verificamos que se haya creado el complemento y si es así lo agregamos al comprobante, tambien debemos agregar el namespace para nómina  if (rtnXmlCompl.IsXmlComplementoCreated) {  cfdi40.addComplementoComprobante(rtnXmlCompl.XMLComplemento);  cfdi40.addNamespace(complCfdi.NsNomina12);  } |

## Complemento Detallista

### Definir complemento Detallista

|  |  |
| --- | --- |
| 01  02  03  04 | //=================>>>>>>>>>>> Inclusión de complemento Detallista  complCfdi.Det131\_Define\_Detallista(detallistaDocumentStatus.ORIGINAL); //DocumentStatus |

### Establecer el requestForPaymentIdentification a Detallista

|  |  |
| --- | --- |
| 01  02  03  04  05 | complCfdi.Det131\_Agrega\_RequestForPaymentIdentification\_Detallista(  detallistaRequestForPaymentIdentificationEntityType.DEBIT\_NOTE //EntityType  ); |

### Agregar un specialInstruction a Detallista

|  |  |
| --- | --- |
| 01  02  03  04 | complCfdi.Det131\_AgregaSpecialInstruction(  detallistaSpecialInstructionCode.AAB, //Code  "TEXT"); //text |

### Establecer el orderIdentification a Detallista

|  |  |
| --- | --- |
| 01  02  03  04  05 | complCfdi.Det131\_SetOrderIdentification(new DateTime(2016, 02, 07), //referenceDate  "81283939"); //ReferenceIdentification |

### Agregar un referenceIdentification al AdditionalInformation de Detallista

|  |  |
| --- | --- |
| 01  02  03  04  05  06 | complCfdi.Det131\_AddReferenceIdentificationToAdditionalInformation(  detallistaReferenceIdentificationType.AAE, //tipo  "000001"); //valor |

### Establecer un DeliveryNote a Detallista

|  |  |
| --- | --- |
| 01  02  03  04  05 | complCfdi.Det131\_SetDeliveryNote(  new DateTime(2016, 04, 07), // reference date  "0000001" //referenceIdIndetification  ); |

### Definir el *Buyer* (comprador) de Detallista

|  |  |
| --- | --- |
| 01  02  03  04  05 | complCfdi.Det131\_SetBuyer(  "7504000107903", //gln  "0001" //personOrDeparmentName  ); |

### Definir el *Seller* (vendedor) a Detallista

|  |  |
| --- | --- |
| 01  02  03  04  05  06 | complCfdi.Det131\_SetSeller(  "7244020982223", //gnl  detallistaSellerAlternatePartyIdentificationType.IEPS\_REFERENCE,  "12382317" //alternatePartyIdentification  ); |

### Establer el shipTo (lugar de envío) de Detallista

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07 | complCfdi.Det131\_SetShipTo(  "8724098266218", //gln  "EMPRESA DEMO", //name  "Calle Peru num 029", //street address one  "Nuevo León", //city  "64590" //cp  ); |

### Establecer el InvoiceCreator a Detallista

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07  08  09 | complCfdi.Det131\_SetInvoiceCreator(  "5529910871321", //gln  detallistaInvoiceCreatorAlternatePartyIdentificationType.IA,  "1281282",  "EMPRESA DEMO", //name  "Calle Niños Heres num 1293",//street address one  "Nuevo León",//city  "64590");//cp |

### Agregar un Customs a Detallista

|  |  |
| --- | --- |
| 01  02  03 | complCfdi.Det131\_AddCustomsGln("0129829934772"); |

### Agregar un currency a Detallista

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07 | complCfdi.Det131\_addCurrency(detallistaCurrencyCurrencyISOCode.MXN,  detallistaCurrencyCurrencyFunction.PRICE\_CURRENCY,  null,  null,  17.10m  ); |

### Establecer el paymentTerms a Detallista

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07  08  09  10  11 | complCfdi.Det131\_SetPaymentTerms(detallistaPaymentTermsPaymentTermsEvent.DATE\_OF\_INVOICE,  detallistaPaymentTermsPaymentTermsRelationTime.REFERENCE\_AFTER,  detallistaPaymentTermsNetPaymentNetPaymentTermsType.BASIC\_NET,  detallistaPaymentTermsNetPaymentPaymentTimePeriodTimePeriodDueTimePeriod.DAYS, "10",  detallistaPaymentTermsDiscountPaymentDiscountType.ALLOWANCE\_BY\_PAYMENT\_ON\_TIME, "0.00"); |

### Establecer un allowanceCharge a Detallista para definir cargos o descuentos

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07 | complCfdi.Det131\_AddAllowanceCharge(  detallistaAllowanceChargeAllowanceChargeType.ALLOWANCE\_GLOBAL,  detallistaAllowanceChargeSettlementType.BILL\_BACK,  "12717262",// numConsec  null, //detallista servetype  1.00m  ); |

### Definir el totalAmount a Detallista

|  |  |
| --- | --- |
| 01  02  03 | complCfdi.Det131\_setTotalAmount(1.0m); |

### Definir el TotalAllowanceCharge a Detallista para definir el monto total de cargos o descuentos

|  |  |
| --- | --- |
| 01  02  03  04  05  06 | complCfdi.Det131\_AddTotalAllowanceCharge(detallistaTotalAllowanceChargeAllowanceOrChargeType.CHARGE,  detallistaTotalAllowanceChargeSpecialServicesType.ADS, 1.00m); |

### Generar el complemento Detallista

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07  08  09  10  11 | //generamos el complemento detallista  ReturnXmlComplemento rtnXmlCompl = complCfdi.Det131\_generaComplemento();  //verificamos que se creo  if (rtnXmlCompl.IsXmlComplementoCreated) {  //le pasamos el complemento generado a CFDI40 y agregamos su correspondiente NameSpace  cfdi40.addComplementoComprobante(rtnXmlCompl.XMLComplemento);  cfdi40.addNamespace(complCfdi.NsDetallista131);  } |

## Complemento Recepción de Pagos 2.0

### Definir el nodo Totales

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07  08  09  10  11  13  1415 | //=================>>>>>>>>>>> Inclusión de complemento Pagos 2.0  complCfdi.PAGOS20\_SET\_Pagos(null, //pTotalRetencionesIVA  null, //pTotalRetencionesISR  null, //pTotalRetencionesIEPS  1m, //pTotalTrasladosBaseIVA16  0.16m, //pTotalTrasladosImpuestoIVA16  null, //pTotalTrasladosBaseIVA8  null, //pTotalTrasladosImpuestoIVA8  null, //pTotalTrasladosBaseIVA0  null, //pTotalTrasladosImpuestoIVA0  null, //pTotalTrasladosBaseIVAExento  49153.09m); //pMontoTotalPagos |

### Agregar un Pago

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07  08  09  10  11  13  1415  16  17  18 | complCfdi.PAGOS20\_ADD\_Pagos\_Pago(  new DateTime(2020, 12, 15, 12, 0, 0), //pFechaPago  "03", //pFormaDePago  "MXN", //pMoneda  1m, //pTipoCambio  49153.09m, //pMonto  null, //pNumOperacion  "BSM970519DU8", //pRfcEmisorCtaOrd  "BANCO SANTANDER MEXICO SA", //pNomBancoOrdExt  "014628655019049888", //pCtaOrdenante  "BSM970519DU8", //pRfcEmisorCtaBen  "014650655057442596", //pCtaBeneficiario  null, //pTipoCadPago  null, //pCertPago  null, //pCadPago  null); //pSelloPago |

### Agregar un DoctoRelacionado al último Pago agregado

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07  08  09  10  11  12 | complCfdi.PAGOS20\_ADD\_Pagos\_Pago\_DoctoRelacionado\_UltimoPago(  "6AD9904B-4954-48C7-980C-FFAF9B09642B", //pIdDocumento  "A", //pSerie  "19225", //pFolio  "MXN", //pMonedaDR  1m, //pEquivalenciaDR  1, //pNumParcialidad  47703.41m, //pImpSaldoAnt  47703.41m, //pImpPagado  0.00m, //pImpSaldoInsoluto  "02"); //pObjetoImpDR |

### Agregar un Impuesto TrasladoDR / RetencionDR al último DoctoRelacionado agregado

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07  08 | complCfdi.PAGOS20\_ADD\_Pagos\_Pago\_DoctoRelacionado\_TrasladoDR\_UltimoDoctoRelacionado(  1m, //pBaseDR  "002", //pImpuestoDR  "Tasa", //pTipoFactorDR  0.160000m, //pTasaOCuotaDR  0.16m); //pImporteDR |

El código anterior agregará un nodo TrasladoDR al último nodo DoctoRelacionado. Para agregar un nodo RetencionDR, se debe usar el método PAGOS20\_ADD\_Pagos\_Pago\_DoctoRelacionado\_RetencionDR\_UltimoDoctoRelacionado con los mismos parámetros que se utilizan para los Traslados.

### Agregar un Impuesto TrasladoP al ultimo Pago agregado.

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07 | complCfdi.PAGOS20\_ADD\_Pagos\_Pago\_TrasladoP\_UltimoPago(1m, //pBaseP  "002", //pImpuestoP  "Tasa", //pTipoFactorP  0.160000m, //pTasaOCuotaP  0.16m); //pImporteP |

### Agregar un Impuesto RetencionP al ultimo Pago agregado.

|  |  |
| --- | --- |
| 01  02  03 | complCfdi.PAGOS20\_ADD\_Pagos\_Pago\_RetencionP\_UltimoPago("001", //pImpuestoP  0.16m); //ImporteP |

### Generar el Complemento Pago

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07  08  09  10  11 | //generamos el complemento pago  ReturnXmlComplemento rtnXmlCompl = complCfdi.Pagos20\_generaComplemento();  //verificamos que se haya creado.  if (rtnXmlCompl.IsXmlComplementoCreated) {  // agregamos el complemento al comprobante  cfdi40.addComplementoComprobante(rtnXmlCompl.XMLComplemento);  //agregamos el namespace del complemento pago al comprobante  cfdi40.addNamespace(complCfdi.NsPagos20);  } |

## Definir y generar el complemento IEDU (a un concepto)

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07  08  09  10  11  12  13  14  15  16 | //debido a que el complemento IEDU es un solo nodo, el método Iedu10\_generaComplemento define y genera el complemento a la vez.  ReturnXmlComplemento rtnXmlCompl = complCfdi.Iedu10\_generaComplemento(  "SOLIS HERRERA VALENTINA", //nombre del alumno  "SOHV031025MHGLRLA0", //CURP  instEducativasNivelEducativo.Primaria,  "13PPR0034A", //autRVOE  "AALE5706229F0" //rfcPago  );  //comprobamos que se creó el complemento  if (rtnXmlCompl.IsXmlComplementoCreated) {  //agregamos el complemento al concepto indicado por @indexConc  cfdi40.addComplementoConcepto(indexConc, rtnXmlCompl.XMLComplemento);  //agregamos el namespace de Iedu al comprobante  cfdi40.addNamespace(complCfdi.NsIedu10);  } |

## Definir y generar el complemento Donatarias 1.1

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07  08  09  10  11  12  13  14 | //debido a que el complemento IEDU es un solo nodo, el método Donat11\_generaComplemento define y genera el complemento a la vez.  string leyenda = "Este comprobante ampara un donativo…";  ReturnXmlComplemento rtnXmlCompl = complCfdi.Donat11\_generaComplemento("325-SAT-28-I-(21)-4109", new DateTime(2017, 1, 9), leyenda);  //comprobamos que se creó el complemento  if (rtnXmlCompl.IsXmlComplementoCreated) {  //agregamos el complemento al comprobante  cfdi40.addComplementoComprobante(rtnXmlCompl.XMLComplemento);  //agregamos el namespace de Iedu al comprobante  cfdi40.addNamespace(complCfdi.NsDonat11);  } |

## Complemento Carta Porte 2.0

### Definir el complemento Carta Porte

|  |  |
| --- | --- |
| 01  02  03  04  05 | complCfdi.CP20\_SET\_CartaPorte("No", //TranspInternac  null, //EntradaSalidaMerc  null, //PaisOrigenDestino  null, //ViaEntradaSalida  52.61m); //TotalDistRec |

### Agregar una Ubicación

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07  08  09  10  11  12  13 | int indx = complCfdi.CP20\_ADD\_CartaPorte\_Ubicacion("Origen", //TipoUbicacion  "OR000001", //IDUbicacion  "PASF690721EL6", //RFCRemitenteDestinatario  "", //NombreRemitenteDestinatario  "", //NumRegIdTrib  "", //ResidenciaFiscal  "", //NumEstacion  "", //NombreEstacion  "", //NavegacionTrafico  new DateTime(2021, 12, 01, 19, 17, 52), //FechaHoraSalidaLlegada  "", //TipoEstacion  null); //DistanciaRecorrida |

### Establecer el Domicilio para una ubicación

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07  08  09  10  11  12  13 | complCfdi.CP20\_SET\_CartaPorte\_Ubicacion\_Domicilio(indx, //indexUbicacion  new CartaPorte10\_Domicilio  {  Calle = "ORQUIDEAS 6320",  Colonia = "0491",  Localidad = "03",  Municipio = "014",  Estado = "CMX",  Pais = "MEX",  CodigoPostal = "03010"  }); |

### Establecer información de Mercancias

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07 | complCfdi.CP20\_SET\_CartaPorte\_Mercancias(17000m, //PesoBrutoTotal  "KGM", //UnidadPeso  null, //PesoNetoTotal  1, //NumTotalMercancias  null); //CargoPorTasacion |

### Agregar una Mercancía

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07  08  09  10  11  12  13  14  15  16  17  18 | complCfdi.CP20\_ADD\_CartaPorte\_Mercancias\_Mercancia(  "12181503", //BienesTransp  null, //ClaveSTCC  "BULTOS DE PARAFINA 50 KG", //Descripcion  340m, //Cantidad  "XGY", //ClaveUnidad  "", //Unidad  null, //Dimensiones  null, //MaterialPeligroso  null, //CveMaterialPeligroso  null, //Embalaje  null, //DescripEmbalaje  17000m, //PesoEnKg  null, //ValorMercancia  "", //Moneda  null, //FraccionArancelaria  null); //UUIDComercioExt |

### Agregar CantidadTransporta a la última Mercancía

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07 | complCfdi.CP20\_ADD\_CartaPorte\_Mercancias\_Mercancia\_CantidadTransporta\_UltimaMercancia(  340m, //Cantidad  "OR000001", //IDOrigen  "DE000001", //IDDestino  ""); //CvesTransporte |

### Establecer elemento Autotransporte para Mercancias

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07 | complCfdi.CP20\_SET\_CartaPorte\_Mercancias\_Autotransporte(  "TPAF01", //PermSCT  "2141TAL31052011021001063", //NumPermisoSCT  "VL", //ConfigVehicular  "465AU2", //PlacaVM  2013); //AnioModeloVM |

### Establecer información de Seguros para el Autotransporte

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07  08  09 | complCfdi.CP20\_SET\_CartaPorte\_Mercancias\_Autotransporte\_Seguros(  "SEGUROS ATLAS S.A", //AseguraRespCivil  "P00-1-7-118240-18", //PolizaRespCivil  "", //AseguraMedAmbiente  "", //PolizaMedAmbiente  "", //AseguraCarga  "", //PolizaCarga  null); //PrimaSeguro |

### Agregar un Tipo de Figura para FiguraTransporte

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07  08  09  10  11  12  13  14  15  16  17  18  19  20  21 | complCfdi.CP20\_ADD\_CartaPorte\_TiposFigura(  "01", //TipoFigura  "AAMI760614929", //RFCFigura  "PUE0036473", //NumLicencia  "", //NombreFigura  "", //NumRegIdTribFigura  null, //ResidenciaFiscalFigura  new CartaPorte10\_Domicilio  {  Calle = "",  NumeroExterior = "",  NumeroInterior = "",  Colonia = "",  Localidad = "",  Referencia = "",  Municipio = "",  Estado = "CMX",  Pais = "MEX",  CodigoPostal = "01985"  }); |

### Generar el Complemento Carta Porte 2.0

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07  08 | ReturnXmlComplemento rtnXmlCompl = complCfdi.CartaPorte20\_generaComplemento();  //comprobamos que se creó el complemento  if (rtnXmlCompl.IsXmlComplementoCreated) {  //agregamos el complemento al comprobante  cfdi40. addComplementoComprobante(rtnXmlCompl.XMLComplemento);  //agregamos el namespace de Impuestos Locales al comprobante  cfdi40.addNamespace(complCfdi.NsCartaPorte20);  } |

## Complemento Impuestos Locales

### Definir el complemento Impuestos Locales

|  |  |
| --- | --- |
| 01  02  03 | complCfdi.ImpLocal10\_Define\_ImpuestoLocal(  0.17m, //total traslado  0.00m); //total retencion |

### Agregar Impuesto Local Trasladado al complemento

|  |  |
| --- | --- |
| 01  02  03  04 | complCfdi.ImpLocal10\_Agrega\_TrasladoImpuestoLocal(  "ISH", //impuesto  0.17m, //importe  2.00m); //tasa |

### Generar el Complemento Impuestos Locales

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07  08 | ReturnXmlComplemento rtnXmlCompl = complCfdi. ImpLocal10\_generaComplemento ();  //comprobamos que se creó el complemento  if (rtnXmlCompl.IsXmlComplementoCreated) {  //agregamos el complemento al comprobante  cfdi40. addComplementoComprobante(rtnXmlCompl.XMLComplemento);  //agregamos el namespace de Impuestos Locales al comprobante  cfdi40.addNamespace(complCfdi. NsImpLocal10);  } |

# Proceso de Cancelación

## Cancelar un CFDI 2018 (actualización)

Este método será eliminado en futura versión, por lo que se recomienda utilizar en su lugar el método cancelaCfdisNuevoProceso()

Este método permite iniciar la cancelación de uno o más CFDIs mediante el nuevo esquema de cancelación, para usarlo se debe configurar su respectivo “Endpoint”, por lo que si ya se utilizaba la dll con anterioridad y no se ha implementado la llamada a este método se deberá de consumir este servicio mediante esta clase [Cancelacion2018](#_Clase_ReturnCancelacion2018) (Nota ver [sección configuración](#_Agregar_información_del) para agregar su respectivo endpoint address). Para mayor detalle ver [Referencias](#_Referencias).

La lista de UUIDs a cancelar debe conformarse con un string para cada UUID a cancelar con el formato

|UUID a cancelar|MotivoCancelacion|UUID que sustituye|, en caso de que no sea necesario incluir el UUID que sustituye, se deberá dejar el espacio en blanco conservando los caracteres pipe especificados.

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07  08  09  10  1112  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  32  33  34  35  36  37  38  39  40  41  42  43  44  45 | Cancelacion2018 cancelacion2018 = new Cancelacion2018(  "JGE080430LN8", //Usuario Sifei  "0e84ec92", //Password Sifei  CFDI33.tipoTimbrado.Pruebas); //Ambiente  string msg = "";  //Lista de uuids del comprobante a cancelar  List<string> uuids = new List<string>();  uuids.Add("|XXXXXX-447C-4F10-BCD7-D1949CDF636CXXX|02||");  try {  ReturnCancelacion2018 returnCancelacion2018 = cancelacion2018.cancelaCfdis(ref msg,  uuids.ToArray(), //arreglo de Uuid  "MAR980114GQA", //rfcEmisor  CERTIFICADO\_PATH, //rutaCert  CERTIFICADOKEY\_PATH, //rutaKey  "12345678a"); //pass del certificado  if (!returnCancelacion2018.Procesado) {  Console.Write(returnCancelacion2018.CodigoError);  MessageBox.Show("Error al cancelar: " + returnCancelacion2018.CodigoError + "detalle" + returnCancelacion2018.ErrorDetalle);  }  else {  //acuse detalle es una lista de objetos AcuseDetalle, uno por cada uuid mandado a cancelar  foreach (var acuse in returnCancelacion2018.AcuseBE.AcuseDetalle) {  //codigo 201 indica que se realizó exitosamente ya sea para aprobación por parte del receptor o si el comprobante se canceló sin autorización  if (acuse.Status == "201") {  //se recomienda verificar con el metodo consulta CFDI  }  }  XmlDocument xml = new XmlDocument();  xml.LoadXml(returnCancelacion2018.AcuseBE.AcuseXml);    xml.Save(WORK\_DIR + "\\AcuseCancelacion" + DateTime.Now.ToString("yyyy-MM-ddTHH.mm.ss") + ".xml");  MessageBox.Show("ok: " + returnCancelacion2018.AcuseBE.AcuseXml);  }  }  catch (Exception ex) {  MessageBox.Show("Error al cancelar: " + ex.Message);  } |

Para mayor detalle ir a la sección 9 Referencia y a la sección “[Códigos de respuesta](#_Codigos_de_respuesta)”.

## Cancelar CFDIs Nuevo Proceso

Este método sustituye al método cancelaCfdis()

Este método realiza la cancelación de los CFDIs especificados, validando inicialmente el estatus del comprobante, sólo aquellos comprobantes que se encuentren en *Estado* Vigente, el valor de *EsCancelable* sea Cancelable (sin aceptación o con aceptación) y que su *EstatusCancelación* no sea En proceso, podrán ser procesados. Los UUIDs que no caigan en dichos supuestos se listarán en la respuesta del método, en el atributo *UuidsNoProcesados*.

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07  08  09  10  1112  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  32  33  34  35  36 | Cancelacion2018 canc2018 = new Cancelacion2018(this.usuarioSifei, this.pwdSifei, this.tipoTimb);                List<estructuraConsultaQr> structUuids = new List<estructuraConsultaQr>();              estructuraConsultaQr uuid = new estructuraConsultaQr();              uuid.uuid = "67CC490A-1AE5-4D07-BCBB-542E17F8CF9D";              uuid.rfcReceptor = "TCM970625MB1";              uuid.total = 1334.09m;              uuid.selloCfd8 = "tBcd7w==";              structUuids.Add(uuid);              uuid = new estructuraConsultaQr();              uuid.uuid = "EA9BCCE8-3617-46E2-9FE0-C8EB0BBC25B9";              uuid.total = 1334.09m;              uuid.rfcReceptor = "TCM970625MB1";              uuid.selloCfd8 = "N4tOTQ==";              uuid.motivoCancelacion = "02";  uuid.uuidSustituye = "";              structUuids.Add(uuid);                string msg = "";              ReturnCancelacion2018 result = canc2018.cancelaCfdisNuevoProceso(ref msg, structUuids.ToArray(), "S&S051221SE2",  @" CSD\_S\_&\_SOFTWARE\_S.A\_DE\_C.V\_S&S051221SE2\_20190627\_135134s.cer",  @" CSD\_S\_&\_SOFTWARE\_S.A\_DE\_C.V\_S&S051221SE2\_20190627\_135134.key",  "12345678a");              if (result.Procesado)              {                  XmlDocument xmlAc = new XmlDocument();                  xmlAc.LoadXml(result.AcuseBE.AcuseXml);                  xmlAc.Save(@"acuseCanc.xml");              }              else              {                  MessageBox.Show("Error al procesar cancelación: " + result.CodigoError + " - " + result.ErrorDetalle);                  return;              } |

Es importante hacer notar que aunque la respuesta en su atributo ***Procesado*** sea ***true***, se debe revisar el valor del atributo ***UuidsNoProcesados***, ya que el proceso de cancelación realiza el proceso de cancelación del al menos 1 UUID, por ejemplo si envía a cancelar 5 UUIDs pero 3 de ellos son No Cancelables, el sistema procesa la cancelación de los 2 restantes.

## Consulta CFDI relacionados

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07  08  09  10  1112  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57 | //instanciando clase cancelacion 2018 que permite consumir el metodo Consulta CFDI relacionados  Cancelacion2018 cancelacion2018 = new Cancelacion2018(  "JGE080430LN8", //Usuario Sifei  "0e84ec92", //Password Sifei  CFDI33.tipoTimbrado.Pruebas); //Ambiente  string msg = "";  //UUID del CFDI a consultar, para saber si tiene CFDI relacionados y cuales son.  string uuidCfdi = INEXISTENTE\_UUID;    try {  ReturnDocumentosRelacionados returnCfdiRelacionados =  cancelacion2018.consultaDocumentosRelacionados(  Cancelacion2018.tipoConsultaRelacionados.Emisor, //Consultar por (Emisor, Receptor)  "LAN7008173R5", //Emisor  "TCM970625MB1", //Receptor  CERTIFICADO\_PATH,  CERTIFICADO\_KEY,  "12345678a", //pass del certificado  uuidCfdi, //uuid del CFDI a consultar  ref msg);  if (!returnCfdiRelacionados.Procesado) {  Console.Write(returnCfdiRelacionados.CodigoError);  }  else {  //en caso de exito (procesado = true)  //se puede saber si tiene CFDI hijos o padres, el cual contiene UUID, RFC Emisor y RFC Receptor  StringBuilder sb = new StringBuilder();  sb.AppendLine(string.Format("Numero de CFDI hijos:{0}", returnCfdiRelacionados.Result.UuidsRelacionadosHijos.Length));  foreach (var padre in returnCfdiRelacionados.Result.UuidsRelacionadosHijos) {  sb.AppendLine( string.Format("UUID:{0}, RFC Emisor:{1}, RFC Receptor:{2}", padre.Uuid, padre.RfcEmisor, padre.RfcReceptor));  }  sb.AppendLine(string.Format("Numero de CFDI Padre:{0}", returnCfdiRelacionados.Result.UuidsRelacionadosHijos.Length));    foreach (var padre in returnCfdiRelacionados.Result.UuidsRelacionadosPadres) {  sb.AppendLine( string.Format("UUID:{0}, RFC Emisor:{1}, RFC Receptor:{2}",padre.Uuid,padre.RfcEmisor,padre.RfcReceptor));  }  //guardando el acuse  XmlDocument xml = new XmlDocument();  xml.LoadXml(returnCfdiRelacionados.AcuseXml);  xml.Save(WORK\_DIR+"\\AcuseCFDIRelacionados" + DateTime.Now.ToString("yyyy-MM-ddTHH.mm.ss") + ".xml");  MessageBox.Show(sb.ToString());  }  }  catch (Exception ex) {  MessageBox.Show("Error al cancelar: " + ex.Message);  } |

Para mayor detalle referirse a la sección 9.17 [códigos de error de CFDI Relacionados.](#_Codigos_de_Error)

## Consulta CFDI Estatus

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07  08  09  10  1112  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  32  33  34  35  36  37  38 | Cancelacion2018 cancelacion2018 = new Cancelacion2018(  "JGE080430LN8", //Usuario Sifei  "0e84ec92", //Password Sifei  CFDI33.tipoTimbrado.Pruebas); //Ambiente  try {  // se consulta el metodo  ReturnQrAcuse returnQr = cancelacion2018.consultaEstatusCfdi(  uuid, //UUID  rfcEmisor, // RFC Emisor  rfcReceptor, //Rfc Receptor  total, // total  sello // sello ultimos 8 digitos  );  //Cancelable con aceptación|Cancelable sin aceptación  //S - Comprobante obtenido satisfactoriamente|  if (returnQr.Procesado) {  //Cancelable sin aceptación|  if (returnQr.EsCancelable == "Cancelable sin aceptación") {  }  if (returnQr.Estado == "Cancelado") { //"Vigente", "Cancelado","No Encontrado"  //Cancelado sin aceptación  Console.WriteLine("", returnQr.EstatusCancelacion);  }  XmlDocument xml = new XmlDocument();  xml.LoadXml(returnQr.AcuseXml);  xml.Save(WORK\_DIR + "\\AcuseConsultaEstatus" + DateTime.Now.ToString("yyyy-MM-ddTHH.mm.ss") + ".xml");  MessageBox.Show(returnQr.AcuseXml);  }  }  catch (Exception ex) {  MessageBox.Show(string.Format("Error al consultar CFDIS {0}: ", ex.Message));  } |

## Consulta peticiones de Cancelación pendientes (que requieren autorización).

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07  08  09  10  1112  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53 | /\*En el rol de rfc receptor,con este metodo se obtienen las peticiones pendientes de Cancelacion que requieren la autorizacion  \* del RFC como Receptor (aceptacion) para aprobar o rechazar la cancelacion de un CFDI.  \* \*/  Cancelacion2018 cancelacion2018 = new Cancelacion2018(  "JGE080430LN8", //Usuario Sifei  "0e84ec92", //Password Sifei  CFDI33.tipoTimbrado.Pruebas); //Ambiente  string msg = "";  try {  //se obtiene las peticiones pendientes de cancelacion al RFC Receptor:  ReturnPeticionesPendientes returnPeticionesPendientes = cancelacion2018.getPeticionesCancelacionPendientes(  "MAR980114GQA" //RFC receptor quien aprueba o rechaza la cancelacion del CFDI  );  //se verifica si fue procesado es false lo que indicaria que hubo un error a nivel de conxion, o excepcion a nivel de WebService)  if (!returnPeticionesPendientes.Procesado) {  MessageBox.Show("Codigo de error:"+returnPeticionesPendientes.CodigoError+"Error detalle: "+returnPeticionesPendientes.ErrorDetalle);  }  //en caso de exito  else {  PrepareListViewForPeticionesPendientes();  uuidPeticionesPendientes.Clear();  // 1100 Se obtuvo petición del RFC receptor de forma exitosa  if (returnPeticionesPendientes.CodEstatus == "1100") {  //si el codigo es 1100, entonces recorrer el UUID de cada peticion.  foreach (var uuid in returnPeticionesPendientes.UUID) {  //en este ejemplo se usa un ListView para mostrar el UUId, pero es posible que se desee guardar a BD como registro  listView2.Items.Add(new ListViewItem(uuid));  uuidPeticionesPendientes.Add(uuid);  }  //guardadndo el acuse a archivo(ejemplo)  XmlDocument xml = new XmlDocument();  xml.LoadXml(returnPeticionesPendientes.AcuseXml);  xml.Save(WORK\_DIR+"\\AcusePeticionesPendientes\_"+DateTime.Now.ToString("yyyy-MM-ddTHH.mm.ss")+".xml");  }  //1101 no existen peticiones pendientes  else if (returnPeticionesPendientes.CodEstatus == "1101") {  Console.WriteLine("No hay peticiones pendientes");  }  }  }  catch (Exception ex) {  MessageBox.Show("Error al obtener peticiones Pendinetes de Cancelacion: " + ex.Message);  } |

Para mayor detalle referirse a la sección códigos de error consulta [Peticiones de Cancelacion pendientes](#_Codigos_de_error_1).

## Procesar Respuesta de Peticion de Cancelacion (Aceptacion o Rechazo de cancelación)

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07  08  09  10  1112  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55 | Cancelacion2018 cancelacion2018 = new Cancelacion2018(  "JGE080430LN8", //Usuario Sifei  "0e84ec92", //Password Sifei  CFDI33.tipoTimbrado.Pruebas); //Ambiente  //lista de objeto folios requerido por el servicio, incluye el UUID y la respuesta:aceptacion o rechazo  List<folios> lstFolios = new List<folios>();  //Se establecen los UUID de las peticiones  List<string> uuidSeleccionados= getUUIDSeleccionados();  foreach (string uuid in uuidSeleccionados) {  //se crea objeto clase folio , que requiere el UUID y la respuesta  folios f = new folios();  //se asigna el UUID de la peticion a la que se le dara respuesta  f.uuid = uuid;  //se asgina la respuesta, que es donde se indica si se acepta o rechaza  f.respuesta = respuestaParaUUIDMarcados;  //se indica que el valor respuesta no es uno por defecto,sino que de lado del cliente, realmente se esta estebleciendo este valor  f.respuestaSpecified = true;  lstFolios.Add(f);//agregamos la respuesta  }  string msg="";  try {  // con este metodo, como receptor se acepta o rechaza  ReturnAceptacionRechazoResult returnAceptacionRechazoResult = cancelacion2018.procesaRespuestaPeticionesCancelacion(  "MAR980114GQA",  CERTIFICADO\_PATH,  CERTIFICADO\_KEY\_PATH,  "12345678a", //pass del certificado  ref msg,  lstFolios.ToArray());  if (!returnAceptacionRechazoResult.Procesado) {  MessageBox.Show("Codigo de error:" + returnAceptacionRechazoResult.CodigoError + "Error detalle: " + returnAceptacionRechazoResult.ErrorDetalle);  }  //en caso de ok  if (returnAceptacionRechazoResult.Procesado) {  if (returnAceptacionRechazoResult.CodEstatus == "1000") {  Console.WriteLine(returnAceptacionRechazoResult.AcuseXML);  }  XmlDocument xml = new XmlDocument();  xml.LoadXml(returnAceptacionRechazoResult.AcuseXML);  xml.Save(WORK\_DIR + "\\AcuseProcesaRespuestaAceptacionRechazo\_" + DateTime.Now.ToString("yyyy-MM-ddTHH.mm.ss") + ".xml");  }  }  catch (Exception ex) {  MessageBox.Show("Error al procesar respuesta: " + ex.Message);  } |

Para mayor detalle referirse a la sección [códigos de error del metodo Aceptacion/Rechazo de Cancelacion](#_Codigos_de_error_2).

# Comprobante de Retención de Información de Pagos

## Instanciar un objeto de Comprobante de Retención e Información de Pagos

La clase retención es la clase principal para generar Comprobantes de Retención e Información de Pagos, para poder utilizar esta clase, se debe pasar como parámetro la cadena de conexión a la base de datos Catálogossat.

|  |  |
| --- | --- |
| 01  02  03 | retencion ret = new retencion( “Server=localhost;Port=3306;Database=catálogossat;Uid=root;pwd=123456;”,  DateTime.Now); |

## Definir la retención

Haciendo uso de la clase retencion que se inicializó, se invoca a los métodos necesarios para generar el comprobante. El primer paso es definir los parámetros generarles del comprobante utilizando el método Reten20\_Define\_Retenciones():

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07  08 | ret.Reten20\_Define\_Retenciones(  this.rutaCert, //pRutaCert  "BZNPLUS339", //pFolioInt  DateTime.Now, //pFechaExp  "70780", //pLugarExpRetenc  "26", //pCveRetenc  "" //pDescRetenc  ); |

Lo cual produce un XML con lo siguiente:

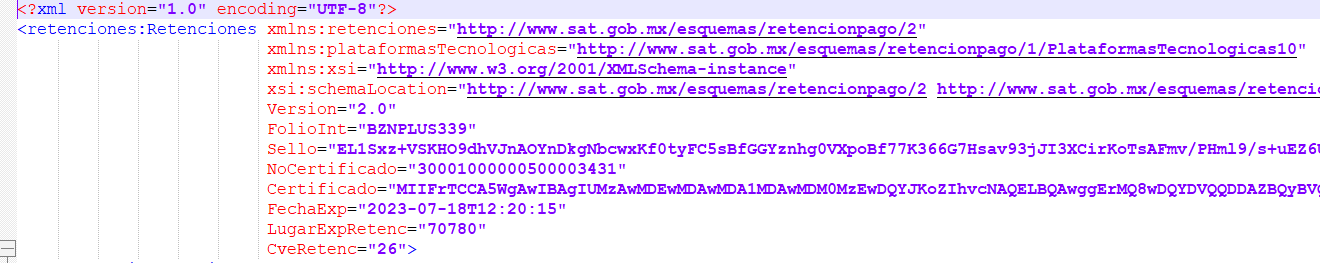


Imagen Atributos de Retenciones agregados

## Definir CfdiRetenRelacionados

Para establecer los datos del comprobante relacionado se debe utilizar el método Reten20\_Define\_CfdiRetenRelacionados().

|  |  |
| --- | --- |
| 01  02  03  04 | ret.Reten20\_Define\_CfdiRetenRelacionados("04", //pTipoRelacion  "AD61BF2E-2D20-478A-9E59-9AE0C9EC2301"); //pUUID |

El nodo generado es Emisor:

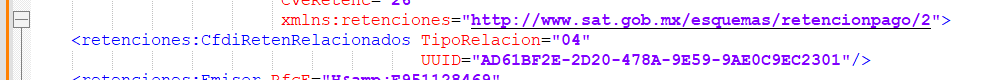


Imagen Nodo CfdiRetenRelacionados agregado

## Definir el Emisor

Para establecer los datos del Emisor del comprobante se debe utilizar el método Reten20\_Set\_Emisor().

|  |  |
| --- | --- |
| 01  02  03  04 | ret.Reten20\_Set\_Emisor(this.rfcEmisor, //pRfcE  "HERRERIA & ELECTRICOS SA DE CV", //pNomDenRazSocE  "601" //pRegimenFiscalE  ); |

El nodo generado es Emisor:

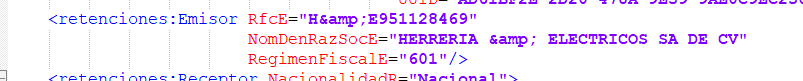


Imagen Nodo Emisor agregado

## Definir Receptor

Para establecer los datos de Receptor, se puede utilizar el método Reten20\_Set\_ReceptorNacional() o Reten20\_Set\_ReceptorExtranjero() según sea el caso, en el siguiente ejemplo se utiliza el primer método:

|  |  |
| --- | --- |
| 01  02  03  04 | ret.Reten20\_Set\_ReceptorNacional("CACX7605101P8", //pRfcR  "Pruebas", //pNomDenRazSocR  "", //pCurpR  "94310" //pDomicilioFiscalR  ); |

El nodo generado es Receptor:

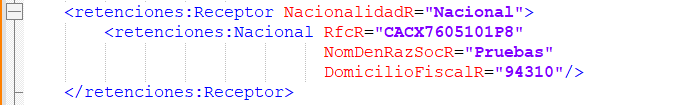


Imagen Nodo Receptor agregado

## Definir el Periodo

Para establecer el periodo del comprobante se debe utilizar el método Reten20\_Set\_Periodo():

|  |  |
| --- | --- |
| 01  02  03  04 | ret.Reten20\_Set\_Periodo("01", //pMesIni  "01", //pMesFin  "2023"); //pEjercicio |

El nodo generado es Periodo:



Imagen Nodo Periodo agregado

## Definir Totales

Para establecer los totales del comprobante de Retenciones e Información de Pagos, se debe utilizar el método Reten20\_Set\_Totales():

|  |  |
| --- | --- |
| 01  02  03  04 | ret.Reten20\_Set\_Totales(1681.06m, //pMontoTotOperacion  1681.06m, //pMontoTotGrav  0.00m, //pMontoTotExent  151.29m, //pMontoTotRet  null, //pUtilidadBimestral  null); //pISRCorrespondiente |

Lo cual genera el siguiente nodo Totales:



Imagen Nodo Totales agregado

## Agregar Impuesto Retenido

Para agregar un impuesto retenido se debe utilizar el método Reten20\_Add\_ImpRetenidos ():

|  |  |
| --- | --- |
| 01  02  03  04 | ret.Reten20\_Add\_ImpRetenidos(1681.06m, //pBaseRet  "001", //pImpuestoRet  16.81m, //pMontoRet  "03"); //pTipoPagoRet |

Lo cual genera el nodo ImpRetenidos:



Imagen Nodo ImpRetenidos agregado

## Autorización para el uso de Sifei Signature

La clase retencion incluye el método setIdentificacionSifei() que debe invocarse antes de invocar a los métodos generaXmlRetencion20() y sellaRetencion20(), referirse a la misma sección para CFDI 3.3.

## Generar el XML

Para generar el XmlDocument debemos invocar al método generaXmlRetencion20(), el objeto XmlDocument no es retornado de forma directa, sino que es devuelto en un objeto *Wrapper* de tipo [ReturnXML](#_Clase_ReturnXML), este objeto nos permite averiguar si se generó el XML o no. (Para más información ver la referencia [ReturnXML](#_Clase_ReturnXML)).

Para utilizar este método antes debió invocar al método setIdentificacionSifei().

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07  08  09  10  11  12 | //generamos el XML, el objeto de tipo ReturnXML, contiene propiedades que indican si se creó el XML, y si es así en su propiedad XML, el objeto XmlDocument  ReturnXML result = this.generaXmlRetencion20();  //comprobamos que se ha creado con la propiedad IsXMLCreated  if (rtnRet.IsXMLCreated) {  // si se creó, obtenemos de la propiedad XML para guardarlo rtnRet.XML.Save(@"C:\Workspace\SifeiSignature\RetGeneradosTest\ret\_creado.xml");  if (!File.Exists(@"C:\Workspace\SifeiSignature\RetGeneradosTest\ret\_creado.xml")) {  MessageBox.Show("No se escribió el archivo");  }  } |

El XML generado sería parecido al siguiente:

|  |
| --- |
| <?xml version=**"1.0"** encoding=**"utf-8"**?>  <retenciones:Retenciones xmlns:plataformasTecnologicas=**"http://www.sat.gob.mx/esquemas/retencionpago/1/PlataformasTecnologicas10"**  xmlns:xsi=**"http://www.w3.org/2001/XMLSchema-instance"**  xsi:schemaLocation=**"http://www.sat.gob.mx/esquemas/retencionpago/2 http://www.sat.gob.mx/esquemas/retencionpago/2/retencionpagov2.xsd http://www.sat.gob.mx/esquemas/retencionpago/1/PlataformasTecnologicas10 http://www.sat.gob.mx/esquemas/retencionpago/1/PlataformasTecnologicas10/ServiciosPlataformasTecnologicas10.xsd"**  Version=**"2.0"**  FolioInt=**"BZNPLUS339"**  Sello=**"@"**  NoCertificado=**"30001000000500003431"**  Certificado=**"MIIFrTCCA5WgA..."**  FechaExp=**"2023-07-18T13:45:39"**  LugarExpRetenc=**"70780"**  CveRetenc=**"26"**  xmlns:retenciones=**"http://www.sat.gob.mx/esquemas/retencionpago/2"**>  <retenciones:CfdiRetenRelacionados TipoRelacion=**"04"**  UUID=**"AD61BF2E-2D20-478A-9E59-9AE0C9EC2301"**/>  <retenciones:Emisor RfcE=**"H&amp;E951128469"**  NomDenRazSocE=**"HERRERIA &amp; ELECTRICOS SA DE CV"**  RegimenFiscalE=**"601"**/>  <retenciones:Receptor NacionalidadR=**"Nacional"**>  <retenciones:Nacional RfcR=**"CACX7605101P8"**  NomDenRazSocR=**"Pruebas"**  DomicilioFiscalR=**"94310"**/>  </retenciones:Receptor>  <retenciones:Periodo MesIni=**"01"**  MesFin=**"01"**  Ejercicio=**"2023"**/>  <retenciones:Totales MontoTotOperacion=**"1681.06"**  MontoTotGrav=**"1681.06"**  MontoTotExent=**"0.00"**  MontoTotRet=**"151.29"**>  <retenciones:ImpRetenidos BaseRet=**"1681.06"**  ImpuestoRet=**"001"**  MontoRet=**"16.81"**  TipoPagoRet=**"03"**/>  <retenciones:ImpRetenidos BaseRet=**"268.96"**  ImpuestoRet=**"002"**  MontoRet=**"134.48"**  TipoPagoRet=**"01"**/>  </retenciones:Totales>  </retenciones:Retenciones> |

## Sellar el Comprobante de Retención e Información de Pagos

Es posible sellar un Comprobante de Retención e Información de Pagos a través del método *sellaRetencion20()* el *cual retorna un objeto tipo* [ReturnXML](#_Clase_ReturnXML), mismo que nos sirvepara saber si el comprobante fue sellado y recuperarlo, el código siguiente muestra como sellar el Comprobante de Retención.

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07  08  09  10  11  12  13  14  15 | //sellamos el comprobante pasándole el path del certificado y de la llave  ReturnXML result= ret. sellaRetencion20(result.XML,  this.rutaCert,  this.rutaKey,  this.pwdKey,  true,  true);  //comprobamos que se halla sellado  if (result.EstaSellado) {  //escribimos el Comprobante sellado:  result.XML.Save(@"C:\Workspace\SifeiSignature\RetGeneradosTest\ret\_sellado.xml");  } |

Comprobante de retenciones sellado atributo sello (línea 6):

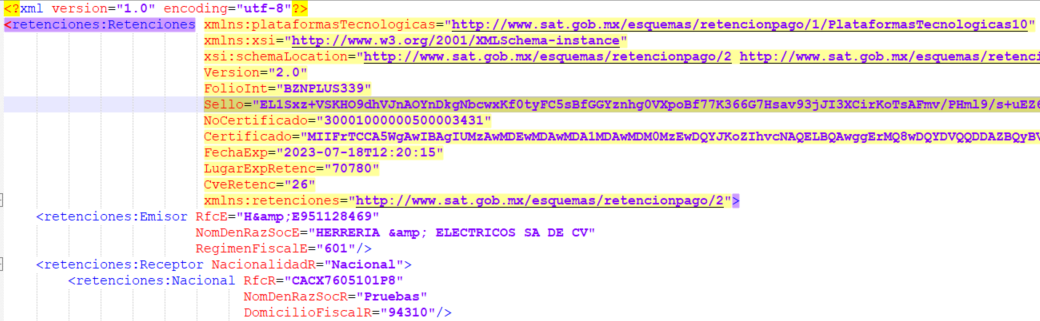


Imagen Comprobante Sellado

Para utilizar este método antes debió invocar al método setIdentificacionSifei().

## Timbrar el Comprobante de Retenciones e Información de Pagos

Es posible timbrar un Comprobante de Retenciones e Información de Pagos a través del método timbraRetencion()el cual retorna un objeto tipo[ReturnTimbre](#_Clase_ReturnTimbre), mismo que sirvepara saber si el comprobante fue timbrado y recuperarlo, tal y como se ve en el siguiente ejemplo:

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07  08  09  10  11 | //timbramos el comprobante  ReturnTimbre rtnTimbre = ret.timbraRetencion(this.serieSifei,  this.idEquipo,  result.XML);  //verificamos que haya sido timbrado:  if (rtnTimbre.IsTimbrado) {  //escribimos el cfdi timbrado rtnTimbre.Xml.Save(@"C:\Workspace\SifeiSignature\ret\_timbrado.xml"); |

Para utilizar este método antes debió invocar al método setIdentificacionSifei().Es importante mencionar que este método ya no solicita el usuario y contraseña de timbrado, ni el ambiente en el que efectuará el timbrado (Pruebas o Producción), ya que estos son tomados de los valores que se le asignaron al invocar el método setIdentificacionSifei().

Al guardar el Comprobante de Retenciones contendrá el nodo TimbreFiscalDigital (líneas 12 a 19):



Imagen Comprobante timbrado

# Complementos de Comprobante de Retenciones e Información de Pagos

Con la misma clase retencion es posible crear complementos del Comprobante de retención e información de pagos. Actualmente el único complemento disponible es Servicios Plataformas Tecnológicas.

## Complemento Servicios Plataformas Tecnológicas

### Definir Complemento Servicios Plataformas Tecnológicas

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07  08  09  10  11  12 | //Complemento servicios plataformas tecnologicas  ret.Plat\_Define\_ServiciosPlataformasTecnologicas(  "02", //pPeriodicidad  1, //pNumServ  1681.06m, //pMonTotServSIVA  268.9696m, //pTotalIVATrasladado  134.48m, //pTotalIVARetenido  16.81m, //pTotalISRRetenido  134.4896m, //pDifIVAEntregadoPrestServ  14.66m, //pMonTotalporUsoPlataforma  null //pMonTotalContribucionGubernamental  ); |

### Agregar DetallesDelServicio

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07  08  09  10  11  12  13  14  15  16  17  18  19 | ret.Plat\_Add\_DetallesDelServicio(  "02", //pFormaPagoServ  "05", //pTipoDeServ  "", //pSubTipServ  "", //pRFCTerceroAutorizado  new DateTime(2023, 1, 6), //pFechaServ  1681.06m, //pPrecioServSinIVA  true, //integraImpTrasServ  1681.06m, //pImpTras\_Base  "02", //pImpTras\_Impuesto  retencion.tipoTasaOCuota.IVA\_16, //pImpTras\_TasaCuota  268.9696m, //pImpTras\_Importe  true, //integraComServ  1681.06m, //pComServ\_Base  null, //pComServ\_Porcentaje  14.66m, //pComServ\_Importe  false, //integraContribucionGubernamental  null, //pContrGub\_ImpContrib  "" //pContrGub\_EntidadDondePaga  ); |

### Generar Complemento Servicios Plataformas Tecnológicas

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07  08  09 | ReturnXmlComplemento rtnCompl = ret.Plat\_generaComplemento();                  if (!rtnCompl.IsXmlComplementoCreated)                  {                      MessageBox.Show("Error al crear el Xml del Complemento Servicios de Plataformas Tecnológicas 1.0: " + Environment.NewLine + rtnCompl.ErrorDetalle);                      return;                  }                  ret.Reten\_AddComplementoComprobante(rtnCompl.XMLComplemento);                  ret.Reten\_AddNamespace(ret.getNsPlataformasTecnologicas10()); |

# Referencia

## *Namespaces*

A continuación, se listas los *namespaces* de la biblioteca:

* *Namespace* [Cancelacion](#_Referencia_del_Namespace)
* *Namespace* [Complementos](#_Namespace_SifeiSignature3_3.Complem)
* *Namespace* [Returns](#_Namespace_SifeiSignature3_3.Returns)
* *Namespace* [Version3\_3](#_Namespace_SifeiSignature3_3.Version)
* *Namespace* [Timbre11](#_Namespace_SifeiSignature3_3.Timbre1)
* *Namespace ClasesRetenciones*

## Clases principales

| Archivo | Descripción |
| --- | --- |
| CFDI40 | Clase que representa el comprobante CFDI versión 4.0, en esta clase se agregan todos los valores que el comprobante a generar tendrá en sus atributos, así como sus respectivos nodos. |
| retencion | Clase que representa al Comprobante de Retenciones e Información de Pagos, así como los métodos para la definición de valores de nodos y atributos. También se encarga de encapsular los métodos correspondientes a los Complementos aplicables a Comprobantes de Retención e Información de Pagos. |
| Complemento | Clase que representa al nodo complemento, en esta clase se contendrán todos los valores que el comprobante a generar tendrá en sus atributos, así como sus respectivos nodos. |
| ReturnXML | Clase que representa el resultado de generar el CFDI tras invocar al método CFDI40.generaXml(), por tanto, esta clase no contiene métodos. Si algún error ocurre al generar el comprobante cuando no es válido, esta clase contendrá los códigos de error y mensajes correspondientes a la matriz de errores, en caso de éxito contiene el comprobante. |
| ReturnTimbre | Clase que representa el resultado de obtener el timbre CFDI40.recuperaTimbre(), por tanto, no contiene métodos. |
| ReturnXmlComplemento | Clase que representa el resultado de generar un complemento a través del método Pagos20\_generaComplemento de clase Complemento, por tanto, esta clase no tiene métodos. |

## *Namespace* SifeiSignature3\_3.Version3\_3

Clases:

* CFDI40

### Clase CFDI33

#### Constructor

| Constructor | Descripción |
| --- | --- |
| public CFDI40(  string connString,  string pSerie,  string pFolio,  DateTime pFecha,  string pFormaPago,  string pCondPago,  decimal pSubtotal,  decimal? pDescuento,  string pMoneda,  decimal? pTipoCambio,  decimal pTotal,  string pTipoCompr,  string pExportacion,  string pMetodoPago,  string pLugarEped,  string pConfirmacion,  string rutaCert) | Inicializa el objeto CFDI40 con los atributos principales del comprobante |

### Métodos

| Método de CFDI40 | Descripción |
| --- | --- |
| public static ReturnTimbre recuperaTimbre(  string usuario,  string password,  XmlDocument xml,  tipoTimbrado TipoTimbrado  ); | Recupera un timbre del servicio web de sifei, el cual si es correcto es devuelto en el objeto de tipo ReturnTimbre. Para más detalle ver la clase ReturnTimbre. |
| public void addAddenda(  XmlElement pAddenda  ); | Agrega un nodo hijo addenda de tipo any (cualquiera) al nodo addenda del comprobante.  xPath:  Cfdi:Addenda/**cfdi:any** |
| public void addCfdirelacionados(  string pTipoRelacion,  params string[] pLsUUIDs  ) | Crea el nodo CfdiRelacionados con el atributo TipoRelacion indicado en el parámetro @pTipoRelacion y agrega los nodos CfdiRelacionado con cada UUID especificado en el array @pLsUUIDs.  xPath:  Cfdi:CfdiRelacionados/**cfdi:Cfdirelacionado**  **Excepciones:**  **Exception**  Si @pUUID no cumple con el formato y restricciones establecidas por el SAT. |
| public void addComplementoComprobante(XmlElement pComplemento); | Agrega un complemento al nodo complemento. |
| public bool addComplementoConcepto(  int indexConcepto,  XmlElement complemento  ); | Agrega el nodo @complemento como hijo al Concepto apuntado por @indexConcepto.  xPath:  Cfdi:Conceptos/**cfdi:concepto[**@indexConcepto**]/ComplementoConcepto**  **Excepciones:**  **Exception**  Si el concepto apuntado por @indexConcepto no existe. |
| public int addConcepto(  string pCveProdServ,  string pNoIdentif,  decimal pCantidad,  string pCveUnidad,  string pUnidad,  string pDescripcion,  decimal pValorUnit,  decimal pImporte,  decimal? pDescuento,  string pObjetoImp  ); | Crea y agrega un nodo hijo concepto al nodo conceptos (Cfdi:Conceptos).  xPath:  Cfdi:Conceptos/**cfdi:concepto**  **Returns** Índice del nodo concepto que se agregó  **Excepciones:**  **Exception**  Si los parámetros no cumplen el formato y restricciones definidas por el SAT  **Returns** Índice del nodo Concepto |
| public void addCuentaPredialToConcepto(  int pIndexConc,  params string[] pLsNumeros  ); | Crea y agrega un nodo hijo CuentaPredial al nodo Concepto).  xPath:  Cfdi:Conceptos/**cfdi:concepto/cfdi:CuentaPredial**  **Excepciones:**  **Exception**  Si los parámetros no cumplen el formato y restricciones definidas por el SAT  Si el concepto apuntado por @ pIndexConc no existe |
| public void addImpRetencion(  string pImpuesto,  decimal pImporte  ); | Crea y agrega un nodo Retencion con el impuesto e importe indicado al nodo Impuestos del comprobante, verifica que exista el impuesto en el catálogo impuesto y que el importe corresponda con tipo importe (definido por el estándar del SAT).  XPath:  cfdi:Impuestos/cfdi:Retenciones/cfdi:Retencion  **Excepciones:**  **Exception**  Si el nodo Impuestos no existe (es null), no ha sido definida.  Lanza excepciones los parámetros no cumplen el formato y restricciones definidas por el SAT.  **ArgumentException**  Si alguno de los parámetros: @pImpuesto no existe en su catálogo. |
| public void addImpRetencionToConcepto(  int pIndexConc,  decimal pBase,  string pImpuesto,  string pTipoFactor,  decimal pTasaOCuota,  decimal pImporte  ); | Crea y agrega un nodo Retencion al Concepto indicado por el valor @pIndexConc, el nodo retencion tendrá los valores: pBase, pImpuesto, pTipoFactor, pTasaOCuota, pImporte.  XPath:  Cfdi:Conceptos/cfdi:concepto[@pIndexConc]/cfdi: Impuestos/cfdi:Retenciones/**cfdi:Retencion**  **Excepciones:**  **Exception**  Si el concepto al que apunta @pIndexConc no existe (es null o está fuera de rango).  Lanza excepciones los parámetros no cumplen el formato y restricciones definidas por el SAT.  **ArgumentException**  Si alguno de los parámetros: @pImpuesto, @pTipoFactor no existe en su catálogo. |
| public void addImpTraslado(  decimal pBase,  string pImpuesto,  string pTipoFactor,  decimal? pTasaCuota,  decimal? pImporte  ); | Crea y agrega un nodo Traslado con el @pImpuesto, @ptipofactor, @pTasaCuota e @pimporte indicado al nodo Impuestos del comprobante, verifica que exista el impuesto en el catálogo impuesto y que el importe corresponda con tipo importe (definido por el estándar del SAT).  XPath:  cfdi:Impuestos/cfdi:Traslados/**cfdi:Traslado**  **Excepciones:**  **Exception**  Lanza excepciones los parámetros no cumplen el formato y restricciones definidas por el SAT.  **ArgumentException**  Si alguno de los parámetros: @pImpuesto, @pTipoFactor no existe en su catálogo. |
| public void addImpTrasladoToConcepto(  int pIndexConc,  decimal pBase,  string pImpuesto,  string pTipoFactor,  decimal? pTasaOCuota,  decimal? pImporte); | Crea y agrega un nodo Traslado al Concepto indicado por el valor @pIndexConc, el nodo traslado tendrá los valores: @pBase, @pImpuesto, @pTipoFactor, @pTasaOCuota, @pImporte.  XPath:  Cfdi:Conceptos/cfdi:concepto[@pIndexConc]/cfdi: Impuestos/cfdi:Traslados/**cfdi:Traslado**  **Excepciones:**  **Exception**  Si el concepto apuntado por @pIndexConc no existe.  Lanza excepciones si no existen los valores en el catálogo o si no cumplen los tipos. |
| public void addInformacionAduaneraToConcepto(  int pIndexConc,  string pNumPedimento  ); | Crea y agrega un nodo InformacionAduanera al Concepto indicado por el valor @pIndexConc, el nodo InformacionAduanera tendrá los valores: @pNumPedimento.  Lanza excepciones si no existen los valores en el catálogo o si no cumplen los tipos.  XPath:  Cfdi:Conceptos/cfdi:concepto[@pIndexConc]/cfdi: InformacionAduanera  **Excepciones:**  **Exception**  Lanza excepciones si no existen los valores en el catálogo o si no cumplen los tipos. |
| public void addNamespace(  namespaces ns  ); | Agrega un *namespace* a la lista de *namespace* que contendrá el comprobante |
| public ReturnXML generaXml(  string pCadConex,  bool aplicaValidacionesMatrizSat = true, MidpointRounding pTipoRedondeo = MidpointRounding.AwayFromZero); | Genera el comprobante con los datos de la instancia, y retorna un objeto tipo ReturnXML, el cual contiene información acerca del resultado de generar el XML, contiene una referencia al objeto xmlDocument. |
| public List<namespaces> getNamespaces(); | Obtiene la lista de *namespaces* del comprobante. |
| public ReturnXML sellaComprobante(  XmlDocument xml,  string rutaCert,  string rutaKey,  string pwdCert,  bool asignaFechaEmisionActual,  bool validaXsds); | Recibe una referencia del de tipo XmlDocument (que es el comprobante por sellar). La ruta del certificado y llave, además de la fecha actual y una bandera para validar contra los XSD. |
| public void setACuentaTercerosToConcepto(  int pIndexConc,  string pRfcACuentaTerceros,  string pNombreACuentaTerceros,  string pRegimenFiscalACuentaTerceros,  string pDomicilioFiscalACuentaTerceros  ); | Crea y agrega un nodo ACuentaTerceros al Concepto indicado por el valor @pIndexConc, el nodo ACuentaTerceros tendrá los valores: @pRfcACuentaTerceros, @pNombreACuentaTerceros, @pRegimenFiscalACuentaTerceros y @pDomicilioFiscalACuentaTerceros.  Lanza excepciones si no existen los valores en el catálogo o si no cumplen los tipos.  XPath:  Cfdi:Conceptos/cfdi:concepto[@pIndexConc]/cfdi: ACuentaTerceros  **Excepciones:**  **Exception**  Lanza excepciones si no existen los valores en el catálogo o si no cumplen los tipos. |
| public void setEmisor(  string pRfc,  string pNombre,  string pRegFiscal  ); | Crea y establece un nodo Emisor como hijo directo al comprobante, verifica que existe el régimen fiscal en el catálogo y lanza una excepción sino existe.  XPath:  **Cfdi:Emisor**  **Excepciones:**  **Exception** Si los parámetros no cumplen con el formato y restricciones establecidas por el SAT.  **ArgumentException**  Si alguno de los parámetros: @pRegFiscalno no existe en su catálogo. |
| public void setInformacionGlobal(  string pPeriodicidad,  string pMeses,  short pAño  ); | Crea y establece un nodo InformacionGlobal como hijo directo al comprobante.  XPath:  **Cfdi:InformacionGlobal**  **Excepciones:**  **Exception** Si los parámetros no cumplen con el formato y restricciones establecidas por el SAT. |
| public void setReceptor(  string pRfc,  string pNombre,  string pDomicilioFiscalReceptor,  string pResidenciaFiscal,  string pNumRegIdTrib,  string pRegimenFiscalReceptor  string pUsoCfdi  ); | Crea y establece un nodo Receptor como hijo directo al comprobante.  XPath:  **Cfdi:Receptor**  **Excepciones:**  **Exception** Si los parámetros no cumplen con el formato y restricciones establecidas por el SAT.  **ArgumentException**  Si alguno de los parámetros: @pUsoCfdi, @pResidenciaFiscal no existe en su catálogo. |
| public void setTotalImpuestos(  decimal? pTotImpRetenidos,  decimal? pTotImpTrasladados  ); | Crea y establece un nodo Impuestos (ver xPath abajo) como hijo directo al comprobante, los parámetros @pTotImpRetenidos y @pTotImpTrasladados son opcionales, pero si se pasan deben cumplir con el tipo t\_Importe, en caso contrario se lanza una excepción.  XPath:  Cfdi:Impuestos  **Excepciones:**  **Exception**  Si los parámetros son establecidos (no null) y no cumplen los formatos y restricciones del SAT. |
| public ReturnTimbre timbraComprobante(  string serie,  string idEquipo,  XmlDocument cfdi); | Genera el timbre del comprobante a partir de XML. Retorna un objeto ReturnTimbre con el Comprobante timbrado y el cbb.  @cfdi XmlDocument con el comprobante a timbrar. |
| public void setIdentificacionSifei(  string pUsuarioSifei,  string pPasswordSifei,  tipoTimbrado pTipo  ) | Establece los valores para autenticarse en los servidores de Sifei. Estos valores se utilizarán para las operaciones de generación de XML, sellado y timbrado.  @usuarioSifei asignado para usar *SifeiSignature*  @passwordSifei asignado para usar *SifeiSignature*  @Tipo es un valor de enumerado tipoTimbrado para indicar si es de tipo pruebas o productivo. |
| public ReturnTimbre depositaFacturaGlobal(  string serie,  string idEquipo,  XmlDocument cfdi,  ref string msg) | Deposita una factura global (cantidad de conceptos mayor o igual a 3000) para su procesamiento para timbrado, retornando un Id de petición, con la que se debe recuperar el timbre.  @cfdi XmlDocument con el comprobante a timbrar. |
| public static ReturnTimbre recuperaTimbreFacturaGlobal(  string usuario,  string password,  XmlDocument xml,  long idPeticion,  tipoTimbrado tipoTimbrado,  ref string msg) | Recupera el timbre de una factura global depositada previamente, especificando el Id de Petición con el cual se hizo el depósito de la factura global.  @idPetición El Id de la Petición por la que se consulta el timbre de la Factura Global. |

### Enumeraciones:

#### Enumeracion tipoTimbrado

|  |  |
| --- | --- |
| Valor | Descripción |
| Productivo | Indica que se debe usar el ambiente de producción. |
| Pruebas | Valor que indicar que se están haciendo pruebas con *SIFEI Signature.* |

## *Namespace* SifeiSignature3\_3.Timbre11

### Clases

* TimbreBE

### Clase TimbreBE

Clase que contiene los datos del Timbre Fiscal versión 1.1.

#### Propiedades de clase

|  |
| --- |
| Propiedades |
| public string UUID { get; set; } |
| public string SelloPAC { get; set; } |
| public string NoCertificado { get; set; } |
| public string FechaTimbre { get; set; } |
| public string CadenaOriginalTimbre { get; set; } |
| public string RfcProvCertif { get; set; } |
| public string Version { get; set; } |
| public string Leyenda { get; set; } |
| public string SelloCFD { get; set; } |

## *Namespace* SifeiSignature3\_3.Complementos

### Clase Complemento

#### Enumeraciones

|  |
| --- |
| Nomina12\_pSiNoNoEspecificado |
| SI = 0, |
| NO = 1, |
| NoEspecificado = 2 |

#### Propiedades de clase

|  |  |
| --- | --- |
| Propiedades de clase | Descripción |
| public namespaces NsComercioExterior1\_1 { get} | Obtiene el *namespace* de Comercio Exterior versión 1.2 |
| public namespaces NsNomina12 { get} | Obtiene el *namespace* de Nómina versión 1.1 |
| public namespaces NsDetallista131 { get} | Obtiene el *namespace* de Detallista versión 1.3.1 |
| public namespaces NsIedu10 { get} | Obtiene el *namespace* de Instituciones Educativas Privadas versión 1.0 |
| public namespaces NsImpLocal10 { get} | Obtiene el *namespace* de Impuestos locales versión 1.0 |
| public namespaces NsPagos10 { get} | Obtiene el *namespace* de Complemento pagos versión 1.0 |
| public namespaces NsCartaPorte20 { get} | Obtiene el *namespace* de Complemento Carta Porte versión 2.0 |
| public namespaces NsPagos20 { get} | Obtiene el *namespace* de Complemento pagos versión 2.0 |
| public namespaces NsDonat11{ get} | Obtiene el *namespace* de Complemento Donatarias versión 1.1 |

| Constructor | Descripción |
| --- | --- |
| public Complemento(  string connString,  DateTime pFecha  ); | Inicializar la clase.  @connString representa la cadena de conexión a la base de datos, ejemplo:  “Server=localhost;Port=3306;Database=catálogossat;Uid=root;Pwd=123456;" |

#### Métodos

| Métodos complemento | Descripción |
| --- | --- |
| public void Cce11\_addDescripcionEspecifica(  int pIndexMercancia,  string pMarca,  string pModelo,  string pSubmodelo,  string pNumSerie  ); | Agrega la descripción al complemento de Comercio Exterior.  xPath:  Cfdi:Complemento/cce11:ComercioExterior/Mercancias/Mercancia/DescripcionesEspecificas  **Excepciones:**  **Exception** Esta excepción ocurre si los parámetros no cumplen con el formato válido establecido por el SAT.  También cuando no se ha definido la instancia Comercio Exterior. |
| public int Cce11\_addDestinatario(  string pNumRegIdTrib, // número de identificación o registro fiscal del país  string pNombre, //nombre del destinaratio  string pDomCalle,  string pDomNumExt,  string pDomNumInt,  string pDomColonia,  string pDomLocalidad,  string pDomReferencia,  string pDomMunicipio,  string pDomEstado,  string pDomPais,  string pDomCP) | Crea el nodo Destinatario y su hijo Domicilio. Se hacen las validaciones correspondientes.  Xpath:  Cfdi:Complemento/cce11:ComercioExterior/**Destinatario**  **Excepciones:**  **Exception** Esta excepción ocurre si no cumple con el formato válido establecido por el SAT.  **ArgumentException** Ocurre cuando pDomPais no existe en el catálogo.  **Returns:** Índice del nodo **Destinatario** que se agregó. |
| public void Cce11\_addDestinatarioDomicilio(int pIndexDestinatario,  string pDomCalle,  string pDomNumExt,  string pDomNumInt,  string pDomColonia,  string pDomLocalidad,  string pDomReferencia,  string pDomMunicipio,  string pDomEstado,  string pDomPais,  string pDomCP  ); | Crea y establece un nodo hijo Domicio al nodo Destinatario.  Xpath: Cfdi:Complemento/cce11:ComercioExterior/Destinatario/**Domicilio** |
| public int Cce11\_addMercancia(  string pNoIdentificacion,  string pFraccionAranc,  decimal? pCantidadAduana,  string pUnidadAduana,  decimal? pValorUnitAduana,  decimal pValorDolares) | Crea y establece un nodo hijo Mercancia a Mercancias.  Xpath: Cfdi:Complemento/cce11:ComercioExterior/Mercancias/**Mercancia**  **Returns:** Índice del nodo Mercancia que se agregó. |
| public void Cce11\_addPropietario(  string pNumRegIdTrib,  string pResidenciaFiscal  ) | Crea y establece un nodo **Propietario** como hijo a ComercioExterior.  Xpath:  Cfdi:Complemento/cce11:ComercioExterior/**Propietario** |
| public void Cce11\_Define\_ComercioExterior(  string pMotivoTrasl,  string pTipoOperacion,  string pClavePedim,  int? pCertifOrigen,  string pNumCertifOrigen,  string pNumExportadorConf, string pIncoterm,  int? pSubdivision, string pObservaciones,  decimal? pTipoCambioUSD,  decimal? pTotalUSD) | Define el complemento ComercioExterior.  Xpath:  Cfdi:Complemento/cce11**:ComercioExterior**  **Excepciones:**  **Exception** Esta excepción ocurre si no cumple con el formato válido establecido por el SAT.  **ArgumentException** Ocurre cuando un parámetro no existe en el catálogo. |
| public ReturnXmlComplemento Cce11\_generaComplemento() | Genera el complemento de comercio exterior, lanza una excepción si no se ha definido la instancia comercio exterior (con el método Cce11\_Define\_ComercioExterior()) |
| public void Cce11\_setEmisor(string pCurp,  string pDomCalle,  string pDomNumExt,  string pDomNumInt,  string pDomColonia,  string pDomLocalidad,  string pDomReferencia,  string pDomMunicipio,  string pDomEstado,  string pDomPais,  string pDomCP) | Crea y establece un nodo **Emisor** (como hijo de ComercioExterior).  Xpath:  Cfdi:Complemento/cce11**:ComercioExterior/**Emisor  **Excepciones:**  **Exception** Si algún atributo no cumple con el formato definido por el SAT, o si no se ha definido el complemento **ComercioExterior11**. |
| public void Cce11\_setReceptor(string pNumRegIdTrib, string pDomCalle, string pDomNumExt, string pDomNumInt, string pDomColonia, string pDomLocalidad, string pDomReferencia, string pDomMunicipio, string pDomEstado, string pDomPais, string pDomCP) | Crea y establece un nodo **Receptor** (como hijo de ComercioExterior).  Xpath:  Cfdi:Complemento/cce11**:ComercioExterior/**Emisor  **Excepciones:**  **Exception** Si algún atributo no cumple con el formato definido por el SAT, o si no se ha definido el complemento **ComercioExterior11** |
| public void Det131\_AddAllowanceCharge(detallistaAllowanceChargeAllowanceChargeType allChargeType, detallistaAllowanceChargeSettlementType settType, string numConsec, detallistaAllowanceChargeSpecialServicesType? specialServType, decimal? porcent) | Crea y establece un nodo allowanceCharge (como hijo de Detallista).  Xpath:  Cfdi:Complemento/Detallista**/** **allowanceCharge**  **Excepciones:**  **Exception** Si algún atributo no cumple con el formato definido por el SAT, o si no se ha definido el complemento Detallista (ver Det131\_Define\_Detallista()), o si se ha sobrepasado el límite de 99. |
| public void Det131\_addCurrency(  detallistaCurrencyCurrencyISOCode isoCode, /\*enumeración \*/  detallistaCurrencyCurrencyFunction funcion1,  /\*enumeración \*/  detallistaCurrencyCurrencyFunction? funcion2,  /\*enumeración \*/ detallistaCurrencyCurrencyFunction? funcion3,  /\*enumeración \*/  decimal? rateOfChange) | Crea y agrega un nodo currency como hijo al nodo detallista.  Xpath:  Cfdi:Complemento/Detallista**/** **currency[]**  **Excepciones:**  **Exception** Esta excepción ocurre si no se ha establecido en complemento Detallista o si se superó el número máximo de nodos Currency. |
| public void Det131\_AddCustomsGln(  string gln/\* número global de localización (GLN) de la aduana \*/  ) | Crea y agrega un nodo Customs como hijo al nodo detallista.  Xpath:  Cfdi:Complemento/Detallista**/** Customs[]  **Excepciones:**  **Exception** Esta excepción ocurre si no si no se ha establecido en complemento Detallista o si se superó el número máximo de nodos Customs. |
| public void Det131\_AddReferenceIdentificationToAdditionalInformation(  detallistaReferenceIdentificationType tipo,  /\* el número de referencia adicional\*/  string valor  ) | Crea y agrega un nodo referenceIdentification como hijo al nodo AdditionalInformation.  Xpath:  Cfdi:Complemento/Detallista**/** AdditionalInformation/ **referenceIdentification**  **Excepciones:**  **Exception** Esta excepción ocurre si no se ha establecido en complemento Detallista o si se superó el número máximo de nodos (99) **referenceIdentification**. |
| public void Det131\_AddReferenceIdentificationToDeliveryNote(  string referenceIdentification /\* folio de folio de recepción de mercancia\*/  ) | Crea y agrega un nodo referenceIdentification como hijo al nodo DeliveryNote.  Xpath:  Cfdi:Complemento/Detallista**/** DeliveryNote/ **referenceIdentification**  **Excepciones:**  **Exception** Esta excepción ocurre si no se ha establecido en complemento Detallista o si se superó el número máximo de nodos (30) referenceIdentification. Por último, si no se ha establecido DeliveryNote (que es opcional). |
| public void Det131\_AddReferenceIdentificationToOrderIdentification(string referenceIdentification /\* número de orden de compra (comprador) a la que hace referencia la factura \*/  ) | Crea y agrega un nodo referenceIdentification como hijo al nodo orderIdentification.  Xpath:  Cfdi:Complemento/Detallista**/** orderIdentification / **referenceIdentification**  **Excepciones:**  **Exception** Esta excepción ocurre si no se ha establecido en complemento Detallista o si se superó el número máximo de nodos (30) referenceIdentification. Por último, si no se ha establecido orderIdentification (que es requerido). |
| public void Det131\_AddTotalAllowanceCharge(  detallistaTotalAllowanceChargeAllowanceOrChargeType allOrChrgType, /\*enum para indicar si es cargo o descuento\*/  detallistaTotalAllowanceChargeSpecialServicesType? specServType, /\* enum para indicar el tipo de servicio\*/  decimal? Amnt /\* total de los cargos o descuentos\*/  ) | Crea y agrega un nodo TotalAllowanceCharge como hermano al nodo TotalAllowanceCharge.  Se debe indicar si es descuento o cargo; es opcional indicar el tipo de servicio y el total de los cargos o descuentos.  Xpath:  Cfdi:Complemento/Detallista**/** **TotalAllowanceCharge**[]  **Excepciones:**  **Exception** Esta excepción ocurre si no se ha establecido en complemento Detallista o si se superó el número máximo de nodos (30) referenceIdentification. Por último, si no se ha establecido orderIdentification (que es requerido). |
| public int Det131\_AgregaSpecialInstruction(  detallistaSpecialInstructionCode Code,/\* tipo de instrucciones comerciales\*/  string text  ) | Crea y agrega un nodo specialInstruction como hermano al nodo specialInstruction.  Xpath:  Cfdi:Complemento/Detallista**/** specialInstruction[]  **Returns:** Índice del nodo specialInstruction que se agregó  **Excepciones:**  **Exception** Esta excepción ocurre si no se ha establecido en complemento Detallista o si se superó el número máximo de nodos (4) specialInstruction. |
| public void Det131\_AgregaTextASpecialInstruction(  int indexSpecialInstr, /\*index del nodo **specialInstruction** \*/  string text  ) | Crea y agrega un nodo text a specialInstruction (como hijo) dado un índice (el cual indica a qué nodo specialInstruction será el padre).  Xpath:  Cfdi:Complemento/Detallista**/** specialInstruction[indexSpecialInstr]/text  **Excepciones:**  **Exception** Esta excepción ocurre si no se ha establecido en complemento Detallista, si no existe el nodo specialInstruction o si se superó el número máximo de nodos (15) text. |
| public void Det131\_Agrega\_RequestForPaymentIdentification\_Detallista(  detallistaRequestForPaymentIdentificationEntityType EntityType /\* tipo de transacción\*/  ) | Crea y agrega (o remplaza si existe) un requestForPaymentIdentification a detallista (como hijo) para indicar la transacción.  Xpath:  Cfdi:Complemento/Detallista**/** **requestForPaymentIdentification/ entityType**  **Excepciones:**  **Exception** Esta excepción ocurre si no se ha establecido en complemento Detallista. |
| public void Det131\_Define\_Detallista(detallistaDocumentStatus DocumentStatus) | Define el complemento **detallista** del comprobante.  Xpath:  Cfdi:Complemento/Detallista |
| public ReturnXmlComplemento Det131\_generaComplemento() | Retorna el complemento detallista en el objeto ReturnXmlComplemento si todo fue bien.  **Excepciones:**  **Exception** Ocurre si no se ha establecido en complemento Detallista. |
| public void Det131\_SetBuyer(  string gln, /\* el número global de localización \*/  string personOrDepartmentName /\* contacto de compras \*/  ) | Crea y agrega (o remplaza si existe) un buyer a detallista (como hijo) para indicar la información del comprador.  Xpath:  Cfdi:Complemento/Detallista**/** **buyer/gln, buyer/ contactInformation/ personOrDepartmentName/text**  **Excepciones:**  **Exception** Esta excepción ocurre si no se ha establecido en complemento Detallista. |
| public void Det131\_SetDeliveryNote(  DateTime? referenceDate, /\*opcional, pasar null para no agregar atributo\*/  string ReferenceIdentification) | Crea y establece un DeliveryNote a detallista (como hijo) para indicar la información recepción de mercancía.  Xpath:  Cfdi:Complemento/Detallista**/** **DeliveryNote**  **Excepciones:**  **Exception** Esta excepción ocurre si no se ha establecido en complemento Detallista. |
| public void Det131\_SetInvoiceCreator(  string gln, detallistaInvoiceCreatorAlternatePartyIdentificationType? tipo, /\* tipo de identificación secundaria se le asigno al proveedor \*/  string value,  string name,  string streetAddressOne,/\* domicilio del emisor de la factura \*/  string city,  string cp /\* codigo postal del domicilio del emisor de la factura \*/  ) | Crea y establece el InvoiceCreator a detallista (como hijo).  Xpath:  Cfdi:Complemento/Detallista**/** **InvoiceCreator/nameAndAddress, InvoiceCreator/** **alternatePartyIdentification**  **Excepciones:**  **Exception** Esta excepción ocurre si no se ha establecido el complemento Detallista. |
| public void Det131\_SetOrderIdentification(  DateTime? referenceDate,  string ReferenceIdentification  ) | Crea y establece el orderIdentification a detallista (como hijo).  Xpath:  Cfdi:Complemento/Detallista**/** **orderIdentification/ referenceIdentification**  **Excepciones:**  **Exception** Esta excepción ocurre si no se ha establecido el complemento Detallista. |
| public void Det131\_SetPaymentTerms(  detallistaPaymentTermsPaymentTermsEvent? termsEvent, detallistaPaymentTermsPaymentTermsRelationTime? termsRelationTime,  detallistaPaymentTermsNetPaymentNetPaymentTermsType? netPaymntTermsType, /\*para el nodo paymentTimePeriod\*/  detallistaPaymentTermsNetPaymentPaymentTimePeriodTimePeriodDueTimePeriod? pTimePeriod, /\*para el nodo paymentTimePeriod\*/  string valueTimePeriod, detallistaPaymentTermsDiscountPaymentDiscountType? descType, /\*para el nodo discountPayment \*/  string porcent /\*para el nodo discountPayment \*/  ) | Crea y establece el paymentTerms a detallista (como hijo) para especificar los términos de pago de la factura  Xpath:  Cfdi:Complemento/Detallista**/** **paymentTerms/** y/o  Cfdi:Complemento/Detallista**/** **paymentTerms/paymentTimePeriod**  y/o  Cfdi:Complemento/Detallista**/** **paymentTerms/discountPayment**  **Excepciones:**  **Exception** Esta excepción ocurre si no se ha establecido el complemento Detallista o si se estableció @pTimePeriod o valueTimePeriod cuando se pasó @netPaymntTermsType como null (ya que estos pertenecen a dicho nodo. |
| public void Det131\_SetSeller(string strGln, detallistaSellerAlternatePartyIdentificationType tipo, string alternatePartyIdentification) | Crea y establece el seller a detallista (como hijo) para especificar la información del vendedor.  Xpath:  Cfdi:Complemento/Detallista**/** **seller**  **Excepciones:**  **Exception** Esta excepción ocurre si no se ha establecido el complemento Detallista. |
| public void Det131\_SetShipTo(  string gln,  string name,  string streetAddressOne,  string city,  string cp  ) | Crea y establece el shipTo a detallista (como hijo) para especificar la ubicación donde debe realizarse la entrega de la mercancía.  Xpath:  Cfdi:Complemento/Detallista**/** **shipTo**  **Excepciones:**  **Exception** Esta excepción ocurre si no se ha establecido el complemento Detallista. |
| public void Det131\_setTotalAmount(  decimal amnt  ) | Crea y establece el totalAmount a detallista (como hijo).  Xpath:  Cfdi:Complemento/Detallista**/** **totalAmount**  **Excepciones:**  **Exception** Esta excepción ocurre si no se ha establecido el complemento Detallista. |
| public ReturnXmlComplemento Iedu10\_generaComplemento(  string nombreAlumno,  string curp,  instEducativasNivelEducativo nivelEducativo,  string autRVOE,  string RfcPago  ) | Genera el complemento concepto de instituciones educativas privadas. El complemento viene la propiedad XMLComplemento de ReturnXmlComplemento.  **Excepciones:**  **Exception** Esta excepción ocurre si los parámetros no cumplen la especificación del SAT en formato o restricción: @curp, @rfcPago, @autRVOE. |
| public void ImpLocal10\_Agrega\_RetencionImpuestoLocal(  string nombre,/\*Nombre del impuesto local retenido\*/  decimal importe,  decimal tasa) | Crea y establece el RetencionesLocales a ImpuestosLocales (como hijo).  Xpath:  Cfdi:Complemento/ImpuestosLocales**/**RetencionesLocales  **Excepciones:**  **Exception** Esta excepción ocurre si no se ha establecido el complemento ImpuestosLocales.  Si @nombre es null o cadena vacía. |
| public void ImpLocal10\_Agrega\_TrasladoImpuestoLocal(  string nombre,  decimal importe,  decimal tasa  ) | Crea y establece el RetencionesLocales a ImpuestosLocales (como hijo).  Xpath:  Cfdi:Complemento/ImpuestosLocales**/RetencionesLocales**  **Excepciones:**  **Exception** Esta excepción ocurre si no se ha establecido el complemento ImpuestosLocales.  Si @nombre es null o cadena vacía. |
| public void ImpLocal10\_Define\_ImpuestoLocal(  decimal totalTraslado,  decimal totalRetencion  ) | Crea y establece el complemento ImpuestosLocales.  Xpath:  Cfdi:Complemento/**ImpuestosLocales**  **Excepciones:**  **Exception** Ninguna. |
| public ReturnXmlComplemento ImpLocal10\_generaComplemento() | Genera el complemento impuestos locales que es devuelto en el objeto ReturnXmlComplemento, si existe un error no fatal es incluido en el mismo objeto. Ver propiedades:   * ReturnXmlComplemento.IsXmlComplementoCreated * ReturnXmlComplemento.ErrorDetalle   **Excepciones:**  **Exception**  Si los parámetros no cumplen con las especificaciones definidas por el SAT en formato o combinación.  Si no se pudo formar el nodo IEDU. |
| public void Nomina12\_addDeduccion(  string pTipoDeducc,  string pClave,  string pConcepto,  decimal pImporte  ) | Crea y establece una Deduccion a Deducciones (como hijo).  Xpath:  Cfdi:Complemento/nomina12: Nomina/nomina12: Deducciones/**Deduccion[]**  **Excepciones:**  **Exception** Esta excepción ocurre si no se ha establecido el complemento Nómina 1.2.  Si no se ha establecido el nodo deducciones.  Si los parámetros no cumplen con las especificaciones definidas por el SAT en formato o combinación.  **ArgumentException** si @pTipoDeducc no existe en el catálogo definido por el SAT. |
| public void Nomina12\_addHorasExtra(  int indexPercepcion,  int pDias,  string pTipoHoras,  int pHorasExtra,  decimal pImportePagado  ) | Crea y establece un nodo HorasExtra (como hijo) a la Percepcion apuntada por @indexPercepcion.  Xpath:  Cfdi:Complemento/nomina12: Nomina/nomina12:Percepciones/ Percepcion[@indexPercepcion]/**HorasExtra**[]  **Excepciones:**  **Exception** Esta excepción ocurre si no se ha establecido el complemento Nómina 1.2.  Si no se ha establecido el nodo Percepcion.  Si los parámetros no cumplen con las especificaciones definidas por el SAT en formato o combinación.  **ArgumentException** si @pTipoHoras no existe en el catálogo definido por el SAT. |
| public void Nomina12\_addIncapacidad(  int pDias,  string pTipo,  decimal? pImporteMon  ) | Crea y establece un nodo Incapacidad como hijo a Incapacidades.  Xpath:  Cfdi:Complemento/nomina12: Nomina/nomina12:Incapacidades/ **Incapacidad**[]  **Excepciones:**  **Exception** Esta excepción ocurre si no se ha establecido el complemento Nómina 1.2.  Si los parámetros no cumplen con las especificaciones definidas por el SAT en formato o combinación.  **ArgumentException** si @pTipo no existe en el catálogo definido por el SAT. |
| public int Nomina12\_addOtroPago(  string pTipo, /\*Tipo de otro pago(catálogo)\*/  string pClave,  string pConcepto,  decimal pImporte  ) | Crea y establece un nodo OtroPago como hijo a OtrosPagos para expresar otros pagos aplicables.  Xpath:  Cfdi:Complemento/nomina12: Nomina/nomina12:OtrosPagos/ **OtroPago**[]  **Returns:** Índice del nodo **OtroPago** que se agregó  **Excepciones:**  **Exception** Esta excepción ocurre si no se ha establecido el complemento Nómina 1.2.  Si los parámetros no cumplen con las especificaciones definidas por el SAT en formato o combinación.  **ArgumentException** si @pTipo no existe en el catálogo definido por el SAT |
| public int Nomina12\_addPercepcion(  string pTipoPercepcion,  string pClave,  string pConcepto,  decimal pImporteGravado,  decimal pImporteExento  ) | Crea y establece un nodo Percepcion como hijo a Percepciones.  Xpath:  Cfdi:Complemento/nomina12: Nomina/nomina12:Percepciones/ **Percepcion[]**  **Returns:** Índice del nodo **Percepcion** que se agregó  **Excepciones:**  **Exception** Esta excepción ocurre si no se ha establecido el complemento Nómina 1.2.  Si los parámetros no cumplen con las especificaciones definidas por el SAT en formato o combinación.  **ArgumentException** si @pTipoPercepcion no existe en el catálogo definido por el SAT |
| public void Nomina12\_addSubContratacion(  string pRfcLabora,  decimal pPorcentajeTiempo  ) | Crea y establece un nodo SubContratacion como hijo a Receptor.  Xpath:  Cfdi:Complemento/nomina12: Nomina/nomina12:Receptor/ SubContratacion**[]**  **Excepciones:**  **Exception** Esta excepción ocurre si no se ha definido el complemento Nómina 1.2.  Si los parámetros no cumplen con las especificaciones definidas por el SAT en formato o combinación. |
| public void Nomina12\_DefineNomina(  string pTipoNomina,  DateTime pFechaPago,  DateTime fechaInicPago,  DateTime fechaFinPago,  decimal numDiasPag,  decimal? totPercepciones,  decimal? totDeducciones,  decimal? totOtrosPagos  ) | Define (establece) el complemento Nómina versión 1.2.  Xpath:  Cfdi:Complemento/**nomina12: Nomina**/  **Excepciones:**  **Exception**  Si los parámetros no cumplen con las especificaciones definidas por el SAT en formato o combinación.  **ArgumentException**  Si @pTipoNomina no existe en el catálogo definido por el SAT. |
| public ReturnXmlComplemento Nomina12\_generaComplemento() | Genera el complemento Nómina versión 1.2 que es devuelto en el objeto ReturnXmlComplemento, si existe un error no fatal es incluido en el mismo objeto. Ver propiedades:   * ReturnXmlComplemento.IsXmlComplementoCreated * ReturnXmlComplemento.ErrorDetalle   **Excepciones:**  **Exception**  Si los parámetros no cumplen con las especificaciones definidas por el SAT en formato o combinación.  Si la instancia complemento Nómina no está definida. |
| public void Nomina12\_setAccionesOTitulos(  int indexPercepcion,  decimal pValorMercado,  decimal pPrecioAlOtorgarse  ) | Crea y establece el nodo AccionesOTitulos como hijo a la Percepcion apuntada por @indexPercepcion.  Xpath:  Cfdi:Complemento/nomina12: Nomina/nomina12:Percepcion[@indexPercepcion]/**AccionesOTitulos**  **Excepciones:**  **Exception** Esta excepción ocurre si no se ha definido el complemento Nómina 1.2.  Si la Percepcion padre apuntada no existe.  Si los parámetros no cumplen con las especificaciones definidas por el SAT en formato o combinación. |
| public void Nomina12\_setCompensacionSaldosAFavor(  int indexOtroPago,  decimal pSaldoAFavor,  short pAño,  decimal pRemanenteAFavor) | Crea y define la CompensacionSaldosAFavor al OtroPago apuntado por @indexOtroPago.  Xpath:  Cfdi:Complemento/nomina12: Nomina/nomina12:/OtrosPagos/OtroPago [@indexOtroPago]/ **CompensacionSaldosAFavor**  **Excepciones:**  **Exception** Esta excepción ocurre si no se ha definido el complemento Nómina 1.2.  Si OtrosPagos no está definida.  Si los parámetros no cumplen con las especificaciones definidas por el SAT en formato o combinación. |
| public void Nomina12\_setDeducciones(  decimal? pTotalOtrasDeducc, decimal? pTotalImpuestosReten) | Crea y define el nodo Deducciones como hijo a nomima.  Xpath:  Cfdi:Complemento/nomina12: Nomina/nomina12:/**Deducciones**  **Excepciones:**  **Exception** Esta excepción ocurre si no se ha definido el complemento Nómina 1.2. |
| public void Nomina12\_setEmisor(  string pCurp,  string pRegPatronal,  string pRfcPatronOrigen  ) | Crea y define el nodo Emisor como hijo a nomima.  Xpath:  Cfdi:Complemento/nomina12: Nomina/nomina12:/**Emisor**  **Excepciones:**  **Exception** Esta excepción ocurre si no se ha definido el complemento Nómina 1.2.  Si los parámetros no cumplen con las especificaciones definidas por el SAT en formato o combinación. |
| public void Nomina12\_setEntidadSNCF(string pOrigenRecurso,  decimal pMontoRecursoPropio,  bool aplicaMontoRecursoPropio  ) | Crea y define el nodo EntidadSNCF como hijo a Emisor.  Xpath:  Cfdi:Complemento/nomina12: Nomina/nomina12:/Emisor/EntidadSNCF  **Excepciones:**  **Exception** Esta excepción ocurre si no se ha definido el complemento Nómina 1.2.  Si no se ha definido la instancia Emisor.  **ArgumentException**  Si @pOrigenRecurso no existe en el catálogo definido por el SAT. |
| public void Nomina12\_setJubilacionPensionRetiro(  decimal? pTotalUnaExhibicion,  decimal? pTotalParcialidad,  decimal? pMontoDiario,  decimal pIngresoAcumulable, decimal pIngresoNoAcumulable  ) | Crea y define el nodo JubilacionPensionRetiro como hijo a Percepciones.  Xpath:  Cfdi:Complemento/nomina12: Nomina/nomina12:/Percepciones/JubilacionPensionRetiro  **Excepciones:**  **Exception** Esta excepción ocurre si no se ha definido el complemento Nómina 1.2.  Si no se ha definido la instancia nodo Percepciones.  **ArgumentException**  Ninguna. |
| public void Nomina12\_setPercepciones(  decimal? pTotalSueldos,  decimal? pTotalSeparacionIndemn,  decimal? pTotalJubilacionPensionRetiro,  decimal pTotalGravado,  decimal pTotalExento  ) | Crea y define el nodo Percepciones como hijo a nomina.  Xpath:  Cfdi:Complemento/nomina12: Nomina/nomina12:/**Percepciones**  **Excepciones:**  **Exception** Esta excepción ocurre si no se ha definido el complemento Nómina 1.2.  **ArgumentException**  Ninguna. |
| public void Nomina12\_setReceptor(  string pCurp,  string pNumSegSocial,  DateTime? pFechaInicRelLab,  string pAntiguedad,  string pTipoContrato,  Nomina12\_pSiNoNoEspecificado esSindicalizado,  string pTipoJornada,  string pTipoRegimen,  string pNumEmpleado,  string pDepartamento,  string pPuesto,  string pRiesgo,  string pPeriodiciadPago,  string pBanco,  string pCuentaBanco,  decimal? pSalarioBaseCotApor,  decimal? pSalarioDiario,  string pEstado) | Crea y define el nodo Emisor como hijo a Receptor.  Xpath:  Cfdi:Complemento/nomina12: Nomina/nomina12:/Receptor  **Excepciones:**  **Exception** Esta excepción ocurre si no se ha definido el complemento Nómina 1.2.  Si los parámetros no cumplen con las especificiones definidas por el SAT en formato o combinación.  **ArgumentException**  Si @pTipoContrato, @pTipoRegimen, @pTipoJornada, @pRiesgo, @pEstado, @pPeriodiciadPago o @pBanco no existe en el catálogo del SAT. |
| public void Nomina12\_setSeparacionIndemnizacion(  decimal pTotalPagado,  int pNumAñosServicio,  decimal pUltimoSueldoMensOrd,  decimal pIngresoAcumulable,  decimal pIngresoNoAcumulable  ) | Crea y define el nodo SeparacionIndemnizacion como hijo a Percepciones.  Xpath:  Cfdi:Complemento/nomina12: Nomina/nomina12:/Percepciones/ **SeparacionIndemnizacion**  **Excepciones:**  **Exception** Esta excepción ocurre si no se ha definido el complemento Nómina 1.2.  Si no se ha definido la instancia nodo Percepciones.  Si @pNumAñosServicio es mayor a 99.  **ArgumentException**  Ninguna. |
| public void Nomina12\_setSubsidioAlEmpleo(int indexOtroPago, decimal pSubsidioCausado) | Crea y define el nodo SubsidioAlEmpleo como hijo a OtroPago.  Xpath:  Cfdi:Complemento/nomina12: Nomina/nomina12:/OtrosPagos/OtroPago/**SubsidioAlEmpleo**  **Excepciones:**  **Exception** Esta excepción ocurre si no se ha definido el complemento Nómina 1.2.  Si no se ha definido la instancia nodo OtrosPagos.  Si @pNumAñosServicio es mayor a 99.  **ArgumentException**  Ninguna. |
| public void Pagos10\_addDoctoRelacionadoToPago(  int indexPago,  string pIdDocumento,  string pSerie,  string pFolio,  string pMonedaDR,  decimal? pTipoCambioDR,  string pMetodoDePagoDR,  int? pNumParcialidad,  decimal? pImpSaldoAnt,  decimal? pImpPagado,  decimal? pImpSaldoInsoluto  ) | Crea y define el nodo DoctoRelacionado como hijo de Pago.  Xpath:  Cfdi:Complemento/pagos/pago/ **DoctoRelacionado**[]  **Excepciones:**  **Exception** Esta excepción ocurre si no se ha definido el complemento Pago 1.0.  Si no se ha definido la instancia nodo pago.  Si los parámetros no cumplen con las especificiones definidas por el SAT en formato o combinación.  **ArgumentException**  Si @pMonedaDRes ó @pMetodoDePagoDR no existen en el catálogo del SAT. |
| public int Pagos10\_addPago(  DateTime pFechaPago,  string pFormaDePago,  string pMoneda,  decimal? pTipoCambio,  decimal pMonto,  string pNumOperacion,  string pRfcEmisorCtaOrd,  string pNomBancoOrdExt,  string pCtaOrdenante,  string pRfcEmisorCtaBen,  string pCtaBeneficiario,  string pTipoCadPago,  byte[] pCertPago,  string pCadPago,  byte[] pSelloPago  ) | Crea y define el nodo Pago como hijo de Pagos.  Xpath:  Cfdi:Complemento/pagos/pago[]  **Returns:** Índice del nodo pago que se agregó  **Excepciones:**  **Exception** Esta excepción ocurre si no se ha definido el complemento Pago 1.0.  Si los parámetros no cumplen con las especificiones definidas por el SAT en formato o combinación.  **ArgumentException**  Si @pFormaDePago, @pMoneda ó @pTipoCadPago no existen en el catálogo del SAT. |
| public ReturnXmlComplemento Pagos10\_generaComplemento() | Genera el complemento pago 1.0 que es devuelto en el objeto ReturnXmlComplemento, si existe un error no fatal es incluido en el mismo objeto. Ver propiedades:   * ReturnXmlComplemento.IsXmlComplementoCreated * ReturnXmlComplemento.ErrorDetalle   **Excepciones:**  **Exception**  Si los parámetros no cumplen con las especificiones definidas por el SAT en formato o combinación.  Si no se tiene el mínimo de elementos pagos (1).  Si se tiene establecido el nodo impuestos (no soportado para esta versión de librería). |
| public ReturnXmlComplemento CartaPorte20\_generaComplemento() | Genera el complemento Carta Porte 2.0 que es devuelto en el objeto ReturnXmlComplemento, si existe un error no fatal es incluido en el mismo objeto. Ver propiedades:   * ReturnXmlComplemento.IsXmlComplementoCreated * ReturnXmlComplemento.ErrorDetalle   **Excepciones:**  **Exception**  Si los parámetros no cumplen con las especificiones definidas por el SAT en formato o combinación. |
| public void CP20\_ADD\_CartaPorte\_Mercancias\_Autotransporte\_Remolques(string SubTipoRem, string Placa) | Crea y define el nodo Remolques como hijo de Autotransporte.  Xpath:  /cartaporte20:CartaPorte/cartaporte20:Mercancias/cartaporte20:Autotransporte/cartaporte20:Remolques  **Excepciones:**  **Exception** Esta excepción ocurre si no se ha definido el complemento Carta Porte 2.0, el elemento Mercancias o el elemento Autotransporte.  Si los parámetros no cumplen con las especificiones definidas por el SAT en formato o combinación. |
| public int CP20\_ADD\_CartaPorte\_Mercancias\_Mercancia(string BienesTransp, string ClaveSTCC, string Descripcion, decimal Cantidad,  string ClaveUnidad, string Unidad, string Dimensiones, string MaterialPeligroso,  string CveMaterialPeligroso, string Embalaje, string DescripEmbalaje, decimal PesoEnKg, decimal? ValorMercancia, string Moneda, string FraccionArancelaria, string UUIDComercioExt) | Crea y define el nodo Mercancia como hijo de Mercancias.  Xpath:  /cartaporte20:CartaPorte/cartaporte20:Mercancias/cartaporte20:Mercancia  **Returns:** Índice del nodo Mercancia que se agregó  **Excepciones:**  **Exception** Esta excepción ocurre si no se ha definido el complemento Carta Porte 2.0, el elemento Mercancias.  Si los parámetros no cumplen con las especificiones definidas por el SAT en formato o combinación. |
| public void CP20\_ADD\_CartaPorte\_Mercancias\_Mercancia\_CantidadTransporta\_UltimaMercancia(decimal pCantidad, string pIDOrigen, string pIDDestino, string pCvesTransporte) | Crea y define el nodo CantidadTransporta como hijo del último nodo Mercancia.  Xpath:  /cartaporte20:CartaPorte/cartaporte20:Mercancias/cartaporte20:Mercancia/cartaporte20:CantidadTransporta  **Excepciones:**  **Exception** Esta excepción ocurre si no se ha definido el complemento Carta Porte 2.0, el elemento Mercancias o no existe ningún elemento Mercancía.  Si los parámetros no cumplen con las especificiones definidas por el SAT en formato o combinación. |
| public void CP20\_ADD\_CartaPorte\_Mercancias\_Mercancia\_GuiasIdentificacion\_UltimaMercancia(string pNumeroGuiaIdentificacion, string pDescripGuiaIdentificacion, decimal pPesoGuiaIdentificacion) | Crea y define el nodo GuiasIdentificacion como hijo del último nodo Mercancia.  Xpath:  /cartaporte20:CartaPorte/cartaporte20:Mercancias/cartaporte20:Mercancia/cartaporte20:GuiasIdentificacion  **Excepciones:**  **Exception** Esta excepción ocurre si no se ha definido el complemento Carta Porte 2.0, el elemento Mercancias o no existe ningún elemento Mercancía.  Si los parámetros no cumplen con las especificiones definidas por el SAT en formato o combinación. |
| public void CP20\_ADD\_CartaPorte\_Mercancias\_Mercancia\_Pedimento\_UltimaMercancia(string pedim) | Crea y define el nodo Pedimentos como hijo del último nodo Mercancia.  Xpath:  /cartaporte20:CartaPorte/cartaporte20:Mercancias/cartaporte20:Mercancia/cartaporte20:Pedimentos  **Excepciones:**  **Exception** Esta excepción ocurre si no se ha definido el complemento Carta Porte 2.0, el elemento Mercancias o no existe ningún elemento Mercancía.  Si los parámetros no cumplen con las especificiones definidas por el SAT en formato o combinación. |
| public int CP20\_ADD\_CartaPorte\_Mercancias\_TransporteFerroviario\_Carro(string TipoCarro, string MatriculaCarro, string GuiaCarro, decimal ToneladasNetasCarro) | Crea y define el nodo Carro como hijo de TransporteFerroviario.  Xpath:  /cartaporte20:CartaPorte/cartaporte20:Mercancias/cartaporte20:TransporteFerroviario/cartaporte20:Carro  **Returns:** Índice del nodo Carro que se agregó  **Excepciones:**  **Exception** Esta excepción ocurre si no se ha definido el complemento Carta Porte 2.0, el elemento Mercancias o el elemento TransporteFerroviario.  Si los parámetros no cumplen con las especificiones definidas por el SAT en formato o combinación. |
| public void CP20\_ADD\_CartaPorte\_Mercancias\_TransporteFerroviario\_Carro\_Contenedor\_UltimoCarro(string TipoContenedor, decimal PesoContenedorVacio, decimal PesoNetoMercancia) | Crea y define el nodo Contenedor como hijo del último nodo Carro.  Xpath:  /cartaporte20:CartaPorte/cartaporte20:Mercancias/cartaporte20:TransporteFerroviario/cartaporte20:Carro/cartaporte20:Contenedor  **Excepciones:**  **Exception** Esta excepción ocurre si no se ha definido el complemento Carta Porte 2.0, el elemento Mercancias, elemento TransporteFerroviario, o no existe ningún elemento Carro.  Si los parámetros no cumplen con las especificiones definidas por el SAT en formato o combinación. |
| public void CP20\_ADD\_CartaPorte\_Mercancias\_TransporteFerroviario\_DerechosDePaso(string TipoDerechoDePaso, decimal KilometrajePagado) | Crea y define el nodo DerechosDePaso como hijo del nodo TransporteFerroviario.  Xpath:  /cartaporte20:CartaPorte/cartaporte20:Mercancias/cartaporte20:TransporteFerroviario/cartaporte20:DerechosDePaso  **Excepciones:**  **Exception** Esta excepción ocurre si no se ha definido el complemento Carta Porte 2.0, el elemento Mercancias, o el elemento TransporteFerroviario.  Si los parámetros no cumplen con las especificiones definidas por el SAT en formato o combinación. |
| public void CP20\_ADD\_CartaPorte\_Mercancias\_TransporteMaritimo\_Contenedor(string MatriculaContenedor, string TipoContenedor, string NumPrecinto) | Crea y define el nodo Contenedor como hijo del nodo TransporteMaritimo.  Xpath:  /cartaporte20:CartaPorte/cartaporte20:Mercancias/cartaporte20:TransporteMaritimo/cartaporte20:Contenedor  **Excepciones:**  **Exception** Esta excepción ocurre si no se ha definido el complemento Carta Porte 2.0, el elemento Mercancias, o el elemento TransporteMaritimo.  Si los parámetros no cumplen con las especificiones definidas por el SAT en formato o combinación. |
| public void CP20\_ADD\_CartaPorte\_TiposFigura(string TipoFigura, string RFCFigura, string NumLicencia,  string NombreFigura, string NumRegIdTribFigura, string ResidenciaFiscalFigura,  CartaPorte10\_Domicilio domicilio) | Crea y define el nodo TiposFigura como hijo del nodo FiguraTransporte. Si se incluye el parámetro @domicilio, se crea el nodo Domicilio como hijo del nodo TiposFigura.  Xpath:  /cartaporte20:CartaPorte/cartaporte20:FiguraTransporte/cartaporte20:TiposFigura  **Excepciones:**  **Exception** Esta excepción ocurre si no se ha definido el complemento Carta Porte 2.0.  Si los parámetros no cumplen con las especificiones definidas por el SAT en formato o combinación. |
| public void CP20\_ADD\_CartaPorte\_TiposFigura\_PartesTransporte\_UltimoTipoFigura(string partesTransp) | Crea y define el nodo PartesTransporte como hijo del nodo TiposFigura.  Xpath:  /cartaporte20:CartaPorte/cartaporte20:FiguraTransporte/cartaporte20:TiposFigura/cartaporte20:PartesTransporte  **Excepciones:**  **Exception** Esta excepción ocurre si no se ha definido el complemento Carta Porte 2.0 o el elemento FiguraTransporte/TiposFigura.  Si los parámetros no cumplen con las especificiones definidas por el SAT en formato o combinación. |
| public int CP20\_ADD\_CartaPorte\_Ubicacion(string TipoUbicacion, string IDUbicacion, string RFCRemitenteDestinatario, string NombreRemitenteDestinatario, string NumRegIdTrib, string ResidenciaFiscal, string NumEstacion, string NombreEstacion, string NavegacionTrafico, DateTime FechaHoraSalidaLlegada, string TipoEstacion, decimal? DistanciaRecorrida) | Crea y define el nodo Ubicacion como hijo del nodo Ubicaciones.  Xpath:  /cartaporte20:CartaPorte/cartaporte20:Ubicaciones/cartaporte20: Ubicacion  **Returns:** Índice del nodo Ubicacion que se agregó  **Excepciones:**  **Exception** Esta excepción ocurre si no se ha definido el complemento Carta Porte 2.  Si los parámetros no cumplen con las especificiones definidas por el SAT en formato o combinación. |
| public void CP20\_REMOVEALL\_CartaPorte\_Mercancias\_Autotransporte\_Remolques() | Remueve todos los elementos Remolques del nodo Autotransporte.  **Excepciones:**  **Exception** Esta excepción ocurre si no se ha definido el complemento Carta Porte 2, el elemento Mercancias o el elemento Autotransporte. |
| public void CP20\_SET\_CartaPorte(string TranspInternac, string EntradaSalidaMerc, string PaisOrigenDest, string ViaEntradaSalida, decimal? TotalDistRec) | Crea y define el nodo CartaPorte.  Xpath:  Cfdi:Complemento/cartaporte20:CartaPorte |
| public void CP20\_SET\_CartaPorte\_Mercancias(decimal PesoBrutoTotal, string UnidadPeso, decimal? PesoNetoTotal, int NumTotalMercancias, decimal? CargoPorTasacion) | Crea y define el nodo Mercancias como hijo del nodo CartaPorte.  Xpath:  /cartaporte20:CartaPorte/cartaporte20:Mercancias  **Excepciones:**  **Exception** Esta excepción ocurre si no se ha definido el complemento Carta Porte 2.  Si los parámetros no cumplen con las especificiones definidas por el SAT en formato o combinación. |
| public void CP20\_SET\_CartaPorte\_Mercancias\_Autotransporte(string PermSCT, string NumPermisoSCT, string ConfigVehicular, string PlacaVM, int AnioModeloVM) | Crea y define el nodo Autotransporte como hijo del nodo Mercancias. y a su vez el nodo IdentificacionVehicular como hijo del nodo Autotransporte.  Xpath:  /cartaporte20:CartaPorte/cartaporte20:Mercancias/cartaporte20:Autotransporte  **Excepciones:**  **Exception** Esta excepción ocurre si no se ha definido el complemento Carta Porte 2.0 o el elemento Mercancias.  Si los parámetros no cumplen con las especificiones definidas por el SAT en formato o combinación. |
| public void CP20\_SET\_CartaPorte\_Mercancias\_Autotransporte\_Seguros(string AseguraRespCivil, string PolizaRespCivil, string AseguraMedAmbiente, string PolizaMedAmbiente, string AseguraCarga, string PolizaCarga, decimal? PrimaSeguro) | Crea y define el nodo Seguros como hijo del nodo Autotransporte.  Xpath:  /cartaporte20:CartaPorte/cartaporte20:Mercancias/cartaporte20:Autotransporte/cartaporte20:Seguros  **Excepciones:**  **Exception** Esta excepción ocurre si no se ha definido el complemento Carta Porte 2.0, el elemento Mercancias o el elemento Autotransporte.  Si los parámetros no cumplen con las especificiones definidas por el SAT en formato o combinación. |
| public void CP20\_SET\_CartaPorte\_Mercancias\_Mercancia\_DetalleMercancia\_UltimaMercancia(string pUnidadPesoMerc, decimal pPesoBruto, decimal pPesoNeto,  decimal pPesoTara, int? pNumPiezas) | Crea y define el nodo DetalleMercancia como hijo del ultimo nodo Mercancia.  Xpath:  /cartaporte20:CartaPorte/cartaporte20:Mercancias/cartaporte20:Mercancia/cartaporte20:DetalleMercancia  **Excepciones:**  **Exception** Esta excepción ocurre si no se ha definido el complemento Carta Porte 2.0, el elemento Mercancias o al menos un elemento Mercancía.  Si los parámetros no cumplen con las especificiones definidas por el SAT en formato o combinación. |
| public void CP20\_SET\_CartaPorte\_Mercancias\_TransporteAereo(string PermSCT, string NumPermisoSCT, string MatriculaAeronave, string NombreAseg, string NumPolizaSeguro, string NumeroGuia, string LugarContrato, string CodigoTransportista, string RFCEmbarcador, string NumRegIdTribEmbarc,string ResidenciaFiscalEmbarc, string NombreEmbarcador) | Crea y define el nodo TransporteAereo como hijo del nodo Mercancias.  Xpath:  /cartaporte20:CartaPorte/cartaporte20:Mercancias/cartaporte20:TransporteAereo  **Excepciones:**  **Exception** Esta excepción ocurre si no se ha definido el complemento Carta Porte 2.0, o el elemento Mercancias.  Si los parámetros no cumplen con las especificiones definidas por el SAT en formato o combinación. |
| public void CP20\_SET\_CartaPorte\_Mercancias\_TransporteFerroviario(string TipoDeServicio, string TipoDeTrafico, string NombreAseg, string NumPolizaSeguro) | Crea y define el nodo TransporteFerroviario como hijo del nodo Mercancias.  Xpath:  /cartaporte20:CartaPorte/cartaporte20:Mercancias/cartaporte20:TransporteFerroviario  **Excepciones:**  **Exception** Esta excepción ocurre si no se ha definido el complemento Carta Porte 2.0, o el elemento Mercancias.  Si los parámetros no cumplen con las especificiones definidas por el SAT en formato o combinación. |
| public void CP20\_SET\_CartaPorte\_Mercancias\_TransporteMaritimo(string PermSCT, string NumPermisoSCT, string NombreAseg, string NumPolizaSeguro, string TipoEmbarcacion, string Matricula, string NumeroOMI, int? AnioEmbarcacion, string NombreEmbarc, string NacionalidadEmbarc, decimal UnidadesDeArqBruto, string TipoCarga, string NumCertITC, decimal? Eslora, decimal? Manga, decimal? Calado, string LineaNaviera, string NombreAgenteNaviero, string NumAutorizacionNaviero, string NumViaje, string NumConocEmbarc, string MatriculaContenedor, string TipoContenedor, string NumPrecinto) | Crea y define el nodo TransporteMaritimo como hijo del nodo Mercancias.  Xpath:  /cartaporte20:CartaPorte/cartaporte20:Mercancias/cartaporte20:TransporteMaritimo  **Excepciones:**  **Exception** Esta excepción ocurre si no se ha definido el complemento Carta Porte 2.0, o el elemento Mercancias.  Si los parámetros no cumplen con las especificiones definidas por el SAT en formato o combinación. |
| public void CP20\_SET\_CartaPorte\_Ubicacion\_Domicilio(int indexUbicacion, CartaPorte10\_Domicilio domic) | Crea y define el nodo Domicilio como hijo del nodo Ubicacion.  Xpath:  /cartaporte20:CartaPorte/cartaporte20:Ubicaciones/cartaporte20:Ubicacion/cartaporte20:Domicilio  **Excepciones:**  **Exception** Esta excepción ocurre si no se ha definido el complemento Carta Porte 2.0, o no existe el nodo Ubicacion en el índice especificado por @indexUbicacion.  Si los parámetros no cumplen con las especificiones definidas por el SAT en formato o combinación. |
| public ReturnXmlComplemento Donat11\_generaComplemento(string noAutorizacion, DateTime fechaAutorizacion, string leyenda) | Genera el complemento de Donatarias 1.1. El complemento viene la propiedad XMLComplemento de ReturnXmlComplemento.  **Excepciones:**  **Exception** Esta excepción ocurre si los parámetros no cumplen la especificación del SAT en formato o restricción. |
| public void PAGOS20\_ADD\_Pagos\_Pago(DateTime pFechaPago, string pFormaDePago, string pMoneda, decimal? pTipoCambio, decimal pMonto, string pNumOperacion, string pRfcEmisorCtaOrd, string pNomBancoOrdExt, string pCtaOrdenante, string pRfcEmisorCtaBen, string pCtaBeneficiario, string pTipoCadPago, byte[] pCertPago, string pCadPago, byte[] pSelloPago) | Crea y define el nodo Pago como hijo del nodo Pagos.  Xpath:  /pago20:Pagos/pago20:Pago  **Excepciones:**  **Exception** Esta excepción ocurre si no se ha definido el complemento Pagos 2.0.  Si los parámetros no cumplen con las especificiones definidas por el SAT en formato o combinación. |
| public void PAGOS20\_ADD\_Pagos\_Pago\_DoctoRelacionado\_RetencionDR\_UltimoDoctoRelacionado(decimal pBaseDR, string pImpuestoDR, string pTipoFactorDR, decimal pTasaOCuotaDR, decimal pImporteDR) | Crea y define el nodo RetencionDR como hijo del nodo RetencionesDR en el último nodo DoctoRelacionado.  Xpath:  /pago20:Pagos/pago20:Pago/pago20:DoctoRelacionado/pago20:ImpuestosDR/pago20:RetencionesDR/pago20:RetencionDR  **Excepciones:**  **Exception** Esta excepción ocurre si no se ha definido el complemento Pagos 2.0, el elemento Pago o no existe ningún elemento DoctoRelacionado.  Si los parámetros no cumplen con las especificiones definidas por el SAT en formato o combinación. |
| public void PAGOS20\_ADD\_Pagos\_Pago\_DoctoRelacionado\_TrasladoDR\_UltimoDoctoRelacionado(decimal pBaseDR, string pImpuestoDR, string pTipoFactorDR, decimal? pTasaOCuotaDR, decimal? pImporteDR) | Crea y define el nodo TrasladoDR como hijo del nodo TrasladosDR en el último nodo DoctoRelacionado.  Xpath:  /pago20:Pagos/pago20:Pago/pago20:DoctoRelacionado/pago20:ImpuestosDR/pago20:TrasladosDR/pago20:TrasladoDR  **Excepciones:**  **Exception** Esta excepción ocurre si no se ha definido el complemento Pagos 2.0, el elemento Pago o no existe ningún elemento DoctoRelacionado.  Si los parámetros no cumplen con las especificiones definidas por el SAT en formato o combinación. |
| public void PAGOS20\_ADD\_Pagos\_Pago\_DoctoRelacionado\_UltimoPago(string pIdDocumento, string pSerie, string pFolio, string pMonedaDR, decimal? pEquivalenciaDR, int pNumParcialidad, decimal pImpSaldoAnt, decimal pImpPagado, decimal pImpSaldoInsoluto, string pObjetoImpDR) | Crea y define el nodo DoctoRelacionado como hijo del último nodo Pago.  Xpath:  /pago20:Pagos/pago20:Pago/pago20:DoctoRelacionado  **Excepciones:**  **Exception** Esta excepción ocurre si no se ha definido el complemento Pagos 2.0, o el elemento Pago.  Si los parámetros no cumplen con las especificiones definidas por el SAT en formato o combinación. |
| public void PAGOS20\_ADD\_Pagos\_Pago\_RetencionP\_UltimoPago(string pImpuestoP, decimal ImporteP) | Crea y define el nodo RetencionP como hijo del nodo RetencionesP del último nodo Pago.  Xpath:  /pago20:Pagos/pago20:Pago/pago20:ImpuestosP/pago20:RetencionesP/pago20:RetencionP  **Excepciones:**  **Exception** Esta excepción ocurre si no se ha definido el complemento Pagos 2.0, o el elemento Pago.  Si los parámetros no cumplen con las especificiones definidas por el SAT en formato o combinación. |
| public void PAGOS20\_ADD\_Pagos\_Pago\_TrasladoP\_UltimoPago(decimal pBaseP, string pImpuestoP, string pTipoFactorP, decimal? pTasaOCuotaP, decimal? pImporteP) | Crea y define el nodo TrasladoP como hijo del nodo TrasladosP del último nodo Pago.  Xpath:  /pago20:Pagos/pago20:Pago/pago20:ImpuestosP/pago20:TrasladosP/pago20:TrasladoP  **Excepciones:**  **Exception** Esta excepción ocurre si no se ha definido el complemento Pagos 2.0, o el elemento Pago.  Si los parámetros no cumplen con las especificiones definidas por el SAT en formato o combinación. |
| public ReturnXmlComplemento Pagos20\_generaComplemento() | Genera el complemento Pagos 2.0 que es devuelto en el objeto ReturnXmlComplemento, si existe un error no fatal es incluido en el mismo objeto. Ver propiedades:   * ReturnXmlComplemento.IsXmlComplementoCreated * ReturnXmlComplemento.ErrorDetalle   **Excepciones:**  **Exception**  Si los parámetros no cumplen con las especificiones definidas por el SAT en formato o combinación. |
| public void PAGOS20\_SET\_Pagos(decimal? pTotalRetencionesIVA, decimal? pTotalRetencionesISR, decimal? pTotalRetencionesIEPS, decimal? pTotalTrasladosBaseIVA16, decimal? pTotalTrasladosImpuestoIVA16, decimal? pTotalTrasladosBaseIVA8, decimal? pTotalTrasladosImpuestoIVA8,  decimal? pTotalTrasladosBaseIVA0, decimal? pTotalTrasladosImpuestoIVA0,  decimal? pTotalTrasladosBaseIVAExento, decimal pMontoTotalPagos) | Crea y define el nodo Pagos y el nodo Totales como hijo del nodo Pagos.  Xpath:  /pago20:Pagos/pago20:Totales  **Excepciones:**  **Exception**  Si los parámetros no cumplen con las especificiones definidas por el SAT en formato o combinación. |

## *Namespace* SifeiSignature3\_3.Returns

### Clase ReturnXML

#### Constructores

|  |  |
| --- | --- |
| Constructor | Descripción |
| public ReturnXML(); | Constructor vacío. |

#### Propiedades

|  |  |
| --- | --- |
| Nombre de propiedad | Descripción |
| public bool IsXMLCreated { get; set; } | *Booleano* que indica si el XML fue creado. |
| public string ErrorCodigo { get; set; } | Indica el código de error. |
| public string ErrorDetalle { get; set; } | Propiedad que indicará el error detallado, esto incluye los errores de la “[Matriz de Errores” del SAT](#_Códigos_de_Error). |
| public XmlDocument XML { get; set; } | Contendrá el comprobante en XML generado o un complemento. Esto varía dependiendo del método que lo retorne. |
| public bool EstaSellado { get; set; } | *Booleano* que indicará si el comprobante está sellado. |
| public string CadenaOriginal { get; set; } | *String* que alberga la cadena original del comprobante. |
| public string Sello { get; set; } | Sello del comprobante. |
| public string NoCert { get; set; } | Número de certificado. |

### Clase ReturnTimbre

#### Constructores

|  |  |
| --- | --- |
| Constructor | Descripción |
| public ReturnTimbre(); | Constructor vacío. |

#### Propiedades

| Nombre de propiedad | Descripción |
| --- | --- |
| public bool IsTimbrado { get; set; } | Indica si se se timbró el CFDI. |
| public string ErrorDetalle { get; set; } | Indica el detalle del error (si es que hubo) durante el proceso de timbrado. |
| public string CodigoError { get; set; } | Código de error (si es que hubo) durante el proceso de timbrado. |
| public XmlDocument Xml { get; set; } | Objeto XmlElement que contiene Timbre en XML. |
| public byte[] Cbb { get; set; } | Arreglo de bytes del archivo imagen jpg del código de barras bidimensional. |
| public TimbreBE Timbre { get; set; } | Datos del timbre fiscal versión 1.1. |
| public XmlNode NodoTimbre { get; set; } | Nodo del complemento timbre. |
| public bool EsFacturaGlobalDepositada { get; set; } | Si se procesó factura global indica si se depositó en el servicio de timbrado. |
| public long IdPeticionFacturaGlobal { get; set; } | Id de Petición de timbrado de factura globla, debe utilizarse este valor para recuperar el timbre de la factura global. |
| public int FoliosConsumidosFacturaGlobal { get; set; } | Indica la cantidad de folios que fueron utilizados para timbrar la factura globa. |

### Clase ReturnXmlComplemento

#### Constructores

|  |  |
| --- | --- |
| Constructor | Descripción |
| public ReturnXmlComplemento(); | Constructor vacío. |

#### Propiedades

|  |  |
| --- | --- |
| Constructor | Descripción |
| public bool IsXmlComplementoCreated { get; set; } | *Boleano* que indica si se creó el complemento en XML. |
| public string ErrorDetalle { get; set; } | Incluye el detalle del error cuando IsXmlComplementoCreated es false. |
| public XmlElement XMLComplemento { get; set; } | Objeto XmlElement que contiene el Complemento. |

### Clase ReturnCancelacion2018

#### Constructores

|  |  |
| --- | --- |
| Constructor | Descripción |
| public ReturnCancelacion2018(); | Constructor vacío. |

#### Propiedades

| Constructor | Descripción |
| --- | --- |
| public bool Procesado{ get; set; } | Indica si fue recibida la petición al servicio de cancelacion. |
| public Acuse AcuseBE { get; set; } | Objeto de tipo Acuse, indica el estado de la operación “Cancelacion” ver clase Acuse. |
| public string CodigoError { get; set; } | Indica el código de error (si es que hubo). |
| public string ErrorDetalle { get; set; } | *String* con el detalle del error. |
| public List<string> UuidsNoProcesados { get; set; } | Lista de los UUIDs que no fueron procesados para su cancelación ya que los únicos que se procesan son los que tienen Estado Vigente y su estatus de Ccancelación es diferente a “En proceso”.  Siempre se debe revisar este campo independientemente del resultado de la cancelación. |

### Clase ReturnDocumentosRelacionados

#### Constructores

|  |  |
| --- | --- |
| Constructor | Descripción |
| public ReturnDocumentosRelacionados (); | Constructor vacío. |

#### Propiedades

| Constructor | Descripción |
| --- | --- |
| public bool Procesado{ get; set; } | Indica si fue recibida la petición al servicio consulta CFDI relacionados. |
| public string AcuseXml{ get; set; } | Acuse de metodo |
| public string CodigoError { get; set; } | Indica el código de error (si es que hubo). |
| public string ErrorDetalle { get; set; } | *String* con el detalle del error. |
| public ConsultaRelacionados Result { get; set; } | ObjetoConsultaRelacionados que contiene los datos de los CFDI padre e hijo. |

### Clase ReturnQrAcuse

#### Constructores

|  |  |
| --- | --- |
| Constructor | Descripción |
| public ReturnQrAcuse(); | Constructor vacío. |

#### Propiedades

| Constructor | Descripción |
| --- | --- |
| public bool Procesado{ get; set; } | Indica si fue recibida la petición al servicio consulta CFDI relacionados. |
| public string AcuseXml{ get; set; } | Acuse de metodo |
| public string CodigoError { get; set; } | Indica el código de error (si es que hubo). |
| public string ErrorDetalle { get; set; } | *String* con el detalle del error. |
| public string CodigoEstatus { get; set; } | Valor relacionado a la solicitud de consulta(no al CFDI) |
| public string EsCancelable { get; set;} | Indica si el CFDI es cancelable:  No cancelable  Cancelable sin aceptació  Cancelable con aceptación |
| public string Estado { get; set; } | Estado del CFDI:Vigente,Cancelado, No Encontrado |
| public string EstatusCancelacion { get; set; } | Estatus de la cancelación:  Cancelado sin aceptación  En proceso  Plazo vencido  Cancelado con aceptación  Solicitud rechazada |

### Clase ReturnPeticionesPendientes

#### Constructores

|  |  |
| --- | --- |
| Constructor | Descripción |
| public ReturnPeticionesPendientes (); | Constructor vacío. |

#### Propiedades

| Constructor | Descripción |
| --- | --- |
| public bool Procesado{ get; set; } | Indica si fue recibida la petición al servicio consulta CFDI relacionados. |
| public string AcuseXml{ get; set; } | Acuse de metodo |
| public string CodigoError { get; set; } | Indica el código de error (si es que hubo). |
| public string ErrorDetalle { get; set; } | *String* con el detalle del error. |
| public string CodEstatus { get; set; } | *Codigo del WS* |
| public string CodEstatusDescripcion { get; set; } | Descripción del código del WS |
| public string[] UUID { get; set; } | Lista de UUID de peticiones pendientes. |

### Clase ReturnAceptacionRechazoResult

#### Constructores

|  |  |
| --- | --- |
| Constructor | Descripción |
| public ReturnAceptacionRechazoResult(); | Constructor vacío. |

#### Propiedades

| Constructor | Descripción |
| --- | --- |
| public bool Procesado{ get; set; } | Indica si fue recibida la petición al servicio consulta CFDI relacionados. |
| public string AcuseXml{ get; set; } | Acuse de metodo |
| public string CodigoError { get; set; } | Indica el código de error (si es que hubo). |
| public string ErrorDetalle { get; set; } | *String* con el detalle del error. |
| public string CodEstatus { get; set; } | *Codigfo de estatus donde,1000 =* Se recibió la respuesta de la petición de forma exitosa |
| public string CodEstatusDescripcion { get; set; } | Descripcion del código de estatus |
| public string RfcReceptor { get; set; } | RFC receptor que envio a cancelar los CFDI. |
| public string RfcPac { get; set; } | RFC del PAC por el cual son cancelados los CFDI. |
| public DateTime Fecha { get; set; } | Fecha |
| public string Sello { get; set; } | Sello |
| public List<DetalleFolios> Folios { get; set; } | Colección de objetos DatelleFolio |

### Clase DetalleFolios

#### Constructores

|  |  |
| --- | --- |
| Constructor | Descripción |
| public DetalleFolios (); | Constructor vacío. |

#### Propiedades

| Constructor | Descripción |
| --- | --- |
| public string UUID { get; set; } | UUID la petición |
| public string EstatusUUID { get; set; } | Estatus especifico para la petición Cancelada |
| public string EstatusUUIDDescripcion { get; } | Descripcion del estatus UUID. |
| public TipoRespuesta Respuesta { get; set; } | Respuesta correspondiente al folio que se envio. |

#### Enumeracion TipoRespuesta

|  |
| --- |
| TipoRespuesta |
| Aceptacion = 0, |
| Rechazo = 1 |

## *Namespace* SifeiSignature3\_3.Complementos.Detallista131

### Enumeraciones de complemento Detallista versión 1.3.1

| Enumeraciones de Detallista |
| --- |
| detallistaAllowanceChargeAllowanceChargeType |
| detallistaAllowanceChargeMonetaryAmountOrPercentageRateBase |
| detallistaAllowanceChargeSettlementType |
| detallistaAllowanceChargeSpecialServicesType |
| detallistaCurrencyCurrencyFunction |
| detallistaCurrencyCurrencyISOCode |
| detallistaDocumentStatus |
| detallistaInvoiceCreatorAlternatePartyIdentificationType |
| detallistaLineItemAdditionalInformationReferenceIdentificationType |
| detallistaLineItemAditionalQuantityQuantityType |
| detallistaLineItemAllowanceChargeAllowanceChargeType |
| detallistaLineItemAllowanceChargeSettlementType |
| detallistaLineItemAllowanceChargeSpecialServicesType |
| detallistaLineItemAlternateTradeItemIdentificationType |
| detallistaLineItemCustomsAlternatePartyIdentificationType |
| detallistaLineItemLogisticUnitsSerialShippingContainerCodeType |
| detallistaLineItemPalletInformationDescriptionType |
| detallistaLineItemPalletInformationTransportMethodOfPayment |
| detallistaLineItemTradeItemDescriptionInformationLanguage |
| detallistaLineItemTradeItemTaxInformationTaxCategory |
| detallistaLineItemTradeItemTaxInformationTaxTypeDescription |
| detallistaOrderIdentificationReferenceIdentificationType |
| detallistaPaymentTermsDiscountPaymentDiscountType |
| detallistaPaymentTermsNetPaymentNetPaymentTermsType |
| detallistaPaymentTermsNetPaymentPaymentTimePeriodTimePeriodDueTimePeriod |
| detallistaPaymentTermsPaymentTermsEvent |
| detallistaPaymentTermsPaymentTermsRelationTime |
| detallistaReferenceIdentificationType |
| detallistaRequestForPaymentIdentificationEntityType |
| detallistaSellerAlternatePartyIdentificationType |
| detallistaSpecialInstructionCode |
| detallistaTotalAllowanceChargeAllowanceOrChargeType |
| detallistaTotalAllowanceChargeSpecialServicesType |

## *Namespace* SifeiSignature3\_3.Complementos.Iedu10

### Enumeraciones de complemento Instituciones Educativas versión 1.0

|  |
| --- |
| Namespace: SifeiSignature3\_3.Complementos.Iedu10 |
| instEducativasNivelEducativo |

### Enumeraciones de complemento Nómina versión 1.2

|  |
| --- |
| Namespace: SifeiSignature3\_3.Complementos.Nomina12 |
| NominaReceptorSindicalizado |

## Referencia del *Namespace* SifeiSignature3\_3.Cancelacion

### Clase Acuse

#### Constructores

|  |  |
| --- | --- |
| Constructor | Descripción |
| public Acuse(); | Constructor vacío. |

#### Propiedades

| Constructor | Descripción |
| --- | --- |
| public string AcuseXml { get; set; } | *String* que contiene el XML del acuse de cancelación. |
| public DateTime Fecha { get; set; } | Fecha en que se generó el acuse. |
| public string Rfc { get; set; } | RFC del emisor. |
| public string Status { get; set; } | Código del estatus de la operación cancelar SAT. |
| public string Sello { get; set; } | Sello. |
| public string Mensaje { get; set; } | Mensaje que contiene la descripción del *status*. |
| public List<AcuseDetalle> AcuseDetalle { get; set; } | Lista de detalles de acuse (indicando en UUID y estatus) ver clase [AcuseDetalle](#_Clase_AcuseDetalle). |

### Clase AcuseDetalle

#### Constructores

|  |  |
| --- | --- |
| Constructor | Descripción |
| public AcuseDetalle(); | Constructor vacío. |

#### Propiedades

|  |  |
| --- | --- |
| Propiededades | Descripción |
| public string UUID { get; set; } | UUID del CFDI. |
| public string Status { get; set; } | Código del estatus de la operacion (ver numeral 0  Códigos de Respuesta de Cancelación). |
| public string Mensaje { get; set; } | Mensaje relacionado al estatus. |

### Clase Cancelacion (deprecada, obsoleta)

Esta clase no debe usarse, favor de referirse a [Cancelacion2018](#_Clase_Cancelacion_2018).

#### Constructores

|  |  |
| --- | --- |
| Constructor | Descripción (deprecada,obsoleta) |
| public Cancelacion (); | Constructor vacío. |

#### Métodos

| Metodos de Cancelacion | Descripción |
| --- | --- |
| public ReturnCancelacion cancela(  ref string msg,  string[] listaUUIDS, /\*arreglo de UUID a cancelar\*/  string rfcSifei,  string password,  string rfcEmisor,  string rutaCert, /\*ruta del certificado\*/  string rutaKey, /\*ruta del key\*/  string pwdCert, /\*Contraseña del certificado\*/  Definiciones.TipoTimbrado pTipoTimbrado  ); | Cancela todos los CFDI pasados en el parámetro @listaUUIDS  **Excepciones:**  **Exception**  Si no existe el certificado indicado por el path @rutaCert.  Si el certificado es inválido.  Si hubo un error al crear el PFX.  **ArgumentException**  Ninguna.  **Returns:**  Objeto de tipo ReturnCancelacion que indica el resultado de la operación cancelar (ver clase ReturnCancelacion). |

### Clase Cancelacion 2018

#### Constructores

|  |  |
| --- | --- |
| Constructor | Descripción (deprecada,obsoleta) |
| public Cancelacion2018 (); | Constructor vacío. |

#### Métodos

| Metodos de Cancelacion2018 | Descripción |
| --- | --- |
| public ReturnCancelacion2018 cancelaCfdis(  ref string msg,  string[] listaUUIDS,  string rfcSifei,  string password,  string rfcEmisor,  string rutaCert,  string rutaKey,  string pwdCert, Definiciones.TipoTimbrado pTipoTimbrado); | Cancela todos los CFDI pasados en el parámetro @listaUUIDS, siguiendo el nuevo esquema de cancelación, es decir, se sigue el nuevo mecanismo implemnetado para la cancelación.  **Excepciones:**  **Exception**  Si no existe el certificado indicado por el path @rutaCert.  Si el certificado es inválido.  Si hubo un error al crear el PFX.  **ArgumentException**  Ninguna.  **Returns:**  Objeto de tipo ReturnCancelacion que indica el resultado de la operación cancelar (ver clase ReturnCancelacion). |
| public ReturnDocumentosRelacionados consultaDocumentosRelacionados(  string rfcUsuarioSifei,  string password,  tipoConsultaRelacionados consultar,  string rfcEmisor, string rfcReceptor,  string rutaCert, string rutaKey, string pwdCert,  string uuid,  ref string msg,  Definiciones.TipoTimbrado pTipoTimbrado); | Retorna todos los documentos relacionados al CFDI. El parámetro @consultar especifica qué sujeto está realizando la consulta (Emisor o Receptor) y por lo tanto los certificados deben corresponder al rfc del sujeto que esta realizando la consulta. El parámetro @rfcEmisor se vuelve requerido si el parámetro @consultar corresponde a Emisor, por el contrario el parámetro @rfcReceptor se vuelve requerido si el parámetro @consultar corresponde a Receptor  **Returns:**  Objeto de tipo ReturnDocumentosRelacionados que indica el resultado de la operación, contiene tanto los UUID, y RFC de los CFDI padre e hijos. |
| public ReturnQrAcuse consultaEstatusCfdi(  string uuid,  string rfcEmisor,  string rfcReceptor,  decimal total,  string selloCfd8,  Definiciones.TipoTimbrado tipoServicio  ); | Consulta el estado de un CFDI.  @selloCfd8 Ultimos 8 caracteres del sello  **Returns:**  Objeto de tipo ReturnDocumentosRelacionados que indica el resultado de la operación, contiene tanto los UUID, y RFC de los CFDI padre e hijos. |
| public ReturnPeticionesPendientes getPeticionesCancelacionPendientes(  string rfcUsuarioSifei,  string password,  string rfcReceptor, Definiciones.TipoTimbrado pTipoTimbrado  ); | Consulta las peticiones de cancelación pendiente que requieren autorización para ser canceladas por el @ rfcReceptor  **Returns:**  Objeto de tipo ReturnPeticionesPendientes, |
| public ReturnAceptacionRechazoResult procesaRespuestaPeticionesCancelacion(  string rfcUsuarioSifei,  string password,  string rfcReceptor,  string rutaCert,  string rutaKey,  string pwdCert,  ref string msg,  folios[] respFolios, //array de objetos folio donde se indica si se acepta o no la peticion  Definiciones.TipoTimbrado pTipoTimbrado); | Metodo que realiza la operación de aceptación y rechazo de peticiones de cancelación. Para hacerlo se deben pasar el parámetro @folios, que es el arreglo de objetos tipo folios, en el se indica el UUID a la vez que la respuesta (Aceptacion o Rechazo)  **Returns:**  ReturnAceptacionRechazoResult |

## Referencia del *Namespace* SifeiSignature3\_3.Complementos.Cce11

### Clases

* class ComercioExterior
* class ComercioExteriorDestinatario
* class ComercioExteriorDestinatarioDomicilio
* class ComercioExteriorEmisor
* class ComercioExteriorEmisorDomicilio
* class ComercioExteriorMercancia
* class ComercioExteriorMercanciaDescripcionesEspecificas
* class ComercioExteriorPropietario
* class ComercioExteriorReceptor
* class ComercioExteriorReceptorDomicilio

### Clase ComercioExterior

#### Constructores

| Constructor | Descripción |
| --- | --- |
| public ComercioExterior(); | Constructor vacío. |

#### Propiedades

| Propiedades |
| --- |
| public bool TipoCambioUSDSpecified { get; set; } |
| public decimal TipoCambioUSD { get; set; } |
| public string Observaciones { get; set; } |
| public bool SubdivisionSpecified { get; set; } |
| public int Subdivision { get; set; } |
| public bool IncotermSpecified { get; set; } |
| public string Incoterm { get; set; } |
| public string NumeroExportadorConfiable { get; set; } |
| public string NumCertificadoOrigen { get; set; } |
| public bool CertificadoOrigenSpecified { get; set; } |
| public decimal TotalUSD { get; set; } |
| public int CertificadoOrigen { get; set; } |
| public string ClaveDePedimento { get; set; } |
| public string TipoOperacion { get; set; } |
| public bool MotivoTrasladoSpecified { get; set; } |
| public string MotivoTraslado { get; set; } |
| public string Version { get; set; } |
| public ComercioExteriorMercancia[] Mercancias { get; set; } |
| public ComercioExteriorDestinatario[] Destinatario { get; set; } |
| public ComercioExteriorReceptor Receptor { get; set; } |
| public ComercioExteriorPropietario[] Propietario { get; set; } |
| public ComercioExteriorEmisor Emisor { get; set; } |
| public bool ClaveDePedimentoSpecified { get; set; } |
| public bool TotalUSDSpecified { get; set; } |

### Clase ComercioExteriorDestinatario

#### Constructores

|  |  |
| --- | --- |
| Constructor | Descripción |
| public ComercioExteriorDestinatario(); | Constructor vacío. |

#### Propiedades

|  |
| --- |
| Propiedades |
| public ComercioExteriorDestinatarioDomicilio[] Domicilio { get; set; } |
| public string NumRegIdTrib { get; set; } |
| public string Nombre { get; set; } |

### Clase ComercioExteriorDestinatarioDomicilio

#### Constructores

| Constructor | Descripción |
| --- | --- |
| public ComercioExteriorDestinatarioDomicilio (); | Constructor vacío. |

#### Propiedades

|  |
| --- |
| Propiedades |
| public string Calle { get; set; } |
| public string NumeroExterior { get; set; } |
| public string NumeroInterior { get; set; } |
| public string Colonia { get; set; } |
| public string Localidad { get; set; } |
| public string Referencia { get; set; } |
| public string Municipio { get; set; } |
| public string Estado { get; set; } |
| public string Pais { get; set; } |
| public string CodigoPostal { get; set; } |

### Clase ComercioExteriorEmisor

#### Constructores

|  |  |
| --- | --- |
| Constructor | Descripción |
| public ComercioExteriorEmisor (); | Constructor vacío. |

#### Propiedades

|  |
| --- |
| Propiedades |
| public ComercioExteriorEmisorDomicilio Domicilio { get; set; } |
| public string Curp { get; set; } |

### Clase ComercioExteriorEmisor

#### Constructores

|  |  |
| --- | --- |
| Constructor | Descripción |
| public ComercioExteriorEmisor (); | Constructor vacío. |

#### Propiedades

| Propiedades |
| --- |
| public string Calle { get; set; } |
| public string NumeroExterior { get; set; } |
| public string NumeroInterior { get; set; } |
| public string Colonia { get; set; } |
| public bool ColoniaSpecified { get; set; } |
| public string Localidad { get; set; } |
| public bool LocalidadSpecified { get; set; } |
| public string Referencia { get; set; } |
| public string Municipio { get; set; } |
| public bool MunicipioSpecified { get; set; } |
| public string Estado { get; set; } |
| public string Pais { get; set; } |
| public string CodigoPostal { get; set; } |

### Clase ComercioExteriorMercancia

#### Constructores

|  |  |
| --- | --- |
| Constructor | Descripción |
| public ComercioExteriorMercancia(); | Constructor vacío. |

#### Propiedades

|  |
| --- |
| Propiedades |
| public ComercioExteriorMercanciaDescripcionesEspecificas[] DescripcionesEspecificas { get; set; } |
| public string NoIdentificacion { get; set; } |
| public string FraccionArancelaria { get; set; } |
| public bool FraccionArancelariaSpecified { get; set; } |
| public decimal CantidadAduana { get; set; } |
| public bool CantidadAduanaSpecified { get; set; } |
| public string UnidadAduana { get; set; } |
| public bool UnidadAduanaSpecified { get; set; } |
| public decimal ValorUnitarioAduana { get; set; } |
| public bool ValorUnitarioAduanaSpecified { get; set; } |
| public decimal ValorDolares { get; set; } |

### Clase ComercioExteriorMercanciaDescripcionesEspecificas

#### Constructores

| Constructor | Descripción |
| --- | --- |
| public ComercioExteriorMercanciaDescripcionesEspecificas (); | Constructor vacío. |

#### Propiedades

|  |
| --- |
| Propiedades |
| public string Marca { get; set; } |
| public string Modelo { get; set; } |
| public string SubModelo { get; set; } |
| public string NumeroSerie { get; set; } |

### Clase ComercioExteriorPropietario

#### Constructores

|  |  |
| --- | --- |
| Constructor | Descripción |
| public ComercioExteriorPropietario (); | Constructor vacío. |

#### Propiedades

| Propiedades |
| --- |
| public string NumRegIdTrib { get; set; } |
| public string ResidenciaFiscal { get; set; } |

### Clase ComercioExteriorReceptor

#### Constructores

|  |  |
| --- | --- |
| Constructor | Descripción |
| public ComercioExteriorReceptor (); | Constructor vacío. |

#### Propiedades

|  |
| --- |
| Propiedades |
| public ComercioExteriorReceptorDomicilio Domicilio { get; set; } |
| public string NumRegIdTrib { get; set; } |

### Clase ComercioExteriorReceptorDomicilio

#### Constructores

|  |  |
| --- | --- |
| Constructor | Descripción |
| public ComercioExteriorReceptorDomicilio (); | Constructor vacío. |

#### Propiedades

| Propiedades |
| --- |
| public string Calle { get; set; } |
| public string NumeroExterior { get; set; } |
| public string NumeroInterior { get; set; } |
| public string Colonia { get; set; } |
| public string Localidad { get; set; } |
| public string Referencia { get; set; } |
| public string Municipio { get; set; } |
| public string Estado { get; set; } |
| public string Pais { get; set; } |
| public string CodigoPostal { get; set; } |

## Referencia del *Namespace* SifeiSignature3\_3.Complementos.ImpLocal10

### Clase ImpuestosLocales

#### Constructores

|  |  |
| --- | --- |
| Constructor | Descripción |
| public ImpuestosLocales (); | Constructor vacío. |

#### Propiedades

| Propiedades |
| --- |
| public ImpuestosLocalesRetencionesLocales[] RetencionesLocales { get; set; } |
| public ImpuestosLocalesTrasladosLocales[] TrasladosLocales { get; set; } |
| public string version { get; set; } |
| public decimal TotaldeRetenciones { get; set; } |
| public decimal TotaldeTraslados { get; set; } |

### Clase ImpuestosLocalesRetencionesLocales

#### Constructores

|  |  |
| --- | --- |
| Constructor | Descripción |
| public ImpuestosLocalesRetencionesLocales (); | Constructor vacío. |

#### Propiedades

|  |
| --- |
| Propiedades |
| public string ImpLocRetenido { get; set; } |
| public decimal TasadeRetencion { get; set; } |
| public decimal Importe { get; set; } |

### Clase ImpuestosLocalesTrasladosLocales

#### Constructores

|  |  |
| --- | --- |
| Constructor | Descripción |
| public ImpuestosLocalesTrasladosLocales (); | Constructor vacío. |

#### Propiedades

| Propiedades |
| --- |
| public string ImpLocTrasladado { get; set; } |
| public decimal TasadeTraslado { get; set; } |
| public decimal Importe { get; set; } |

## Referencia del *Namespace* SifeiSignature3\_3.Complementos.Nomina12

### Clases

* class [Nomina](#_Clase_Nomina)
* class [NominaDeducciones](#_Clase_NominaDeducciones)
* class [NominaDeduccionesDeduccion](#_Clase_NominaEmisor)
* class [NominaEmisor](#_Clase_NominaEmisor)
* class [NominaEmisorEntidadSNCF](#_Clase_NominaEmisorEntidadSNCF)
* class [NominaIncapacidad](#_Clase_NominaIncapacidad)
* class [NominaOtroPago](#_Clase_NominaOtroPago)
* class [NominaOtroPagoCompensacionSaldosAFavor](#_Clase_NominaOtroPagoCompensacionSal)
* class [NominaOtroPagoSubsidioAlEmpleo](#_Clase_NominaOtroPagoSubsidioAlEmple)
* class [NominaPercepciones](#_Clase_NominaPercepciones)
* class [NominaPercepcionesJubilacionPensionRetiro](#_Clase_NominaPercepcionesJubilacionP)
* class [NominaPercepcionesPercepcion](#_Clase_NominaPercepcionesPercepcion)
* class [NominaPercepcionesPercepcionAccionesOTitulos](#_Clase_NominaPercepcionesPercepcionA)
* class [NominaPercepcionesPercepcionHorasExtra](#_Clase_NominaPercepcionesPercepcionH)
* class [NominaPercepcionesSeparacionIndemnizacion](#_Clase_NominaPercepcionesSeparacionI)
* class [NominaReceptor](#_Clase_NominaReceptor)
* class [NominaReceptorSubContratacion](#_Clase_NominaReceptorSubContratacion)

### Enumeración NominaReceptorSindicalizado

| Enumeraciones de NominaReceptorSindicalizado |
| --- |
| NominaReceptorSindicalizado |

### Clase Nomina

#### Propiedades

|  |
| --- |
| Propiedades |
| public bool TotalDeduccionesSpecified { get; set; } |
| public decimal TotalDeducciones { get; set; } |
| public bool TotalPercepcionesSpecified { get; set; } |
| public decimal TotalPercepciones { get; set; } |
| public decimal NumDiasPagados { get; set; } |
| public DateTime FechaFinalPago { get; set; } |
| public DateTime FechaInicialPago { get; set; } |
| public decimal TotalOtrosPagos { get; set; } |
| public DateTime FechaPago { get; set; } |
| public string Version { get; set; } |
| public NominaIncapacidad[] Incapacidades { get; set; } |
| public NominaOtroPago[] OtrosPagos { get; set; } |
| public NominaDeducciones Deducciones { get; set; } |
| public NominaPercepciones Percepciones { get; set; } |
| public NominaReceptor Receptor { get; set; } |
| public NominaEmisor Emisor { get; set; } |
| public string TipoNomina { get; set; } |
| public bool TotalOtrosPagosSpecified { get; set; } |

### Clase NominaDeducciones

#### Propiedades

|  |
| --- |
| Propiedades |
| public NominaDeduccionesDeduccion[] Deduccion { get; set; } |
| public decimal TotalOtrasDeducciones { get; set; } |
| public bool TotalOtrasDeduccionesSpecified { get; set; } |
| public decimal TotalImpuestosRetenidos { get; set; } |
| public bool TotalImpuestosRetenidosSpecified { get; set; } |

### Clase NominaDeduccionesDeduccion

#### Propiedades

|  |
| --- |
| Propiedades |
| public string TipoDeduccion { get; set; } |
| public string Clave { get; set; } |
| public string Concepto { get; set; } |
| public decimal Importe { get; set; } |

### Clase NominaEmisor

#### Propiedades

|  |
| --- |
| Propiedades |
| public string TipoDeduccion { get; set; } |
| public string Clave { get; set; } |
| public string Concepto { get; set; } |
| public decimal Importe { get; set; } |

### Clase NominaEmisorEntidadSNCF

### Propiedades

|  |
| --- |
| Propiedades |
| public string OrigenRecurso { get; set; } |
| public decimal MontoRecursoPropio { get; set; } |
| public bool MontoRecursoPropioSpecified { get; set; } |

### Clase NominaIncapacidad

#### Propiedades

|  |
| --- |
| Propiedades |
| public int DiasIncapacidad { get; set; } |
| public string TipoIncapacidad { get; set; } |
| public decimal ImporteMonetario { get; set; } |
| public bool ImporteMonetarioSpecified { get; set; } |

### Clase NominaOtroPago

#### Propiedades

|  |
| --- |
| Propiedades |
| public NominaOtroPagoSubsidioAlEmpleo SubsidioAlEmpleo { get; set; } |
| public NominaOtroPagoCompensacionSaldosAFavor CompensacionSaldosAFavor { get; set; } |
| public string TipoOtroPago { get; set; } |
| public string Clave { get; set; } |
| public string Concepto { get; set; } |
| public decimal Importe { get; set; } |

### Clase NominaOtroPagoCompensacionSaldosAFavor

#### Propiedades

|  |
| --- |
| Propiedades |
| public decimal SaldoAFavor { get; set; } |
| public short Año { get; set; } |
| public decimal RemanenteSalFav { get; set; } |

### Clase NominaOtroPagoSubsidioAlEmpleo

#### Propiedades

| Propiedades |
| --- |
| public decimal SubsidioCausado { get; set; } |

### Clase NominaPercepciones

#### Propiedades

|  |
| --- |
| Propiedades |
| public NominaPercepcionesPercepcion[] Percepcion { get; set; } |
| public NominaPercepcionesJubilacionPensionRetiro JubilacionPensionRetiro { get; set; } |
| public NominaPercepcionesSeparacionIndemnizacion SeparacionIndemnizacion { get; set; } |
| public decimal TotalSueldos { get; set; } |
| public bool TotalSueldosSpecified { get; set; } |
| public decimal TotalSeparacionIndemnizacion { get; set; } |
| public bool TotalSeparacionIndemnizacionSpecified { get; set; } |
| public decimal TotalJubilacionPensionRetiro { get; set; } |
| public bool TotalJubilacionPensionRetiroSpecified { get; set; } |
| public decimal TotalGravado { get; set; } |
| public decimal TotalExento { get; set; } |

### Clase NominaPercepcionesJubilacionPensionRetiro

#### Propiedades

| Propiedades |
| --- |
| public decimal TotalUnaExhibicion { get; set; } |
| public bool TotalUnaExhibicionSpecified { get; set; } |
| public decimal TotalParcialidad { get; set; } |
| public bool TotalParcialidadSpecified { get; set; } |
| public decimal MontoDiario { get; set; } |
| public bool MontoDiarioSpecified { get; set; } |
| public decimal IngresoAcumulable { get; set; } |
| public decimal IngresoNoAcumulable { get; set; } |

### Clase NominaPercepcionesPercepcion

#### Propiedades

|  |
| --- |
| Propiedades |
| public NominaPercepcionesPercepcionAccionesOTitulos AccionesOTitulos { get; set; } |
| public NominaPercepcionesPercepcionHorasExtra[] HorasExtra { get; set; } |
| public string TipoPercepcion { get; set; } |
| public string Clave { get; set; } |
| public string Concepto { get; set; } |
| public decimal ImporteGravado { get; set; } |
| public decimal ImporteExento { get; set; } |

### Clase NominaPercepcionesPercepcionAccionesOTitulos

#### Propiedades

| Propiedades |
| --- |
| public decimal ValorMercado { get; set; } |
| public decimal PrecioAlOtorgarse { get; set; } |

### Clase NominaPercepcionesPercepcionHorasExtra

#### Propiedades

|  |
| --- |
| Propiedades |
| public int Dias { get; set; } |
| public string TipoHoras { get; set; } |
| public int HorasExtra { get; set; } |
| public decimal ImportePagado { get; set; } |

### Clase NominaPercepcionesSeparacionIndemnizacion

#### Propiedades

|  |
| --- |
| Propiedades |
| public decimal TotalPagado { get; set; } |
| public int NumAñosServicio { get; set; } |
| public decimal UltimoSueldoMensOrd { get; set; } |
| public decimal IngresoAcumulable { get; set; } |
| public decimal IngresoNoAcumulable { get; set; } |

### Clase NominaReceptor

#### Propiedades

| Propiedades |
| --- |
| public decimal SalarioDiarioIntegrado { get; set; } |
| public bool SalarioBaseCotAporSpecified { get; set; } |
| public decimal SalarioBaseCotApor { get; set; } |
| public string CuentaBancaria { get; set; } |
| public bool BancoSpecified { get; set; } |
| public string Banco { get; set; } |
| public string PeriodicidadPago { get; set; } |
| public bool RiesgoPuestoSpecified { get; set; } |
| public string RiesgoPuesto { get; set; } |
| public string Puesto { get; set; } |
| public string Departamento { get; set; } |
| public bool SalarioDiarioIntegradoSpecified { get; set; } |
| public string NumEmpleado { get; set; } |
| public bool TipoJornadaSpecified { get; set; } |
| public string TipoJornada { get; set; } |
| public bool SindicalizadoSpecified { get; set; } |
| public NominaReceptorSindicalizado Sindicalizado { get; set; } |
| public string TipoContrato { get; set; } |
| public string Antigüedad { get; set; } |
| public bool FechaInicioRelLaboralSpecified { get; set; } |
| public DateTime FechaInicioRelLaboral { get; set; } |
| public string NumSeguridadSocial { get; set; } |
| public string Curp { get; set; } |
| public NominaReceptorSubContratacion[] SubContratacion { get; set; } |
| public string TipoRegimen { get; set; } |
| public string ClaveEntFed { get; set; } |

### Clase NominaReceptorSubContratacion

#### Propiedades

|  |
| --- |
| Propiedades |
| public string RfcLabora { get; set; } |
| public decimal PorcentajeTiempo { get; set; } |

## Referencia del *Namespace* SifeiSignature3\_3.Complementos.Pagos10

### Clases

* class [Pagos](#_Clase_Pagos)
* class [PagosPago](#_Clase_PagosPago)
* class [PagosPagoDoctoRelacionado](#_Clase_PagosPagoDoctoRelacionado)
* class [PagosPagoImpuestos](#_Clase_PagosPagoImpuestos)
* class [PagosPagoImpuestosRetencion](#_Clase_PagosPagoImpuestosRetencion)
* class [PagosPagoImpuestosTraslado](#_Clase_PagosPagoImpuestosTraslado)

### Clase Pagos

#### Propiedades

|  |
| --- |
| Propiedades |
| public PagosPago[] Pago { get; set; } |
| public string Version { get; set; } |

### Clase PagosPago

#### Propiedades

| Propiedades |
| --- |
| public byte[] CertPago { get; set; } |
| public bool TipoCadPagoSpecified { get; set; } |
| public string TipoCadPago { get; set; } |
| public string CtaBeneficiario { get; set; } |
| public string RfcEmisorCtaBen { get; set; } |
| public string CtaOrdenante { get; set; } |
| public string NomBancoOrdExt { get; set; } |
| public string RfcEmisorCtaOrd { get; set; } |
| public string NumOperacion { get; set; } |
| public decimal Monto { get; set; } |
| public bool TipoCambioPSpecified { get; set; } |
| public decimal TipoCambioP { get; set; } |
| public string MonedaP { get; set; } |
| public string FormaDePagoP { get; set; } |
| public DateTime FechaPago { get; set; } |
| public PagosPagoImpuestos[] Impuestos { get; set; } |
| public PagosPagoDoctoRelacionado[] DoctoRelacionado { get; set; } |
| public string CadPago { get; set; } |
| public byte[] SelloPago { get; set; } |

### Clase PagosPagoDoctoRelacionado

#### Propiedades

|  |
| --- |
| Propiedades |
| public string IdDocumento { get; set; } |
| public string Serie { get; set; } |
| public string Folio { get; set; } |
| public string MonedaDR { get; set; } |
| public decimal TipoCambioDR { get; set; } |
| public bool TipoCambioDRSpecified { get; set; } |
| public string MetodoDePagoDR { get; set; } |
| public string NumParcialidad { get; set; } |
| public decimal ImpSaldoAnt { get; set; } |
| public bool ImpSaldoAntSpecified { get; set; } |
| public decimal ImpPagado { get; set; } |
| public bool ImpPagadoSpecified { get; set; } |
| public decimal ImpSaldoInsoluto { get; set; } |
| public bool ImpSaldoInsolutoSpecified { get; set; } |

### Clase PagosPagoImpuestos

#### Propiedades

|  |
| --- |
| Propiedades |
| public PagosPagoImpuestosRetencion[] Retenciones { get; set; } |
| public PagosPagoImpuestosTraslado[] Traslados { get; set; } |
| public decimal TotalImpuestosRetenidos { get; set; } |
| public bool TotalImpuestosRetenidosSpecified { get; set; } |
| public decimal TotalImpuestosTrasladados { get; set; } |
| public bool TotalImpuestosTrasladadosSpecified { get; set; } |

### Clase PagosPagoImpuestosRetencion

#### Propiedades

|  |
| --- |
| Propiedades |
| public string Impuesto { get; set; } |
| public decimal Importe { get; set; } |

### Clase PagosPagoImpuestosTraslado

#### Propiedades

| Propiedades |
| --- |
| public string Impuesto { get; set; } |
| public string TipoFactor { get; set; } |
| public decimal TasaOCuota { get; set; } |
| public decimal Importe { get; set; } |

## Referencia del *Namespace* SifeiSignature3\_3 .Complementos.cartaporte20

### Clases

* class CartaPorte
* class CartaPorteUbicacion
* class CartaPorteUbicacionDomicilio
* class CartaPorteMercancias
* class CartaPorteMercanciasMercancia
* class CartaPorteMercanciasMercanciaPedimentos
* class CartaPorteMercanciasMercanciaGuiasIdentificacion
* class CartaPorteMercanciasMercanciaCantidadTransporta
* class CartaPorteMercanciasMercanciaDetalleMercancia
* class CartaPorteMercanciasAutotransporte
* class CartaPorteMercanciasAutotransporteIdentificacionVehicular
* class CartaPorteMercanciasAutotransporteSeguros
* class CartaPorteMercanciasAutotransporteRemolque
* class CartaPorteMercanciasTransporteMaritimo
* class CartaPorteMercanciasTransporteMaritimoContenedor
* class CartaPorteMercanciasTransporteAereo
* class CartaPorteMercanciasTransporteFerroviario
* class CartaPorteMercanciasTransporteFerroviarioDerechosDePaso
* class CartaPorteMercanciasTransporteFerroviarioCarro
* class CartaPorteMercanciasTransporteFerroviarioCarroContenedor
* class CartaPorteTiposFigura
* class CartaPorteTiposFiguraPartesTransporte
* class CartaPorteTiposFiguraDomicilio

### Clase CartaPorte

#### Propiedades

| Propiedades |
| --- |
| public string EntradaSalidaMerc { set; get; } |
| public bool EntradaSalidaMercSpecified { set; get; } |
| public CartaPorteTiposFigura[] FiguraTransporte { set; get; } |
| public CartaPorteMercancias Mercancias { set; get; } |
| public string PaisOrigenDestino { set; get; } |
| public bool PaisOrigenDestinoSpecified { set; get; } |
| public decimal TotalDistRec { set; get; } |
| public bool TotalDistRecSpecified { set; get; } |
| public string TranspInternac { set; get; } |
| public CartaPorteUbicacion[] Ubicaciones { set; get; } |
| public string Version { set; get; } |
| public string ViaEntradaSalida { set; get; } |
| public bool ViaEntradaSalidaSpecified { set; get; } |

### Clase CartaPorteMercancias

#### Propiedades

| Propiedades |
| --- |
| public CartaPorteMercanciasAutotransporte Autotransporte { set; get; } |
| public decimal CargoPorTasacion { set; get; } |
| public bool CargoPorTasacionSpecified { set; get; } |
| public CartaPorteMercanciasMercancia[] Mercancia { set; get; } |
| public int NumTotalMercancias { set; get; } |
| public decimal PesoBrutoTotal { set; get; } |
| public decimal PesoNetoTotal { set; get; } |
| public bool PesoNetoTotalSpecified { set; get; } |
| public CartaPorteMercanciasTransporteAereo TransporteAereo { set; get; } |
| public CartaPorteMercanciasTransporteFerroviario TransporteFerroviario { set; get; } |
| public CartaPorteMercanciasTransporteMaritimo TransporteMaritimo { set; get; } |
| public string UnidadPeso { set; get; } |

### Clase CartaPorteMercanciasAutotransporte

#### Propiedades

| Propiedades |
| --- |
| public **CartaPorteMercanciasAutotransporteIdentificacionVehicular** **IdentificacionVehicular** { set; get; } |
| public **string** **NumPermisoSCT** { set; get; } |
| public **string** **PermSCT** { set; get; } |
| public **CartaPorteMercanciasAutotransporteRemolque[]** **Remolques** { set; get; } |
| public **CartaPorteMercanciasAutotransporteSeguros** **Seguros** { set; get; } |

### Clase CartaPorteMercanciasAutotransporteIdentificacionVehicular

#### Propiedades

| Propiedades |
| --- |
| public **int** **AnioModeloVM** { set; get; } |
| public **string** **ConfigVehicular** { set; get; } |
| public **string** **PlacaVM** { set; get; } |

### Clase CartaPorteMercanciasAutotransporteRemolque

#### Propiedades

| Propiedades |
| --- |
| public **string** **Placa** { set; get; } |
| public **string** **SubTipoRem** { set; get; } |

### Clase CartaPorteMercanciasAutotransporteSeguros

#### Propiedades

| Propiedades |
| --- |
| public **string** **AseguraCarga** { set; get; } |
| public **string** **AseguraMedAmbiente** { set; get; } |
| public **string** **AseguraRespCivil** { set; get; } |
| public **string** **PolizaCarga** { set; get; } |
| public **string** **PolizaMedAmbiente** { set; get; } |
| public **string** **PolizaRespCivil** { set; get; } |
| public **decimal** **PrimaSeguro** { set; get; } |
| public **bool** **PrimaSeguroSpecified** { set; get; } |

### Clase CartaPorteMercanciasMercancia

#### Propiedades

| Propiedades |
| --- |
| public **string** **BienesTransp** { set; get; } |
| public **decimal** **Cantidad** { set; get; } |
| public **CartaPorteMercanciasMercanciaCantidadTransporta[]** **CantidadTransporta** { set; get; } |
| public **string** **ClaveSTCC** { set; get; } |
| public **string** **ClaveUnidad** { set; get; } |
| public **string** **CveMaterialPeligroso** { set; get; } |
| public **bool** **CveMaterialPeligrosoSpecified** { set; get; } |
| public **string** **Descripcion** { set; get; } |
| public **string** **DescripEmbalaje** { set; get; } |
| public **CartaPorteMercanciasMercanciaDetalleMercancia** **DetalleMercancia** { set; get; } |
| public **string** **Dimensiones** { set; get; } |
| public **string** **Embalaje** { set; get; } |
| public **bool** **EmbalajeSpecified** { set; get; } |
| public **string** **FraccionArancelaria** { set; get; } |
| public **bool** **FraccionArancelariaSpecified** { set; get; } |
| public **CartaPorteMercanciasMercanciaGuiasIdentificacion[]** **GuiasIdentificacion** { set; get; } |
| public **string** **MaterialPeligroso** { set; get; } |
| public **bool** **MaterialPeligrosoSpecified** { set; get; } |
| public **string** **Moneda** { set; get; } |
| public **bool** **MonedaSpecified** { set; get; } |
| public **CartaPorteMercanciasMercanciaPedimentos[]** **Pedimentos** { set; get; } |
| public **decimal** **PesoEnKg** { set; get; } |
| public **string** **Unidad** { set; get; } |
| public **string** **UUIDComercioExt** { set; get; } |
| public **decimal** **ValorMercancia** { set; get; } |
| public **bool** **ValorMercanciaSpecified** { set; get; } |

## Referencia del *Namespace* SifeiSignature3\_3.srvSifeiCancelacion2018

### Clases

* class folios

### Enumeraciones

* Respuesta

|  |
| --- |
| Respuesta |
| Aceptacion = 0, |
| Rechazo = 1 |

### Clase folios

#### Propiedades

| Propiedades |
| --- |
| public respuesta respuesta { get; set; } |
| public bool respuestaSpecified { get; set; } |
| public string uuid { get; set; } |

## Códigos de Respuesta de Cancelación

Estos son los códigos de respuesta que devuelve la operación cancelar CFDI.

| Código | Descripción |
| --- | --- |
| 201 | Cancelado |
| 202 | Previamente cancelado |
| 203 | UUID no corresponde a Emisor |
| 204 | UUID no aplicable para cancelacion |
| 205 | UUID no existe |
| 300 | Autenticación inválida |
| 301 | XML mal formado |
| 302 | Sello mal formado o inválido |
| 303 | Sello no corresponde a Emisor o caduco |
| 304 | Certificado Revocado o Caduco |
| 305 | La Fecha de Emision no está dentro de la vigencia del CSD del Emisor |
| 306 | El Certificado no es de tipo de CSD |
| 307 | El CFDI contiene un timbre previo |
| 308 | Certificado no expedido por el SAT |

## Códigos de Error de la Matriz de validaciones para CFDI versión 4.0

Para CFDI 4.0 se están siguiendo las reglas de validación que el SAT especificó en la Matriz de códigos de error para el Comprobante Fiscal Digital por Internet, version 4.0, disponible en <http://omawww.sat.gob.mx/tramitesyservicios/Paginas/anexo_20.htm>

## Codigos de Error del SAT para el metodo Consulta CFDI Relacionado.

|  |  |
| --- | --- |
| Código | Descripción |
| 2000 | Existen CFDI relacionados al folio fiscal |
| 2001 | No existen CFDI relacionados al folio fiscal |
| 2002 | El folio fiscal no pertenece al receptor |
| 1101 | No existen peticiones pendientes para el RFC Receptor |

## Codigos de error de SAT para el metodo obtener Peticiones de Cancelacion Pendientes.

|  |  |
| --- | --- |
| Código | Descripción |
| 300 | Usuario no valido |
| 301 | XML mal formado |
| 1100 | Se obtuvo petición del RFC receptor de forma exitosa |
| 1101 | No existen peticiones para el RFC del receptor |

## Codigos de error del SAT para el metodo Aceptacion/Rechazo de cancelacion (procesa Respuesta).

|  |  |
| --- | --- |
| Código | Descripción |
| 300 | Usuario no válido |
| 301 | XML mal formado |
| 302 | Sello mal formado |
| 304 | Certificado revocado o caduco |
| 305 | Certificado inválido |
| 309 | Patrón de folio inválido |
| 310 | CSD inválido |
| 1000 | Se recibió la respuesta de la petición de forma exitosa |
| 1001 | No existen peticiones de cancelación en espera de respuesta para el UUID |
| 1002 | Ya se recibió una respuesta para la petición de cancelación del UUID |
| 1003 | Sello no corresponde al RFC receptor |
| 1004 | Existen más de una petición de cancelación para el mismo UUID |
| 1005 | El UUID es nulo no posee el formato correcto |
| 1006 | Se rebasó el número máximo de solicitudes permitidas |

## Codigos de respuesta del servicio de consulta CFDI.

|  |  |
| --- | --- |
| Atributo | Descripción |
| Estado | “Vigente”,”Cancelado”,”No encontrado” |
| CodigoEstatus | “S - Comprobante obtenido satisfactoriamente”,” N - 601: La expresión impresa proporcionada no es válida”,” N - 602: Comprobante no encontrado” |
| EsCancelable | “No cancelable”,” Cancelable sin aceptación”,” Cancelable con aceptación” |

# Información de Contacto con SIFEI

Centro de Soporte Técnico SIFEI

Acceso a recursos de Soporte Técnico de los productos y servicios de SIFEI, Preguntas Frecuentes, Manuales de Usuario, Manuales Técnicos, Notas Técnicas, entre otros.

|  |  |
| --- | --- |
| Dirección electrónica | [Centro de Soporte Técnico SIFEI](http://www.sifei.com.mx/soporte) |

Teléfonos de Contacto

|  |  |
| --- | --- |
| Orizaba, Ver. | +52 (272) 726 6999 |
| CDMX | (55) 4624 0146 |
| Puebla, Pue. | (222) 211 6156 |
| con 10 líneas | |

Atención a Incidentes

La atención a incidentes se realizará mediante una herramienta de gestión de incidentes y la comunicación se realizará mediante correo electrónico.

|  |  |
| --- | --- |
| Correo Electrónico | [helpdesk@sifei.com.mx](mailto:helpdesk@sifei.com.mx?subject=Solicitud%20de%20Atención) |

Horario de atención

El horario de atención a clientes y de Soporte Técnico para para preguntas, dudas o problemas de la aplicación es:

|  |  |
| --- | --- |
| Lunes a viernes | De 09:00 a 19:00 hrs. |

Páginas oficiales de SIFEI

|  |  |
| --- | --- |
| Sitio web | <http://www.sifei.com.mx/> |
| Facebook | <http://www.facebook.com/sifeimexico> |
| Twitter | <http://twitter.com/#!/SIFEIMexico> |
| YouTube | <https://www.youtube.com/channel/UCDSLHrFOg2tGvG3AruHNmaw> |
| LinkedIn | <https://www.linkedin.com/company/18265037/> |

Ubicación de Oficina Matriz

Primera Privada de Oriente 17 No. 32

Col. Centro, Orizaba, Veracruz, México

CP 94300

<Fin del Documento>