PROTOCOLO TCP

INSIGNA

Contenido

1. Objetivo

Con el fin de facilitar el proceso de timbrado de CFDIs, se ha desarrollado el protocolo INSIGNA. En este documento se presentan las especificaciones del protocolo INSIGNA versión 1.0, un protocolo de texto XML que permite interactuar con el Servicio de Timbrado INSIGNA.

1.1. Tipos de CFDIs

En el protocolo INSIGNA se identifica como un CFDI a los siguientes tipos de comprobantes:

- Comprobante Fiscal Digital por Internet (CFDI V3.3)
- Documento Electrónico de Retenciones e información de Pagos (Retención Pago V1.0)

Por lo que al hacer referencia a un CFDI dentro de este manual tomaremos en cuenta los comprobantes previamente mencionados.

2. Servicio de Timbrado INSIGNA

El Servicio de Timbrado INSIGNA es un servicio que corre sobre el protocolo TCP y permite timbrar, cancelar y obtener información relativa a CFDIs ya timbrados o cancelados en los sistemas de INSIGNA.

2.1. Prerrequisitos para conectarse

Con el fin de proteger el acceso a los datos de sus clientes, INSIGNA solicita a sus clientes que cumplan los siguientes pasos antes de conectarse al Servicio de Timbrado:

a) Proporcionar a INSIGNA la(s) dirección(es) IP desde la(s) cual(es) se conectará(n) al Servicio de Timbrado
b) Solicitar a INSIGNA un par de certificados TLS/SSL, los cuales servirán para encriptar la comunicación entre el cliente e INSIGNA. Un certificado es personalizado para el cliente y el otro es el certificado público de INSIGNA

Cualquier intento de conexión hecha desde una dirección IP no registrada o sin utilizar los certificados SSL proporcionados por INSIGNA será rechazado.

2.2. Conexión (greeting)

Para conectarse al Servicio de Timbrado INSIGNA se necesita establecer una conexión TCP/TLS, esto implica la utilización de un certificado TLS/SSL generado conjuntamente por el cliente e INSIGNA.

Si el certificado TLS utilizado por el cliente fue proporcionado y autorizado por INSIGNA, y la dirección IP del cliente también está dada de alta ante INSIGNA, el Servicio de Timbrado responderá con una estructura en XML llamada “greeting”, la cual desplegará la siguiente información sobre el Servicio de Timbrado.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<pac xmlns="urn:ietf:params:xml:ns:pac-1.0">
	<greeting>
		<svID>signer.insigna.mx</svID>
		<svDate>2018-01-01T14:10:11.12</svDate>
		<svVersion>
			<version>1.0</version>
```

```
                </svVersion>
                <svSession>
                        <svSessionTTL>86400</svSessionTTL>
                        <svSessionTimeout>7200</svSessionTimeout>
                </svSession>
        </greeting>
</pac>
```

Los elementos del greeting son los siguientes:

- <svID> - despliega el ID de la máquina
- <svDate> - fecha y hora
- <svVersion> - versiones del protocolo (no confundir con la versión del CFDI). Actualmente solo existe la versión 1.0
    - <version> - versión
- <svSession> - información sobre las sesiones
    - <svSessionTTL> - tiempo máximo que puede durar una sesión en segundos
    - <svSessionTimeout> - tiempo máximo de inactividad en la sesión en segundos

2.3. Envío de comandos

La manera de interactuar con el Servicio de Timbrado INSIGNA es a través de comandos. Dichos comandos se dividen en dos categorías:

- Para manejar la sesión con el Servicio de Timbrado
- Para hacer operaciones y consultas sobre CFDIs (siempre y cuando se tenga sesión abierta)

Los comandos de manejo de sesión se detallan en la sección 2.4, mientras que los comandos de CFDI son cubiertos en la sección 3 del presente documento.

2.3.1 Aspectos técnicos del envío de comandos a INSIGNA

El protocolo INSIGNA es un protocolo definido en XML. Esto implica que el cliente formará el XML de acuerdo al comando o acción que desea realizar, y lo enviará al Servicio de Timbrado de INSIGNA vía una conexión TCP/TLS.

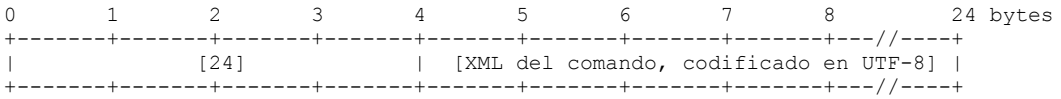
Adicionalmente, al enviar un comando al Servicio de Timbrado INSIGNA vía TCP/TLS, se deben tener en cuenta las siguientes consideraciones:

- **Encabezado.** Antes de cada comando se incluye un encabezado con el total de los bytes que se envíen. Dicho encabezado añade 4 bytes extra.
  Ejemplo: Si el XML del comando que se va a enviar mide 20 bytes, el encabezado correspondiente será un 24 codificado en bytes.
  Importante: No se enviará una cadena de texto que diga "24", sino un 24 representado en un arreglo de bytes de longitud 4
- **Encoding.** La codificación utilizada para enviar el texto del comando en XML debe ser UTF-8
- **Esquema**. Las reglas que INSIGNA define para la formación de comandos se encuentran en el esquema proporcionado en el presente documento. INSIGNA valida cada comando recibido contra dicho esquema; si el comando no se apega a las reglas del esquema, INSIGNA rechazará el comando

Una manera recomendada de generar este formato es la siguiente:

1. Se forma el comando XML (en texto) de acuerdo a la acción que se desea ejecutar
2. Se valida que el comando XML cumpla con el esquema definido por INSIGNA, esto es, que sea un comando válido de acuerdo al protocolo INSIGNA. Esto se hace con ayuda del esquema (ver sección 8)
3. Se codifica el comando XML (texto) en UTF-8, de forma que se tenga un arreglo de bytes listo para ser enviado
4. Se mide la longitud del arreglo de bytes generado en el punto 2, el cual contiene la representación en bytes (codificados en UTF-8) del comando, y se le suma 4 para tener la longitud total de los datos a enviar
5. Se genera un arreglo de bytes de tamaño 4, y se le asigna el valor obtenido en el punto 3
6. Se envía el encabezado junto con el comando

Ejemplo de los datos enviados, con un encabezado de 4 bytes y un comando de 20 bytes:



Es importante recalcar que la conversión de número a arreglo de bytes (para el encabezado), y de cadena de texto a arreglo de bytes (para el comando) dependen del lenguaje de programación utilizado. Se anexan estos ejemplos en Java, sólo para fines ilustrativos:

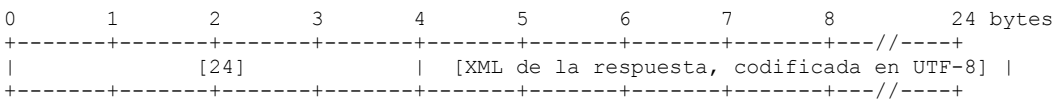
```
// Convertir un "String comando" a un arreglo de bytes (usando codificación UTF-8)
byte[] comandoBytes = comando.getBytes("UTF-8");

// Convertir un número "longitud" a un arreglo de bytes. Se usa BigInteger porque tiene
// un método que transforma automáticamente a arreglo de bytes
BigInteger longitud = new BigInteger(comandoBytes.length);
byte[] longitudBytes = longitud.toByteArray();
```

NOTA: Si el encabezado del comando indica que el tamaño del mismo es mayor a 5 MB, el Servicio de Timbrado de INSIGNA rechazará el comando por razones de seguridad. Como referencia, se espera que el comando más extenso sea el invoicer:sign, y que el tamaño de dicho comando para una factura ordinaria sea de entre 4 kb y 8 kb.

2.3.2 Respuestas de INSIGNA

De igual manera, una vez que el cliente ha enviado el comando al Servicio de Timbrado, este comando será procesado e INSIGNA enviará una respuesta al cliente informando del resultado de la transacción. Cuando el cliente procese la respuesta de INSIGNA deberá tener en cuenta las mismas recomendaciones de la sección 2.3.1, es decir:

- **Encabezado**: INSIGNA antepone al inicio de sus respuestas el tamaño de dicha respuesta, en un encabezado de 4 bytes. Es decir, si la respuesta mide 20 bytes, INSIGNA enviará información equivalente a 24 bytes:



- **Encoding**. La respuesta enviada por INSIGNA está codificada en UTF-8, es necesario decodificarla en este formato para poder hacer una correcta lectura de los datos
- **Esquema**. INSIGNA se asegura de que las respuestas que envía corresponden a la sintaxis definida en su esquema (ver sección 8)

2.4. Sesiones

En este apartado se detallan los comandos relativos al manejo de sesiones. Es importante recalcar que antes de hacer alguna operación con CFDIs se debe contar con una sesión abierta.

2.4.1 Inicio de sesión.

Al enviar un comando login, un cliente podrá abrir una sesión. El comando login requiere de los siguientes parámetros, siendo todos obligatorios:

- <clID> - ID de la cuenta o nombre de usuario INSIGNA
- <pw> - contraseña de la cuenta del cliente
- <options> - este elemento contiene los siguientes parámetros:
  - <version> - versión a utilizar del protocolo (valor fijo en 1.0)

```
<? xml version="1.0" encoding="UTF-8" standalone="no"?>
<pac xmlns="urn:ietf:params:xml:ns:pac-1.0">
	<command>
		<login>
			<clID>ClientX</clID>
			<pw>foo-BAR</pw>
			<options>
				<version>1.0</version>
			</options>
		</login>
	</command>
</pac>
```

INSIGNA decodifica el XML utilizando UTF-8, por lo cual los comandos enviados por el cliente a través de TCP/TLS deben emplear dicha codificación.

Si los datos proporcionados en el login son correctos el Servicio de Timbrado regresará un código de resultado 1000.

```
<? xml version="1.0" encoding="UTF-8" standalone="no"?>
<pac xmlns="urn:ietf:params:xml:ns:pac-1.0">
	<response>
		<result code="1000">
			<msg lang="es">Comando ejecutado exitosamente</msg>
		</result>
		<trID>
			<svTRID>8806</svTRID>
		</trID>
		<account>
			<signatures>10100312</signatures>
		</account>
		<stats>
			<svTimestamp>2018-01-01T13:46:24</svTimestamp>
			<cmdReceived>2018-01-01T13:46:24</cmdReceived>
```

```
                        <cmdExecutionTime>33</cmdExecutionTime>
                        <clConnectionTime>0</clConnectionTime>
                        <clOpenConnections>1</clOpenConnections>
                </stats>
        </response>
</pac>
```

Además del código de resultado, el Servicio de Timbrado INSIGNA envía datos informativos sobre la sesión del cliente.

2.4.2 Cierre de sesión

Para cerrar la sesión se envía comando de tipo logout, el cual no requiere de datos adicionales.

```
<? xml version="1.0" encoding="UTF-8" standalone="no"?>
<pac xmlns="urn:ietf:params:xml:ns:pac-1.0">
        <command>
                <logout/>
        </command>
</pac>
```

Si el comando logout es procesado correctamente, el Servicio de Timbrado de INSIGNA responderá con un código de resultado 1500, que confirma que la sesión fue cerrada.

```
<? xml version="1.0" encoding="UTF-8" standalone="no"?>
<pac xmlns="urn:ietf:params:xml:ns:pac-1.0">
        <response>
                <result code="1500">
                        <msg lang="es">
                                Comando ejecutado exitosamente; terminando la sesión
                        </msg>
                </result>
                <trID>
                        <svTRID>8832</svTRID>
                </trID>
                <account>
                        <signatures>10100312</signatures>
                </account>
                <stats>
                        <svTimestamp>2018-01-01T13:52:23</svTimestamp>
                        <cmdReceived>2018-01-01T13:52:23</cmdReceived>
                        <cmdExecutionTime>16</cmdExecutionTime>
                        <clConnectionTime>2000</clConnectionTime>
                        <clOpenConnections>0</clOpenConnections>
                </stats>
        </response>
</pac>
```

2.4.3 Mantener viva la sesión

Cuando una sesión establecida con el Servicio de Timbrado de INSIGNA rebasa las 2 horas de inactividad, dicha sesión expira.
Es posible mantener viva la sesión enviando periódicamente un comando denominado hello. Dicho comando no requiere datos adicionales:

```
<? xml version="1.0" encoding="UTF-8" standalone="no"?>
<pac xmlns="urn:ietf:params:xml:ns:pac-1.0">
        <command>
                <hello/>
        </command>
</pac>
```

La respuesta al hello es idéntica a la del login y muy parecida a la del logout:

```
<? xml version="1.0" encoding="UTF-8" standalone="no"?>
<pac xmlns="urn:ietf:params:xml:ns:pac-1.0">
        <response>
                <result code="1000">
                        <msg lang="es">Comando ejecutado exitosamente</msg>
                </result>
                <trID>
                        <svTRID>8591</svTRID>
                </trID>
                <account>
                        <signatures>10098213</signatures>
                </account>
                <stats>
                        <svTimestamp>2018-01-01T23:25:28</svTimestamp>
                        <cmdReceived>2018-01-01T23:25:28</cmdReceived>
                        <cmdExecutionTime>17</cmdExecutionTime>
                        <clConnectionTime>37800</clConnectionTime>
                        <clOpenConnections>1</clOpenConnections>
                </stats>
        </response>
</pac>
```


Si el cliente envía un comando hello sin tener sesión abierta, no recibirá una respuesta como la anteriormente mostrada; en lugar de ello recibirá un greeting con el nombre del servidor, etc.

2.4.4. Tiempo máximo de vida de la sesión.

Cuando transcurren 24 horas de que una sesión fue establecida con el Servicio de Timbrado de INSIGNA, la sesión expira (ver elemento svSessionTTL del greeting, en la sección 2.2). Esto es con el fin de que el cliente proporcione su contraseña al menos una vez al día.

Si el cliente desea mantener comunicación continua con el Servicio de Timbrado de INSIGNA debe renovar su sesión al menos una vez al día. El cliente también tiene la posibilidad de mantener varias sesiones a la vez.

2.4.5 Sesiones concurrentes

Un cliente puede tener hasta 10 sesiones con el Servicio de Timbrado INSIGNA de manera concurrente. Solamente se puede tener una sesión por cada conexión. Es decir, un cliente no puede abrir dos sesiones en una misma conexión.

Si el cliente intenta abrir dos sesiones en una misma conexión ocurre lo siguiente:

a) El cliente se conecta al Servicio de Timbrado INSIGNA

b) El cliente manda un comando login e INSIGNA contesta con un código 1000, abriendo así una sesión
c) El cliente manda otro comando login e INSIGNA le responde que ya tiene una sesión abierta

2.4.6 Fallos al iniciar sesión.

Si se envía algún dato equivocado en el comando login, el Servicio de Timbrado de INSIGNA responderá con un código diferente al 1000, lo cual indica que hubo un error en el inicio de sesión. Por ejemplo, cuando la contraseña no es correcta, se envía la siguiente respuesta:

```
<? xml version="1.0" encoding="UTF-8" standalone="no"?>
<pac xmlns="urn:ietf:params:xml:ns:pac-1.0">
        <response>
                <result code="2200">
                        <msg lang="es">Error de autenticación</msg>
                </result>
                <trID>
                        <svTRID>11318</svTRID>
                </trID>
                <stats>
                        <svTimestamp>2018-01-01T23:13:27</svTimestamp>
                        <cmdReceived>2018-01-01T23:13:26</cmdReceived>
                        <cmdExecutionTime>1138</cmdExecutionTime>
                </stats>
        </response>
</pac>
```

2.4.7 Bloqueos

INSIGNA establece un máximo de intentos para conectarse, de modo que si se rebasa este límite el cliente quedará bloqueado.

2.4.7.1 Bloqueo por intentos de inicio de sesión fallidos.

El valor máximo es de 5 intentos por cliente y/o dirección IP. Esto significa que si se reciben 5 intentos fallidos desde la misma dirección IP, esta será bloqueada. De la misma manera, si se reciben 5 intentos fallidos del mismo cliente la cuenta del cliente será bloqueada.

Para desbloquear una cuenta y/o una dirección IP es necesario comunicarse con INSIGNA.

2.4.7.2 Bloqueos por Firewall

El Servicio de Timbrado de INSIGNA se encuentra protegido por un firewall. Si en un momento dado la dirección IP de un cliente tiene más de 30 conexiones simultáneas al Servicio de Timbrado, será bloqueado. Para desbloquear una dirección IP es necesario comunicarse con INSIGNA, o bien esperar una hora y el bloqueo desaparecerá de forma automática.

3. Comandos de CFDIs

Los comandos de consulta, timbrado y cancelación de CFDIs disponibles son válidos para CFDIs procesados en INSIGNA.

Los comandos son contenidos en esta estructura:

```
<? xml version="1.0" encoding="UTF-8" standalone="no"?>
<pac xmlns="urn:ietf:params:xml:ns:pac-1.0">
        <command>
        </command>
      <trID>
            <clTRID>#ID#</clTRID>
      </trID>
</pac>
```

Destaca ahora la presencia del elemento trID. El cliente tiene la posibilidad de asignar a cada comando que envía un ID único (el *client transaction ID*, o abreviado: clTRID). Este ID facilitará el rastreo de los comandos en caso de que exista algún problema con alguno de ellos, sin embargo, no es obligatorio, por lo cual la estructura mencionada en la sección 2.4 –sin el trID- es válida también.

3.1. Timbrado de CFDIs

El comando ***sign*** se utiliza para realizar el timbrado de un CFDI. Dentro del elemento ***sign*** se requiere solamente el siguiente parámetro:

- <cfdi> Indica que dentro de este elemento se encuentra el CFDI a timbrar. INSIGNA no pide el XML del CFDI en texto plano, sino codificado en formato base64, como se observa en el siguiente ejemplo:

```
<? xml version="1.0" encoding="UTF-8" standalone="no"?>
<pac xmlns="urn:ietf:params:xml:ns:pac-1.0">
        <command>
                <sign>
                        <cfdi>
                PC9FeHBlZGlkb0VPD94bmU9Im5vIj8+PENvbXByb2JhbnRlIHhtbG5zPSJodHRwOi8vd
                3d3LnNhbWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiIHN0YW5kYWxvdC
                5nb2IubXgvY2ZkLzMiIHhtbG5zOnhzaT0iaHR0cDovL3d3dy53My5vcmcvMjAwMS9Y
                WFMb2NhdGlvbj0iaHR0cDovL3d3dy5zYXQuZ29iLm14L2NmZC8zIGh0dHATUxTY2hl
                bWEtaW5zdGFuY2UiIHhzaTpzY2hlb6Ly93d3cuc2F0LmdvYi5teC9zaXRpb19pbnRlcm5l
                dC9jZmQvMy9jZmR2MzIueHNkIiB2ZXJzaW9uPSIzLjIiIGZlY2hhPSIyMDEzLTAzLTI2
                VDEyOjU3OjQ1IiBzZWxsbz0iQ25xN2IyaUdlYjN4dStSK1h5YjZ0bUJtbFJIS0I1WHRtaF
                JGSWFRUnpOQkE3Ymk2VjFCcWFham9DN0RFa2NEalVXY2RMbXhwNnQyWmpwe
                WZGZ05tNDJBekc1Q25VMnA2dEtEeS82cEFlcVdnczFZNDYvVi9uanZ6MnBRMk5nb
                Wo0M3hLMVNhWlh0RDVL1J5WVdOcHc3TnVJRlJ5YVdKMWRHRnlhV0V4SHpBZE
                Jna3Foa2lHOXcwQkNRRVdFR0ZqYjJSelFITmhkQzVuYjJJdWJYZ3hKakFrQmdOVkJB
                a01IVUYyTGlCSWFXUmhiR2R2SURjM0xDQkRiMnd1SUVkMVpYSnlaWEp2TVE0d0
                RBWURWUVFSREFVd05qTXdNREVMTUFrR0ExVUVCaE1DVFZneEdUQVhCZ05W
                QkFnTUVFUnBjM1J5YVhSdklFWmxaR1Z5WVd3eEV6QVJCZ05WQkFjTUNrTjFZWF
                ZvZEdWdGIyTXhNekF4QmdrcWhraUc5dzBCQ1FJTUpGSmxjM0J2Ym5OaFlteGxPaUJ
                HWlhKdVlXNWtieUJOWVhKMHc2MXVaWG9nUTI5emN6QWVGdzB4TVRFd016QXl
                NalV4TXpCYUZ3MHhNekV3TWpreU1qVXhNekJhTUlIUk1TZ3dKZ1lEVlFRREV4OU
                VWVXhEUlNCQ1JWUklTVTVCSUcEhlTW9JaFluRmdDVjNSeU9EdnZ5SHYwdDU4V
                nZZPSIgdG90YWw9IjEuMTYiIHN1YlRvdGFsPSIxLjAwIiBjZXJ0aWZpY2Fkbz0iTUlJR
```

U1UQ0NBeG1nQXdJQkFnSREF4TURRMk1qQXhNekF3RFFZSktvWklodmNOQVFFRk
JRQXdnZ0UyTVRnd05nWURWUVFEREM5QkxrTXVJR1JsYkNCVFpYSjJhV05wYnlC
a1pTQkJaRzFwYm1semRISmhZMm5EczI0Z1ZISnBIGVuIHVuYSBzb2xhIGV4aGliaWN
pw7NuIiBub0NlcnRpZmljYWRvPSIwMDAwMTAwMDAwMDEwNDYyMDEzMCIgdG
lwb0RlQ29tcHJvYmFudGU9ImluZ3Jlc28iIGRlc2N1ZW50bz0iMC4wMCIgbWV0b2RvR
GVQYWdvPSJFZmVjdGl2byIgTHVnYXJFeHBlZGljaW9uPSJHdWFkYWx1cGUsIE51Z
XZvIExlSU0JDUlZSSVNVNUJJRkpQUkZKSlIxVkZXaUJCVGtkRlRFVlRNU2d3SmdZ
RFZRUUtFeDlFVlV4RFJTQkNSVlJJU1U1QklGSlBSRkpKUjFWRldpQkJUa2RGVEVW
VE1SWXdGQVlEVlFRdEV3MVNUMEZFTnpjeE1USTFRVTAwTVJzd0dRWURWUVF
GRXhKU1QwRkVOemNZblYwWVhKcFlURXZNQzBHQTFVRUNnd21VMlZ5ZG1sam
FXOGdaR1VnUVdSdGFXNXBjMZKUFJGSkpSMVZGV2lCQlRrZEZURVZUTVNnd0p
nWURWUVFwRXg5RVZVeERMDQ4ejZsc2d2SkRhTHcxK3NnY2VJK2h5cm51TWRN
bmZUNjhzZWtJb0wrY0tvQzU0QjhMYzArMVNxOUxqRVhta3VCMHJsY0hxcEJ4cEljW
EJvL3I5UnQ2c24rUnFRKzNlNVpzb1FmSTF5L2lvSDlaVzhnaEMzRHNXcHlkVU5wZHJ
mdGJHVkFpcDVXUXp2VlpVVWI1azFZNldoUUpuTHBSengvbmRpNS9WTzYrSjRhSW
JTUy9xZmlXRjQxOHg1RWt2SHBDL2gwNWhVNEU2eWU1Ymp0bHE5K1M3TjZqNlV
4L1Y3TWVKTDArOXpxVzhsY3FZRUoiIGZvcm1hRGVQYWdvPSJQYWdv4TVRJMV
RVZFVSRTVNTURFeEhEQWFCZ05WQkFzVUUwMWhkSEpwZWlCTVlYTWdRVzN
wY21sallYTXdnWjh3RFFZSktvWklodmNOQVFFQkJRQURnWTBBTUlHSkFvR0JBTD
UvcTlldEtDako4ZkhERVZJNHhVd2ZWMDJMcEdJZXlwTmxwZFMwSHlRUUdvampk
WUdnYW50dC9TRmgrcVhOb050dTZWS0ZOeGQzYlI0bzFpbHNNSHFFYVR3QVFzY0
RrR2ZpOUgwZFJZZVUrZlo3eVQ1b2FrUkVkVHNWSWRPUDc3b1pqTXFUVjNVd3lD
UW5GbUM0bi9zSnc2eTB0M09lWGdBd2ZzWFQ2dld4QWdNQkFBR2pIVEFiTUF3R0E
xVWRFd0VCL3dRQ01BQXdDd1lEVlIwUEJBUURBZ2JBTUEwR0NTVVNREF3TURF
d01EQXdNcUdTSWIzRFFFQkJRVUFBNElCQVFDL1EzRWdDM3FlQzdDNWpTNHRl
ZmVWUkZrVHV6eXBGdzZkNjNoUUdBM3NqVWUvSDlhZ01HMndCeDhBZzg2RTJF
ZWIzckp1VktWcVpVZkVBZmZYeTZhMXU3L0t4MEl6dWhUOG9Vw7NuIj48RW1pc29
yIHJmYz0iUk9BRDc3MTEyNUFNNCIgbm9tYnJlPSJEdWxjZSBCZXRoaW5hIFJvZHJp
Z3VleiBBbmdlbGVzIj48RG9taWNpbGlvRmlzY2FsIHBhaXM9Ik1YIiBjYWxsZT0iSnVsa
W8gQ29ydGF6YXIiIG11bmljaXBpbz0iR3VhZGFsdXBlIiBlc3RhZG89Ik51ZXZvIExlw7
NuIiBjb2RpZ29Qb3N0YWw9IjY3MTQwIiBub0V4dGVyaW9yPSIxMDAiIG5vSW50ZXJ
pb3I9IkMiIGNvbG9uaWE9IlJpbmNvbiBDb250cnkiIGxvY2FsaWRhZD0iR3VhZGFsdXBl
Ij48L0RvbWljaWxpb0Zpc2NhbD48RXhwZWRpZG9FbiBwYWlzPSJNWCI+uPjxSZWdp
bWVuRmlzY2FsIFJlZ2ltZW49IlJlZ2ltZW4gSW50ZXJtZWRpbyI+PC9SZWdpbWVuRml
zY2FsPjwvRW1pc29yPjxSZWNlcHRvciByZmM9IlhBWFgwMTAxMDEwMDAiPjwvU
mVjZXB0b3I+PENvbmNlcHRvcz48Q29uY2VwdG8gdW5pZGFkPSJzZXJ2KHMpIiBpb
XBvcnRlPSIxLjAwIiBjYW50aWRhZD0iMS4wMDAiIGRlc2NyaXBjaW9uPSJTZXJ2aW
NpbyBkZSBDb25zdWx0b3LDrWEgZGUgVmlhamVzIiB2YWxvclVuaXRhcmlvPSIxLjA
wIj48L0NvbmNlcHRvPjwvQ29uY2VwdG9zPjxJbXB1ZXN0b3M+PFRyYXNsYWRvcz4
8VHJhc2xhZG8gdGFzYT0iMTYuMDAiIGltcG9ydGU9IjAuMTYiIGltcHVlc3RvPSJJVkE
iPjwvVHJ3M+PC9JbXB1ZXN0b3M+PC9Db21wcm9iYW50ZThc2xhZG8+PC9UcmFzbG
Fb4=
</cfdi >
</sign>

```
        </command>
        <trID>
                <clTRID>123454321</clTRID>
        </trID>
</pac>
```

Como referencia, en la sección 9 se puede apreciar un CFDI sin codificar en base 64.

NOTA: INSIGNA no acepta comandos con una longitud mayor a 5 MB.

Respuesta

Por otro lado, los datos de respuesta, dentro del elemento <signData> son los siguientes:

- <signID> Folio fiscal generado por INSIGNA a la transacción del timbrado. Este identificador se puede utilizar posteriormente para realizar consultas
- <cfdi> XML del CFDI ya timbrado en formato base64

Ejemplo de una respuesta exitosa a un comando ***sign***:

```
<? xml version="1.0" encoding="UTF-8" standalone="no"?>
<pac xmlns="urn:ietf:params:xml:ns:pac-1.0">
<response>
        <result code="1000">
                <msg lang="es">Comando ejecutado exitosamente</msg>
        </result>
        <resData>
                <signData>
                        <signID>f47ac10b-58cc-4372-a567-0e02b2c3d479</signID>
                        <cfdi>
                        PC9FeHBlZGlkb0VPD94bmU9Im5vIj8+PENvbXByb2JhbnRlIHhtbG5zPSJodHR
wOi8vd3d3LnNhbWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiIHN0YW5kY
WxvdC5nb2IubXgvY2ZkLzMiIHhtbG5zOnhzaT0iaHR0cDovL3d3dy53My5vcmcvMjAw
MS9YWFMb2NhdGlvbj0iaHR0cDovL3d3dy5zYXQuZ29iLm14L2NmZC8zIGh0dHATU
xTY2hlbWEtaW5zdGFuY2UiIHhzaTpzY2hlb6Ly93d3cuc2F0LmdvYi5teC9zaXRpb19pbn
Rlcm5ldC9jZmQvMy9jZmR2MzIueHNkIiB2ZXJzaW9uPSIzLjIiIGZlY2hhPSIyMDEzLT
AzLTI2VDEyOjU3OjQ1IiBzZWxsbz0iQ25xN2IyaUdlYjN4dStSK1h5YjZ0bUJtbFJIS0l1
WHRtaFJGSWFRUnpOQkE3Ymk2VjFCcWFham9DN0RFa2NEalVXY2RMbXhwNnQy
WmpweWZGZ05tNDJBekc1Q25VMnA2dEtEeS82cEFlcVdnczFZNDYvVi9uanZ6MnBR
Mk5nbWo0M3hLMVNhWlh0RDVL1J5WVdOcHc3TnVJRlJ5YVdKMWRHRnlhV0V4S
HpBZEJna3Foa2lHOXcwQkNRRVdFR0ZqYjJSelFITmhkQzVuYjJJdWJYZ3hKakFrQmd
OVkJBa01IVUYyTGlCSWFXUmhiR2R2SURjM0xDQkRiMnd1SUVkMVpYSnlaWEp2T
VE0d0RBWURWUVFSREFVd05qTXdNREVMTUFrR0ExVUVCaE1DVFZneEdUQVhC
Z05WQkFnTUVFUnBjM1J5YVhSdklFWmxaR1Z5WVd3eEV6QVJCZ05WQkFjTUNrTj
FZWFZvZEdWdGIyTXhNekF4QmdrcWhraUc5dzBCQ1FJTUpGSmxjM0J2Ym5OaFlteG
xPaUJHWlhKdVlXNWtieUJOWVhKMHc2MXVaWG9nUTI5emN6QWVGdzB4TVRFd0
16QXlNalV4TXpCYUZ3MHhNekV3TWpreU1qVXhNekJhTUlIUk1TZ3dKZ1lEVlFRRE
V4OUVWVXhEUlNCQ1JWUklTVTVCSUcEhlTW9JaFluRmdDVjNSeU9EdnZ5SHYwd
DU4VnZZPSIgdG90YWw9IjEuMTYiIHN1YlRvdGFsPSIxLjAwIiBjZXJ0aWZpY2Fkbz0i
TUlJRU1UQ0NBeG1nQXdJQkFnSREF4TURRMk1qQXhNekF3RFFZSktvWklodmNOQ
VFFRkJRQXdnZ0UyTVRnd05nWURWUVFEREM5QkxrTXVJR1JsYkNCVFpYSjJhV05
wYnlCa1pTQkJaRzFwYm1semRISmhZMm5EczI0Z1ZISnBIGVuIHVuYSBzb2xhIGV4a
GliaWNpw7NuIiBub0NlcnRpZmljYWRvPSIwMDAwMTAwMDAwMDEwNDYyMDEz
MCIgdGlwb0RlQ29tcHJvYmFudGU9ImluZ3Jlc28iGRlc2N1ZW50bz0iMC4wMCIgbWV
0b2RvRGVQYWdvPSJFZmVjdGl2byIgTHVnYXJFeHBlZGljaW9uPSJHdWFkYWx1cG
UsIE51ZXZvIExlSU0JDUlZSSVNVNUJJRkpQUkZKSlIxVkZXaUJCVGtkRlRFVlRNU2
d3SmdZRFZRUUtFeDlFVlV4RFJTQkNSVlJJU1U1QklGSlBSRkpKUjFWRldpQkJUa2R
GVEVWVE1SWXdGQVlEVlFRdEV3MVNUMEZFTnpjeE1USTFRVTAwTVJzd0dRWU
RWUVFGRXhKU1QwRkVOemNZblYwWVhKcFlURXZNQzBHQTFVRUNnd21VMlZ5
ZG1samFXOGdaR1VnUVdSdGFXNXBjMZKUFJGSkpSMVZGV2lCQlRrZEZURVZUT
VNnd0pnWURWUVFwRXg5RVZVeERMDQ4ejZsc2d2SkRhTHcxK3NnY2VJK2h5cm5
1TWRNbmZUNjhzZWtJb0wrY0tvQzU0QjhMYzArMVNxOUxqRVhta3VCMHJsY0hxcE
J4cEljWEJvL3I5UnQ2c24rUnFRKzNlNVpzb1FmSTF5L2lvSDlaVzhnaEMzRHNXcHlkV
U5wZHJmdGJHVkFpcDVXUXp2VlpVVWI1azFZNldoUUpuTHBSengvbmRpNS9WTzY
rSjRhSWJTUy9xZmlXRjQxOHg1RWt2SHBDL2gwNWhVNEU2eWU1Ymp0bHE5K1M3
```

TjZqNlV4L1Y3TWVKTDArOXpxVzhsY3FZRUoiIGZvcm1hRGVQYWdvPSJQYWdv4T
VRJMVRVZFVSRTVNTURFeEhEQWFCZ05WQkFzVUUwMWhkSEpwZWlCTVlYTW
dRVzNwY21sallYTXdnWjh3RFFZSktvWklodmNOQVFFQkJRQURnWTBBTUlHSkFvR
0JBTDUvcTlldEtDako4ZkhERVZJNHhVd2ZWMDJMcEdJZXlwTmxwZFMwSHlRUUdv
ampkWUdnYW50dC9TRmgrcVhOb050dTZWS0ZOeGQzYlI0bzFpbHNNSHFFYVR3QV
FzY0RrR2ZpOUgwZFJZZVUrZlo3eVQ1b2FrUkVkVHNWSWRPUDc3b1pqTXFUVjNV
d3lDUW5GbUM0bi9zSnc2eTB0M09lWGdBd2ZzWFQ2dld4QWdNQkFBR2pIVEFiTUF3
R0ExVWRFd0VCL3dRQ01BQXdDd1lEVlIwUEJBUURBZ2JBTUEwR0NTVVNREF3T
URFd01EQXdNcUdTSWIzRFFFQkJRVUFBNElCQVFDL1EzRWdDM3FlQzdDNWpTN
HRlZmVWUkZrVHV6eXBGdzZkNjNoUUdBM3NqVWUvSDlhZ01HMndCeDhBZzg2R
TJFZWIzckp1VktWcVpVZkVBZmZYeTZhMXU3L0t4MEl6dWhUOG9Vw7NuIj48RW1
pc29yIHJmYz0iUk9BRDc3MTEyNUFNNCIgbm9tYnJlPSJEdWxjZSBCZXRoaW5hIFJvZ
HJpZ3VleiBBbmdlbGVzIj48RG9taWNpbGlvRmlzY2FsIHBhaXM9Ik1YIiBjYWxsZT0iSn
VsaW8gQ29ydGF6YXIiIG11bmljXBpbz0iR3VhZGFsdXBlIiBlc3RhZG89Ik51ZXZvIExl
w7NuIiBjb2RpZ29Qb3N0YWw9IjY3MTQwIiBub0V4dGVyaW9yPSIxMDAiIG5vSW50Z
XJpb3I9IkMiIGNvbG9uaWE9IlJpbmNvbiBDb250cnkiIGxvY2FsaWRhZD0iR3VhZGFsd
XBlIj48L0RvbWljaWxpb0Zpc2NhbD48RXhwZWRpZG9FbiBwYWlzPSJNWCI+uPjxSZ
WdpbWVuRmlzY2FsIFJlZ2ltZW49IlJlZ2ltZW4gSW50ZXJtZWRpbyI+PC9SZWdpbWVu
RmlzY2FsPjwvRW1pc29yPjxSZWNlcHRvciByZmM9IlhBWFgwMTAxMDEwMDAiPjw
vUmVjZXB0b3I+PENvbmNlcHRvcz48Q29uY2VwdG8gdW5pZGFkPSJzZXJ2KHMpIiB
pbXBvcnRlPSIxLjAwIiBjYW50aWRhZD0iMS4wMDAiIGRlc2NyaXBjaW9uPSJTZXJ2a
WNpbyBkZSBDb25zdWx0b3LDrWEgZGUgVmlhamVzIiB2YWxvclVuaXRhcmlvPSIxLj
AwIj48L0NvbmNlcHRvPjwvQ29uY2VwdG9zPjxJbXB1ZXN0b3M+PFRyYXNsYWRvcz
48VHJhc2xhZG8gdGFzYT0iMTYuMDAiIGltcG9ydGU9IjAuMTYiIGltcHVlc3RvPSJJVk
EiPjwvVHJ3M+PC9JbXB1ZXN0b3M+PC9Db21wcm9iYW50ZThc2xhZG8+PC9UcmFzb
GFkb4=
</cfdi >
</signData>
</resData>
<trID>
<svTRID>11336</svTRID>
</trID>
<stats>
<svTimestamp>2018-01-01T15:19:18</svTimestamp>
<cmdReceived>2018-01-01T15:19:18</cmdReceived>
<cmdExecutionTime>1126</cmdExecutionTime>
<clConnectionTime>33100</clConnectionTime>
<clOpenConnections>1</clOpenConnections>
</stats>
</response>
</pac>

3.1.1 CFDIs enviados en más de una ocasión.

Cuando INSIGNA detecta que el CFDI a timbrar ya se encuentra en su Base de Datos (esto es, que la cadena original del CFDI a timbrar es idéntica a un CFDI que ya timbró INSIGNA), regresará un error 307: Error de Timbre Previo.

Los datos que conforman la cadena original de un CFDI se pueden verificar en los siguientes documentos:

- CFDI V3.3 se puede verificar en el Anexo 20 del SAT: http://www.dof.gob.mx/nota_detalle.php?codigo=5468849&fecha=10/01/2017.
- Retención Pago V1.0 se puede verificar en el Estándar de Retenciones y pago del SAT :

http://www.sat.gob.mx/informacion_fiscal/factura_electronica/Paginas/cfdi_retenciones_pagos.aspx.

```
<? xml version="1.0" encoding="UTF-8" standalone="no"?>
<pac xmlns="urn:ietf:params:xml:ns:pac-1.0">
        <response>
                <result code="307">
                        <msg lang="es">
                        El CFDI contiene un timbre previo
                        </msg>
                        <value>
                            <cfdi>
                PC9FeHBlZGlkb0VPD94bmU9Im5vIj8+PENvbXByb2JhbnRlIHhtbG5zPSJodHRwOi8vd
                …(resto del CFDI en base 64)…
                +PC9JbXB1ZXN0b3M+PC9Db21wcm9iYW50ZThc2xhZG8+PC9UcmFzbGFkb4=
                            </cfdi>
                        </value>
                </result>
                <trID>
                        <svTRID>11355</svTRID>
                </trID>
                <stats>
                        <svTimestamp>2018-01-01T15:19:18</svTimestamp>
                        <cmdReceived>2018-01-01T15:19:18</cmdReceived>
                        <cmdExecutionTime>330</cmdExecutionTime>
                </stats>
        </response>
</pac>
```

3.1.2 Ejemplos de CFDIs con datos incorrectos.

Si el CFDI enviado a INSIGNA no cumple con alguno de los requisitos del SAT, el Servicio de Timbrado rechazará la solicitud de timbrado. Por ejemplo, cuando el RFC del emisor no está dado de alta ante el SAT el Servicio responde lo siguiente:

```
<? xml version="1.0" encoding="UTF-8" standalone="no"?>
<pac xmlns="urn:ietf:params:xml:ns:pac-1.0">
        <response>
                <result code="402">
                        <msg lang="es">
                        RFC del emisor no se encuentra en el régimen de contribuyentes
                        </msg>
                </result>
                <trID>
                        <svTRID>11336</svTRID>
                </trID>
                <stats>
                        <svTimestamp>2018-01-01T15:19:18</svTimestamp>
                        <cmdReceived>2018-01-01T15:19:18</cmdReceived>
                        <cmdExecutionTime>130</cmdExecutionTime>
                        <clConnectionTime>33100</clConnectionTime>
                        <clOpenConnections>1</clOpenConnections>
                </stats>
        </response>
</pac>
```

Otra posible causa de error es que el cliente no ha adquirido timbres con INSIGNA o bien su saldo se ha agotado:

```
<? xml version="1.0" encoding="UTF-8" standalone="no"?>
<pac xmlns="urn:ietf:params:xml:ns:pac-1.0">
	<response>
		<result code="2309">
			<msg lang="es">Saldo de timbres insuficiente</msg>
		</result>
		<trID>
			<svTRID>11353</svTRID>
		</trID>
		<stats>
			<svTimestamp>2018-01-01T16:02:07</svTimestamp>
			<cmdReceived>2018-01-01T16:02:07</cmdReceived>
			<cmdExecutionTime>153</cmdExecutionTime>
			<clConnectionTime>33100</clConnectionTime>
			<clOpenConnections>1</clOpenConnections>

		</stats>
	</response>
</pac>
```

3.2. Cancelación de CFDIs

Un cliente de INSIGNA puede cancelar un CFDI que haya sido timbrado previamente con INSIGNA o con un proveedor externo, la cancelación de CFDI externos consume un timbre. Para cancelar su CFDI, el cliente necesita enviar un comando cancel al Servicio de Timbrado. Los parámetros del comando cancel son los siguientes:

- <uuid> El UUID del CFDI timbrado
- <date La fecha de cancelación del CFDI
- <signature> El signatureValue requerido por el SAT. Para más detalles revisar el Anexo 9 de este manual
- <certificate> El contenido del certificado en una cadena de texto codificada en base64 y en charset UTF-8
- <cfdiTotal> Atributo requerido solo si se cancela un CFDI externo. Monto total del CFDI
- <cfdiType> Atributo requerido solo si se cancela un CFDI externo. Identificador del tipo del comprobante CFDI [I (Ingreso), E (Egreso), N (Nomina), T (Traslado), P (Pagos)]
- <receptorRfc> Atributo requerido solo si se cancela un CFDI externo. EL RFC del receptor del CFDI

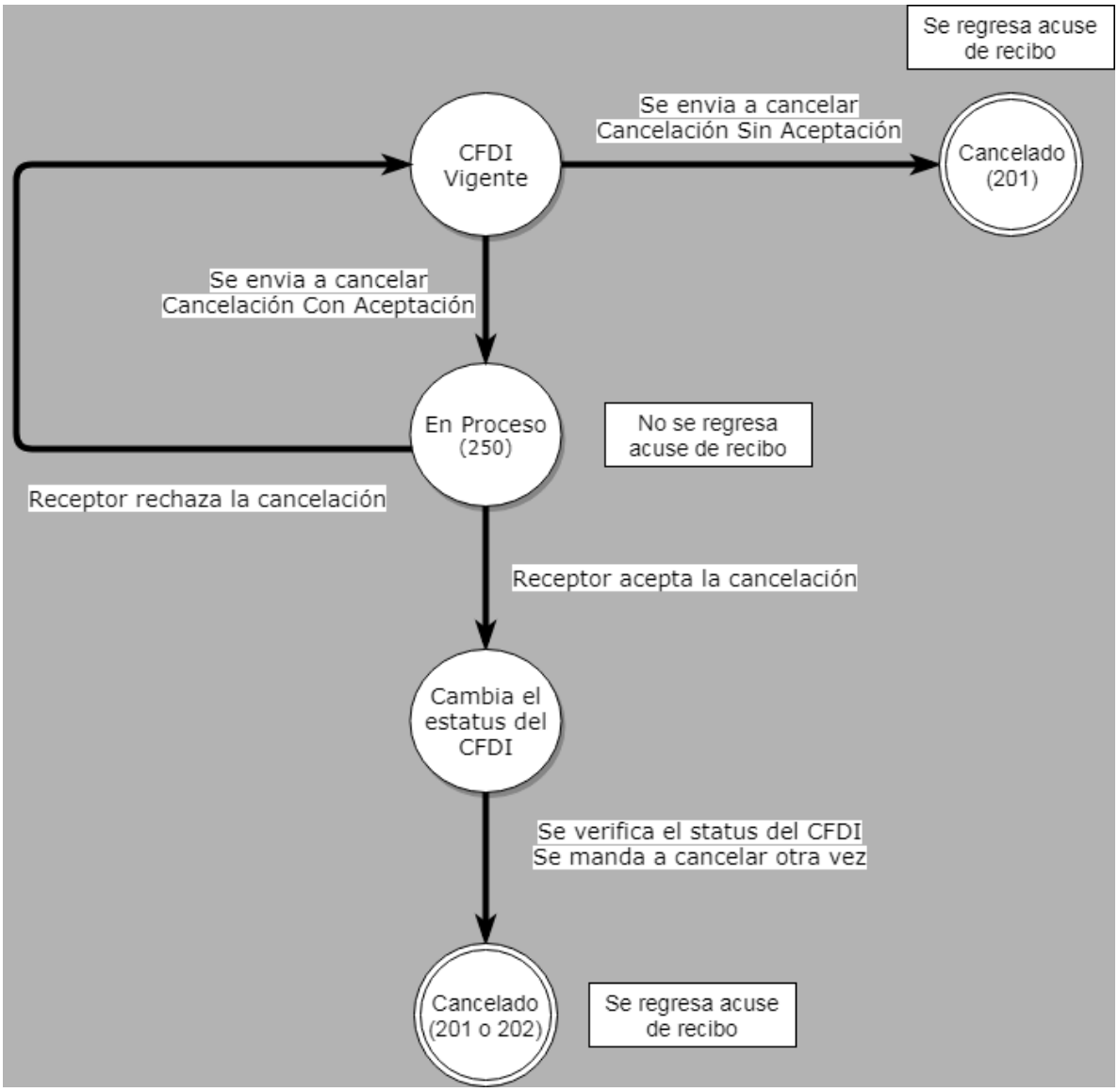
```
<? xml version="1.0" encoding="UTF-8" standalone="no"?>
<pac xmlns="urn:ietf:params:xml:ns:pac-1.0">
	<command>
		<cancel>
			<date>2018-01-01T16:34:52</date>
			 <uuid>f47ac10b-58ee-4372-a567-0x02b4c5a468</uuid>
		<signature>SEdMWWlqM0JsOEZCMlBFV2dzeFJub2FLVTM4SWR0YXA1dU9hSmJTR
		XN2T1lnZjIyTkNmNm5QSkxPa3QXJIUlhiaDJJKzFhR2daR2ZuUmM1N2VPNzdyelBvZi
		tVbzdLZFN5Vm5iMGdESW0xUlJtU0t0SVhtTklTYTUzb2R1OVFodC9IT0Y5eXkwdVAz
		c2s4WnFYdUdwRWJuZnU0NlhiZXlpWk10R0JHNG93pjPQ==</signature>

		<certificate>MIIEMTCCAxmgAwIBAgIUMDAwMDEwMDAwMDAxMDQ2MjAxMzA
```

```
                wDQYJKoZIhvcNAQEFBQAwggE2MTgwNgYDVQQDDC9BLkMuIGRlbCBTZXJ2aW
                NpbyBkZSBBZG1pbmlzdHJhY2nDs24gVHJpYnV0YXJpYTEvMC0GA1UECgwmU2Vy
                dmljaW8gZGUgQWRtaW5pc3RyYWNpw7NuIFRyaWJ1dGFyaWExHzAdBgkqhkiG9w0
                BCQEWEGFjb2RzQHNhdC5nb2IubXgxJjAkBgNVBAkMHUF2LiBIaWRhbGdvIDc3LC
                BDb2wuIEd1ZXJyZXJvMQ4wDAYDVQQRDAUwNjMwMDELMAkGA1UEBhMCTVg
                xGTAXBgNVBAgMEERpc3RyaXRvIEZlZGVyYWwxEzARBgNVBAcMCkN1YXVodG
                Vtb2MxMzAxBgkqhkiG9w0BCQIMJFJlc3BvbnNhYmxlOiBGZXJuYW5kbyBNYXJ0w6
                +fZ7yT5oakREdTsVIdOP77oZjMqTV3UwyCQnFmC4n/sJw6y0t3OeXgAwfsXT6vWxAg
                MBAAGjHTAbMAwGA1UdEwEB/wQCMAAwCwYDVR0PBAQDAgbAMA0GCSqGSI
                b3DQEBBQUAA4IBAQC/Q3EgC3qeC7C5jS4tefeVRFkTuzypFw6d63hQGA3sjUe/H9ag
                MG2wBx8Ag86E2Eeb3rJuVKVqZUfEAffXy6a1u7/Kx0IzuhT8o
                1uZXogQ29zczAeFw0xMTEwMzAyMjUxMzBaFw0xMzEwMjkyMjUxMzBaMIHRMSg
                wJgYDVQQDEx9EVUxDRSBCRVRISU5BIFJPRFJJR1VFWiBBTkdFTEVTMSgwJgYD
                VQQpEx9EVUxDRSBCRVRISU5BIFJPRFJJR1VFWiBBTkdFTEVTMSgwJgYDVQQK
                Ex9EVUxDRSBCRVRISU5BIFJPRFJJR1VFWiBBTkdFTEVTMRYwFAYDVQQtEw1S
                T0FENzcxMTI1QU00MRswGQYDVQQFExJST0FENzcxMTI1TUdURE5MMDExHDAa
                BgNVBAsUE01hdHJpeiBMYXMgQW3pcmljYXMwgZ8wDQYJKoZIhvcNAQEBBQAD
                gY0AMIGJAoGBAL5/q9etKCjJ8fHDEVI4xUwfV02LpGIeypNlpdS0HyQQGojjdYGgantt
                /SFhLc0+1Sq9LjEXmkuB0rlcHqpBxpIcXBo/r9Rt6sn+RqQ+qXNoNtu6VKFNxd3bR4o1il
                sMHqEaTwAQscDkGfi9H0dRYeUU048z6lsgvJDaLw1+sgceI+hyrnuMdMnfT68sekIoL+c
                KoC54B8+3e5ZsoQfI1y/ioH9ZW8ghC3DsWpydUNpdrftbGVAip5WQzvVZUUb5k1Y6W
                hQJnLpRzx/ndi5/VO6+J4aIbSS/qfiWF418x5EkvHpC/h05hU4E6ye5bjtlq9+S7N6j6
                Ux/V7MeJL0+9zqW8lcqYEJ</certificate>
                <cfdiTotal>0<cfdiTotal>
                <cfdiType>I<cfdiType>
                <receptorRFC> LAN7008173R5<receptorRFC>
                </cancel >
        </command>
</pac>
```

El proceso de cancelación se aplicara a todos los CFDIs excepto a los que tengan alguna de estas características:

- CFDI que aparen ingresos hasta $5,000.00
- Nómina
- Egreso
- Traslado
- Emitido desde el portal web del SAT “Mis Cuentas”
- CFDI de retenciones e información de pagos
- CFDI de Ingreso expedidos a contribuyentes el RIF
- Dentro de los tres días siguientes a la emisión
- Operaciones con público en general
- Recibidos por residentes en el extranjero
- CFDI a través del adquiriente y sector primario
- CFDI emitido por integrantes del Sistema Financiero

Cuando se manda un CFDI a cancelar se recibirá un código de resultado proveniente del SAT (201 o 250) y dependiendo del tipo de cancelación que se realice (Cancelación sin aceptación o Cancelación con aceptación) se devolverá un acuse de recibo. Como se muestra a continuación:



Con acuse de recibo

```
<? xml version="1.0" encoding="UTF-8" standalone="no"?>
<pac xmlns="urn:ietf:params:xml:ns:pac-1.0">
<response>
        <result code="201">
                <msg lang="es">UUID Cancelado exitosamente</msg>
        </result>
        <resData>
                <cancelData>
                        <uuid>f47ac10b-58ee-4372-a567-0x02b4c5a468</uuid>
                        <receipt>PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0idXRmLTgiPz48czp
                FbnZlbG9wZSB4bWxuczpzPSJodHRwOi8vc2NoZW1hcy54bWxzb2FwLm9yZy9zb2FwL
                2VudmVsb3BlLyI+PHM6Qm9keSB4bWxuczp4c2Q9Imh0dHA6Ly93d3cudzMub3JnLzIw
                MDEvWE1MU2NoZW1hIiB4bWxuczp4c2k9Imh0dHA6Ly93d3cudzMub3JnLzIwMDEv
                WE1MU2NoZW1hLWluc3RhbmNlIj48Q2FuY2VsYUNGRFJlc3BvbnNlIHhtbG5zPSJod
                HRwOi8vY2FuY2VsYWNmZC5zYXQuZ29iLm14Ij48Q2FuY2VsYUNGRFJlc3VsdCBG
```

ZWNoYcml0aG09Imh0dHA6Ly93d3cudzMub3JnL1RSLzE5OTkvUkVDLXhwYXRT0iM
jAxMy0wMy0yNlQxNjoxMzoxNy41ODUzNDEzIiBSZmNFbWlzb3I9IlJPQUQ3NzExMj
VBTTQiPjxGb2xpb3M+PFVVSUQ+RjU5MkU0MDgtN0VCRi00NENGLUFGQTQtNTV
DOENBNEM2NkZBPC9VVUlEPjxFc3RhdHVzVVVJRD4yMDE8L0VzdGF0dXNVVUl
EPjwvRm9saW9zPjxTaWduYXR1cmUgeG1sbnM9Imh0dHA6Ly93d3cudzMub3JnLzIw
MDAvMDkveG1sZHNpZyMiIElkPSJTZWxsb1NBVCI+PFNpZ25lZEluZm8+PENhbm9u
aWNhbGl6YXRpb25NZXRob2QgQWxnb3JpdGhtPSJodHRwOi8vd3d3LnczLm9yZy9UU
i8yMDAxL1JFQy14bWwtYzE0bi0yMDAxMDMxNSI+PC9DYW5vbmljYWxpemF0aW9
uTWV0aG9kPjxTaWduYXR1cmVNZXRob2QgQWxnb3JpdGhtPSJodHRwOi8vd3d3Lncz
Lm9yZy8yMDAxLzA0L3htbGRzaWctbW9yZSNobWFjLXNoYTUxMiI+PC9TaWduYX
R1cmVNZXRob2Q+PFJlZmVyZW5jZSBVUkk9IiI+PFRyYW5zZm9ybXM+PFRyYW5z
Zm9ybSBBbGdvcml0aG09Imh0dHA6Ly93d3cudzMub3JnL1RSLzE5OTkvUkVDLXhwY
XRoLTE5OTkxMTE2Ij48WFBhdGg+bm90KGFuY2VzdG9yLW9yLXNlbGY6OipbbG9j
YWwtbmFtZSgpPSdTaWduYX0hiUVY3TDdIMFBQZkk0MTIzaU16VXNVWGEyRGRC
S1ZlbR1cmUnXSk8L1hQYXRoPjwvVHJhbnNmb3JtPjwvVHJhbnNmb3Jtcz48RGlnZXN
0TWV0aG9kIEFsZ29yaXRobT0iaHR0cDovL3d3dy53My5vcmcvMjAwMS8wNC94bWxl
bmMjc2hhNTEyIj48L0RpZ2VzdE1ldGhvZD48RGlnZXN0VmFsdWU+d1E3TXFrVlhoayt
lQkNMN1FrVEdpS3BSUkpDS1dSQW04SURuZzNseDlDUVY4TXkvdVNYV0NTU244
MHVybW51d3JRVllKZytINFRGVWFqejd5VTErZ1E9PTwvRGlnZXN0VmFsdWU+PC9
SZWZlcmVuY2U+PC9TaWduZWRJbmZvPjxTaWduYXR1NHlFTT08L01vZHVsdXM+P
EV4cG9uZW50PkFRQUI8cmVWYWx1ZT5ITkE5WGRJOWpvZGlhVnhjd3hrWDZhS1V
TTDQrU3hWOXExTVRtT1phY2tsZEE2Y2U1bHcrQjVJeHpOT0dTZ3lGZkRoZnkwZStJ
OWsxSkl3TVlaWWt6QT09PC9TaWduYXR1cmVWYWx1ZT48S2V5SW5mbz48S2V5T
mFtZT4wMDAwMTA4ODg4ODgwMDAwMDAwMzwvS2V5TmFtZT48S2V5VmFsdW
U+PFJTQUtleVZhbHVlPjxNb2R1bHVzPjVXOFBOdWdML0hiUVY3TDdIMFBQZkk0M
TIzaU16VXNVWGEyRGRCS1ZlbXlHV0dGZGpobnpzK0xMZFU0Qm5LbmUyVU1CSF
ByT0UwbjJySzQ0RGZkVEZMQmdNaFJoekxzc3RpYUM0ck1zbFc1QVdsL2RYd2d2YTJ
FVlZoRkF1VFAzMUxBR1Y1c2hrYlBicDc1WkNyZUZFMDByMTRvUXY0RXBtWnVv
eGh6NHlFTT08L01vZHVsdXM+PEV4cG9uZW50PkFRQUI8L0V4cG9uZW50PjwvUlN
BS2V5VmFsdWU+PC9LZXlWYWx1ZT48L0tleUluZm8+PC9TaWduYXR1cmU+PC9DY
W5jZWxhQ0ZEUmVzdWx0PjwvQ2FuY2VsYUNGRFJlc3BvbnNlPjwvczpCb2R5Pjwvcz
pFbnZlbG9wZT4=</receipt>
</cancelData>
</resData>
<trID>
<svTRID>11336</svTRID>
</trID>
<stats>
<svTimestamp>2018-01-01T15:19:18</svTimestamp>
<cmdReceived>2018-01-01T15:19:22</cmdReceived>
<cmdExecutionTime>3626</cmdExecutionTime>
<clConnectionTime>33100</clConnectionTime>
<clOpenConnections>1</clOpenConnections>
</stats>
</response>
</pac>

Sin acuse de recibo

```
<? xml version="1.0" encoding="UTF-8" standalone="no"?>
<pac xmlns="urn:ietf:params:xml:ns:pac-1.0">
<response>
        <result code="250">
                <msg lang="es">Factura en proceso de cancelación</msg>
        </result>
        <resData>
                <cancelData>
                        <uuid>f47ac10b-58cc-4372-a567-0e02b2c3d479</uuid>
                        <receipt> </receipt>
                </cancelData>
        </resData>
        <trID>
                <svTRID>11336</svTRID>
        </trID>
        <stats>
                <svTimestamp>2018-01-01T15:19:18</svTimestamp>
                <cmdReceived>2018-01-01T15:19:22</cmdReceived>
                        <cmdExecutionTime>3626</cmdExecutionTime>
                        <clConnectionTime>33100</clConnectionTime>
                        <clOpenConnections>1</clOpenConnections>
        </stats>
</response>
</pac>
```


Para obtener el acuse de recibo cuando se obtenga un código 250 es necesario saber el estatus en el que se encuentra la cancelación, para ello se dan dos opciones:

1. Configurar el servicio de **Notificaciones de estatus de cancelación** en https://www.insigna.mx/, la cual se encuentra en la sección **Mi Cuenta** > **Configuración Adicional** en el cual se deberá ingresar el correo en el cual se desea recibir las notificaciones del cambio de estatus y seleccionar el checkbox para activar el servicio.
2. Usar el comando de **Consulta de estatus de CFDI** el cual regresara el estatus actual del CFDI.

Cuando el estatus de la cancelación pase a **Cancelación aceptada** o **Cancelación por vencimiento** se deberá enviar otra vez el comando de cancelación el cual ahora regresara un código 201 o 202 y el cual tendrá en el elemento de <receipt> el acuse de recibo (Cabe a mencionar que en algunos casos en particular no podrá ser posible recuperar el recibo de cancelación).

Para obtener el acuse de recibo por parte del SAT es necesario decodificar el texto en base 64 dentro del elemento <cancelData>. La codificación o decodificación en base 64 es bastante común y existen varias herramientas en línea para realizarlas, por ejemplo http://www.base64decode.org/

Al intentar cancelar un CFDI es posible que el Servicio de Timbrado de INSIGNA responda con un resultado erróneo. Por ejemplo, en este caso el CFDI fue cancelado anteriormente:

```
<? xml version="1.0" encoding="UTF-8" standalone="no"?>
<pac xmlns="urn:ietf:params:xml:ns:pac-1.0">
	<response>
		<result code="205">
			<msg lang="es">
				UUID No existe
			</msg>
		</result>
		<trID>
			<svTRID>11336</svTRID>
		</trID>
		<stats>
			<svTimestamp>2018-01-01T15:19:18</svTimestamp>
			<cmdReceived>2018-01-01T15:19:22</cmdReceived>
			<cmdExecutionTime>3626</cmdExecutionTime>
			<clConnectionTime>33100</clConnectionTime>
			<clOpenConnections>1</clOpenConnections>
		</stats>
	</response>
</pac>
```

3.3.Consulta de CFDIs

El comando info permite consultar un CFDI timbrado anteriormente en INSIGNA, y los parámetros son los siguientes:

- <signID> Folio fiscal del CFDI a consultar


```
<? xml version="1.0" encoding="UTF-8" standalone="no"?>
<pac xmlns="urn:ietf:params:xml:ns:pac-1.0">
	<command>
		<info>
			<signID>f47ac10b-58cc-4372-a567-0e02b2c3d479</signID>
		</info>
	</command>
</pac>
```

Por otro lado, los datos de respuesta, o <signData> son los siguientes:

- <signID> Folio fiscal del CFDI consultado
- <cfdi> XML del CFDI ya timbrado en formato base64
- <status> Estatus del CFDI (Activo o Cancelado)

Ejemplo de una respuesta exitosa a un comando info:

```
<? xml version="1.0" encoding="UTF-8" standalone="no"?>
<pac xmlns="urn:ietf:params:xml:ns:pac-1.0">
<response>
	<result code="1000">
		<msg lang="es">Comando ejecutado exitosamente</msg>
	</result>
	<resData>
		<infoData>
			<signID>f47ac10b-58cc-4372-a567-0e02b2c3d479</signID>
```

```
                        <cfdi>
                                #XML en base 64 del CFDI ya timbrado#
                        </cfdi >
                        <status>Activo</status>
                </</ infoData >
        </resData>
        <trID>
                <svTRID>11336</svTRID>
        </trID>
        <stats>
                <svTimestamp>2018-01-01T15:19:18</svTimestamp>
                <cmdReceived>2018-01-01T15:19:22</cmdReceived>
                <cmdExecutionTime>3626</cmdExecutionTime>
                <clConnectionTime>33100</clConnectionTime>
                <clOpenConnections>1</clOpenConnections>
        </stats>
</response>
</pac>
```

Como ya se mencionó, la decodificación de base 64 es relativamente sencilla, y se puede observar un ejemplo en la siguiente página: http://www.base64decode.org/

También es posible que la consulta de CFDIs a INSIGNA no regrese un resultado exitoso. Por ejemplo, en este caso el UUID enviado no existe:

```
<? xml version="1.0" encoding="UTF-8" standalone="no"?>
<pac xmlns="urn:ietf:params:xml:ns:pac-1.0">
        <response>
                <result code="205">
                        <msg lang="es">UUID no existe</msg>
                </result>
                <trID>
                        <svTRID>11328</svTRID>
                </trID>
                <stats>
                        <svTimestamp>2018-01-01T23:18:55</svTimestamp>
                        <cmdReceived>2018-01-01T23:18:55</cmdReceived>
                        <cmdExecutionTime>92</cmdExecutionTime>
                </stats>
        </response>
</pac>
```

3.4. Recuperación de CFDIs

En algunas circunstancias (fallos de red del lado de INSIGNA o del cliente, por ejemplo) es posible que el cliente no reciba la confirmación de que su CFDI fue timbrado correctamente.

INSIGNA ofrece la recuperación de dichos CFDIs mediante el comando <verify>, el cual funciona de la siguiente manera:

a) El cliente envía la cadena original codificada en base 64 del CFDI que desea recuperar

```
<? xml version="1.0" encoding="UTF-8" standalone="no"?>
<pac xmlns="urn:ietf:params:xml:ns:pac-1.0">
        <command>
                <verify>
                <chain>
                Q1VPVEFTIElNU1MiIEltcG9ydGVFeGVudG89IjU5LjYyIiBJbXBvcnRlR3JhdmFkbz0i
                MC4wMCIgVGlwb0RlZHVjY2lvbj0iMDAxIi8+PG5vbWluYTpEZWR1Y2Npb24gQ2xhd
                mU9IjUyNSIgQ29uY2VwdG89IkZPTkRPIERFIEFIT1JSTyBFTVBMRUFETyIgSW1wb3
                J0ZUV4ZW50bz0iMjM4LjQ1IiBJbXBvcnRlR3JhdmFkbz0iMC4wMCIgVGlwb0RlZHVj
                Y2lvbj0iMDA0Ii8+
                </chain>
                </verify>
        </command>
</pac>
```

La información necesaria para construir la cadena original de un CFDI está disponible en:

- CFDI V3.3. Apartado B del Anexo 20 del SAT: http://www.dof.gob.mx/nota_detalle.php?codigo=5468849&fecha=10/01/2017
- Retención Pago V1.0. Apartado 2 del Estándar de Retenciones y pago del SAT: http://www.sat.gob.mx/informacion_fiscal/factura_electronica/Documents/retenciones/RetencionPago.pdf

b) El cliente envía el id de transacción del comando en el que originalmente mandó a timbrar el CFDI

```
<? xml version="1.0" encoding="UTF-8" standalone="no"?>
<pac xmlns="urn:ietf:params:xml:ns:pac-1.0">
        <command>
                <verify>
                        <signClTRID>123454321</signClTRID>
                </verify>
        </command>
</pac>
```

En este ejemplo se desea recuperar el CFDI timbrado en la sección 3.1, el cual tiene un id de transacción (<clTRID>) de 123454321, mismo que se proporciona en el elemento <signClTRID> del comando <verify>.

Es responsabilidad del cliente manejar ids de transacciones únicos, ya que si existen dos o más CFDIs asociados a un mismo id de transacción del cliente, INSIGNA no podrá garantizar que el CFDI recuperado es el correcto.

Los incisos a y b son excluyentes. Es decir, el comando <verify> solo admite la cadena o el id, no ambos a la vez. Por otro lado, los datos de respuesta, o <signData> son los siguientes:

- <signID> Folio fiscal del CFDI consultado
- <cfdi> XML del CFDI ya timbrado en formato base64
- <status> Estatus del CFDI (Activo o Cancelado)

Ejemplo de una respuesta exitosa a un comando verify:

```
<? xml version="1.0" encoding="UTF-8" standalone="no"?>
<pac xmlns="urn:ietf:params:xml:ns:pac-1.0">
        <command>
        <resData>
                <verifyData>
                        <signID>f47ac10b-58cc-4372-a567-0e02b2c3d479</signID>
                        <cfdi>
                                #XML en base 64 del CFDI ya timbrado#
                        </cfdi >
                        <status>Activo</status>
                </verifyData>
        </resData>
        <trID>
                <svTRID>11336</svTRID>
        </trID>
        <stats>
                <svTimestamp>2018-01-01T15:19:18</svTimestamp>
                <cmdReceived>2018-01-01T15:19:22</cmdReceived>
                <cmdExecutionTime>3626</cmdExecutionTime>
                <clConnectionTime>33100</clConnectionTime>
                <clOpenConnections>1</clOpenConnections>
        </stats>
        </command>
</pac>
```

Si la cadena original o el id de transacción no corresponden a un CFDI que haya sido timbrado por el cliente que envía el comando, INSIGNA responderá que no encontró dicho CFDI en su Base de Datos:

```
<? xml version="1.0" encoding="UTF-8" standalone="no"?>
<pac xmlns="urn:ietf:params:xml:ns:pac-1.0">
        <response>
                <result code="205">
                        <msg lang="es">UUID no existe</msg>
                </result>
                <trID>
                        <svTRID>11328</svTRID>
                </trID>
                <stats>
                        <svTimestamp>2018-01-01T23:18:55</svTimestamp>
                        <cmdReceived>2018-01-01T23:18:55</cmdReceived>
                        <cmdExecutionTime>92</cmdExecutionTime>
                </stats>
        </response>
</pac>
```

3.5. Validación de CFDIs

El comando ***validate*** se utiliza para validar la estructura de un CFDI y su existencia en los sistemas del SAT, la validación de CFDI consume un timbre. Dentro del elemento ***validate*** se requiere solamente el siguiente parámetro:

- <cfdi> Indica que dentro de este elemento se encuentra el CFDI a validar ante el SAT. INSIGNA no pide el XML del CFDI en texto plano, sino codificado en formato base64, como se observa en el siguiente ejemplo:

```
<? xml version="1.0" encoding="UTF-8" standalone="no"?>
<pac xmlns="urn:ietf:params:xml:ns:pac-1.0">
        <command>
                <validate>
                        <cfdi>
                PC9FeHBlZGlkb0VPD94bmU9Im5vIj8+PENvbXByb2JhbnRlIHhtbG5zPSJodHRwOi8vd
                3d3LnNhbWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiIHN0YW5kYWxvdC
                5nb2IubXgvY2ZkLzMiIHhtbG5zOnhzaT0iaHR0cDovL3d3dy53My5vcmcvMjAwMS9Y
                WFMb2NhdGlvbj0iaHR0cDovL3d3dy5zYXQuZ29iLm14L2NmZC8zIGh0dHATUxTY2hl
                bWEtaW5zdGFuY2UiIHhzaTpzY2hlb6Ly93d3cuc2F0LmdvYi5teC9zaXRpb19pbnRlcm5l
                dC9jZmQvMy9jZmR2MzIueHNkIiB2ZXJzaW9uPSIzLjAiIGZlY2hhPSIyMDEzLTAzLTI2
                VDEyOjU3OjQ1IiBzZWxsbz0iQ25xN2IyaUdlYjN4dStSK1h5YjZ0bUJtbFJIS0I1WHRtaF
                JGSWFRUnpOQkE3Ymk2VjFCcWFham9DN0RFa2NEalVXY2RMbXhwNnQyWmpwe
                WZGZ05tNDJBekc1Q25VMnA2dEtEeS82cEFlcVdnczFZNDYvVi9uanZ6MnBRMk5nb
                Wo0M3hLMVNhWlh0RDVL1J5WVdOcHc3TnVJRlJ5YVdKMWRHRnlhV0V4SHpBZE
                Jna3Foa2lHOXcwQkNRRVdFR0ZqYjJSelFITmhkQzVuYjJJdWJYZ3hKakFrQmdOVkJB
                a01IVUYyTGlCSWFXUmhiR2R2SURjM0xDQkRiMnd1SUVkMVpYSnlaWEp2TVE0d0
                RBWURWUVFSREFVd05qTXdNREVMTUFrR0ExVUVCaE1DVFZneEdUQVhCZ05W
                QkFnTUVFUnBjM1J5YVhSdklFWmxaR1Z5WVd3eEV6QVJCZ05WQkFjTUNrTjFZWF
                ZvZEdWdGIyTXhNekF4QmdrcWhraUc5dzBCCQ1FJTUpGSmxjM0J2Ym5OaFlteGxPaUJ
                HWlhKdVlXNWtieUJOWVhKMHc2MXVaWG9nUTI5emN6QWVGdzB4TVRFd016QXl
                NalV4TXpCYUZ3MHhNekV3TWpreU1qVXhNekJhTUlIUk1TZ3dKZ1lEVlFRREV4OU
                VWVXhEUlNCQ1JWUklTVTVCSUcEhlTW9JaFluRmdDVjNSeU9EdnZ5SHYwdDU4V
                nZZPSIgdG90YWw9IjEuMTYiIHN1YlRvdGFsPSIxLjAwIiBjZXJ0aWZpY2Fkbz0iTUlJR
                U1UQ0NBeG1nQXdJQkFnSREF4TURRMk1qQXhNekF3RFFZSktvWklodmNOQVFFRk
                JRQXdnZ0UyTVRnd05nWURWUVFEREM5QkxrTXVJR1JsYkNCVFpYSjJhV05wYnlC
                a1pTQkJaRzFwYm1semRISmhZMm5EczI0Z1ZISnBIGVuIHVuYSBzb2xhIGV4aGliaWN
                pw7NuIiBub0NlcnRpZmljYWRvPSIwMDAwMTAwMDAwMDEwNDYyMDEzMCIgdG
                lwb0RlQ29tcHJvYmFudGU9ImluZ3Jlc28iIGRlc2N1ZW50bz0iMC4wMCIgbWV0b2RvR
                GVQYWdvPSJFZmVjdGl2byIgTHVnYXJFeHBlZGljaW9uPSJHdWFkYWx1cGUsIE51Z
                XZvIExlSU0JDUlZSSVNVNUJJRkpQUkZKSlIxVkZXaUJCVGtkRlRFVlRNU2d3SmdZ
                RFZRUUtFeDlFVlV4RFJTQkNSVlJJU1U1QklGSlBSRkpKUjFWRldpQkJUa2RGVEVW
                VE1SWXdGQVlEVlFRdEV3MVNUMEZFTnpjeE1USTFRVTAwTVJzd0dRWURWUVF
                GRXhKU1QwRkVOemNZblYwWVhKcFlURXZNQzBHQTFVRUNnd21VMlZ5ZG1sam
                FXOGdaR1VnUVdSdGFXNXBjMZKUFJGSkpSMVZGV2lCQlRrZEZURVZUTVNnd0p
                nWURWUVFwRXg5RVZVeERMDQ4ejZsc2d2SkRhTHcxK3NnY2VJK2h5cm51TWRN
                bmZUNjhzZWtJb0wrY0tvQzU0QjhMYzArMVNxOUxqRVhta3VCMHJsY0hxcEJ4cEljW
                EJvL3I5UnQ2c24rUnFRKzNlNVpzb1FmSTF5L2lvSDlaVzhnaEMzRHNXcHlkVU5wZHJ
                mdGJHVkFpcDVXUXp2VlpVVWI1azFZNldoUUpuTHBSengvbmRpNS9WTzYrSjRhSW
                JTUy9xZmlXRjQxOHg1RWt2SHBDL2gwNWhVNEU2eWU1Ymp0bHE5K1M3TjZqNlV
                4L1Y3TWVKTDArOXpxVzhsY3FZRUoiIGZvcm1hRGVQYWdvPSJQYWdv4TVRJMV
                RVZFVSRTVNTURFeEhEQWFCZ05WQkFzVUUwMWhkSEpwZWlCTVlYTWdRVzN
                wY21sallYTXdnWjh3RFFZSktvWklodmNOQVFFQkJRQURnWTBBTUlHSkFvR0JBTD
                UvcTlldEtDako4ZkhERVZJNHhVd2ZWMDJMcEdJZXlwTmxwZFMwSHlRUUdvampk
                WUdnYW50dC9TRmgrcVhOb050dTZWS0ZOeGQzYlI0bzFpbHNNSHFFYVR3QVFzY0
                RrR2ZpOUgwZFJZZVUrZlo3eVQ1b2FrUkVkVHNWSWRPUDc3b1pqTXFUVjNVd3lD
                UW5GbUM0bi9zSnc2eTB0M09lWGdBd2ZzWFQ2dld4QWdNQkFBR2pIVEFiTUF3R0E
                xVWRFd0VCL3dRQ01BQXdDd1lEVlIwUEJBUURBZ2JBTUEwR0NTVVNREF3TURF
                d01EQXdNcUdTSWIzRFFFQkJRVUFBNElCQVFDL1EzRWdDM3FlQzdDNWpTNHRl
                ZmVWUkZrVHV6eXBGdzZkNjNoUUdBM3NqVWUvSDlhZ01HMndCeDhBZzg2RTJF
                ZWIzckp1VktWcVpVZkVBZmZYeTZhMXU3L0t4MEl6dWhUOG9Vw7NuIj48RW1pc29
                yIHJmYz0iUk9BRDc3MTEyNUFNNCIgbm9tYnJlPSJEdWxjZSBCZXRoaW5hIFJvZHJp
                Z3VleiBBbmdlbGVzIj48RG9taWNpbGlvRmlzY2FsIHBhaXM9Ik1YIiBjYWxsZT0iSnVsa
```

```
                W8gQ29ydGF6YXIiIG11bmljaXBpbz0iR3VhZGFsdXBlIiBlc3RhZG89Ik51ZXZvIExlw7
                NuIiBjb2RpZ29Qb3N0YWw9IjY3MTQwIiBub0V4dGVyaW9yPSIxMDAiIG5vSW50ZXJ
                pb3I9IkMiIGNvbG9uaWE9IlJpbmNvbiBDb250cnkiIGxvY2FsaWRhZD0iR3VhZGFsdXBl
                Ij48L0RvbWljaWxpb0Zpc2NhbD48RXhwZWRpZG9FbiBwYWlzPSJNWCI+uPjxSZWdp
                bWVuRmlzY2FsIFJlZ2ltZW49IlJlZ2ltZW4gSW50ZXJtZWRpbyI+PC9SZWdpbWVuRml
                zY2FsPjwvRW1pc29yPjxSZWNlcHRvciByZmM9IlhBWFgwMTAxMDEwMDAiPjwvU
                mVjZXB0b3I+PENvbmNlcHRvcz48Q29uY2VwdG8gdW5pZGFkPSJzZXJ2KHMpIiBpb
                XBvcnRlPSIxLjAwIiBjYW50aWRhZD0iMS4wMDAiIGRlc2NyaXBjaW9uPSJTZXJ2aW
                NpbyBkZSBDb25zdWx0b3LDrWEgZGUgVmlhamVzIiB2YWxvclVuaXRhcmlvPSIxLjA
                wIj48L0NvbmNlcHRvPjwvQ29uY2VwdG9zPjxJbXB1ZXN0b3M+PFRyYXNsYWRvcz4
                8VHJhc2xhZG8gdGFzYT0iMTYuMDAiIGltcG9ydGU9IjAuMTYiIGltcHVlc3RvPSJJVkE
                iPjwvVHJ3M+PC9JbXB1ZXN0b3M+PC9Db21wcm9iYW50ZThc2xhZG8+PC9UcmFzbG
                Fb4=
                </cfdi >
                </validate>
        </command>
        <trID>
                <clTRID>123454394</clTRID>
        </trID>
</pac>
```

Respuesta

Cuando un CFDI es validado exitosamente, INSIGNA devuelve un código de resultado 1000 (Comando ejecutado exitosamente). En caso contrario, se regresará alguna de las siguientes respuestas:

- 301: XML mal formado. Cuando el XML del CFDI no cumple con la estructura definida por el SAT
- 601: La información para buscar el comprobante no es válida. Error en la sintaxis del UUID que se quiere validar
- 602: El CFDI no fue encontrado en la base de datos del SAT. Este error es común si el CFDI fue emitido hace pocas horas. El SAT recomienda esperar hasta 72 horas
- 1002: El comprobante no se encuentra en el sistema del SAT. El sistema del SAT no cuenta con el comprobante. Le sugerimos intentar más tarde
- 2600: El CFDI contiene una estructura inválida y no fue enviado a validarse ante el SAT
- 2603: El CFDI contiene un sello de comprobante inválido y no fue enviado a validarse ante el SAT
- 2604: Sello de Certificación Inválido. El sello de certificación del CFDI a validar es inválido. Este sello fue asignado por el PAC que timbró el CFDI
- 2800: El CFDI contiene un sello de certificación inválido y no fue enviado a validarse ante el SAT

Ejemplo de una respuesta exitosa a un comando ***validate***:

```
<? xml version="1.0" encoding="UTF-8" standalone="no"?>
<pac xmlns="urn:ietf:params:xml:ns:pac-1.0">
        <response>
                <result code="602">
                        <msg lang="es">El comprobante no se encuentra en el sistema del SAT</msg>
                </result>
                <trID>
                        <svTRID>11328</svTRID>
                </trID>
                <stats>
                        <svTimestamp>2018-01-01T15:19:18</svTimestamp>
                        <cmdReceived>2018-01-01T15:19:22</cmdReceived>
```

```
                        <cmdExecutionTime>3626</cmdExecutionTime>
                        <clConnectionTime>33100</clConnectionTime>
                        <clOpenConnections>1</clOpenConnections>
                </stats>
        </response>
</pac>
```

3.6. Consulta de estatus de CFDIs

El comando **cfdiStatus** se usa para consultar el estatus de un CFDI en el sistema del SAT, su función principal es para conocer cuando el estatus de un CFDI que se ha mandado a cancelar ha cambiado a Cancelado (Cancelación Aceptada) o Vigente (Cancelación Rechazada). Dentro del elemento **cfdiStatus** se requieren los siguientes parámetros:

- <uuid> El UUID del CFDI timbrado
- <issuerRFC> El rfc del emisor del CFDI timbrado
- <receptorRfc> El rfc del receptor del CFDI timbrado
- <cfdiTotal> Monto total del CFDI timbrado

```
<? xml version="1.0" encoding="UTF-8" standalone="no"?>
<pac xmlns="urn:ietf:params:xml:ns:pac-1.0">
        <command>
        <cfdiStatus>
                <uuid>f47ac10b-58cc-4372-a567-0e02b2c3d479</uuid>
                <issuerRFC>LAN7008173R5</issuerRFC>
                <receptorRfc>MAG041126GT8</receptorRfc>
                <cfdiTotal>2500</cfdiTotal>
        </cfdiStatus>
        <clTRID>123456789</clTRID>
        <stats>
                <svTimestamp>2018-01-01T15:19:18</svTimestamp>
                <cmdReceived>2018-01-01T15:19:22</cmdReceived>
                <cmdExecutionTime>3626</cmdExecutionTime>
                <clConnectionTime>33100</clConnectionTime>
                <clOpenConnections>1</clOpenConnections>
        </stats>
        </command>
</pac>
```

Respuesta

```
<? xml version="1.0" encoding="UTF-8" standalone="no"?>
<pac xmlns="urn:ietf:params:xml:ns:pac-1.0">
        <response>
                <result code="1000">
                        <msg lang="es">Comando ejecutado exitosamente</msg>
                </result>
                <resData>
                        <cfdiStatusData>
                                <cfdiStatus>Cancelado</cfdiStatus>
                                <cancelationType>Cancelable con aceptación</cancelationType>
                                <cancelationStatus>Cancelado aceptado</cancelationStatus>
```

```
                    </ cfdiStatusData >
                </resData>
                <trID>
                    <clTRID>12345</ clTRID >
                    <svTRID>54321</svTRID>
                </trID>
                <stats>
                    <svTimestamp>2018-01-01T23:18:55</svTimestamp>
                    <cmdReceived>2018-01-01T23:18:55</cmdReceived>
                    <cmdExecutionTime>92</cmdExecutionTime>
                </stats>
        </response>
</pac>
```

4. Comandos de utilería

En este apartado se detallan los comandos de utilería del sistema. Es importante recalcar que antes de hacer alguna operación se debe contar con una sesión abierta.

4.1. Validación de certificado de sello digital

Este comando se usa para verificar la vigencia y/o el estatus de un certificado de sello digital para un RFC que se encuentre registrado en el LCO (Lista de Contribuyentes con obligación). El comando validateCertificateRFC requiere de los siguientes parámetros, siendo ambos obligatorios:

- <rfc> - RFC asociado al certificado
- <serie> - Número de serie del certificado de sello digital

```
<? xml version="1.0" encoding="UTF-8" standalone="no"?>
<pac xmlns="urn:ietf:params:xml:ns:pac-1.0">
        <command>
            <validateCertificateRFC>
                <rfc>XAXX010101000</rfc>
                <serie>00000000000000000000</serie>
            </validateCertificateRFC >
        </command>
</pac>
```

La estructura de respuesta exitosa del comando es la siguiente:

```
<? xml version="1.0" encoding="UTF-8" standalone="no"?>
<pac xmlns="urn:ietf:params:xml:ns:pac-1.0">
        <command>
            <resData>
                <certificateRFCData>
                    <csdFinalDate>2019-01-01T15:19:18</csdFinalDate>
                    <csdStatus>A</csdStatus >
                </certificateRFCData>
```

```
                </resData>
                <trID>
                        <svTRID>11336</svTRID>
                </trID>
                <stats>
                        <svTimestamp>2018-01-01T15:19:18</svTimestamp>
                        <cmdReceived>2018-01-01T15:19:22</cmdReceived>
                        <cmdExecutionTime>3626</cmdExecutionTime>
                        <clConnectionTime>33100</clConnectionTime>
                        <clOpenConnections>1</clOpenConnections>
                </stats>
        </command>
</pac>
```

El atributo *csdFinalDate* representa la fecha de vencimiento del certificado, mientras que *csdStatus* indica el estado del certificado en la LCO del SAT (A = activo, C = caducado, R = revocado).

4.2. Validación de RFC

Este comando permite verificar si un RFC se encuentra en la lista de RFCs inscritos no cancelados (LRFC) del SAT. El comando validateLRFC requiere únicamente de un parámetro:

- <rfc> - RFC a buscar en la lista LRFC del SAT

```
<? xml version="1.0" encoding="UTF-8" standalone="no"?>
<pac xmlns="urn:ietf:params:xml:ns:pac-1.0">
        <command>
                <validateLRFC>
                        <rfc>XAXX010101000</rfc>
                </validateLRFC>
        </command>
</pac>
```

La estructura de respuesta del comando es la siguiente:

```
<? xml version="1.0" encoding="UTF-8" standalone="no"?>
<pac xmlns="urn:ietf:params:xml:ns:pac-1.0">
        <command>
                <resData>
                        <lrfcData>
                                <sncf>N</sncf >
                                <outsourcing>N</outsourcing >
                        </lrfcData>
                </resData>
                <trID>
                        <svTRID>11336</svTRID>
                </trID>
                <stats>
                        <svTimestamp>2018-01-01T15:19:18</svTimestamp>
                        <cmdReceived>2018-01-01T15:19:22</cmdReceived>
                        <cmdExecutionTime>3626</cmdExecutionTime>
                        <clConnectionTime>33100</clConnectionTime>
                        <clOpenConnections>1</clOpenConnections>
```

```
                </stats>
        </command>
</pac>
```

El atributo *sncf* determina si es una entidad adherida al Sistema Nacional de Coordinación Fiscal. El atributo *outsourcing* indica si el régimen es Subcontratación. Los valores posibles para ambos casos son: Y, N.

5. Respuestas de INSIGNA

Todas las respuestas por parte del Servicio de Timbrado INSIGNA se forman con los siguientes elementos:

- <response> - elemento principal de la respuesta
- <result> - resultado de la operación, cuenta con un atributo "code" que indica el código de respuesta, el cual indica éxito o error en el proceso del comando. Ver Códigos de Respuesta
    - <msg> - mensaje en texto legible que describe el código de respuesta. Puede contener un atributo "lang" indicando el idioma del mensaje
    - <value> - valor asociado a la respuesta, el contenido es XML
    - <reason> - razón detallada de la respuesta, en texto legible para una persona
- <resData> - datos de la respuesta específicos según el tipo de comando enviado (OPCIONAL, según el comando)
    - <signData> Incluye los elementos específicos del proceso de timbrado (revisar sección 3 del presente documento)
- <trID> - indica el inicio de la sección de identificadores de transacción
    - <clTRID> - identificador de transacción por parte del cliente, es devuelta por el Servicio si fue especificada por el cliente en el request del comando. Se recomienda ampliamente que el cliente asigne a cada comando un ID único, con el fin de facilitar su identificación en caso de que exista algún error al enviar un comando determinado. Esto se consigue al agregar un elemento <trID><clTRID>#ID#</clTRID></trID>
    - <svTRID> - identificador de transacción por parte del Servicio. Este código no es asignado en los comandos <hello>, <login> y <logout>
- <account> - información sobre la cuenta del cliente. (OPCIONAL, según el comando)
    - <signatures> - indica cuantos timbres le quedan disponibles al cliente
- <stats> - estadísticas del comando
    - <svTimestamp> - fecha y hora del Servicio en el momento de la respuesta
    - <cmdReceived> - fecha y hora de cuando el Servicio recibió el comando
    - <cmdExecutionTime> - cantidad de milisegundos que tardó el Servicio en procesar el comando
    - <clConnectionTime> -tiempo en milisegundos que la sesión del cliente lleva abierta
    - <clOpenConnections> - número de sesiones simultáneamente abiertas que tiene el cliente al momento de la ejecución del comando

Ejemplo de una respuesta del Servicio de Timbrado de INSIGNA

```
<? xml version="1.0" encoding="UTF-8" standalone="no"?>
<pac xmlns="urn:ietf:params:xml:ns:pac-1.0">
        <response>
                <result code="1000">
                        <msg lang="es">Comando completado exitosamente</msg>
```

```
                </result>
                <resData>
                        #resData#
                </resData>
                <trID>
                        <clTRID>foo-123</clTRID>
                        <svTRID>BAR-987</svTRID>
                </trID>
                <account>
                        <signatures>1000</signatures>
                </account>
                <stats>
                        <svTimestamp>2018-01-01T14:21:53</svTimestamp>
                        <cmdReceived>2018-01-01T14:21:53</cmdReceived>
                        <cmdExecutionTime>1515</cmdExecutionTime>
                        <clConnectionTime>14900</clConnectionTime>
                        <clOpenConnections>2</clOpenConnections>
                </stats>
        </response>
</pac>
```

6. Códigos de Respuesta INSIGNA

Los siguientes códigos de respuesta están definidos por el SAT:

301	XML mal formado	Cuando el XML del CFDI no cumple con la estructura definida por el SAT.
302	Sello mal formado o inválido	El sello del emisor no es válido.
303	Sello no corresponde a emisor	El CSD del emisor no corresponde al RFC emisor del comprobante.
304	Certificado revocado o caduco	El CSD del emisor se encuentra revocado de acuerdo a la lista LCO.
305	La fecha de emisión no está dentro de la vigencia del CSD del emisor	El CSD del emisor no está vigente para la fecha de emisión del CFDI.
306	El certificado no es de tipo CSD	El certificado no tiene la estructura de uno de tipo CSD.
307	El CFDI contiene un timbre previo	El XML enviado a INSIGNA ya contiene un timbre.
308	Certificado no expedido por el SAT	El CSD del emisor no fue firmado por un Certificado de Autoridad del SAT.
401	Fecha y hora de generación fuera del rango	El rango entre la fecha de emisión y fecha de timbrado es mayor a 72 horas.
402	RFC del emisor no se encuentra en el régimen de contribuyentes	El RFC del emisor no se encuentra en la lista LCO del SAT.
403	La fecha de emisión no es posterior al 01 de enero 2012	La fecha de emisión no es posterior al 01 de enero 2012.

601	Información de búsqueda inválida para el servicio de validación.	Error en la sintaxis del UUID que se quiere validar.
602	El CFDI no existe.	El CFDI no se encuentra en la BD del SAT. Se recomienda esperar 72 horas a la fecha de emisión.

Los siguientes códigos son definidos por INSIGNA y pertenecen exclusivamente a su protocolo de comunicación y reglas de negocio.

1000	Comando completado exitosamente	Respuesta a un comando exitoso.
1002	El comprobante no se encuentra en el sistema del SAT.	El sistema del SAT no cuenta con el comprobante. Le sugerimos intentar más tarde.
1500	Comando completado exitosamente; terminando la sesión	Comando completado exitosamente; terminando la sesión.
2001	Error de sintaxis en el comando	Respuesta a un comando mal formado según el esquema definido de XML para el mismo. Por ejemplo, si en un timbrado se envían dos elementos <sign>. Este error se refiere al XML definido por INSIGNA, y es diferente al 301.
2002	Error de uso del comando	Respuesta a un comando de <login> o <logout> enviado fuera de tiempo. Por ejemplo, si se envía un <login> cuando ya se tiene iniciada una sesión.
2010	Error al validar la cantidad del parámetro	Cuando las cantidades utilizadas en el CFDI no concuerdan, por ejemplo, la suma de los conceptos no es igual a la cantidad del subtotal. [1]
2100	Versión del protocolo no implementada	Cuando en un comando <login> el cliente especifica en el elemento versión una versión del protocolo que no es reconocida por el Servicio.
2200	Error de autenticación	Cuando el cliente no puede identificarse para iniciar una sesión por medio de un comando <login> por proporcionar credenciales inválidas.
2201	Error de autorización	Cuando el certificado SSL del cliente no está habilitado en los sistemas de INSIGNA.
2309	Timbres no disponibles / El RFC está bloqueado	Cuando el Servicio recibe un comando para timbrar pero la cuenta no tiene timbres disponibles, o bien, el RFC emisor se encuentra bloqueado.
2500	Comando falló; servidor cerrando conexión	Cuando el Servicio no puede ejecutar un comando debido a un error interno no relacionado con el

[1] INSIGNA valida los totales y subtotales de acuerdo a lo especificado en el Anexo 20.

* En un concepto o parte el valor unitario multiplicado por la cantidad no puede diferir del total más de 50 centavos.

* El subtotal de los nodos de impuestos o conceptos debe ser igual a la suma de cada impuesto/concepto de dicho nodo. Se permite un margen de tolerancia de 50 centavos por cada impuesto/concepto del nodo.

* El total de la factura debe ser igual a la suma del subtotal menos los descuentos, menos los impuestos retenidos y más los impuestos trasladados (ver Anexo 20, pag. 71). Se permite un margen de tolerancia de 50 centavos por cada impuesto/descuento.

		protocolo. Para evitar posteriores errores, el Servicio termina la sesión y cierra la conexión.
2501	Error de autenticación; servidor cerrando conexión	Cuando el Servicio nota un error al validar las credenciales del cliente y espera que este no pueda recuperarse de este en una misma sesión, por ejemplo cuando la cuenta o la IP del cliente se encuentra bloqueada.
2502	Límite de sesiones excedida; servidor cerrando conexión	Cuando el Servicio recibe un comando login y el comando no puede completarse porque el cliente ha excedido un límite para el número de sesiones que puede mantener.
2600	Estructura Inválida	La estructura del CFDI a validar es incorrecta de acuerdo al estándar del Anexo 20.
2603	Sello del Comprobante Inválido	El sello del CFDI a validar es inválido. Este sello se generó con el CSD del emisor del CFDI.
2604	Sello de Certificación Inválido	El sello de certificación del CFDI a validar es inválido. Este sello fue asignado por el PAC que timbró el CFDI.
2901	Certificado no encontrado	El certificado de sello digital no fue encontrado en la lista LCO.
2902	El RFC no coincide con el certificado	El RFC proporcionado no coincide con el registrado en el certificado de sello digital.
2910	El RFC no fue encontrado en la lista	El RFC proporcionado no está registrado en la lista LRFC del SAT al momento de la consulta.

7. Proceso de Timbrado

Proceso de timbrado	
1	El Servicio recibe el comando de timbrado <sign>
2	El Servicio verifica que el cliente tenga timbres disponibles y que el RFC emisor no se encuentre bloqueado (En caso de error se envía el código de error 2309)
3	El Servicio verifica que el XML sea válido (En caso de error se envía el código de error 301)
4	El Servicio verifica que la fecha de emisión sea posterior al 1 de enero 2012 (En caso de error se envía el código de error 403)
5	El Servicio verifica que el certificado CSD del cliente haya sido firmado por el SAT (En caso de error se envía el código de error 308)
6	El Servicio verifica que el certificado corresponda a un CSD (no a un FIEL) (En caso de error se envía el código de error 306)
7	El Servicio verifica que el RFC del certificado sea igual al del XML (En caso de error se envía el código de error 303)
8	El Servicio verifica que el número de certificado sea válido según el LCO del SAT (En caso de error se envía el código de error 402)
9	El Servicio verifica que el certificado no haya sido revocado según el LCO del SAT (En caso de error se envía el código de error 304)
10	El Servicio verifica que el certificado no haya expirado (En caso de error se envía el código de error 304)
11	El Servicio verifica que la fecha de CFDI esté dentro de la validez del certificado (En caso de error se envía el código de error 305)
12	El Servicio verifica que no haya más de 72 horas entre la fecha de emisión y la de timbrado (En caso de error se envía el código de error 401)
13	El Servicio verifica que el CFDI no se haya timbrado anteriormente (En caso de error se envía el código de error 307)
14	El Servicio valida el sello del CFDI (En caso de error se envía el código de error 302)
15	El Servicio realiza el timbrado
16	En caso de ser necesario se envía un aviso de timbres disponibles
17	Se envía la respuesta con el XML timbrado

8. Esquema INSIGNA

```
<?xml version="1.0" encoding="utf-8"?>
<schema targetNamespace="urn:ietf:params:xml:ns:pac-1.0"
	xmlns:pac="urn:ietf:params:xml:ns:pac-1.0"
xmlns="http://www.w3.org/2001/XMLSchema"
	elementFormDefault="qualified">

	<annotation>
		<documentation>PAC-MX Protocol v1.0 schema.</documentation>
	</annotation>

	<!-- Every PAC XML instance must begin with this element. -->
	<element name="pac" type="pac:pacType" />

	<!-- An PAC XML instance must contain a greeting, hello, command, response,
		or extension. -->
	<complexType name="pacType">
		<choice>
			<element name="greeting" type="pac:greetingType" />
			<element name="command" type="pac:commandType" />
			<element name="response" type="pac:responseType" />
		</choice>
	</complexType>

	<!-- A greeting is sent by a server in response to a client connection or
		<hello>. -->
	<complexType name="greetingType">
		<sequence>
			<element name="svID" type="normalizedString" />
			<element name="svDate" type="dateTime" />
			<element name="svVersion" type="pac:svVersion" />
			<element name="svSession" type="pac:svSession" />
		</sequence>
	</complexType>

	<!-- A server greeting identifies available object services. -->
	<complexType name="svVersion">
		<sequence>
			<element name="version" type="pac:versionType" />
		</sequence>
	</complexType>

	<!-- A server greeting identifies available object services. -->
	<complexType name="svSession">
		<sequence>
			<element name="svSessionTTL" type="long" />
			<element name="svSessionTimeout" type="long" />
		</sequence>
	</complexType>

	<!-- Command types. -->
	<complexType name="commandType">
		<sequence>
			<choice>
				<element name="hello" />
				<element name="sign" type="pac:signType" />
				<element name="info" type="pac:infoType" />
				<element name="verify" type="pac:verifyType" />
				<element name="cancel" type="pac:cancelType" />
				<element name="login" type="pac:loginType" />
				<element name="logout" />
				<element name="validate" type="pac:validateType" />
				<element name="validateCertificateRFC"
type="pac:validateCertificateRFCType" />
```

```
                        <element name="validateLRFC" type="pac:validateLRFCType" />
                </choice>
                <element name="clTRID" type="pac:trIDStringType" minOccurs="0" />
        </sequence>
</complexType>

<!-- The <sign> command. -->
<complexType name="signType">
        <sequence>
                <element name="cfdi" type="string" />
        </sequence>
</complexType>

<!-- The <info> command. -->
<complexType name="infoType">
        <sequence>
                <element name="signID" type="string" />
        </sequence>
</complexType>

<!-- The <verify> command. -->
<complexType name="verifyType">
        <sequence>
                <choice>
                        <element name="chain" type="string" />
                        <element name="signClTRID" type="string" />
                </choice>
        </sequence>
</complexType>

<!-- The <cancel> command. -->
<complexType name="cancelType">
        <sequence>
                <element name="date" type="dateTime" />
                <element name="uuid" type="string" />
                <element name="signature" type="string" />
                <element name="certificate" type="string" />
        </sequence>
</complexType>

<!-- The <login> command. -->
<complexType name="loginType">
        <sequence>
                <element name="clID" type="pac:clIDType" />
                <element name="pw" type="pac:pwType" />
                <element name="options" type="pac:optionsType" />
        </sequence>
</complexType>

<simpleType name="clIDType">
        <restriction base="token">
                <minLength value="3" />
                <maxLength value="16" />
        </restriction>
</simpleType>

<simpleType name="pwType">
        <restriction base="token">
                <minLength value="8" />
                <maxLength value="16" />
        </restriction>
</simpleType>

<complexType name="optionsType">
        <sequence>
                <element name="version" type="pac:versionType" />
        </sequence>
</complexType>

<!-- The <validate> command. -->
```

```
        <complexType name="validateType">
                <sequence>
                        <element name="cfdi" type="string" />
                </sequence>
        </complexType>

        <!-- The <validateCertificateRFC> command. -->
        <complexType name="validateCertificateRFCType">
                <sequence>
                        <element name="rfc" type="string" />
                        <element name="serie" type="string" />
                </sequence>
        </complexType>

        <!-- The <validateLRFC> command. -->
        <complexType name="validateLRFCType">
                <sequence>
                        <element name="rfc" type="string" />
                </sequence>
        </complexType>

        <!-- Response types. -->
        <complexType name="responseType">
                <sequence>
                        <element name="result" type="pac:resultType" minOccurs="1"
                                maxOccurs="unbounded" />
                        <element name="resData" type="pac:resultDataType" minOccurs="0" />
                        <element name="trID" type="pac:trIDType" />
                        <element name="account" type="pac:accountType" minOccurs="0" />
                        <element name="stats" type="pac:statsType" />
                </sequence>
        </complexType>

        <complexType name="resultType">
                <sequence>
                        <element name="msg" type="pac:msgType" />
                        <element name="value" type="pac:valueType" minOccurs="0"
                                maxOccurs="unbounded" />
                        <element name="reason" type="string" minOccurs="0" />
                </sequence>
                <attribute name="code" type="pac:resultCodeType" use="required" />
        </complexType>

        <complexType name="valueType" mixed="true">
                <sequence>
                        <any namespace="##any" processContents="skip" />
                </sequence>
                <anyAttribute namespace="##any" processContents="skip" />
        </complexType>

        <complexType name="resultDataType">
                <choice>
                        <element name="signData" type="pac:signDataType" />
                        <element name="infoData" type="pac:infoDataType" />
                        <element name="verifyData" type="pac:verifyDataType" />
                        <element name="cancelData" type="pac:cancelDataType" />
                        <element name="certificateRFCData"
type="pac:certificateRFCDataType" />
                        <element name="lrfcData" type="pac:lrfcDataType" />
                </choice>
        </complexType>

        <complexType name="signDataType">
                <sequence>
                        <element name="signID" type="string" />
                        <element name="cfdi" type="string" />
                </sequence>
        </complexType>

        <complexType name="infoDataType">
```

```
		<sequence>
			<element name="signID" type="string" />
			<element name="cfdi" type="string" />
			<element name="status" type="string" />
		</sequence>
	</complexType>

	<complexType name="verifyDataType">
		<sequence>
			<element name="signID" type="string" />
			<element name="cfdi" type="string" />
			<element name="status" type="string" />
		</sequence>
	</complexType>

	<complexType name="cancelDataType">
		<sequence>
			<element name="uuid" type="pac:uuidType" />
			<element name="receipt" type="string" />
		</sequence>
	</complexType>

	<complexType name="certificateRFCDataType">
		<sequence>
			<element name="csdFinalDate" type="string" />
			<element name="csdStatus" type="string" />
		</sequence>
	</complexType>

	<complexType name="lrfcDataType">
		<sequence>
			<element name="sncf" type="string" />
			<element name="outsourcing" type="string" />
		</sequence>
	</complexType>

	<complexType name="trIDType">
		<sequence>
			<element name="clTRID" type="pac:trIDStringType" minOccurs="0" />
			<element name="svTRID" type="pac:trIDStringType" />
		</sequence>
	</complexType>

	<complexType name="accountType">
		<sequence>
			<element name="signatures" type="long" minOccurs="0" />
		</sequence>
	</complexType>

	<complexType name="statsType">
		<sequence>
			<element name="svTimestamp" type="dateTime" />
			<element name="cmdReceived" type="dateTime" />
			<element name="cmdExecutionTime" type="long" />
			<element name="clConnectionTime" type="long" minOccurs="0" />
			<element name="clOpenConnections" type="long" minOccurs="0" />
		</sequence>
	</complexType>

	<!-- PAC result codes. -->
	<simpleType name="resultCodeType">
		<restriction base="unsignedShort">
			<enumeration value="101" />
			<enumeration value="102" />
			<enumeration value="103" />
			<enumeration value="104" />
			<enumeration value="105" />
			<enumeration value="106" />
			<enumeration value="107" />
			<enumeration value="108" />
```

```
                    <enumeration value="109" />
                    <enumeration value="110" />
                    <enumeration value="111" />
                    <enumeration value="112" />
                    <enumeration value="113" />
                    <enumeration value="114" />
                    <enumeration value="115" />
                    <enumeration value="116" />
                    <enumeration value="117" />
                    <enumeration value="118" />
                    <enumeration value="119" />
                    <enumeration value="120" />
                    <enumeration value="121" />
                    <enumeration value="122" />
                    <enumeration value="123" />
                    <enumeration value="124" />
                    <enumeration value="125" />
                    <enumeration value="126" />
                    <enumeration value="127" />
                    <enumeration value="128" />
                    <enumeration value="129" />
                    <enumeration value="130" />
                    <enumeration value="131" />
                    <enumeration value="132" />
                    <enumeration value="133" />
                    <enumeration value="134" />
                    <enumeration value="135" />
                    <enumeration value="136" />
                    <enumeration value="137" />
                    <enumeration value="138" />
                    <enumeration value="139" />
                    <enumeration value="140" />
                    <enumeration value="141" />
                    <enumeration value="142" />
                    <enumeration value="143" />
                    <enumeration value="144" />
                    <enumeration value="145" />
                    <enumeration value="146" />
                    <enumeration value="147" />
                    <enumeration value="148" />
                    <enumeration value="149" />
                    <enumeration value="150" />
                    <enumeration value="151" />
                    <enumeration value="152" />
                    <enumeration value="153" />
                    <enumeration value="154" />
                    <enumeration value="155" />
                    <enumeration value="156" />
                    <enumeration value="157" />
                    <enumeration value="158" />
                    <enumeration value="159" />
                    <enumeration value="160" />
                    <enumeration value="161" />
                    <enumeration value="162" />
                    <enumeration value="163" />
                    <enumeration value="164" />
                    <enumeration value="165" />
                    <enumeration value="166" />
                    <enumeration value="167" />
                    <enumeration value="168" />
                    <enumeration value="169" />
                    <enumeration value="170" />
                    <enumeration value="171" />
                    <enumeration value="172" />
                    <enumeration value="173" />
                    <enumeration value="174" />
                    <enumeration value="175" />
                    <enumeration value="176" />
                    <enumeration value="177" />
                    <enumeration value="178" />
```

```
<enumeration value="179" />
<enumeration value="180" />
<enumeration value="181" />
<enumeration value="182" />
<enumeration value="183" />
<enumeration value="184" />
<enumeration value="185" />
<enumeration value="186" />
<enumeration value="187" />
<enumeration value="188" />
<enumeration value="189" />
<enumeration value="190" />
<enumeration value="191" />
<enumeration value="192" />
<enumeration value="193" />
<enumeration value="194" />
<enumeration value="195" />
<enumeration value="196" />
<enumeration value="201" />
<enumeration value="202" />
<enumeration value="203" />
<enumeration value="205" />
<enumeration value="301" />
<enumeration value="302" />
<enumeration value="303" />
<enumeration value="304" />
<enumeration value="305" />
<enumeration value="306" />
<enumeration value="307" />
<enumeration value="308" />
<enumeration value="401" />
<enumeration value="402" />
<enumeration value="403" />
<enumeration value="501" />
<enumeration value="502" />
<enumeration value="503" />
<enumeration value="504" />
<enumeration value="505" />
<enumeration value="506" />
<enumeration value="507" />
<enumeration value="508" />
<enumeration value="509" />
<enumeration value="510" />
<enumeration value="511" />
<enumeration value="512" />
<enumeration value="513" />
<enumeration value="514" />
<enumeration value="601" />
<enumeration value="602" />
<enumeration value="1000" />
<enumeration value="1001" />
<enumeration value="1002" />
<enumeration value="1004" />
<enumeration value="1006" />
<enumeration value="1008" />
<enumeration value="1009" />
<enumeration value="1010" />
<enumeration value="1011" />
<enumeration value="1012" />
<enumeration value="1013" />
<enumeration value="1014" />
<enumeration value="1015" />
<enumeration value="1016" />
<enumeration value="1017" />
<enumeration value="1201" />
<enumeration value="1202" />
<enumeration value="1203" />
<enumeration value="1205" />
<enumeration value="1300" />
<enumeration value="1301" />
```

```
                        <enumeration value="1302" />
                        <enumeration value="1303" />
                        <enumeration value="1304" />
                        <enumeration value="1305" />
                        <enumeration value="1306" />
                        <enumeration value="1307" />
                        <enumeration value="1308" />
                        <enumeration value="1309" />
                        <enumeration value="1500" />
                        <enumeration value="2001" />
                        <enumeration value="2002" />
                        <enumeration value="2010" />
                        <enumeration value="2100" />
                        <enumeration value="2200" />
                        <enumeration value="2201" />
                        <enumeration value="2309" />
                        <enumeration value="2500" />
                        <enumeration value="2501" />
                        <enumeration value="2502" />
                        <enumeration value="2600" />
                        <enumeration value="2601" />
                        <enumeration value="2602" />
                        <enumeration value="2603" />
                        <enumeration value="2604" />
                        <enumeration value="2700" />
                        <enumeration value="2800" />
                        <enumeration value="2901" />
                        <enumeration value="2902" />
                        <enumeration value="2910" />
                </restriction>
        </simpleType>

        <!-- Datatypes -->

        <simpleType name="trIDStringType">
                <restriction base="token">
                        <minLength value="1" />
                        <maxLength value="64" />
                </restriction>
        </simpleType>

        <!-- Human-readable text may be expressed in languages other than English. -->
        <complexType name="msgType">
                <simpleContent>
                        <extension base="normalizedString">
                                <attribute name="lang" type="language" default="es" />
                        </extension>
                </simpleContent>
        </complexType>

        <!-- A PAC version number is a dotted pair of decimal numbers. -->
        <simpleType name="versionType">
                <restriction base="token">
                        <pattern value="[1-9]+\.[0-9]+" />
                        <enumeration value="1.0" />
                </restriction>
        </simpleType>

        <!-- Universally Unique Identifier type, represented by 32 hexadecimal characters
-->
        <simpleType name="uuidType">
                <restriction base="string">
                        <pattern
                                value="[a-f0-9A-F]{8}-[a-f0-9A-F]{4}-[a-f0-9A-F]{4}-[a-f0-
9A-F]{4}-[a-f0-9A-F]{12}" />
                </restriction>
        </simpleType>
        <!-- End of schema. -->
</schema>
```

9. Ejemplo de un CFDI

El siguiente es un ejemplo de un CFDI V3.3:

```
<?xml version="1.0" encoding="UTF-8"?>
<cfdi:Comprobante
Certificado="MIIFxTCCA62gAwIBAgIUMjAwMDEwMDAwMDAzMDAwMjI4MTUwDQYJKoZIhvcNAQELBQAw
ggFmMSAwHgYDVQQDDBdBLkMuIDIgZGUgcHJ1ZWJhcyg0MDk2KTEvMC0GA1UECgwmU2VydmljaW8gZGUg
QWRtaW5pc3RyYWNpw7NuIFRyaWJ1dGFyaWExODA2BgNVBAsML0FkbWluaXN0cmFjacOzbiBkZSBTZWd1cml
kYWQgZGUgbGEgSW5mb3JtYWNpw7NuMSkwJwYJKoZIhvcNAQkBFhphc2lzbmV0QHBydWViYXMuc2F0Lmdv
Yi5teDEmMCQGA1UECQwdQXYuIEhpZGFsZ28gNzcsIENvbC4gR3VlcnJlcm8xDjAMBgNVBBEMBTA2MzAwMQ
swCQYDVQQGEwJNWDEZMBcGA1UECAwQRGlzdHJpdG8gRmVkZXJhbDESMBAGA1UEBwwJQ295b2Fjw6Fu
MRUwEwYDVQQtEwxTQVQ5NzA3MDFOTjMxITAfBgkqhkiG9w0BCQIMElJlc3BvbnNhYmxlOiBBQ0RNQTAeFw
0xNjEwMjUyMTUyMTFaFw0yMDEwMjUyMTUyMTFaMIGxMRowGAYDVQQDExFDSU5ERU1FWCBTQSBERS
BDVjEaMBgGA1UEKRMRQ0lOREVNRVggU0EgREUgQ1YxGjAYBgNVBAoTEUNJTkRFTUVYIFNBIERFIENW
MSUwIwYDVQQtExxMQU43MDA4MTczUjUgLyBGVUFCNzcwMTE3QlhBMR4wHAYDVQQFExUgLyBGVUFC
NzcwMTE3TURGUk5OMDkxFDASBgNVBAsUC1BydWViYV9DRkRJMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8A
MIIBCgKCAQEAgvvCiCFDFVaYX7xdVRhp/38ULWto/LKDSZy1yrXKpaqFXqERJWF78YHKf3N5GBoXgzwFPuDX
+5kvY5wtYNxx/Owu2shNZqFFh6EKsysQMeP5rz6kE1gFYenaPEUP9zj+h0bL3xR5aqoTsqGF24mKBLoiaK44pXBzGz
gsxZishVJVM6XbzNJVonEUNbI25DhgWAd86f2aU3BmOH2K1RZx41dtTT56UsszJls4tPFODr/caWuZEuUvLp1M3nj7
Dyu88mhD2f+1fA/g7kzcU/1tcpFXF/rIy93APvkU72jwvkrnprzs+SnG81+/F16ahuGsb2EZ88dKHwqxEkwzhMyTbQIDA
QABox0wGzAMBgNVHRMBAf8EAjAAMAsGA1UdDwQEAwIGwDANBgkqhkiG9w0BAQsFAAOCAgEAJ/xkL8I+f
pilZP+9aO8n93+20XxVomLJjeSL+Ng2ErL2GgatpLuN5JknFBkZAhxVIgMaTS23zzk1RLtRaYvH83lBH5E+M+kEjFG
p14Fne1iV2Pm3vL4jeLmzHgY1Kf5HmeVrrp4PU7WQg16VpyHaJ/eonPNiEBUjcyQ1iFfkzJmnSJvDGtfQK2TiEolDJA
pYv0OWdm4is9Bsfi9j6lI9/T6MNZ+/LM2L/t72Vau4r7m94JDEzaO3A0wHAtQ97fjBfBiO5M8AEISAV7eZidIl3iaJJHk
QbBYiiW2gikreUZKPUX0HmlnIqqQcBJhWKRu6Nqk6aZBTETLLpGrvF9OArV1JSsbdw/ZH+P88RAt5em5/gjwwtFlN
HyiKG5w+UFpaZOK3gZP0su0sa6dlPeQ9EL4JlFkGqQCgSQ+NOsXqaOavgoP5VLykLwuGnwIUnuhBTVeDbzpgrg9L
uF5dYp/zs+Y9ScJqe5VMAagLSYTShNtN8luV7LvxF9pgWwZdcM7lUwqJmUddCiZqdngg3vzTactMToG16gZA4CWn
MgbU4E+r541+FNMpgAZNvs2CiW/eApfaaQojsZEAHDsDv4L5n3M1CC7fYjE/d61aSng1LaO6T1mh+dEfPvLzp7zyzz
+UgWMhi5Cs4pcXx1eic5r7uxPoBwcCTt3YI1jKVVnV7/w="
  Fecha="2018-01-01T09:12:18" LugarExpedicion="68495" Moneda="MXN"
  NoCertificado="20001000000300022815"

Sello="H5GSi4qURXr8QmNc4f5meQZ9Y9VgIfk5DAUvLXtBKjiuqhqT6nF1Tlo3smcnF/FxOxcd3NtwyhRZbuLkAbiFq
jv94qzYH4OKdw5cX3GqfvXVN2TAUopmTZeJ2WK5mdXLVRnUzpdM6iGEPcOkhWL3YLW5pbjtz9UpMnzhH+L0t
RnHaOknwtUwxPsF6/oK4Czce5L2Z6y2+SZipTKVF48ujGsspKgwxAdadcQpQUR4L3DNCQyRGTMMTn/N2ceAuRJF
mfOKJvSLquTnBkbx9ExBblNUmnEHL/T2H7A6piOynv2fA6DV3VrzluXeEwryTt6U+4s1sr+THtonbhV1cnUEPQ=="
  SubTotal="1000.00" TipoDeComprobante="I" Total="1160.00" Version="3.3"
  xmlns:cfdi="http://www.sat.gob.mx/cfd/3"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.sat.gob.mx/cfd/3
http://www.sat.gob.mx/sitio_internet/cfd/3/cfdv33.xsd">
  <cfdi:Emisor Nombre="HORACIO LLANOS" RegimenFiscal="601" Rfc="LAN7008173R5"/>
  <cfdi:Receptor Nombre="RAFAEL ALEJANDRO HERNÁNDEZ PALACIOS"
    Rfc="ITE430714KI0" UsoCFDI="G01"/>
  <cfdi:Conceptos>
    <cfdi:Concepto Cantidad="1" ClaveProdServ="01010101"
      ClaveUnidad="F52" Descripcion="ACERO" Descuento="0.00"
      Importe="1000.00" NoIdentificacion="00001" Unidad="TONELADA" ValorUnitario="1000.00">
      <cfdi:Impuestos>
        <cfdi:Traslados>
          <cfdi:Traslado Base="1000.00" Importe="160.00" Impuesto="002"
            TasaOCuota="0.160000" TipoFactor="Tasa"/>
        </cfdi:Traslados>
      </cfdi:Impuestos>
    </cfdi:Concepto>
  </cfdi:Conceptos>
  <cfdi:Impuestos TotalImpuestosTrasladados="160.00">
    <cfdi:Traslados>
      <cfdi:Traslado Importe="160.00" Impuesto="002"
        TasaOCuota="0.160000" TipoFactor="Tasa"/>
```

```
    </cfdi:Traslados>
  </cfdi:Impuestos>
  <cfdi:Complemento>
    <tfd:TimbreFiscalDigital FechaTimbrado="2018-01-01T17:19:20"
      NoCertificadoSAT="20001000000300022323"
      RfcProvCertif="SCD110105654"

SelloCFD="H5GSi4qURXr8QmNc4f5meQZ9Y9VgIfk5DAUvLXtBKjiuqhqT6nF1Tlo3smcnF/FxOxcd3NtwyhRZbuLk
AbiFqjv94qzYH4OKdw5cX3GqfvXVN2TAUopmTZeJ2WK5mdXLVRnUzpdM6iGEPcOkhWL3YLW5pbjtz9UpMnzh
H+L0tRnHaOknwtUwxPsF6/oK4Czce5L2Z6y2+SZipTKVF48ujGsspKgwxAdadcQpQUR4L3DNCQyRGTMMTn/N2ce
AuRJFmfOKJvSLquTnBkbx9ExBblNUmnEHL/T2H7A6piOynv2fA6DV3VrzluXeEwryTt6U+4s1sr+THtonbhV1cnUEP
Q=="

SelloSAT="MmDhn4b5BsJ/QiPW+hBvU3S+rW3BSltx0rzUCJxstwNcV+CrMy+rfrgYntNOvWpVLvrx4sW9ZRs3pxjR
QaDwklfkpw4Cl7mvYnWwUeKIqbEYDs5chxlTrBT4wNhLB6YAFY0zezTuAMGMp+MGzYTFmlSlrXPgczFYKgs+Z
mWPL3+eEaOejf/xtlGDGURKnCIfnnbE+qeEBvQTS5m+U4VuBx945Eg0TpnDTHDGmIdxrJNxh3fLtHx1ZsBBXTC47
oj7cIuje9qsnYwbY/W1K6Hb8F6jspUvncst/yoopkI0JyNndvtVI2Tb+PYiIzJ2NEU3NDLZ40aqJJyvze49vKCEYw=="
      UUID="A0D49B21-0614-6A4E-8D6B-B777C8B8AB86" Version="1.1"
      xmlns:tfd="http://www.sat.gob.mx/TimbreFiscalDigital"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.sat.gob.mx/TimbreFiscalDigital
http://www.sat.gob.mx/sitio_internet/cfd/TimbreFiscalDigital/TimbreFiscalDigitalv11.xsd"/>
  </cfdi:Complemento>
</cfdi:Comprobante>
```

- Nota: se recomienda no declarar atributos xlmns (o namespaces) en los elementos interiores (datos de Emisor, Receptor, etc) y dejarlos todos en el elemento principal (es decir, en el elemento <Comprobante>), esto para evitar errores de parseo del XML.

El siguiente es un ejemplo de una Retención Pago V1.0:

```
<?xml version="1.0" encoding="utf-8"?>
<retenciones:Retenciones
        xmlns:retenciones="http://www.sat.gob.mx/esquemas/retencionpago/1"
        xmlns:xsi="http://www.w3.org./2001/XMLSchema-instance" CveRetenc="01"
        FechaExp="2018-01-01T12:45:53-06:00"
        Cert="MIIFljCCA36gAwIBAgIUMjAwMDEwMDAwMDAzMDAwMDM3MDIwDQYJKoZIhvcNAQELBQ
AwggFmMSAwHgYDVQQDDBdBLkMuIDIgZGUgcHJ1ZWJhcyg0MDk2KTEvMC0GA1UECgwmU2VydmljaW8gZ
GUgQWRtaW5pc3RyYWNpw7NuIFRyaWJ1dGFyaWExODA2BgNVBAsML0FkbWluaXN0cmFjacOzbiBkZSBTZWd
1cmlkYWQgZGUgbGEgSW5mb3JtYWNpw7NuMSkwJwYJKoZIhvcNAQkBFhphc2lzbmV0QHBydWViYXMuc2F0L
mdvYi5teDEmMCQGA1UECQwdQXYuIEhpZGFsZ28gNzcsIENvbC4gR3VlcnJlcm8xDjAMBgNVBBEMBTA2MzAw
MQswCQYDVQQGEwJNWDEZMBcGA1UECAwQRGlzdHJpdG8gRmVkZXJhbDESMBAGA1UEBwwJQ295b2Fjw6
FuMRUwEwYDVQQtEwxTQVQ5NzA3MDFOTjMxITAfBgkqhkiG9w0BCQIMElJlc3BvbnNhYmxlOiBBQ0RNQTAe
Fw0xNDA2MDYxNDU5MDZaFw0xODA2MDYxNDU5MDZaMIIBBTE3MDUGA1UEAxMuRElTVFJJQlVJRE9SQS
BERSBDQVJORVMgU0FOVEEgTUFSQ0VMQSBTQSBERSBDVjE3MDUGA1UEKRMuRElTVFJJQlVJRE9SQSBE
RSBDQVJORVMgU0FOVEEgTUFSQ0VMQSBTQSBERSBDVjE3MDUGA1UEChMuRElTVFJJQlVJRE9SQSBERS
BDQVJORVMgU0FOVEEgTUFSQ0VMQSBTQSBERSBDVjElMCMGA1UELRMcRENTMDAxMjIxMTY0IC8gRlV
BQjc3MDExN0JYQTEeMBwGA1UEBRMVIC8gRlVBQjc3MDExN01ERlJOTjA5MREwDwYDVQQLFAhQcnVlYm
FfMTCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEAhVicJ41LJkM/sROtrirXc5z1H10UTNYubjBg3SCAVtW
NRA150FX7V1wYx1y+7oUW8ZxR8MZuysP2RyUHRy64j0A0gqvwCJIkJMIqcl9DTSoxDJtiMkudypSiJ3Gdut/dlFo1sI
yFrCcafoSn3EXcNz2AIQXV+NK2RQ0DYIDlVV8CAwEAAaMdMBswDAYDVR0TAQH/BAIwADALBgNVHQ8EB
AMCBsAwDQYJKoZIhvcNAQELBQADggIBAF2ABRqiyPA4hsuBqvzjMP5X25oQfEYr+gXOtsIEoJsBd2m2A6x6oCb
WkoLw29Ey8uQHwKzAW/sBSxuZjagx5ybMxmL4BQ5PFuH7+y2lqRm02Jt0oA9fphsaCPQSGXE/dei3nWIEoBx5Up0
5Ux7S8RD+PH5NjtZIXDXEVkaYNZ/zNNyXOlqhnFa7YTwNUt5g4lvvXNkb2godyXbFBwZPUY48FP93H+0oY8WsH
yghBdL65xkb0AO+oZmcGoYA7AjjxFWCuL3hxHOlrffNuMlzxc279H7BprbMPKbaiMfo2UPjNNxNZ+CtSlJ82ITIFJ2S
V9c2uGM5XjDWTrBxxYqxxZHERQEYkHFM76qGwiuoA1pMAsTWWe1KvAY3I7n0qNbMnldlVdZQ1jOtTTsSU/lHt
SkQJ6E4m9jiyKga/LkmjNLBa/NMbqbYN0d+NbBCWfWtshXx9go1EDbTfnFFlDPZ6yiY2/U25WozcLS4Y3ILbOH9KJ
1J/3f+CD3iSQxnCdHfeSQY2+7sYBFdFwEVxlbzjBSsS5Uod40ETrhwi+1hhV0Um92m9AHnK3OJS5+ZrFqFNosoLrvk
XnHcbCf9G0PbV+kZjo0NKCKFnK6AEnhYM1Y9Ldnznc9ZBzaDJ0uxxM/wXbFlpjHJW3K+X1Cd4GPCHcepsJK0w8
```

```
v3vdlcAxLCtq9t"
            NumCert="20001000000300003702"
            Sello="Dn2p1bnV43KxThmHo9GOIDm+f0krvEUD/OnyYGWu+4m82yJR+8fYwBvb3+MtTohZJZ//+JhnDpo
a+GF8yLR/IqE+cd6bsTaOhFJN/rqTprh+7njyElyS3sOf15ybaiJrUKnTY7wn1DrPBO9/I9L+RtamLZslCKbRCN+5PUJL
UZw=" Version="1.0">
 <retenciones:Emisor RFCEmisor="DCS001221164" />
 <retenciones:Receptor Nacionalidad="Nacional">
     <retenciones:Nacional RFCRecep="AAQM610917QJA" />
 </retenciones:Receptor>
 <retenciones:Periodo Ejerc="2014" MesFin="1" MesIni="1" />
 <retenciones:Totales montoTotRet="30" montoTotExent="70" montoTotGrav="90" montoTotOperacion="100" />
 <retenciones:Complemento>
      <tfd:TimbreFiscalDigital xmlns:tfd="http://www.sat.gob.mx/TimbreFiscalDigital"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.sat.gob.mx/TimbreFiscalDigital
http://www.sat.gob.mx/TimbreFiscalDigital/TimbreFiscalDigital.xsd"
version="1.0" UUID="22d19f5a-d4ec-4347-95f0-b7670ae4e218"
FechaTimbrado="2018-01-01T13:07:45"
selloCFD="Dn2p1bnV43KxThmHo9GOIDm+f0krvEUD/OnyYGWu+4m82yJR+8fYwBvb3+MtTohZJZ//+JhnDpoa+GF
8yLR/IqE+cd6bsTaOhFJN/rqTprh+7njyElyS3sOf15ybaiJrUKnTY7wn1DrPBO9/I9L+RtamLZslCKbRCN+5PUJLUZw=
"
noCertificadoSAT="20001000000300003693"
selloSAT="okAdgzCGJRPoXlkMaugInjmccIrV855BDPjZpBm/fAYQXpZFCJ52naOaQBR3gsf+cM69iv0pVz50bAaZ/O
J7Qc9GjZ8daY0L2qDu6W3+3sYT0pE0rrt2c6DoliC5J+Hrl/dD0H9CXOObiKFzAs0+Jx7jCBjXFIOHCN8ML4gslI8="
/>
</retenciones:Complemento></retenciones:Retenciones>
```

- Nota: se recomienda no declarar atributos xlmns (o namespaces) en los elementos interiores (datos de Emisor, Receptor, etc) y dejarlos todos en el elemento principal (es decir, en el elemento <Retencion>), esto para evitar errores de parseo del XML.

10. Elementos para cancelación de un CFDI

10.1. Signature

Para obtener el valor del <signature> en el comando <cancel> se necesitan los siguientes datos:

- El RFC emisor de la factura en mayúsculas (RfcEmisor="AAA010101AAA").
- La fecha y hora actual (no la de emisión de la factura) en formato ISO (Fecha="2018-01-01T14:14:40"). Es importante asegurarse de que se está utilizando la hora de México, y sin time zones extranjeros.
- El UUID de la factura a cancelar, en mayúsculas (<UUID>AA97B177-9383-4934-8543-0F91A7A02836</UUID>). Es importante no confundir con el UUID enviado en el comando <cancel> descrito en esta sección 3.2. El UUID dentro del comando <cancel> deberá estar en minúsculas, mientras que la generación del signatureValue requiere este UUID en mayúsculas.
- El certificado CSD (no FIEL) con extensión .cer autorizado por el SAT y correspondiente al RFC que emitió la factura.
- La llave .key otorgada por el SAT y correspondiente al RFC que emitió la factura, así como la contraseña de la misma.

Una vez se cuente con estos datos, se procede a llenar la siguiente estructura en XML con la fecha y hora actual, el RFC Emisor, así como el UUID en mayúsculas. Se recomienda generar el XML sin saltos de línea ni espacios en blanco entre los elementos. Se recomienda también

respetar el orden de los elementos xmlns (namespaces), como se muestra en el siguiente ejemplo:

```
<Cancelacion xmlns="http://cancelacfd.sat.gob.mx"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" Fecha="2018-01-
01T14:14:40" RfcEmisor="AAA010101AAA"><Folios><UUID>AA97B177-9383-4934-
8543-0F91A7A02836</UUID></Folios></Cancelacion>
```

Una vez se tenga esta estructura en XML, se realiza un digest SHA-1 de tal manera que este digest arrojará como resultado *4e660ed992bec29aa1e8e2e2add6e7b5db3d7458* (representación Hex).

Algunas herramientas para calcular el digest SHA-1 requiren del *hexdump* del mensaje a procesar (http://www.fileformat.info/tool/hexdump.htm), en este caso el *hexdump* correspondiente a la estructura xml se muestra a continuación:

```
000000  3c 43 61 6e 63 65 6c 61 63 69 6f 6e 20 78 6d 6c  <Cancelacion xml
000010  6e 73 3d 22 68 74 74 70 3a 2f 2f 63 61 6e 63 65  ns="http://cance
000020  6c 61 63 66 64 2e 73 61 74 2e 67 6f 62 2e 6d 78  lacfd.sat.gob.mx
000030  22 20 78 6d 6c 6e 73 3a 78 73 64 3d 22 68 74 74  " xmlns:xsd="htt
000040  70 3a 2f 2f 77 77 77 2e 77 33 2e 6f 72 67 2f 32  p://www.w3.org/2
000050  30 30 31 2f 58 4d 4c 53 63 68 65 6d 61 22 20 78  001/XMLSchema" x
000060  6d 6c 6e 73 3a 78 73 69 3d 22 68 74 74 70 3a 2f  mlns:xsi="http:/
000070  2f 77 77 77 2e 77 33 2e 6f 72 67 2f 32 30 30 31  /www.w3.org/2001
000080  2f 58 4d 4c 53 63 68 65 6d 61 2d 69 6e 73 74 61  /XMLSchema-insta
000090  6e 63 65 22 20 46 65 63 68 61 3d 22 32 30 31 32  nce" Fecha="2012
0000a0  2d 30 39 2d 33 30 54 31 34 3a 31 34 3a 34 30 22  -09-30T14:14:40"
0000b0  20 52 66 63 45 6d 69 73 6f 72 3d 22 41 41 41 30   RfcEmisor="AAA0
0000c0  31 30 31 30 31 41 41 41 22 3e 3c 46 6f 6c 69 6f  10101AAA"><Folio
0000d0  73 3e 3c 55 55 49 44 3e 41 41 39 37 42 31 37 37  s><UUID>AA97B177
0000e0  2d 39 33 38 33 2d 34 39 33 34 2d 38 35 34 33 2d  -9383-4934-8543-
0000f0  30 46 39 31 41 37 41 30 32 38 33 36 3c 2f 55 55  0F91A7A02836</UU
000100  49 44 3e 3c 2f 46 6f 6c 69 6f 73 3e 3c 2f 43 61  ID></Folios></Ca
000110  6e 63 65 6c 61 63 69 6f 6e 3e                    ncelacion>
```

El digest que se calcula debe ser codificado en base 64, para el ejemplo el resultado es *TmYO2ZK+wpqh6OLirdbntds9dFg=*.

Ya que tenemos el digest codificado en base 64, se procede a agregarlo en la estructura <SignedInfo>, definida a continuación (dicha estructura es estándar, lo único variable es el valor del digest, dentro del elemento <DigestValue>).

```
<SignedInfo xmlns="http://www.w3.org/2000/09/xmldsig#"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"><CanonicalizationMethod
Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-
20010315"></CanonicalizationMethod><SignatureMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-
sha1"></SignatureMethod><Reference URI=""><Transforms><Transform
Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-
signature"></Transform></Transforms><DigestMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"></DigestMethod><Digest
Value>TmYO2ZK+wpqh6OLirdbntds9dFg=</DigestValue></Reference></SignedInfo>
```

Ya que se tiene esta estructura, se aplica un digest en SHA-1 (de la misma forma que con la estructura <Cancelacion>). El resultado es *1dccb689cf418bbb0ee55bd0231b8de0c608d57f* (representación Hex).

Una vez que tenemos el digest del <SignedInfo> se procede a firmarlo con la llave privada (archivo .key) del emisor del CFDI. Esto nos arrojará el siguiente valor:

OjXZOASLum2/DYwBr6IhUnNhsws/CN7ppXOojRu0WK0iAiu7v484h8zVB3P+jzNL80agSyNgjebpJZ4Rs w3FG8dtOR7+A2wRumNVXr5O5imCh+MyUJsemQz1h0Ple/gNwJ3dpLfzYab/2UGnOFepCxelO9Oavn5 9iw2hcGT7+Fk=

Si bien este valor ya se encuentra codificado en base 64, se requiere codificarlo una vez más, de forma que se obtenga el siguiente valor:

T2pYWk9BU0x1bTIvRFl3QnI2SWhVbk5oc3dzL0NON3BwWE9valJ1MFdLMGlBaXU3djQ4NGg4elZCM1Ar anpOTDgwYWdTeU5namVicEpaNFJzdzNGRzhkdE9SNytBMndSdW1OVlhyNU81aW1DaCtNeVVKc2VtUX oxaDBQbGUvZ053SjNkcExmellhYi8yVUduT0ZlcEN4ZWxPOU9hdm41OWl3MmhjR1Q3K0ZrPQ==

Este es finalmente el valor que se proporcionará dentro del elemento <signature> del comando <cancel>:

<signature>T2pYWk9BU0x1bTIvRFl3QnI2SWhVbk5oc3dzL0NON3BwWE9valJ1MFdLMGlBaXU3djQ4NGg4elZ CM1AranpOTDgwYWdTeU5namVicEpaNFJzdzNGRzhkdE9SNytBMndSdW1OVlhyNU81aW1DaCtNeVVKc2VtU XoxaDBQbGUvZ053SjNkcExmellhYi8yVUduT0ZlcEN4ZWxPOU9hdm41OWl3MmhjR1Q3K0ZrPQ==</signatur e>

9.1.1 Ejemplo de generación de signature en Java

Para generar la estructura de cancelación, se procede a vaciar los datos requeridos (fecha, RFC emisor en mayúsculas y UUID en mayúsculas).
Como se puede apreciar, se genera el XML con el RFC, la fecha y el UUID requeridos (ver el objeto preSign).

```
SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd'T'HH:mm:ss");

emiterRfc = emiterRfc.toUpperCase();

// Genera xml canonizado de prefirma
StringBuilder preSign = new StringBuilder();
preSign.append("<Cancelacion xmlns=\"http://cancelacfd.sat.gob.mx\"
xmlns:xsd=\"http://www.w3.org/2001/XMLSchema\"
xmlns:xsi=\"http://www.w3.org/2001/XMLSchema-instance\" ");
preSign.append("Fecha=\"" + sdf.format(fecha) + "\" ");
preSign.append("RfcEmisor=\"" + emiterRfc + "\">");
for (String uuid : uuids) {
      preSign.append("<Folios><UUID>" + uuid.toUpperCase() +
"</UUID></Folios>");
}
preSign.append("</Cancelacion>");
```

Posteriormente se realiza un digest en SHA-1 a este XML (objeto preSign). En este ejemplo usaremos la librería Bouncy Castle Provider versión 1.57 (https://www.bouncycastle.org/latest_releases.html) registrandolo como Provider previamente:

```
import java.security.Security;
```

```
import org.bouncycastle.jce.provider.BouncyCastleProvider;

public class CopyOfSha1Util {

	public static void main(String[] args) {
		Security.addProvider(new BouncyCastleProvider());
		//.
		//.
		//.
	}
}
```

Una vez registrado el provider las siguientes líneas generarán el digest SHA-1 y se almacena en un arreglo de bytes:

```
MessageDigest hasher = MessageDigest.getInstance("SHA-1", "BC");
hasher.update(preSign.toString().getBytes("utf-8"));
byte[] digest = hasher.digest();
```

Posteriormente se codifica este digest en Base64 (lo cual arrojaría la cadena *TmYO2ZK+wpqh6OLirdbntds9dFg=)* y se almacena en la variable *preSignHash*, como se ve aprecia en el siguiente código:

```
String preSignHash;
	try {
		MessageDigest hasher = MessageDigest.getInstance("SHA-1", "BC");
		hasher.update(input.toString().getBytes("utf-8"));
		byte[] digest = hasher.digest();

		preSignHash = new String(Base64.encode(digest), "utf-8");
	} catch (UnsupportedEncodingException | NoSuchAlgorithmException |
NoSuchProviderException e) {
		throw new RuntimeException("Error al generar digest para
cancelación", e);
	}
```

Ya con el valor del digest en base 64 (*TmYO2ZK+wpqh6OLirdbntds9dFg=*), contenido en la variable preSignHash, se construye la estructura <SignedInfo>:

```
// Genera xml canonizado de SignedInfo
StringBuilder signedInfo = new StringBuilder();
signedInfo.append("<SignedInfo xmlns=\"http://www.w3.org/2000/09/xmldsig#\"
xmlns:xsd=\"http://www.w3.org/2001/XMLSchema\"
xmlns:xsi=\"http://www.w3.org/2001/XMLSchema-instance\">");
signedInfo.append("<CanonicalizationMethod Algorithm=\"http://www.w3.org/TR/2001/REC-xml-
c14n-20010315\">");
signedInfo.append("</CanonicalizationMethod>");
signedInfo.append("<SignatureMethod Algorithm=\"http://www.w3.org/2000/09/xmldsig#rsa-
sha1\">");
signedInfo.append("</SignatureMethod>");
signedInfo.append("<Reference URI=\"\">");
signedInfo.append("<Transforms>");
signedInfo.append("<Transform Algorithm=\"http://www.w3.org/2000/09/xmldsig#enveloped-
signature\">");
signedInfo.append("</Transform>");
signedInfo.append("</Transforms>");
signedInfo.append("<DigestMethod Algorithm=\"http://www.w3.org/2000/09/xmldsig#sha1\">");
signedInfo.append("</DigestMethod>");
signedInfo.append("<DigestValue>");
signedInfo.append(preSignHash);
```

```
signedInfo.append("</DigestValue>");
signedInfo.append("</Reference>");
signedInfo.append("</SignedInfo>");
```

Resta obtener el digest (en SHA-1) de la estructura <SignedInfo> y firmarla con la llave privada (archivo .key) del emisor del CFDI. En este ejemplo el objeto privateKey es de la clase RSAPrivateKey y previamente se inicializó con el archivo .key y su contraseña. Como se puede apreciar, el digest en SHA-1 se hace internamente, al especificar el algoritmo "SHA1withRSA":

```
// firmar signedInfo con llave privada
try {
      byte[] data = signedInfo.toString().getBytes("utf-8");
      Signature signature = Signature.getInstance("SHA1withRSA", "BC");
      signature.initSign(privateKey);
      signature.update(data);
      byte[] base64signatureValue = Base64.encode(signature.sign());

      return new String(base64signatureValue, "utf-8");
```

En este punto ya se cuenta con el hash de la estructura <SignedInfo>, ya firmado con la llave privada (archivo .key) del emisor de la factura a cancelar:

OjXZOASLum2/DYwBr6IhUnNhsws/CN7ppXOojRu0WK0iAiu7v484h8zVB3P+jzNL80agSyNgjebpJZ4Rsw3FG8dtOR7+A2wRumNVXr5O5imCh+MyUJsemQz1h0Ple/gNwJ3dpLfzYab/2UGnOFepCxelO9Oavn59iw2hcGT7+Fk=

Sin embargo, es necesario codificarla una vez más en base 64 antes de enviarla en el elemento <signature> del comando <cancel>:

<signature>T2pYWk9BU0x1bTIvRFl3QnI2SWhVbk5oc3dzL0NON3BwWE9valJ1MFdLMGlBaXU3djQ4NGg4elZCM1AranpOTDgwYWdTeU5namVicEpaNFJzdzNGRzhkdE9SNytBMndSdW1OVlhyNU81aW1DaCtNeVVKc2VtUXoxaDBQbGUvZ053SjNkcExmellhYi8yVUduT0ZlcEN4ZWxPOU9hdm41OWl3MmhjR1Q3K0ZrPQ==</signature>

Es así como se construye el elemento signature. Se puede encontrar una especificación aún más detallada en la siguiente dirección: http://www.cryptosys.net/pki/satcancelcfd.html

10.2. Certificate

El comando <cancel> definido por INSIGNA requiere del archivo .CER (CSD, no FIEL) del emisor del CFDI a cancelar, en formato base 64. Para ello basta con cargar dicho archivo a un arreglo de bytes y realizar la codificación con el charset UTF-8 y en base 64.

A continuación se muestra un ejemplo en Java:

```
X509Certificate x509Certificate = null;
try {
    x509Certificate = SecurityUtil.readCertificate(byteArrayCertificate);
} catch (CertificateException e) {
    throw new RuntimeException("Error al obtener el certificado en el proceso de firmado de la factura: ", e);
}

String certificate = null;
Charset charset = Charset.forName("UTF-8");
try {
    certificate = new String(org.bouncycastle.util.encoders.Base64.encode(x509Certificate.getEncoded()), charset);
} catch (CertificateEncodingException e) {
    throw new RuntimeException("Error al obtener el certificado en el proceso de firmado de la factura: ", e);
}
```

Hecho esto, el contenido del String certificate es lo que se incluye dentro del elemento <certificate>. Ejemplo:

<certificate>MIIEMTCCAxmgAwIBAgIUMDAwMDEwMDAwMDAxMDQ2MjAxMzAwDQYJKoZIhvcN
AQEFBQAwggE2MTgwNgYDVQQDDC9BLkMuIGRlbCBTZXJ2aWNpbyBkZSBBZG1pbmlzdHJhY2nDs
24gVHJpYnV0YXJpYTEvMC0GA1UECgwmU2VydmljaW8gZGUgQWRtaW5pc3RyYWNpw7NuIFRyaW
J1dGFyaWExHzAdBgkqhkiG9w0BCQEWEGFjb2RzQHNhdC5nb2IubXgxJjAkBgNVBAkMHUF2LiBIaW
RhbGdvIDc3LCBDb2wuIEd1ZXJyZXJvMQ4wDAYDVQQRDAUwNjMwMDELMAkGA1UEBhMCTVgx
GTAXBgNVBAgMEERpc3RyaXRvIEZlZGVyYWwxEzARBgNVBAcMCkN1YXVodGVtb2MxMzAxBgk
qhkiG9w0BCQIMJFJlc3BvbnNhYmxlOiBGZXJuYW5kbyBNYXJ0w6+fZ7YT5oakREdTsVIdOP77oZjMqT
V3UwyCQnFmC4n/sJw6y0t3OeXgAwfsXT6vWxAgMBAAGjHTAbMAwGA1UdEwEB/wQCMAAwCwY
DVR0PBAQDAgbAMA0GCSqGSIb3DQEBBQUAA4IBAQC/Q3EgC3qeC7C5jS4tefeVRFkTuzypFw6d63h
QGA3sjUe/H9agMG2wBx8Ag86E2Eeb3rJuVKVqZUfEAffXy6a1u7/Kx0IzuhT8o
1uZXogQ29zczAeFw0xMTEwMzAyMjUxMzBaFw0xMzEwMjkyMjUxMzBaMIHRMSgwJgYDVQQDEx9
EVUxDRSBCRVRISU5BIFJPRFJJR1VFWiBBTkdFTEVTMSgwJgYDVQQpEx9EVUxDRSBCRVRISU5B
IFJPRFJJR1VFWiBBTkdFTEVTMSgwJgYDVQQKEx9EVUxDRSBCRVRISU5BIFJPRFJJR1VFWiBBTkd
FTEVTMRYwFAYDVQQtEw1ST0FENzcxMTI1QU00MRswGQYDVQQFExJST0FENzcxMTI1TUdURE
5MMDExHDAaBgNVBAsUE01hdHJpeiBMYXMgQW3pcmljYXMwgZ8wDQYJKoZIhvcNAQEBBQADg
Y0AMIGJAoGBAL5/q9etKCjJ8fHDEVI4xUwfV02LpGIeypNlpdS0HyQQGojjdYGgantt/SFhLc0+1Sq9LjE
XmkuB0rlcHqpBxpIcXBo/r9Rt6sn+RqQ+qXNoNtu6VKFNxd3bR4o1ilsMHqEaTwAQscDkGfi9H0dRYeU
U048z6lsgvJDaLw1+sgceI+hyrnuMdMnfT68sekIoL+cKoC54B8+3e5ZsoQfI1y/ioH9ZW8ghC3DsWpydUN
pdrftbGVAip5WQzvVZUUb5k1Y6WhQJnLpRzx/ndi5/VO6+J4aIbSS/qfiWF418x5EkvHpC/h05hU4E6ye5bj
tlq9+S7N6j6Ux/V7MeJL0+9zqW8lcqYEJ</certificate>