



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

MODELO DE SOLUCIÓN PARA DOCUMENTOS TRIBUTARIOS ELECTRÓNICOS EN
DISPOSITIVOS MÓVILES

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL EN COMPUTACIÓN

CRISTIÁN ALFREDO ACUÑA MUÑOZ

PROFESORA GUÍA:
CECILIA BASTARRICA PYÑEIRO

MIEMBROS DE LA COMISIÓN:
BENJAMIN BUSTOS CARDENAS
PABLO GONZALEZ JURE

Este trabajo ha sido parcialmente financiado por Acepta S.A.

SANTIAGO DE CHILE

2014

Resumen

Si bien, los documentos tributarios electrónicos (DTEs) forman parte de la realidad nacional hace más de 10 años, hasta la fecha las soluciones móviles para emisión de DTEs no se han masificado y en general se han desarrollado en industrias muy particulares, sobre dispositivos de elevado costo tipo hand held o bajo modelos de operación que de alguna manera están sujetos a la disponibilidad de la red.

Buscando masificar el uso de DTEs, a través de una alternativa más económica desde el punto de vista de la infraestructura y, a la vez, buscando un modelo que otorgue niveles de autonomía y seguridad a los usuarios, es que por medio de este trabajo se propone un modelo de operación para dispositivos móviles, smartphones o tablets, con un ejemplo aplicado en SO Android.

El modelo se basa en separar las acciones de firma y timbre del DTE, logrando la generación del timbre en el dispositivo móvil y la firma a nivel de un servidor centralizado. Todo lo demás es agregar funcionalidades y servicios para hacer crecer un producto de emisión de documentos electrónicos, cumpliendo con el objetivo de operar en dispositivos más económicos, brindando autonomía operacional.

En este sentido, la empresa Acepta, Autoridad Certificadora y proveedora de soluciones de factura electrónica, cuenta con el soporte y la infraestructura necesarios para complementar la emisión del documento tributario, otorgando elementos de gestión que permiten dar cumplimiento a los requerimientos legales de custodia e intercambio que define la normativa del Servicio de Impuestos Internos, para la emisión de DTEs.

Agradecimientos

A mis padres, Oscar y Betty, fuente inagotable de apoyo y consejo.

A don Pablo Izquierdo, jefe y maestro, quien impulsó y facilitó poder retomar esta última etapa de la carrera.

Quiero agradecer a María Cecilia por su cordial guía, aceptando representar a este memorista que apareció a última hora.

También, un especial agradecimiento a Angélica, por su orientación, ayuda y permanente aliento, durante todo este proceso.

A mis hijos, Amanda, Camila, Felipe y Alfredo, quienes muchas veces sin saberlo tuvieron que ceder paseos, fines de semana o aceptar que papá estaba muy ocupado.

Encarecidamente, gracias a Pilar, mi mujer, mi amada compañera, a ella agradezco cada día su presencia, su incondicional entrega, su fuerza, ánimo y empuje que se transforman en la energía necesaria para todo lo que hay que hacer.

Tabla de contenido

1	Introducción	1
2	Antecedentes	3
	Documentos tributarios electrónicos	3
	Uso de XML	3
	Timbre del documento tributario	4
	Firma del documento tributario	6
	Emisión Móvil	7
3	Especificación del problema	9
	Pérdida de información	9
	Interfaz de usuario	9
	Asignación de folios	12
	Gestión de documentos	12
4	Descripción de la solución	14
	Solución planteada.....	14
	4.1.1 Timbre en el dispositivo	14
	4.1.2 Servicio central	17
	Aplicación móvil e interfaz de usuario	19
	Asignación de folios	23
	Generación del DTE.....	25
	Gestión de los documentos: Integración con Acepta	27
5	Resultados	28
	Análisis de envío de documentos hacia la API REST	28
	Validación del timbre.....	29
	Generación de documentos	30
6	Conclusiones	32
7	Glosario	34
8	Bibliografía.....	36

Anexo A. Código de Autorización de Folios (SII).....	38
A 1. Proceso de generación de una Autorización de Folios.	38
A 2. Estructura de la Autorización	39
A 3. Almacenamiento y Uso del CAF	44
Anexo B. Timbre Electrónico del DTE (SII)	45
B 1. Introducción.....	45
B 2. Generación De Un Timbre	45
B 3. Estructura.....	45
B 4. Consideraciones para la Generación y Firma del Timbre Electrónico.....	48
B 5. Reglas Para La Generación e Impresión Del Timbre PDF417.....	49

Índice de tablas

Tabla 1: Códigos para DTE	3
Tabla 2: Estructura del timbre electrónico digital <TED>	5
Tabla 3: XML resumido de un DTE, para destacar timbre <TED> y firma <Signature> ...	6
Tabla 4: Elementos de la firma electrónica	6
Tabla 5: Datos mínimos para generar un DTE	11
Tabla 6: Datos mínimos para generar una Boleta Electrónica	11
Tabla 7: Generación del TED	15
Tabla 8: Extracto clase CreaPDF417 y el método escribePDF	16
Tabla 9: Extracto API RESTful en Python utilizando micro-framework Flask.....	18
Tabla 10: Clase Documento	20
Tabla 11: Solicitud de folios.....	24
Tabla 12: Respuesta a solicitud de folios	24
Tabla 13: XML generado desde la API RESTMóvil.....	26
Tabla 14: XSLT: Ejemplo definición de variables	26
Tabla 15: XSLT: Extracto de generación del DTE.....	26
Tabla 16: Tiempo de respuesta de los servicios web.....	29
Tabla 17: Extracto: Comparación timbre Android vs timbre en el servidor.....	30

Índice de ilustraciones

Ilustración 1: Evolución de los contribuyentes inscritos en Facturación Electrónica (fuente: www.sii.cl)	2
Ilustración 2: Modelo de emisión de documentos electrónicos.....	14
Ilustración 3: Timbre PDF417 del TED	16
Ilustración 4: Comunicación móvil – servidor	19
Ilustración 5: Campos principales.....	20
Ilustración 6: Diagrama de flujo definido para la aplicación.....	21
Ilustración 7: Primera pantalla, selecciona tipo de documento.....	22
Ilustración 8: Aplicación controla estado de conexión	23
Ilustración 9: Asignación de folios	23
Ilustración 10: Generación del DTE.....	25
Ilustración 11: Modelo final de emisión de documentos electrónicos	27
Ilustración 12: Gráfico, tiempo de respuesta WS, peticiones internas.....	28
Ilustración 13: Gráfico, tiempo de WS, peticiones desde la red	28
Ilustración 14: Boleta electrónica generada desde el móvil.....	31
Ilustración 15: Factura electrónica generada desde el móvil.....	31

1 Introducción

La factura electrónica, normada por el Servicio de Impuesto Internos (SII), es un documento digital, firmado y timbrado electrónicamente, que tiene validez legal tributaria como medio de respaldo de las operaciones comerciales entre contribuyentes, y reemplaza a las facturas tradicionales de papel.

En este sentido, el SII, a través de su portal, provee un servicio gratuito para las empresas MiPyme. Para las demás empresas, el mercado ofrece alternativas que van desde soluciones in-house hasta outsourcing total a través de portales y servicios Web.

Luego de 10 años de la puesta en marcha de esta iniciativa, hoy existe un amplio abanico de soluciones de factura electrónica, ya sea incluidas en el ERP de la empresa o que otorgan la funcionalidad a los sistemas tradicionales y cumplen el objetivo de permitir la emisión de los Documentos Tributarios Electrónicos (DTEs).

Sin embargo, aún no se logra el objetivo de la masificación total de los DTEs, ya que como se muestra en la figura 1, hay poco más de 95.000 empresas que son contribuyentes electrónicos de un total superior a 800.000, o sea, aproximadamente, sólo el 12% de las empresas del país es contribuyente electrónico.

Y si de este 90% restante, el 95%, según el SII, es MiPyme, entonces, ¿por qué no se ha logrado una mayor penetración del modelo de factura electrónica?

En parte, esto se debe a que la solución gratuita del SII no es aplicable a la realidad de muchas empresas ya sea porque no cuentan con un computador con acceso a Internet, como la mayoría de los comercios medianos a pequeños; o porque operacionalmente no aplica a la realidad de la empresa, como es el caso de la venta en ruta y el POS¹, por dar un par de ejemplos.

Considerando que Chile lidera el uso de tablets y smartphones en Latinoamérica², gracias a los bajos precios que han favorecido la compra de estos aparatos, y que de acuerdo a la Subsecretaría de Telecomunicaciones, las conexiones a Internet desde smartphones representan cerca del 81% de la navegación móvil, la cual está en permanente aumento, alcanzando un 770% de crecimiento en los últimos 4 años³, es esperable que la mayoría de las empresas que no cuenten con equipamiento computacional, sí cuenten con un smartphone con acceso a Internet o al menos puedan adquirirlo en el corto plazo.

En consecuencia, construir aplicaciones móviles que ayuden a resolver problemas de negocio, que abaraten costos, faciliten la gestión y sean un aporte operacional, es uno de los desafíos y tendencia en la actualidad desde el punto de vista tecnológico.

¹ POS – Point of Sale (Punto de venta).

² [Diario Pyme](#).cl, 27-06-2013/Tendencias

³ <http://www.subtel.gob.cl/index...>

Esta memoria presenta un modelo de solución para la generación de documentos tributarios electrónicos desde dispositivos móviles y como objetivo práctico, una aplicación en Android basada en el modelo.

La solución final se integra con la infraestructura del proveedor de factura electrónica Acepta, para complementarla a través de los servicios de intercambio y custodia que ofrece la empresa. Esto es posible ya que Acepta, brinda un servicio orientado a la gestión de los DTEs, sin ser el foco la emisión sino el servicio, para ser un intermediario entre la empresa, sus clientes y proveedores.

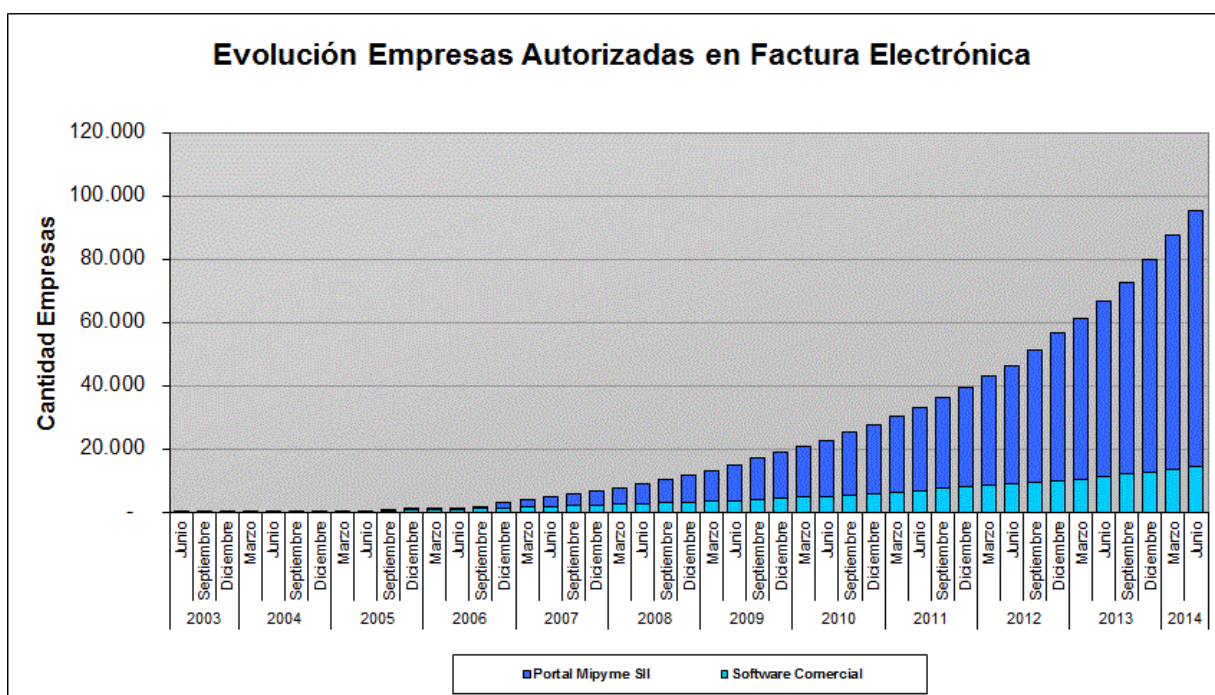


Ilustración 1: Evolución de los contribuyentes inscritos en Facturación Electrónica (fuente: www.sii.cl)

2 Antecedentes

Documentos tributarios electrónicos

La factura electrónica, normada por el SII, corresponde a un documento digital en formato XML, firmado y timbrado electrónicamente, que tiene validez legal tributaria como medio de respaldo de las operaciones comerciales entre contribuyentes y reemplaza a las facturas tradicionales de papel.

Esta definición se extiende para un listado amplio de documentos como por ejemplo facturas afectas y exentas, notas de crédito, notas de débito, boletas afectas y exentas, documentos de exportación y guías de despacho, que están dentro de los más utilizados. Así cada documento es representado por un código, como se puede ver en la siguiente tabla:

Código	Tipo Documento
33	Factura Electrónica
34	Factura No Afecta o Exenta Electrónica
39	Boleta Electrónica
41	Boleta Exenta Electrónica
43	Liquidación-Factura Electrónica
46	Factura de Compra Electrónica.
52	Guía de Despacho Electrónica
56	Nota de Débito Electrónica
61	Nota de Crédito Electrónica
110	Factura de Exportación Electrónica
111	Nota de Débito de Exportación Electrónica
112	Nota de Crédito de Exportación Electrónica

Tabla 1: Códigos para DTE

En particular, este trabajo se enfoca en los documentos: 33, 39, 52, 56 y 61, por tratarse de los de mayor uso en el mercado.

Uso de XML

El Servicio de Impuestos Internos ha decidido que el formato en que se generarán los documentos tributarios electrónicos sea XML o Lenguaje Extensible de "Etiquetado", eXtensible Markup Language. Este metalenguaje proporciona una forma de aplicar etiquetas para describir las partes que componen un documento, permitiendo además el intercambio de documentos entre diferentes plataformas.

La versión 1.0 del lenguaje XML es una recomendación del W3C (W3 Consortium) desde Febrero de 1998, pero está basado en estándares anteriores como SGML (Standard Generalized Markup Language, ISO 8879).

El formato estándar “Extensible Markup Language (XML)”⁴, tiene varias características que lo hacen conveniente, entre las que se puede destacar:

- Es un estándar abierto, flexible y ampliamente utilizado para almacenar, publicar e intercambiar cualquier tipo de información.
- Ofrece portabilidad y utilización de la información a través de las distintas plataformas (permite independizar aplicaciones de datos).
- Es ampliamente soportado por diversas aplicaciones en distintas plataformas y existen múltiples bibliotecas para diversos lenguajes de programación, ya sea gratuitos o comerciales, que permiten el desarrollo de nuevas aplicaciones.
- Es un formato legible por personas y computadores.
- La especificación de documentos XML es simple, rápida, precisa y concisa.

Para validar si un XML es correcto, el SII provee los archivos de definición del esquema (schema) o archivos XSD (XML Schema Definition). XML Schema, es un lenguaje de esquema utilizado para describir la estructura y las restricciones de los contenidos de los documentos XML de una forma muy precisa, más allá de las normas sintácticas impuestas por el propio lenguaje XML. Se consigue así una percepción del tipo de documento con un nivel alto de abstracción. Fue desarrollado por el World Wide Web Consortium (W3C) y alcanzó el nivel de recomendación en mayo de 2001.⁵

El archivo esquema XSD sirve para definir la correcta estructura de los elementos del documento xml. En concreto, a partir de este archivo se define:

- Los elementos que pueden aparecer en el documento xml.
- Los atributos de los elementos que pueden aparecer en el documento xml.
- Qué elementos son hijos de los elementos principales del documento xml.
- La secuencia en la cual los hijos de los elementos pueden aparecer en el documento xml.
- El número de hijos de los elementos.
- Cuando un elemento es vacío o puede incluir texto.
- El tipo de datos para los elementos y sus atributos.
- Los valores predeterminados para algunos elementos y atributos.
- Si el documento xml no concuerda con la estructura definida por el archivo xsd, entonces el documento xml será erróneo.

En particular, el SII, define un schema para Boletas (39, 41), *schema_boletas* y otro para los demás documentos tributarios (33, 34, 43, 46, 52, 56, 61, 110, 111 y 112), *schema_dte*.

Timbre del documento tributario

El Timbre Electrónico del DTE (TED) consiste en una firma electrónica, sobre los campos que se definen como representativos del documento e incluyendo el Código de

⁴ <http://www.w3.org/XML/>

⁵ http://es.wikipedia.org/wiki/XML_Schema

Autorización de Folios (CAF) proporcionado por el SII. A saber, los elementos que componen el timbre son:

- a) Versión del timbre electrónico
- b) Rut del Emisor
- c) Tipo de Documento
- d) Número de Folio
- e) Fecha de emisión
- f) Rut del Receptor
- g) Razón Social Receptor
- h) Monto total
- i) Descripción del primer Ítem del Detalle
- j) Fecha y hora de generación del timbre electrónico,
- k) Código de Autorización de Folios (proporcionado por el SII)
- l) Algoritmo de firma (Hash y encriptación) que se usó en la firma con que generó el timbre
- m) Firma digital sobre los datos anteriores, con la llave privada entregada por el SII para dicho propósito.

```

<TED version="1.0">
  <DD>
    <RE>11111111-1</RE>
    <TD>39</TD>
    <F>100</F>
    <FE>2014-06-11</FE>
    <RR>11425975-6</RR>
    <RSR>Empresas CAAM Limitada</RSR>
    <MNT>17861</MNT>
    <IT1>Canastos de mimbre</IT1>
    <CAF version="1.0">
      <DA>
        <RE>11111111-1</RE>
        <RS>Ejemplo S.A.</RS>
        <TD>39</TD>
        <RNG>
          <D>1</D>
          <H>200</H>
        </RNG>
        <FA>2014-01-12</FA>
        <RSAPK>
          <M>zf/B...cwx</M>
          <E>QBcs</E>
        </RSAPK>
        <IDK>3</IDK>
      </DA>
      <FRMA>yTfHE...ydmh9fgsj3rv86=</FRMA>
    </CAF>
    <TSTED>2002-06-11T07:34:15</TSTED>
  </DD>
  <FRMT algoritmo="SHA1withRSA">GkdhiwT5a4...09UjhGfsR7l/=</FRMT>
</TED>

```

Tabla 2: Estructura del timbre electrónico digital <TED>

La firma que constituye el timbre electrónico <FRMT> debe ser generada con la llave privada entregada en el CAF por el SII, junto con el rango de folios correspondiente.

En el ANEXO 1, se describen los campos y la estructura del CAF y en el ANEXO 2 se detalla la estructura y campos para el Timbre.

Firma del documento tributario

La firma de un DTE se rige por el estándar XML Signature de la W3C y se utiliza para asegurar la integridad de los datos y como una forma de acreditar la identidad del emisor o firmante del documento.

La firma se aplica sobre el documento completo, en modo Detached (separado). Como se muestra a continuación:

```
<DTE>
  <Documento ID="id_00001">
    <...>
    <TED>
  </TED>
  </Documento>
  <Signature>
</Signature>
</DTE>
```

Tabla 3: XML resumido de un DTE, para destacar timbre <TED> y firma <Signature>

En la siguiente tabla se presentan los elementos de la firma electrónica que se utilizan para la firma de un DTE:

```
[01] <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
[02]   <SignedInfo>
[03]     <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315" />
[04]     <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
[05]     <Reference URI="#id_00001">
[06]       <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
[07]       <DigestValue>TWDBMIcH/kmIW0+JkNkVaj68zk8=</DigestValue>
[08]     </Reference>
[09]   </SignedInfo>
[10]   <SignatureValue>...</SignatureValue>
[11]   <KeyInfo>
[12]     <KeyValue>
[13]       <RSAKeyValue>...</RSAKeyValue>
[14]     </KeyValue>
[15]     <X509Data>
[16]       <X509Certificate></X509Certificate>
[17]     </X509Data>
[18]   </KeyInfo>
[19] </Signature>
```

Tabla 4: Elementos de la firma electrónica

El contenido de las etiquetas *SignedInfo* [02-09], corresponde a la información que está siendo firmada, contiene elementos de validación de la firma y de sus referencias.

CanonicalizationMethod [03], representa el método utilizado para obtener un XML de acuerdo al estándar propuesto por la W3C para un XML Canónico⁶.

⁶ <http://www.w3.org/TR/2001/REC-xml-c14n-20010315>

SignatureMethod [04], es el algoritmo de firma utilizado para convertir el *SignedInfo* canónico en *SignatureValue* [10], en este caso es RSA-SHA1.

Reference [05-08], especifica qué elemento se está firmando a través de la referencia URI. En este caso la referencia #id_00001, corresponde al ID del campo <Documento>, que contiene toda la información legal del DTE más el Timbre. Es decir:

DigestMethod, especifica el algoritmo de hash, en este caso es SHA1. *DigestValue*, contiene el resultado de aplicar el algoritmo de hash.

SignatureValue [10], contiene el resultado de la firma en Base 64.

KeyInfo [14-18], entrega información acerca de la clave para validar la firma y el certificado X.509, que contiene la llave pública.

Emisión Móvil

Las soluciones para la emisión móvil de documentos tributarios electrónicos, en sus inicios se basaron en otorgar la funcionalidad a través de complejas integraciones con sistemas de facturación tradicional corriendo en mini-computadores portátiles de alto costo tipo hand held, por ejemplo, con sistema operativo Linux, Windows Ce o Palm OS, principalmente. La mayor dificultad en estos casos ha sido definir dónde realizar la firma y timbre del XML correspondiente al documento tributario.

Finalmente, las soluciones se han agrupado en las siguientes opciones:

1. Recolector de datos. Uno de los primeros modelos, donde sólo se registra la información en modo offline, se almacenan los datos y posteriormente, se entregan a un servidor central y se generan los documentos al final del día.
2. Envío de data y recepción de imagen del DTE. Por documento se envía data a un servicio centralizado a través de internet, que devuelve la imagen del documento a imprimir.
3. Enviar data y recibir timbre del DTE. Por documento se envía data a un servicio centralizado a través de la red, que devuelve XML del timbre o sólo la imagen del timbre. El dispositivo imprime el documento y el timbre en PDF417.

Otra alternativa utilizada, en el último tiempo, consiste en cargar una aplicación web en el dispositivo móvil, aplicación en la nube, en este caso tanto la firma como el timbre se generan en forma remota. Bajo este modelo, más allá de los problemas de compatibilidad del formulario web con el dispositivo, lo más relevante es que se requiere estar siempre conectado, en otras palabras, dado que existe la posibilidad de falla de comunicación al desplazarse de un punto a otro, no se garantiza la disponibilidad del servicio web cuando se requiere. En este mismo sentido, tampoco se asegura la correcta emisión del documento dada una interrupción o latencia de las comunicaciones.

Por lo tanto, para sortear los problemas anteriores se propone una solución que opere directamente en el dispositivo móvil.

Un ejemplo de esto lo ofrece la empresa *facturamóvil*, con una solución comercial que responde adecuadamente a las necesidades de emisión de documentos tributarios electrónicos en dispositivos móviles, sin embargo, se trata de una aplicación que opera como un cliente de un servicio centralizado y depende 100% de una conexión a la red, en este caso, tanto la firma como el timbre se realizan remotamente en el servidor.

Dado lo anterior, es posible afirmar que a nivel nacional, no existen soluciones para la emisión de DTEs que operen directamente en dispositivos móviles masivos, es decir, en smartphones o tablets de bajo costo y que permitan contar con autonomía operacional.

A continuación se presenta el desarrollo de un modelo de solución para la emisión de DTEs en dispositivos móviles, orientado a brindar autonomía operacional, con un ejemplo aplicado sobre plataforma Android. Se optó por Android por ser la plataforma de mayor presencia en teléfonos inteligentes y tablets, abarcando un mercado superior al 80%, según la IDC (International Data Corporation) a noviembre del 2013.

El modelo busca entregar un esquema de solución genérico que puede ser aplicado sobre cualquier plataforma.

3 Especificación del problema

Como se señaló en el capítulo anterior, el objetivo es poder emitir DTE's desde un dispositivo móvil, en particular facturas, notas de crédito, notas de débito, guías de despacho y boletas, con autonomía operacional.

¿Qué significa esto? Simple, definir la infraestructura y funcionalidades necesarias para que desde el dispositivo se pueda emitir un DTE o Boleta, se esté o no conectado a la red.

La emisión depende, básicamente, de que se genere correctamente el timbre electrónico a partir de los datos del documento.

La solución, en su totalidad, debe contemplar los siguientes puntos:

Pérdida de información

Desde el punto de vista de la data, existen riesgos inherentes en los dispositivos móviles como la pérdida de conexión o intermitencia en la red y por otro lado, al tratarse de equipos livianos, en su mayoría de bolsillo, son más propensos a accidentes como caídas o golpes, robos, fallas, etc. Por lo tanto, la solución debe considerar estos inconvenientes para minimizar los casos de pérdida de información.

Como los datos se crean en el dispositivo, es necesario definir en qué instante se respaldan. Considerando que la aplicación pueda operar sin conectividad por algún periodo, se ha propuesto el siguiente protocolo:

1. Sincronizar al inicio. Es decir, al menos al iniciar la aplicación debe existir conectividad que permita enviar la data pendiente a un respaldo centralizado. Si no se cumple con este protocolo y hay data pendiente la aplicación no se puede iniciar.
2. En caso de existir conectividad, sincronizar siempre.
3. Estando en operación, en caso de perder conectividad, limitar la cantidad de documentos a emitir, es decir, puede generar una cantidad máxima, configurable, de documentos no sincronizados.
4. Imprimir con copia. Cuando no exista conexión a la red, a partir del timbre en la copia impresa y los demás datos del documento, es posible volver a generar un archivo XML equivalente. Esto puede realizarse en forma manual o semi-manual, ya que al escanear el código PDF417 se puede recuperar la información base del timbre y completar manualmente lo que falte del documento. Esto es aplicable a boletas electrónicas principalmente, ya que es más complejo para los demás documentos por requerir mayor información.

Interfaz de usuario

La interfaz que permite el registro de los datos de un documento tributario se debe limitar a capturar la información básica y relevante desde el punto de vista de la normativa y, también, desde el punto de vista operativo, ya que mientras menos

información tenga que registrar el usuario, se facilita el uso y disminuyen las fuentes de error. Esto define un punto de partida, ya que el SII norma más de 200 datos posibles para la emisión de un DTE, lo cual no es práctico de implementar en ninguna GUI (Graphical User Interface) y mucho menos en un dispositivo de bolsillo.

Entonces, teniendo en cuenta que para cada tipo de documento tributario existen campos obligatorios definidos por el SII y campos opcionales, la GUI debe permitir el registro de un mínimo de datos para poder emitir ya sea una boleta o un DTE válido.

Analizando ese mínimo para DTE, se tiene que cumplir al menos con los siguientes campos:

Encabezado DTE (obligatorio)	
Campo	Descripción
<i>Tipo de Documento Electrónico</i>	33: Factura electrónica, 56: Nota de Débito y 61: Nota de Crédito
Folio Documento	Número correlativo, autorizado por el SII
<i>Fecha Emisión</i>	Fecha de emisión contable (AAAA-MM-DD)
Rut Emisor	RUT con guión y dígito verificador
Razón Soc. Emisor	Alfanumérico
Giro Emisor	Alfanumérico
ACTECO	Número. Código de Actividad Económica del Emisor
Dirección Emisor	Alfanumérico
Comuna Emisor	Alfanumérico
Ciudad Emisor	Alfanumérico
<i>Rut Receptor</i>	RUT con guión y dígito verificador
<i>Razón Soc. Receptor</i>	Alfanumérico
<i>Dirección Receptor</i>	Alfanumérico
<i>Comuna Receptor</i>	Alfanumérico
Ciudad Receptor	Alfanumérico
Monto Neto	Número. Suma de Total Ítem afecto (sin Indicador de Exención)
Monto Exento	Número. Suma de Total Ítem con Indicador de Exención activado
Tasa IVA	Número 3 enteros y dos decimales (19,00)
Monto IVA	Monto Neto * Tasa IVA / 100
Monto Total	Monto Neto + Monto Exento + Monto IVA
Detalle DTE (obligatorio, se puede repetir hasta 60 veces)	
Campo	Descripción
<i>Indicador de Exención</i> (campo opcional)	1: Ítem No afecto 2: No facturable
Código Ítem	Alfanumérico. Código del producto (opcional)
<i>Nombre Ítem</i>	Alfanumérico. Nombre del producto
<i>Monto Ítem</i>	Valor neto del producto
Referencias DTE (opcional, se puede repetir hasta 40 veces)	
Campo	Descripción
Tipo Documento Ref.	Código del tipo de documento que se referencia, según tabla del SII.
Indicador de Ref. Global	1:Activo (campo opcional)

Folio Documento Ref.	Folio del documento que se referencia. Debe ser cero cuando está el Indicador de Ref. Global activado. Puede ser alfanumérico si se trata de un documento no tributario.
Fecha Documento Ref.	Fecha de emisión contable del documento que se referencia.
Código de Referencia	1: Anula Documento de Referencia 2: Corrige Texto Documento de Referencia 3: Corrige montos

Tabla 5: Datos mínimos para generar un DTE

O sea, hay tres grupos de información: Encabezado, Detalle y Referencias, donde los dos primeros son obligatorios.

En el caso de las boletas, es suficiente el Encabezado y el Detalle y de acuerdo al esquema definido por el SII, los datos mínimos a considerar son:

Encabezado Boleta Electrónica (obligatorio)	
Campo	Descripción
<i>Tipo Documento Electrónico</i>	39: Boleta electrónica y 41: Boleta electrónica exenta
Folio Documento	Número correlativo, autorizado por el SII
<i>Fecha Emisión</i>	Fecha de emisión contable (AAAA-MM-DD)
Indicador de Servicio	1: Servicios periódicos 2: Servicios periódicos domiciliarios 3: Boleta de venta y servicios (por defecto) 4: Boleta de espectáculos emitida por cuenta de terceros (este valor no estará habilitado inicialmente)
Rut Emisor	RUT con guión y dígito verificador
Rut Receptor	RUT con guión y dígito verificador. Si no se conoce, se utiliza el valor genérico fijo 66.666.666-6
Monto Total	Suma de ítems del Detalle
Detalle Boleta Electrónica (obligatorio, se puede repetir hasta 1000 veces)	
Campo	Descripción
<i>Indicador de Exención</i>	1: Ítem No afecto 2: No facturable (campo opcional)
Código Ítem	Alfanumérico. Código del producto. (opcional)
<i>Nombre Ítem</i>	Alfanumérico. Nombre del producto
<i>Monto Ítem</i>	Valor del producto en monto bruto

Tabla 6: Datos mínimos para generar una Boleta Electrónica

En rigor, la GUI no es un elemento relevante para este trabajo, ya que se puede cambiar, mejorar permanentemente y/o aumentar el número de campos que se deba considerar de acuerdo a las necesidades del usuario final, sin afectar la operación de fondo y el modelo que se quiere implementar.

En este caso, las tablas 5 y 6, representan los campos mínimos necesarios para construir un XML válido para el SII. Se destaca en *rojo*, aquellos campos que deben ser ingresados a través de la interfaz de usuario, ya que los demás campos se pueden calcular o dejar establecidos previamente como parte de la configuración inicial. En este sentido, por ejemplo, los montos totales se obtienen de la suma de los montos del

detalle, el monto de IVA se calcula, pero el porcentaje del IVA puede formar parte de la configuración, junto a los datos del emisor.

Asignación de folios

La asignación de folios consiste en asegurar un folio, es decir, un número al documento que se está tratando de emitir. Este número no puede repetirse y tiene que estar autorizado por el SII, a través de un archivo de Código de Autorización de Folios (CAF) el cual forma parte del timbre electrónico a imprimir. El CAF define el tipo de documento, el RUT de la empresa emisora y un rango de folios autorizado, por lo tanto, se tiene al menos un archivo de CAF por cada tipo de documento.

No es práctico entregar números en forma unitaria o a pedido y menos si se quiere que el punto de emisión tenga autonomía. Para esto, la forma de operar consiste en asignar un rango válido que pueda utilizar sólo el dispositivo que lo recibe.

Para efectos de entregar un rango válido a un punto de emisión se puede otorgar un CAF completo, pero también, un CAF se puede subdividir y obtener sub-rangos lógicos y contar con varios puntos de emisión con un solo CAF físico y cada uno con su propio sub-rango lógico. Esto es aplicable en un modelo con varios POS, donde se requiere una administración centralizada para la asignación de folios y que el sistema en el POS, tenga la opción de solicitar nuevos rangos de folios, dado un umbral de riesgo o el cumplimiento de un porcentaje de uso del rango actual.

Gestión de documentos

El DTE, una vez generado, debe ser enviado al SII para su validación, la entidad fiscalizadora puede responder que el documento está Aceptado, Aceptado con Reparos o Rechazado, entendiendo que sólo el rechazo significa que el folio utilizado sigue disponible.

Luego, una vez aceptado por el SII, el documento se envía al receptor electrónico, esta acción denominada *intercambio entre contribuyentes*, se basa en que cada empresa que opera en el mundo electrónico está registrada en el SII y ha inscrito una casilla de correo donde recibirá los DTEs de proveedores. Dicho correo es público y está disponible en el portal del SII.

De acuerdo a la normativa es responsabilidad del receptor comprobar la validez del DTE recibido, en la página web del SII⁷.

Si el receptor del documento no es un contribuyente electrónico, es decir, receptor no electrónico o manual, entonces, el DTE debe ser impreso y entregado al cliente por mano. La impresión incluye, además de los datos representativos del documento, la representación del timbre en formato de código de barra 2D, PDF417.

⁷ https://palena.sii.cl/cgi_dte/UPL/DTEauth?2

En el caso de las Boletas, no hay envío al SII y tampoco intercambio, sólo se puede entregar una representación impresa o vía correo electrónico, si el receptor así lo autoriza, esto último se denomina envío por mandato.

Finalmente, el SII, en su normativa determina que el archivo XML tiene que ser almacenado por un período mínimo de 6 años, periodo durante cual debe quedar disponible ante posibles fiscalizaciones.

El SII, establece una serie de obligaciones adicionales al contribuyente que opera con documentos tributarios electrónicos, que se encuentran disponibles en la página web del SII (www.sii.cl), en particular en la Resolución exenta N°45 del 01/09/2003⁸, que establece las normas y procedimientos de operación de DTE, pero que no es necesario detallar ya que no forman parte y no afectan al foco de este trabajo.

⁸ <http://www.sii.cl/documentos/resoluciones/2003/reso45.htm>

4 Descripción de la solución

Solución planteada

La solución planteada consiste en separar las acciones de firma y timbre en la generación del documento, es decir, sólo se timbra en el dispositivo y se construye el XML del documento tributario en un servidor central, donde se firma y se vuelve a timbrar.

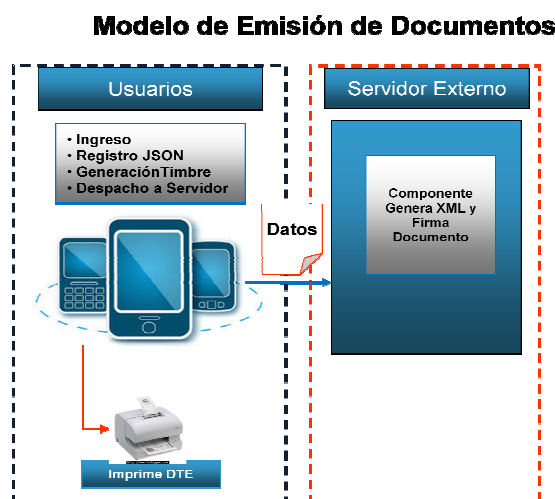


Ilustración 2: Modelo de emisión de documentos electrónicos

Es decir, se divide el problema de la siguiente manera:

4.1.1 Timbre en el dispositivo

Para imprimir el documento desde un celular o tablet, se necesita la representación en código de barra bidimensional PDF417 del TED, es decir, contar con la sección del <TED> del documento, para esto es necesario, primero, derribar el paradigma de construir el XML completo, con firma y timbre de una sola vez.

Dentro de las principales razones para revisar alternativas de dónde generar y firmar el XML del DTE, es evitar la complejidad de administrar la firma electrónica en varios aparatos distintos, dado que ésta tiene un período de vigencia de 1 a 3 años y, además, puede caducar o revocarse antes de tiempo, adicionalmente, se busca con esto minimizar el riesgo de mal uso del certificado de firma.

Por lo demás, el cambio de paradigma consiste en construir exclusivamente la sección del XML llamada TED a partir de los datos básicos que lo constituyen (ver 0). De esta manera se necesita sólo el CAF en el dispositivo para completar el TED y, posteriormente, es posible enviar los datos del documento en un formato más liviano como es JSON, vía HTTP POST hacia un servicio web tipo

REST⁹ (Transferencia de Estado Representacional, por sus siglas en inglés), a partir del cual se construye el XML.

Gracias a la propiedad *determinista* (es decir, dada una cadena de entrada siempre devuelve el mismo valor) de las funciones de hash (SHA1¹⁰) y de firma (RSA¹¹) utilizadas, es posible obtener el mismo valor de timbre a partir de datos idénticos. En consecuencia, a nivel del dispositivo ha sido posible crear un TED que después es generado idéntico en el servidor a partir de los mismos datos del documento ingresado desde el móvil.

Para esto se ha reutilizado la componente Java de Acepta, *CA4TED*, para generación y firma del TED:

```
String ted = CA4TED( tmstFirma, rutEmisor, tipoDTE, fechaEmision, rutReceptor, folio,
                    razonSocialReceptor.substring(1,40), montoTotal, detalle[1].substring(1,40));

// Esto genera una salida del tipo:

"<TEDversion="1.0"><DD><RE>11111111-1</RE><TD>39</TD><F>2141799</F>
<FE>2014-09-01</FE><RR>66666666</RR><RSR>Deprueba</RSR><MNT>60000
</MNT><IT1>PRODUCTO1</IT1><CAFversion="1.0"><DA><RE>11425975-6</RE>
<RS>Empresas CAAM Ltda</RS><TD>39</TD><RNG><D>2141715</D><H>2141814</H>
</RNG><FA>2014-07-09</FA><RSAPK><M>vUJ+ClIFeu2sASBrjRjqp+1/ ...1Nnl4CqAg
PJ2dj5az/a0zFqnrw==</M><E>Aw==</E></RSAPK><IDK>100</IDK></DA>
<FRMAalgoritmo="SHA1withRSA">TNUGITDmJ...AhMgPUjoH1Dg1g==</FRMA></CAF>
<TSTED>2014-08-01T15:44:42</TSTED></DD><FRMTalgoritmo="SHA1withRSA">
skPMxU8q6ONB+ZBg9571aRovvqoyV/9z4cp/6fB6+JdI15tJvYbLHM4AVh3w/FC+d+wgVrSR
9t0kXoWoanoF4A==</FRMT></TED>"
```

Tabla 7: Generación del TED

Luego, para la generación del PDF417, se ha implementado en Android la clase *CreaPDF417* que utiliza la biblioteca en Java de código abierto *zxing*¹² perteneciente al “Zxing Project” proyecto para el procesamiento de imágenes de códigos de barra 1D y 2D en multiplataforma y código abierto. De hecho el código es una adaptación de:

```
com.google.zxing.client.android.encode.QRCodeEncoder.encodeAsBitmap()
```

Para la clase *CreaPDF417*, se han desarrollado los métodos *escribePDF* y *encodeAsBitmap*. El primero utiliza al segundo para retornar una imagen en formato bitmap del código de barra PDF417. El timbre se genera a nivel del dispositivo y la imagen es utilizada por la interfaz para imprimirlo.

⁹ http://es.wikipedia.org/wiki/Representational_State_Transfer

¹⁰ http://es.wikipedia.org/wiki/Secure_Hash_Algorithm

¹¹ [http://en.wikipedia.org/wiki/RSA_\(cryptosystem\)](http://en.wikipedia.org/wiki/RSA_(cryptosystem))

¹² <https://github.com/zxing>

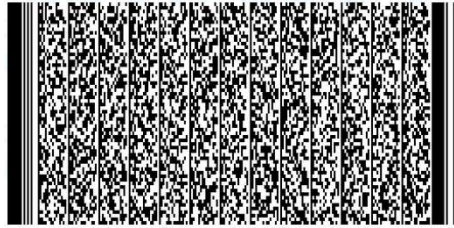


Ilustración 3: Timbre PDF417 del TED

A continuación un extracto del método *escribePDF*, encargado de generar la representación gráfica del timbre:

```
package cl.caam.app;

import java.io.IOException;
import java.util.HashMap;
import java.util.Map;

import android.graphics.Bitmap;

import com.acepta.com.google.zxing.BarcodeFormat;
import com.acepta.com.google.zxing.EncodeHintType;
import com.acepta.com.google.zxing.WriterException;
import com.acepta.com.google.zxing.common.BitMatrix;
import com.acepta.com.google.zxing.pdf417.PDF417Writer;

public class CreaPDF417 {

    public static Bitmap escribePDF(String data) {

        //Se define formato del código de barra: PDF_417 y se
        //asigna a la variable barcodeFormat
        BarcodeFormat barcodeFormat = BarcodeFormat.PDF_417;

        //Se crea objeto del tipo pdf417
        PDF417Writer escribepdf = new PDF417Writer();
        //....
        int width = 300;
        int height = 250;

        //Se crea el formato pdf417 para la "data" (dada como parámetro), utilizando el
        //método "encode" de la clase PDF417Writer.
        //Obteniendo una representación BitMatrix, o sea, una matriz de bits 2D,
        //de ancho width y alto height.
        BitMatrix bitMatrix = escribepdf.encode(data, barcodeFormat, width, height);

        //Devuelve bitmap del pdf417, generado en la línea anterior.
        return encodeAsBitmap(bitMatrix);
    }
}
```

Tabla 8: Extracto clase CreaPDF417 y el método escribePDF

Como se ve en la tabla anterior, el método *escribePDF*, recibe como parámetro el *String* "data" con el valor del TED, equivalente al que se presenta en la Tabla

7: Generación del TED y retorna un *bitmap*,¹³ del PDF417 correspondiente, es decir, la representación en una matriz de bits 2D, de valores blanco o negro ancho *width* y un alto *height*.

De esta manera, se cuenta con el timbre electrónico en el móvil, sin que se haya generado el documento y así la firma reside en un solo punto, en un servidor central donde se completa la operación.

De acuerdo a la normativa tributaria, se permite entregar el DTE al SII, hasta con 12 horas¹⁴ de diferencia, siempre que el documento se emita en una venta en ruta, es decir, en las instalaciones del receptor. Adicionalmente, las boletas no se envían directamente al SII y tampoco se intercambian electrónicamente, por lo tanto, este modelo de operación, con envío desfasado al SII, está dentro de la norma y sólo es necesario controlar que las NC y ND no sean emitidas cuando el móvil se encuentre sin conexión.

4.1.2 Servicio central

Como se mencionó, para completar la operación se cuenta con un servidor donde se levanta un servicio web o API REST, que recibe la información enviada por cada dispositivo. Este WS se ha construido en Python, utilizando el micro framework *Flask*¹⁵, por su portabilidad, flexibilidad y sencillez de programación. A continuación se presenta un extracto del código de la API:

```
#!/pyAV/bin/python
# -*- coding: utf-8 -*-

# Web API: servicio para atender solicitudes de creación de documentos, desde equipos móviles.

from flask import Flask, jsonify, abort, request, json
import urllib, urllib2

app = Flask(__name__)

#Método POST
@app.route('/api/v1.0/pi', methods = ['POST'])

def crea_documento():

    # Valida existencia de mensaje json

    if not request.json or not 'tipo' in request.json:
        abort(400)

    #Asocia datos recibidos en formato JSON
    documento = {'doc' : {
        'rutEmis': request.json['rutEmis'],
        'tipo': request.json['tipo'],
        'folio': request.json['folio'],
        ...
```

¹³ <http://en.wikipedia.org/wiki/Bitmap>

¹⁴ http://www.sii.cl/preguntas_frecuentes/catastro/001_012_1119.htm

¹⁵ <http://flask.pocoo.org/>


```

...
'timestamp': request.json['timestamp'] }
}

#Almacena archivo JSON como XML para entregarlo al servicio ca4xml_http.
nombre = request.json['rutEmis'] + '0' + request.json['tipo'] + '0' + request.json['folio']
doc_xml = json2xml(documento)
archivo = open('respaldo_moviles/' + nombre + '.txt', 'w')
archivo.write("<?xml version='1.0' encoding='ISO-8859-1' standalone='yes'?>\n")
archivo.write(doc_xml)
archivo.close()

#Genera DTE, utilizando el servicio ca4xml_http de Acepta
datos=urllib.urlencode({"docid":nombre, "comando":"envia", "parametros":"","datos":doc_xml})
web=urllib2.urlopen("http://127.0.0.1:5005/ca4xml", datos)
print web.read()

#Responde al cliente que lo invocó
return jsonify( { 'Estado': 'OK' } ), 200, {'Content-Type': 'text/css; charset=utf-8'}

if __name__ == '__main__':
    app.run(host='0.0.0.0')

```

Tabla 9: Extracto API RESTful en Python utilizando micro-framework Flask

En la operación, desde el móvil se realiza una llamada POST, con los datos del documento en formato JSON, al WS en el servidor. Luego, la aplicación en el servidor convierte los datos a un XML básico, equivalente a JSON, para enviarlo vía POST a un segundo servicio web llamado *ca4xml_http*. Este segundo WS es un producto de Acepta, que para este trabajo ha sido configurado de acuerdo a las especificaciones del mensaje de salida que genera el primer WS, dado que *ca4xml_http*, no lee JSON, pero sí XML.

Ambos WS residen en el mismo servidor, el primero Rest4Movil, atiende a los dispositivos móviles y el segundo, *Ca4xml_http*, se encarga de generar el DTE, es decir, crea el XML del DTE, lo firma y timbra. Como resultado devuelve la URL del documento, es decir, la dirección de almacenamiento del DTE en Acepta. El servicio *Ca4xml_http*, crea además un archivo del DTE y lo envía hacia Acepta, para su gestión. Ahí, se mantiene un respaldo en línea y se administran los documentos provenientes de cada móvil. De este modo se evita congestionar el punto de emisión con servicios adicionales orientados al cumplimiento de la normativa tributaria. En caso de no existir conectividad, desde este servidor, la información se encola hasta que se restablece la comunicación, momento en que se sincroniza todo lo pendiente.

Modelo de Emisión de Documentos Comunicación móvil - Servidor

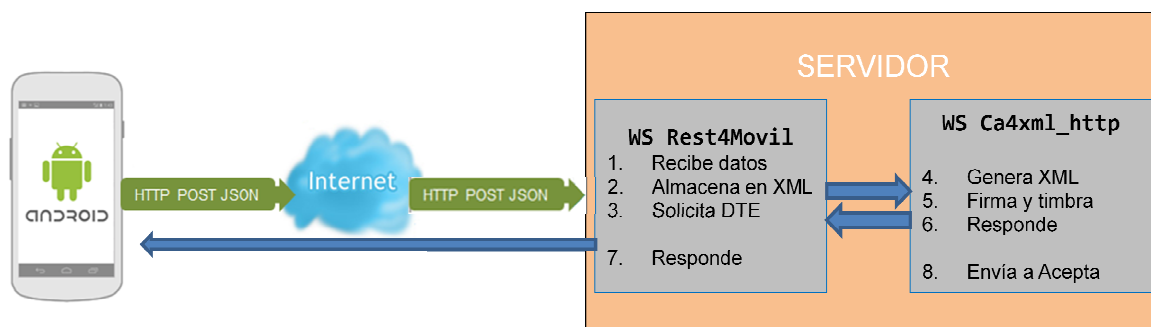


Ilustración 4: Comunicación móvil – servidor

Aplicación móvil e interfaz de usuario

La interfaz de usuario para el ingreso de datos en el dispositivo móvil ha sido externalizada y el núcleo de la solución se ha desarrollado considerando esta división, con el objetivo de presentar lo medular y centrarse en el foco del presente trabajo.

En este sentido se construyó una interfaz simple que permite mostrar y utilizar la funcionalidad del modelo propuesto, ya que éste será utilizado en una aplicación comercial y donde aplicativo del POS lo desarrolla una empresa asociada a Acepta, y debe entregar y controlar más información, como por ejemplo, sincronización con lista de productos, precios, clientes, control de stock, diagrama de ruta, registros de monitoreo del vehículo, entre otros temas que escapan al alcance definido inicialmente.

Además, la aplicación se ha acotado a los documentos más utilizados y factibles de emitir desde un dispositivo móvil, como son las boletas, guías de despacho, facturas, notas de crédito y notas de débito.

Dentro de los requerimientos hacia la interfaz de usuario es básico y necesario validar el contenido de los campos que se ingresan, ya que se trata de una parte sensible del modelo, ya que no hay otra instancia de corrección que permita asegurar la emisión de un documento válido.

Si bien, no es posible controlar el sentido de la información registrada por el usuario final, al menos sí se puede restringir o limitar lo que puede hacer. En este sentido, los datos deben ser chequeados en el origen y no en forma posterior. Los cálculos los realiza la aplicación. Es necesario validar cada campo ingresado, es decir, que la fecha de emisión corresponda a una fecha válida, por ejemplo en mi interface se limita a la fecha del día actual, el RUT del Receptor se aplica módulo 11 con el dígito verificador, las comunas se seleccionan de una lista de comunas, se restringe el tipo de dato que se ingresa en cada campo, así los campos de montos sólo ingresan números. Finalmente, como por cada documento se ha llevado a pantalla el mínimo de datos necesarios, entonces, todos los campos disponibles son obligatorios. Por ejemplo:



Ilustración 5: Campos principales

Internamente, en la aplicación, para registrar los datos ingresados por el usuario y de acuerdo al análisis del punto 0, se ha definido una *clase Java* llamada *Documento*, que contiene los datos esenciales requeridos para un DTE o boleta. Esta clase se utiliza como base para construir la estructura *Json* que luego viaja al servidor.

```

public class Documento {
    private String tipo;
    private String folio;
    private String fechaEmis;
    private String indServicio;
    private String rutEmis;
    private String razonSocEmis;
    private String giroEmis;
    private String acteco;
    private String dirEmis;
    private String comunaEmis;
    private String ciudadEmis;
    private String rutRecep;
    private String razonSocRecep;
    private String dirRecep;
    private String comunaRecep;
    private String ciudadRecep;
    private String montoNeto;
    private String montoExento;
    private String tasalva;
    private String montolva;
    private String montoTotal;
    private String detIndExe;
    private String detCodigoItem;
    private String detNombreItem;
    private String detMontolItem;
    private String refTipoDoc;
    private String refIndRefGlobal;
    private String refFolio;
    private String refFecha;
    private String refCodigo;
    private String refRazonRef;
    private String timeStamp;

    // Getter and Setter ...
}

```

Tabla 10: Clase Documento

Como se puede apreciar, todos los atributos de la *clase* son String, esto se hizo principalmente, para facilitar la conversión de una estructura a otra, es decir, de *Documento* a *JSON* y de éste a *XML*, ya que finalmente todo lo que viaja de un lado para otro es, básicamente, una cadena de caracteres.

A continuación se presenta el diagrama de flujo de la aplicación:

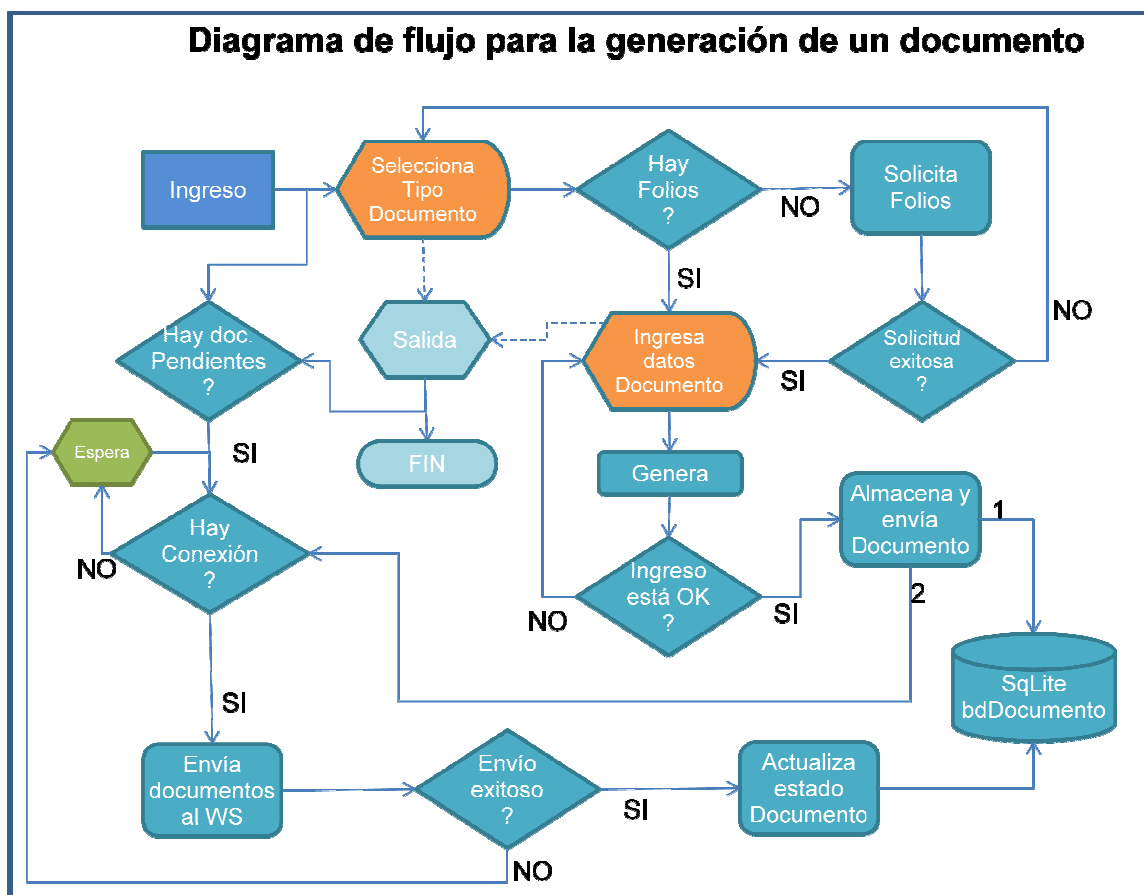


Ilustración 6: Diagrama de flujo definido para la aplicación

Para resguardar los documentos generados desde la aplicación, se utiliza el administrador de base de datos relacional *SQLite*¹⁶, incorporado de serie en Android y que permite llevar un registro y control del estado de publicación de cada uno de los documentos generados. El nombre de la base de datos es *dbDocumentos*.

De acuerdo al diagrama anterior, al iniciar la aplicación se consulta en forma asíncrona la base de datos, para verificar si hay documentos pendientes de envío. En caso de existir conexión, se envían los documentos que están en estado no enviado al servidor y se actualiza su estado en la BD local.

¹⁶ <http://www.sqlite.org/>



Ilustración 7: Primera pantalla, selecciona tipo de documento

Como la acción anterior se ejecuta en forma asíncrona, la aplicación ya se encuentra en la primera pantalla para que el usuario seleccione el tipo de documento a emitir, luego, se valida la existencia de folios disponibles para el tipo de documento seleccionado. En caso exitoso, se pasa a la pantalla de ingreso de datos, donde el usuario completa los campos para solicitar *Generar* el documento. Si no hay folios disponibles, entonces, es necesario solicitar nuevos folios, tema que se analizará en la siguiente sección 0.

El botón *Generar*, valida el correcto ingreso de datos, de acuerdo a lo expuesto anteriormente, completa los atributos de una instancia de la *clase Documento* y construye el objeto *jsonDocumento*.

Posteriormente, se genera un registro en la base de datos local *dbDocumentos*, con los datos [*índice, jsonDocumento.toString(), estado*], es decir, siempre se guarda una copia de los datos del documento y el estado de publicación, inicialmente nulo.

A continuación, se chequea si hay conexión a la red para decidir si se envían los datos del documento hacia el servidor central. Como se muestra en la siguiente ilustración la aplicación Android monitorea el estado de la conexión a la red y lo presenta al usuario. En caso de contar con acceso a la red, entonces, se envía el elemento *jsonDocumento.toString()* vía *HttpPost* al servidor central y se actualiza el registro en la *BD SQLite*.



Asignación de folios

Para poder emitir documentos tributarios electrónicos es necesario asignar un correlativo al documento, este folio debe estar acreditado a través del archivo de autorización de folios otorgado por el SII, para el contribuyente emisor. Es decir, se requiere de un archivo CAF. Además, cabe recordar que el CAF es utilizado para generar el timbre del documento.

La solución más simple, a primera vista, para disponer folios en un dispositivo móvil, consiste en descargar un archivo de CAF por cada equipo o punto de emisión de documentos y llevarlo “de alguna manera” al móvil. Sin embargo, esto pronto se transforma en una dinámica compleja, ya que los CAF entregados por el SII son limitados, se requiere al menos uno por tipo de documento y porque, además, no es sencillo controlar la renovación del CAF en cada equipo.

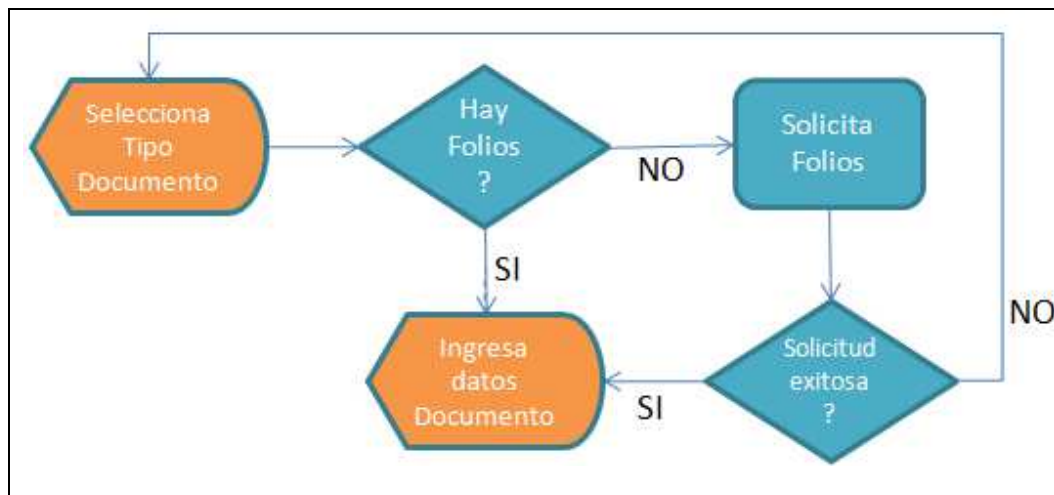


Ilustración 9: Asignación de folios

Según el diagrama anterior, la aplicación consulta si tiene folios disponibles y “Solicita Folios” en caso de ser necesario.

En rigor, el funcionamiento de la solicitud de folios se realiza de acuerdo a un umbral de consumo, esto es, si se ha llegado al umbral disponible para un tipo de documento, entonces, solicita folios para ese tipo de documento.

¿A quién se le solicita folios? Se trata de un servicio de Acepta, que se configura en forma centralizada para descargar CAF del SII y entregar una copia del CAF al punto de emisión asignándole un sub-rango del CAF original. Es decir, se divide el CAF en rangos de menor tamaño (rangos lógicos), para repartir cada sub-rango en los diferentes puntos de emisión asociados el mismo RUT emisor. Por ejemplo, si el archivo de CAF tiene un rango inicial de un millón de folios, entonces, se puede dividir en

rangos más pequeños de mil o 10 mil folios, por ejemplo. En la configuración del servicio, se inscribe el RUT del emisor, el dispositivo con un ID o identificador único en este caso ANDROID_ID y se registran los tipos de documentos que puede emitir y los umbrales para cada uno. Se entiende por umbral, la cantidad remanente a partir de la cual se debe volver a solicitar folios.

Formato de solicitud:

```
<pos:comunicacionCafadminPOScaf>
<rutEmpresa?></rutEmpresa>
<tipoDte?></tipoDte>
<codigoPos?></codigoPos>
<idTransaccion?></idTransaccion>
</pos:comunicacionCafadminPOScaf>
```

Tabla 11: Solicitud de folios

El servicio de asignación de folios devuelve: un código de respuesta, folio inicial, folio final, umbral y el CAF.

El formato de respuesta, está de acuerdo a la siguiente tabla:

```
<ns2:comunicacionCafadminPOScafResponse
xmlns:ns2="http://pos.server.ws.cafadmin.acepta.com/">
<return>
  <![CDATA[
    <respuestaPOS>
      <codigoRespuesta?></codigoRespuesta>
      <folioInicial?></folioInicial>
      <folioFinal?></folioFinal>
      <umbral?></umbral>
      <documentoRespuesta?></documentoRespuesta>
    </respuestaPOS>
  ]]>
</return>
</ns2:comunicacionCafadminPOScafResponse>
```

Tabla 12: Respuesta a solicitud de folios

El código de respuesta, puede ser éxito o error y hay valores dependiente el tipo de error, adicionalmente, el umbral, se refiere a la cantidad remanente de folios que se debe considerar para gatillar una nueva solicitud.

De esta manera, la aplicación en el dispositivo maneja los nuevos valores de rangos de folios, asigna folios a los nuevos documentos y controla stock de acuerdo al umbral.

Generación del DTE

Los datos ingresados en el dispositivo móvil, no tienen valor sino hasta que se crea el DTE correspondiente. Para esto, se envían a un servidor centralizado donde interactúan 2 servicios, tal como se explicó en el punto 4.1.2.

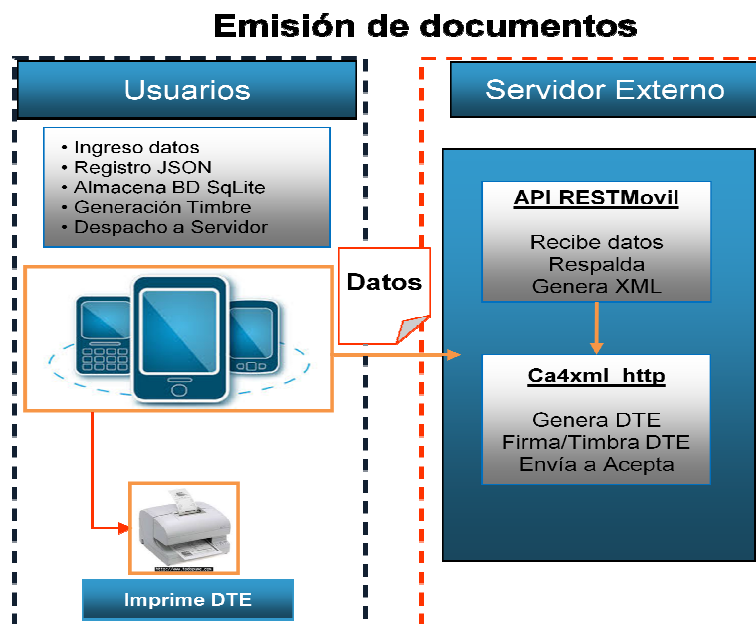


Ilustración 10: Generación del DTE

La API RESTMóvil del servidor recibe los datos desde el móvil, respalda y genera un XML intermedio, homólogo al formato JSON proveniente del equipo móvil, es decir, se estructura un formato con un único nodo padre y varios nodos hijo al mismo nivel. Luego lo envía al servicio de Acepta CA4XML_Http, que se encuentra en el mismo servidor.

La configuración del servicio CA4XML_Http, para leer el XML anterior, se realiza mediante archivos XSLT¹⁷ (Extensible Stylesheet Language Transformation), que pertenece al estándar de la organización W3C¹⁸, y se utiliza para transformar documentos XML en otro formato de salida.

Por ejemplo, el formato de archivo que se emite desde el primer WS es:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<doc><indServicio></indServicio><montoExento>0</montoExento><ciudadEmis>Santiago</ciudadEmis>
<refTipoDoc></refTipoDoc><dirRecep>Alto del Sol 896</dirRecep><montoTotal> 1190000
</montoTotal><giroEmis>Identidad Digital</giroEmis><fechaEmis>2014-09-05</fechaEmis>
<montolva>190000</montolva><folio>125</folio> <tipo>33</tipo><montoNeto>1000000
</montoNeto><rutRecep>11425975-6</rutRecep><detNombreItem>Producto de prueba|Cualquier cosa
mas|</detNombreItem><tasalva> 19</tasalva><refFecha></retFecha><detCodigoItem>
</detCodigoItem><refIndRefGlobal></refIndRefGlobal><dirEmis>Providencia 989 </dirEmis>
<detMontolItem>1000000|1000|</detMontolItem><refFolio> </refFolio><razonSocRecep>Cristian
```

¹⁷ http://es.wikipedia.org/wiki/Extensible_Stylesheet_Language_Transformations

¹⁸ http://es.wikipedia.org/wiki/World_Wide_Web_Consortium


```

Acuna</razonSocRecep><timestamp>2014-09-05T00:14:00 </timestamp><refCodigo></refCodigo>
<rutEmis>96919050-8</rutEmis><razonSocEmis>Acepta </razonSocEmis><comunaEmis>Providencia
</comunaEmis><acteco>123456</acteco><comunaRecep> Talagante</comunaRecep><ciudadRecep>
Santiago</ciudadRecep><detIndExe></detIndExe></doc>

```

Tabla 13: XML generado desde la API RESTMovil

Luego, es transformado a través de un archivo XSLT, que interpreta cada tag del XML y lo utiliza para definir un nuevo formato de salida, que en este caso, corresponde al esquema DTE definido por el SII.

Por ejemplo, veamos una definición de variables en XSLT, de acuerdo a la entrada anterior:

```

<xsl:variable name="RUTEmisor" select="ca4:rut(/doc/rutEmis,'RUT Emisor')"/>
<xsl:variable name="TipoDTE" select="/doc/tipo"/>
<xsl:variable name="Folio" select="/doc/folio"/>
<xsl:variable name="FechaEmision" select="/doc/fechaEmis"/>
<xsl:variable name="RazonSocialEmisor" select="substring(/doc/razonSocEmis, 1, 100)"/>
<xsl:variable name="RUTReceptor" select="ca4:rut(/doc/rutRecep,'RUT Receptor')/

```

Tabla 14: XSLT: Ejemplo definición de variables

Se muestra a continuación, un extracto de la sección del XSLT para generar el DTE:

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0"
<xsl:variable name="DTE">
<DTE xmlns="http://www.sii.cl/SiiDte" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="1.0">
  <Documento ID="{${idDTE}" xmlns="http://www.sii.cl/SiiDte"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <!-- ===== Encabezado ===== -->
    <Encabezado>
      <IdDoc>
        <TipoDTE><xsl:copy-of select="$TipoDTE"/></TipoDTE>
        <Folio><xsl:copy-of select="$Folio"/></Folio>
        <FchEmis><xsl:copy-of select="$FechaEmision"/></FchEmis>
      </IdDoc>
      <Emisor>
        <RUTEmisor><xsl:copy-of select="$RUTEmisor"/></RUTEmisor>
        <RznSoc><xsl:copy-of select="$RazonSocialEmisor"/></RznSoc>
        <GiroEmis><xsl:copy-of select="substring(/doc/giroEmis, 1, 80)"/></GiroEmis>
        <Acteco><xsl:copy-of select="/doc/acteco"/></Acteco>
        <DirOrigen><xsl:copy-of select="substring(/doc/dirEmis, 1, 60)"/></DirOrigen>
        <CmnaOrigen><xsl:copy-of select="substring(/doc/comunaEmis,1, 20)"/></CmnaOrigen>
        <CiudadOrigen><xsl:copy-of select="substring(/doc/ciudadEmis, 1, 20)"/></CiudadOrigen>
      </Emisor>
      <Receptor>
        <RUTRecep><xsl:copy-of select="$RUTReceptor"/></RUTRecep>
        <RznSocRecep><xsl:copy-of select="$RazonSocialReceptor"/></RznSocRecep>
        <GiroRecep><xsl:copy-of select="substring(/doc/giroRecep, 1,40)"/></GiroRecep>
        <DirRecep><xsl:copy-of select="substring(/doc/dirRecep, 1, 60)"/></DirRecep>
        <CmnaRecep><xsl:copy-of select="substring(/doc/comunaRecep,1, 20)"/></CmnaRecep>
        <xsl:copy-of select="dte:optional-tag('CiudadRecep',/doc/ciudadRecep)"/>
      </Receptor>
      ....
    </DTE>
  </xsl:variable>
  <xsl:template match="/">
    <xsl:copy-of select="$DTE"/>
  </xsl:template>
</xsl:stylesheet>

```

Tabla 15: XSLT: Extracto de generación del DTE

De esta manera se configura CA4XML_Http, para leer el formato definido proveniente de los móviles. Adicionalmente, para que el servicio opere es necesario copiar el certificado de firma de la persona autorizada a firmar los documentos, más los archivos de CAF (físicos) para la generación del timbre.

Gestión de los documentos: Integración con Acepta

La gestión de los documentos, como se explicó anteriormente, consiste en cumplir con la normativa definida por el SII, para los DTE.

En particular el envío del DTE al SII, luego al receptor electrónico y garantizar la custodia de los XML por un período mínimo de 6 años. El contribuyente debe contar con una plataforma que le permita recibir y custodiar documentos de proveedores, las respuestas del SII, de los clientes y asociarlas al DTE que les dio origen. No es mi intención detallar en qué consisten estos requerimientos y tampoco seguir enumerando las demás obligaciones que se tienen como contribuyente electrónico.

El objetivo de este punto es presentar el modelo final, haciendo hincapié en que a través de la integración con el proveedor Acepta, se logra dar cumplimiento a los requisitos del SII y a la vez, se obtiene un modelo de emisión de documentos en dispositivos móviles con autonomía operacional.

En resumen, el usuario desde su smartphone o tablet, genera un documento que llega en formato JSON al Servidor Externo donde hay dos servicios que se comunican para generar un DTE válido, el segundo servicio se encarga de enviar el DTE generado a Acepta para completar su gestión ante el SII, de acuerdo al siguiente modelo:

Modelo final de emisión de documentos

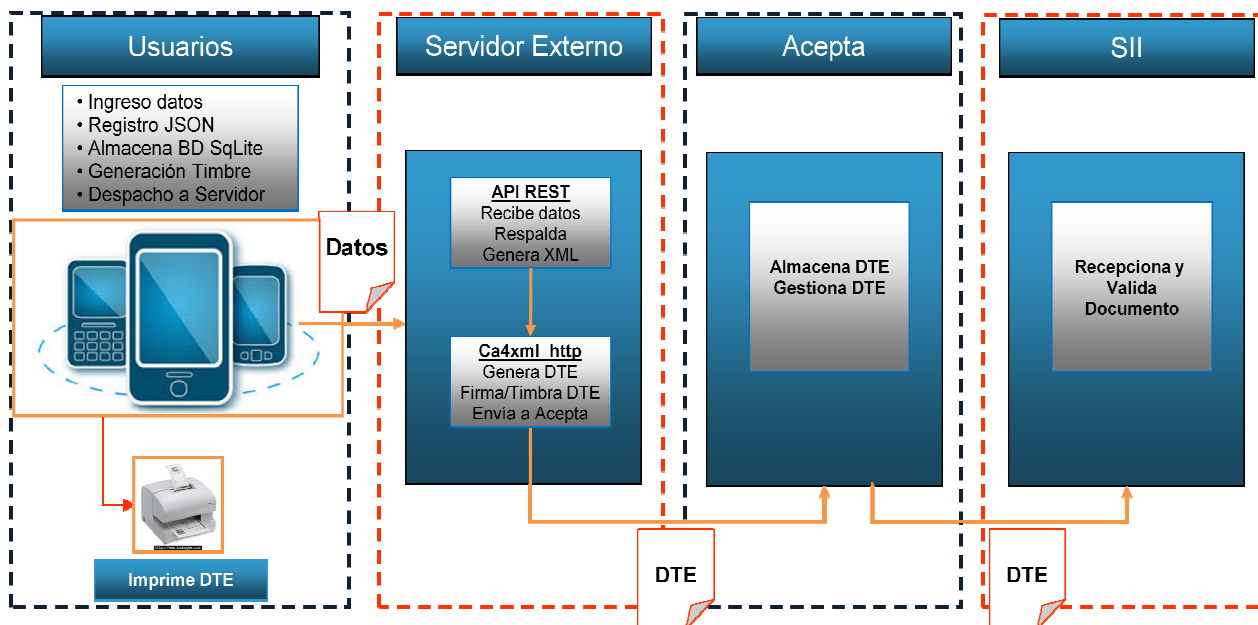


Ilustración 11: Modelo final de emisión de documentos electrónicos

5 Resultados

Análisis de envío de documentos hacia la API REST

Se considera importante validar el tiempo de respuesta del modelo. Para ello se ha realizado un envío masivo de documentos desde la red, midiendo el tiempo de respuesta sólo de la API RESTMóvil, el servicio 1 (S1), es decir, sin la generación del DTE, el servicio 2 (S2), para compararlo luego, con el tiempo de respuesta de ambos servicios operando juntos.

Es decir, S1, corresponde al WS que escucha las peticiones de los dispositivos móviles y, S2, el WS que construye los DTE's y los envía hacia Acepta.

Para tener una referencia de tiempo mínimo de los servicios web, primero se realizó el ejercicio enviando hacia el servicio S1, a través del comando *curl*¹⁹, desde el mismo servidor de aplicaciones, paquetes de 10, 50, 100, 200 y 500 documentos, con el siguiente resultado:

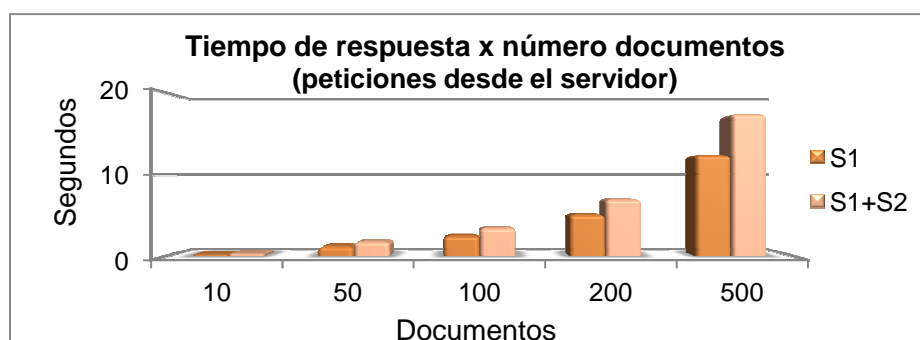


Ilustración 12: Gráfico, tiempo de respuesta WS, peticiones internas

Para este caso, el tiempo promedio de procesamiento es de 39,89 doc/s cuando atiende sólo el S1 y de 28,35 doc/s cuando atienden S1 y S2 juntos.

Cuando se realiza la petición desde el móvil, es decir, desde la red, los tiempos de respuesta aumentan drásticamente, según lo refleja el siguiente gráfico:

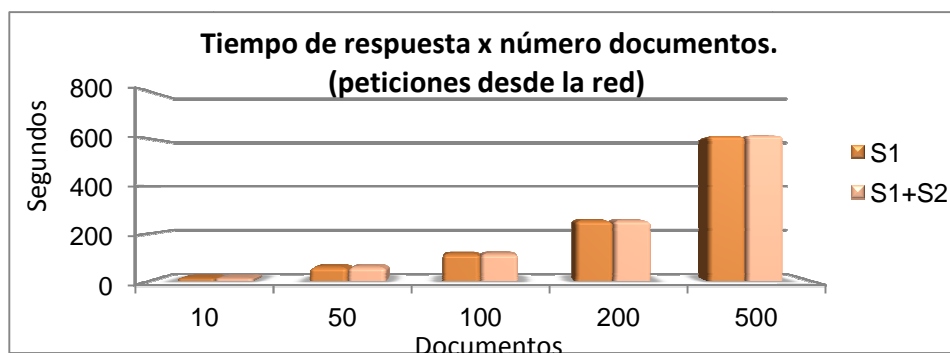


Ilustración 13: Gráfico, tiempo de WS, peticiones desde la red

¹⁹ <http://es.wikipedia.org/wiki/CURL>

Ahora se observan 2 resultados interesantes, primero, que aumenta considerablemente el tiempo de respuesta, en este caso el promedio para S1 es de 0,795 doc/s, y segundo, casi no hay diferencia de tiempo cuando atienden ambos servicios en conjunto, ya que el promedio para S1 + S2, es de 0,790 doc/s. En la práctica, los tiempos para establecer la conexión HTTP, generar la petición, recibir la respuesta (request/response) y luego cerrar la comunicación, superan con creces el tiempo de proceso de los servicios.

Los datos correspondientes a los gráficos anteriores se obtienen de la siguiente tabla:

Tiempo de respuesta de los servicios				
Documentos	S1		S1 + S2	
	Servidor	Red	Servidor	Red
10	0m0.25s	0m 13.98s	0m0.35s	0m 14.17s
50	0m1.24s	1m 0.65s	0m1.82s	1m 1.13s
100	0m2.51s	1m 56.05s	0m3.50s	1m 56.59s
200	0m 5.13s	4m 16.59	0m 7.02s	4m 17.35s
500	0m12.39s	10m 14.65	0m17.43	10m 16.65s

Tabla 16: Tiempo de respuesta de los servicios web

Se desprende, entonces, que el tiempo de respuesta promedio por transacción desde la red es de 1,25 segundos, lo cual, en primera instancia no se ve necesario optimizar.

Validación del timbre

Para validar el timbre, se procede al siguiente tratamiento:

1. Se copia un archivo de prueba en el móvil, que contiene registros en formato de documentos JSON, equivalente al que genera el código local de la aplicación.
2. Se modifica para que la aplicación lea desde el archivo los documentos. Se aplica CA4TED (ver 14 14.1.Timbre en el dispositivo) a cada uno.
3. La salida del CA4TED, se almacena en archivo con nombre *TipoDoc-Folio-android*.
4. Se envía JSON al servicio API RESTMovil.
5. Se rescatan los archivos TED del móvil y se llevan al servidor para comparar la firma FRMT con su correspondiente en los DTE generados, obteniendo secuencias equivalentes en cada caso, a como se muestra en la siguiente tabla:

```

39-136-android:
<FRMT algoritmo="SHA1 withRSA">e1Rw9WHjUHWExCUBFAVT5aomO+y5kl0qqIUxVG1Dol+dr13h9xfYx2H9i8iBcycWd23wdz/Y3BG6gFRDg8RA==</FRMT>
39-136.xml:
<FRMT algoritmo="SHA1 withRSA">e1Rw9WHjUHWExCUBFAVT5aomO+y5kl0qqIUxVG1Dol+dr13h9xfYx2H9i8iBcycWd23wdz/Y3BG6gFRDg8RA==</FRMT>

39-137-android:
<FRMT algoritmo="SHA1 withRSA">WlX6nVgqoqTI3RrLs+TI6QOLoxbFO1fT08O6jQtqdNDVN47ifNJgzhKNPQ++HnQOtaSYykf+AWV3efybkNcA==</FRMT>
39-137.xml:
<FRMT algoritmo="SHA1 withRSA">WlX6nVgqoqTI3RrLs+TI6QOLoxbFO1fT08O6jQtqdNDVN47ifNJgzhKNPQ++HnQOtaSYykf+AWV3efybkNcA==</FRMT>

39-138-android:
<FRMT algoritmo="SHA1 withRSA">CvYHmre3KmOGVuuvgFxFxXg3kzEqxFgvIW3a5tUjQeDqJY11r7Pro47o9AxdAhF2y7f9pvihT6Qy9Pg7Aj0dYEhQ==</FRMT>
39-138.xml:
<FRMT algoritmo="SHA1 withRSA">CvYHmre3KmOGVuuvgFxFxXg3kzEqxFgvIW3a5tUjQeDqJY11r7Pro47o9AxdAhF2y7f9pvihT6Qy9Pg7Aj0dYEhQ==</FRMT>

39-139-android:
<FRMT
algoritmo="SHA1 withRSA">I+6xjaiBTXOWGT050OxyZcpTp8+PoF62YkOFcVQbJEKk0ND2VQMRmAKCVj6kfUr+gk46CPrLS6a1pWJEGPJi0w==</FRMT>
39-139.xml:
<FRMT

```

```

algoritmo="SHA1withRSA">I+6xjaiBTXOWGT050OxyZcpTp8+PoF62YkOfcVQbJEKkOND2VQMRmAkCVj6kfUr+gk46CPrLS6a1pWJEGPJi0w==</FRMT>
39-140-android:
<FRMT
algoritmo="SHA1withRSA">WZwadc8Wf9JjV1a9usMiLNwp00MX3ptXX7EF1WeJAI5C2HMBKpVh3HqZfIMVUoC63JfSBGolg8bGNwH4mPJVxg==</FRMT>
39-140.xml:
<FRMT
algoritmo="SHA1withRSA">WZwadc8Wf9JjV1a9usMiLNwp00MX3ptXX7EF1WeJAI5C2HMBKpVh3HqZfIMVUoC63JfSBGolg8bGNwH4mPJVxg==</FRMT>
33-101-android:
<FRMT algoritmo="SHA1withRSA">IkKnZgLSEm4kkaHb4FzK48jGC16cH89Kn9Rnc+IB4/raxtxb32PyR8S36IUnsJ59ARApLiQZD+vbUWTJiw9ASw==</FRMT>
33-101.xml:
<FRMT algoritmo="SHA1withRSA">IkKnZgLSEm4kkaHb4FzK48jGC16cH89Kn9Rnc+IB4/raxtxb32PyR8S36IUnsJ59ARApLiQZD+vbUWTJiw9ASw==</FRMT>
33-105-android:
<FRMT algoritmo="SHA1withRSA">OaT0ldO3RhIzLi5/LDlZLPkJJCXITFN2IkQT9LEAKOowGk5KSru1SQYf5Kfin74CwNmu854TfO4uvDA7J6rDA==</FRMT>
33-105.xml:
<FRMT algoritmo="SHA1withRSA">OaT0ldO3RhIzLi5/LDlZLPkJJCXITFN2IkQT9LEAKOowGk5KSru1SQYf5Kfin74CwNmu854TfO4uvDA7J6rDA==</FRMT>
33-111-android:
<FRMT
algoritmo="SHA1withRSA">mWILy1fi6N4jVGQzYmu3krZd/uhwF+6/15qnXeB8PY+83XqMeCSWSKQ81IYM67g/XgTw7pYL2yyRb3wVQ94j+Q==</FRMT>
33-111.xml:
<FRMT
algoritmo="SHA1withRSA">mWILy1fi6N4jVGQzYmu3krZd/uhwF+6/15qnXeB8PY+83XqMeCSWSKQ81IYM67g/XgTw7pYL2yyRb3wVQ94j+Q==</FRMT>
33-115-android:
<FRMT
algoritmo="SHA1withRSA">gnyDnisBdV0ccBKoLBSXcNzG8JoQxYfBG+ApbDy1YP766FUQ8NjJuoWFaQaOeWCOjeWdiO/iIPxyU2ZguCdiQ==</FRMT>
33-115.xml:
<FRMT
algoritmo="SHA1withRSA">gnyDnisBdV0ccBKoLBSXcNzG8JoQxYfBG+ApbDy1YP766FUQ8NjJuoWFaQaOeWCOjeWdiO/iIPxyU2ZguCdiQ==</FRMT>
33-116-android:
<FRMT algoritmo="SHA1withRSA">B9unthCS2gpHj8pXtynd4P2ucSart2rW1urg81GIC83s11EYBC0XEhQ7LsTNJKSNtne6c5i3gWbFj9F+hjPYsVQ==</FRMT>
33-116.xml:
<FRMT algoritmo="SHA1withRSA">B9unthCS2gpHj8pXtynd4P2ucSart2rW1urg81GIC83s11EYBC0XEhQ7LsTNJKSNtne6c5i3gWbFj9F+hjPYsVQ==</FRMT>

```

Tabla 17: Extracto: Comparación timbre Android vs timbre en el servidor

Dado que la sección del timbre, FRMT, corresponde al resultado del algoritmo de firma RSA aplicado al SHA1 del contenido en la sección DD del TED, al comparar ambos resultados se puede inferir que el resto del TED es igual para los timbres generados desde la aplicación Android y la aplicación CA4xml_http, que construye el DTE final.

El resultado es equivalente en todos los casos, para distintos tipos de documentos (por ejemplo, 33 y 39 en la tabla anterior), cumpliendo con lo esperado para la generación del timbre electrónico.

Generación de documentos

La aplicación CA4xml_http (S2), de Acepta, a través de las configuraciones realizadas para que pueda interpretar la data proveniente de la APIRestMovil (S1), para generar correctamente el documento tributario electrónico, se encarga de realizar el envío al backoffice de Acepta y devuelve en el proceso la URL de almacenamiento del documento. Por ejemplo:

<http://pruebasacepta1409.acepta.com/v01/2BCD42E41CEEB4123DEDC04B9EE617BDAB46520C?k=5f7091a5abb0874df3e8cb4543a5eb93>

Esta dirección se forma a partir de una función de hash del contenido del documento, por lo tanto es única y no es posible, a partir de ella, inferir otra URL válida.

La respuesta de S2 devuelve la URL a S1, la que se reenvía al dispositivo para completar el estado del documento en el móvil.

Esta URL permite acceder al documento en custodia, para su visualización, reenvío y/o impresión por parte del receptor del documento.


La visualización del documento se logra a través de plantillas en XSL-FO²⁰ (eXtensibleStylesheet Language Formatting Objects), utilizadas para generar salida en formato PDF. En este caso, se usa una plantilla estándar para lograr la siguiente representación de documentos generados desde el móvil:

Boleta:

Empresa de Prueba Identidad Digital Providencia 760 piso 8 - Providencia		R.U.T.: 96.919.050-8 BOLETA ELECTRÓNICA N° 140	
		S.J.I. - SANTIAGO CENTRO	
Señor (es) :	Cristián Acuña M.	Comuna :	Talagante
Dirección :	Alto del Sol 896	Ciudad :	Santiago
R.U.T. :	11.425.975-6	Forma De Pago :	
Ciudad :		Fecha Emis. :	05 SEPTIEMBRE 2014

N°	Cantidad	Unidad	Código	Descripción	Precio	Desc/Recargo	Total
1				Fotocopias			5.350

SON:

		Montos Totales	
Neto	\$	4.500	
I.V.A. 19%	\$	855	
Total	\$	5.355	

Tercera Electrónica SII
Res. 0 del 2012
Verifique documento: www.acpta.com


Ilustración 14: Boleta electrónica generada desde el móvil

Factura:

Empresa de Prueba Identidad Digital Providencia 760 piso 8 - Providencia		R.U.T.: 96.919.050-8 FACTURA ELECTRÓNICA N° 101	
		S.J.I. - SANTIAGO CENTRO	
Señor (es) :	Cristián Acuña M.	Comuna :	Talagante
Dirección :	Alto del Sol 896	Ciudad :	Santiago
R.U.T. :	11.425.975-6	Forma De Pago :	
Ciudad :		Fecha Emis. :	05 SEPTIEMBRE 2014

N°	Cantidad	Unidad	Código	Descripción	Precio	Desc/Recargo	Total
1				Producto de prueba			1.000.000
2				Texto informativo			0

SON:

		Montos Totales	
Neto	\$	1.000.000	
I.V.A. 19%	\$	190.000	
Total	\$	1.190.000	

Tercera Electrónica SII
Res. 0 del 2012
Verifique documento: www.iiif.cl

Ilustración 15: Factura electrónica generada desde el móvil

²⁰ <http://www.w3schools.com/xslfo/>

6 Conclusiones

Se cumple objetivo planteado inicialmente, esto es, confirmar que es posible operar con documentos tributarios electrónicos desde dispositivos móviles masivos, bajo un modelo que permite la generación del timbre electrónico en el móvil, mientras se envían los mismos datos a un servidor central que construye el XML final del documento tributario y lo despacha a un servicio encargado de la gestión conforme a la normativa del SII.

Se demuestra que el modelo propuesto es factible de implementar, con una solución básica que incluye los siguientes aspectos:

- a. Selección de datos mínimos para construir un DTE, lo que permite una interfaz de usuario liviana y sencilla para el registro de los datos. Una solución masiva no requiere operar con tanta información ya que dificulta el uso de la UI y aumenta la posibilidad de errores.
- b. Se registra y transfiere la información en formato de caracteres, no sólo los nombres o direcciones, sino también fechas y montos, utilizando estructuras como Json y XML intermedio, homólogo a Json. Esto facilita pasar de un servicio a otro y, además, se evitan conversiones y errores asociados.
- c. Solicitud de folios autorizados a través de un servicio centralizado de Acepta. Servicio que permite enrolar el dispositivo y asignarle rangos de folios, otorgando seguridad y autonomía.
- d. Generación del timbre electrónico del documento en el dispositivo. Como la generación del timbre electrónico es idéntica o 100% repetible a partir de los mismos datos que le dieron origen y esto unido a la disponibilidad de folios, en conjunto le otorgan autonomía operacional al punto de emisión.
- e. Almacenamiento en SQLite de los documentos pendientes de publicación, otorgando persistencia a la información ingresada por el usuario y de esta manera poder minimizar el riesgo de pérdida de documentos.
- f. Uso de protocolo HTTP Post desde el dispositivo hacia un servidor central con mensaje en Json liviano, con tiempos de respuesta razonables, cercano a 1 segundo por documento y que no afectan la operación, ya que ocurren en modo asíncrono, es decir, no bloqueante de la UI para la experiencia del usuario.
- g. Al tratarse de un protocolo estándar, HTTP Post, se facilita la masificación de la solución, ya que se puede implementar para cualquier SO móvil, ya sea iOS, Android, Windows Phone u otro.

- h. Uso de HTTP Post desde el servidor central a otro servicio que opera en el mismo servidor para la generación del documento tributario. Esto permite contar con tiempos de respuesta inmediatos para la emisión y entregar esta respuesta en línea al móvil. De esta manera, el móvil sabe si debe volver a intentar la publicación de su documento o éste ya ha enviado en forma exitosa.
- i. Configuración y adaptación de aplicación de Acepta, que recibe XML de entrada, genera DTE de salida y lo envía al backoffice de gestión. Parte de la clave de la solución se encuentra en la flexibilidad que tiene este aplicativo, desde el punto de vista de la configuración de su *input*, ya que se pudo leer un XML que ha sido estructurado como un nodo padre con varios elementos, todos al mismo nivel, e interpretarlo en forma exitosa.

Todo lo anterior sirve como punto de partida para implementar una solución comercial, que está respaldada por este modelo y que se puede complementar con un gran potencial de funcionalidades que van desde la preventa, geolocalización de clientes, chequeo de visitas, toma de pedido, etc., todo para facilitar la gestión de la venta en ruta o, simplemente, contar con una solución robusta para la emisión de documentos tributarios electrónicos desde dispositivos móviles masivos.

¿Qué pasa si crece el número de clientes del servicio? Este modelo es fácilmente escalable, es decir, se puede adaptar o mejorar su configuración para atender un mayor número de requerimientos, ya que puede aumentar su capacidad rápidamente, es decir, puede aumentar el número de clientes que lo consumen y la capacidad de atenderlos, agregando servicios y/o servidores que atienden a los distintos dispositivos.

En este sentido, como se trata de servicios muy livianos no se requiere gran infraestructura para soportarlos, ya sea máquinas físicas o a través de máquinas virtuales, lo que sí es necesario en este caso, es poder asegurar un uso equitativo de los recursos o de cada servicio y esto se logra, sencillamente, a través de un proxy balanceador de carga, como *Nginx*²¹, por ejemplo.

²¹ <http://wiki.nginx.org/NginxEs>

7 Glosario

- a. **API:** Interfaz de programación de aplicaciones (IPA) o API (del inglés Application Programming Interface)
- b. **Bitmap:** mapa de bit o arreglo de bit.
- c. **CAF:** Código de autorización de folios, elemento computacional que es entregado por el Servicio de Impuestos Internos vía Internet, al momento de autorizar un determinado rango de folios a un emisor electrónico y que es incorporado en el Timbre Electrónico.
- d. **DTE:** Documento Tributario Electrónico. Archivo XML y firmado electrónicamente por un emisor electrónico, que produce efectos tributarios y cuyo formato está establecido por el SII.
- e. **Emisor Electrónico:** Contribuyente autorizado por el Servicio de Impuestos Internos para generar documentos tributarios electrónicos.
- f. **Firma Electrónica:** Sustituto digital de la firma ológrafa que está constituido por un conjunto de caracteres que acompaña a un documento electrónico, que se origina a partir del documento y que permite verificar con certeza la identidad del emisor, mantener la integridad del documento electrónico e impedir al emisor electrónico desconocer su autoría o repudiarlo.
- g. **Nginx:** servidor HTTP y proxy inverso de alto rendimiento, y un servidor proxy para IMAP/POP3/SMTP.
- h. **Receptor Electrónico:** Todo emisor electrónico que recibe un documento tributario electrónico.
- i. **Receptor No Electrónico:** Todo receptor de un documento tributario electrónico que no es emisor electrónico; también se denomina "receptor manual".
- j. **REST:** Transferencia de Estado Representacional (Representational State Transfer) es una técnica de arquitectura software para sistemas hipermedia distribuidos como la World Wide Web.
- k. **Signatario o Firmante:** Persona que actúa en nombre propio o en el de otra persona natural o jurídica a la que representa, y que habiendo obtenido previamente un certificado digital para uso tributario, tiene la capacidad de firmar un documento digital.
- l. **SQLite:** es un sistema de gestión de bases de datos relacional, contenida en una pequeña biblioteca escrita en C.
- m. **TED:** Timbre Electrónico de un Documento Tributario Electrónico. Conjunto de caracteres que cumple la función de validar la representación impresa de un documento tributario electrónico y que permite verificar:
 - i. que el número con que se generó el documento tributario electrónico ha sido autorizado por el Servicio de Impuestos Internos;
 - ii. que los datos principales del documento tributario electrónico no han sido alterados;
 - iii. que el documento tributario electrónico fue efectivamente generado por el emisor electrónico correspondiente.
- n. **XML:** eXtensible Markup Language ('lenguaje de marcas extensible'), es un lenguaje de marcas desarrollado por el World Wide Web, para documentos electrónicos.
- o. **XSL:** lenguaje extensible de hojas de estilo (Extensible Stylesheet Language).

- p. **XSL-FO:** eXtensibleStylesheet Language Formatting Objects, documento XML en el que se especifica cómo se van a formatear unos datos para presentarlos en pantalla, papel u otros medios. XSL de Objetos de formato.
- q. **XSLT:** Transformaciones XSL (Extensible Stylesheet Language Transformations) es un estándar de la organización W3C que presenta una forma de transformar documentos XML.

8 Bibliografía

Android | Send “POST” JSON Data to Server [En línea] / aut. HMKCode. - <http://hmkcode.com/android-send-json-data-to-server/>.

Android SQLite [En línea] / aut. Vogel Lars. - <http://www.vogella.com/tutorials/AndroidSQLite/article.html>.

API android.database.sqlite [En línea] / aut. Android . - <http://developer.android.com/reference/android/database/sqlite/package-summary.html>.

Aplicaciones para dispositivos móviles = Aplicaciones para dispositivos móviles / aut. González Felipe Martínez. - Valencia : [s.n.], 2011. - <http://riunet.upv.es/bitstream/handle/10251/11538/Memoria.pdf?sequence=1>.

Desde la WEB a los móviles [Libro] / aut. Cádiz R. Alfredo. - Santiago : [s.n.], 2014.

Designing a RESTful API with Python and Flask [En línea] / aut. Grinberg Miguel. - <http://blog.miguelgrinberg.com/post/designing-a-restful-api-with-python-and-flask>.

Developer Android [En línea] / aut. Android. - <http://developer.android.com/index.html>.

Developer Android Training [En línea] / aut. Android. - <http://developer.android.com/training/index.html>.

Editor de documentos XML, usando plantillas y transformaciones [Libro] / aut. Hernández H. Daniel. - Santiago : [s.n.], 2011.

Estadísticas Factura Electrónica [En línea] / aut. SII. - Enero de 2014. - <http://www.sii.cl/portales/efactura/estadistic.htm>.

Flask web development, one drop at a time [En línea] / aut. flask.org. - <http://flask.pocoo.org/>.

How to parse JSON string in Python [En línea] / aut. Nanni Dan. - <http://xmodulo.com/2013/05/how-to-parse-json-string-in-python.html>.

JSON [En línea] / aut. json.org. - <http://json.org/>.

Nginx Community [En línea] / aut. Nginx Org. - <http://wiki.nginx.org/Main>.

PDF417 [En línea] / aut. Grandzebu.net. - <http://grandzebu.net/informatique/codbar-en/pdf417.htm>.

Performance of RESTful Apps [En línea] / aut. subbu.org. - <http://www.subbu.org/blog/2011/03/performance-of-restful-apps>.

Servicio de Impuestos Internos de Chile [En línea] / aut. SII. - 2014. - <http://www.sii.cl>.

Servicios WEB para dispositivos móviles para el soporte de aplicaciones colaborativas / aut. Carrasco Víctor. - 15 de 01 de 2007. - http://www.tesis.uchile.cl/tesis/uchile/2007/carrasco_vh/sources/carrasco_vh.pdf.

SQLite Android Binding [En línea] / aut. SQLite. - <http://www.sqlite.org/android/doc/trunk/www/index.wiki>.

W3C [En línea] / aut. W3C. - <http://www.w3.org/>.

ZXing project [En línea] / aut. Googlegroups. - <https://github.com/zxing>.

Anexo A. Código de Autorización de Folios (SII)

A 1. Proceso de generación de una Autorización de Folios.

La genera el SII en base a una solicitud vía web de autorización de folios enviada por un contribuyente.

a) El contribuyente, al solicitar la autorización debe autenticarse con certificado digital y señalar el tipo de DTE y el número de Folios requeridos. El SII determinará si autorizará el rango solicitado, o una parte de él.

b) El SII genera un par de llaves (pública y privada) asociadas al rango de folios, y genera la sección Datos con la información del contribuyente, de los números autorizados y la llave pública, según la estructura definida en A.1.2.

c) El SII firma digitalmente la sección Datos usando un certificado digital de propiedad de SII construyendo la sección Firma del Código de autorización

d) Se agrega a continuación la llave privada que deberá usar el contribuyente para timbrar electrónicamente.

e) La autorización estará formada por las secciones CAF (Datos, Firma) y Llaves Privada y Pública, según el formato XML que se define en 1.2.

1. Código de Autorización de Folios, CAF (Datos, Firma) que incluye:

a) Versión del timbre electrónico

b) Rut Empresa

c) Razón Social Empresa

d) Tipo de documento

e) Rango de folios autorizados

f) Fecha de autorización de folios

g) Llave pública generada por el SII, para verificar la validez del timbre electrónico.

h) Identificador de la llave pública del SII que permite verificar la firma del SII sobre el CAF.

i) Firma del SII sobre los campos anteriores, con su llave privada. Con la llave pública del SII el contribuyente puede verificar la firma del SII, para asegurarse de la integridad y autenticidad de la información recibida.

2. Llave privada generada por el SII, que permite generar el timbre electrónico

3. Llave pública generada por el SII, que permite verificar la validez del timbre electrónico (es la misma incluida en el CAF (f), pero en un formato distinto).

f) El documento XML resultante de estas operaciones, es enviado al contribuyente.

A 2. Estructura de la Autorización

La autorización es un archivo XML que se compone de 3 secciones: el CAF (código de autorización de folios), la llave privada (RSASK) y la llave pública (RSAPUBK).

El código de autorización de folios a su vez, se compone de dos elementos, datos y firma, según lo siguiente:

```
<AUTORIZACION>
  <CAF version="1.0">
    <DA>
      .
      .
      .
    </DA>
    <FRMA>... </FRMA>
  </CAF>
  <RSASK> .....</RSASK>
  <RSAPUBK> .... </RSAPUBK>
</AUTORIZACION>
```

Sección Datos

La sección Datos (<DA>), contiene los datos relativos al contribuyente, al tipo de DTE que se está autorizando y al rango de folios que abarca esta autorización. La estructura detallada es la siguiente:

```
<CAF version="1.0">
  <DA>
    <RE>.... </RE>
    <RS>.... </RS>
    <TD>... </TD>
    <RNG>
      <D>... </D>
      <H>... </H>
    </RNG>
    <FA>... </FA>
    <RSAPK> ... </RSAPK>
    <IDK> ... </IDK>
  </DA>
  <FRMA>... </FRMA>
</CAF>
```

Donde los campos que se etiquetan son:

a) Versión: es una cadena de caracteres ASCII indicando la versión del código. Por ejemplo "1.0".

b) RUT Empresa (<RE>) : es una cadena de caracteres ASCII indicando el RUT del emisor al cual está autorizando este código en formato XXXXXXXX-X. Por ejemplo “11111111-1”, el cual equivale al contribuyente con RUT 11.111.111-1

c) Razón Social de la Empresa: representa el nombre de la empresa asociado al Rut Empresa, con un máximo de 40 caracteres.

d) Tipo DTE (<TD>): es una cadena de caracteres ASCII que representa el tipo de DTE que se está autorizando (Facturas, Guías de Despacho, etc.). Este valor está conforme a la definición de tipos de DTE impuesta por SII, donde, por citar un ejemplo, el tipo de DTE de factura quedará representado por la cadena de caracteres ASCII “33”.

e) Rango de Folios (<RNG>): es un par de valores indicando el rango de folios autorizados en estructura “Desde-Hasta”. Por ejemplo, un código autorizando el rango de folios desde el 50 al 110, contendría los valores desde “50” hasta “110”.

f) Fecha (<FA>) : es una cadena de caracteres ASCII indicando la fecha en que fue autorizado el rango de folios en formato AAAA-MM-DD, es decir, los primeros 4 caracteres señalando el año, los dos siguientes señalando el mes y los últimos 2 señalando el día, separados por guiones. Si la fecha fuese el 29 de Febrero del 2004, el valor quedaría “2004-02-29”.

g) Llave Pública del contribuyente (<RSAPK>) : Como define el modelo de operación de documentos tributarios electrónicos, cada vez que el contribuyente solicita nuevos folios, el SII le proporciona además el par de llaves (pública y privada) que le permiten generar y verificar el timbre electrónico de los DTEs asociados. Este valor, es una cadena de caracteres ASCII con el valor de la llave pública, generada por el SII. Inicialmente el SII entregará sólo llaves correspondientes al algoritmo criptográfico de llave pública RSA.

Una llave pública RSA tiene dos valores numéricos que la definen, un módulo y un exponente. El valor de “Llave Pública del contribuyente” queda definido por estos 2 valores, como se muestra a continuación:

<RSAPK> <M>... </M> <E>... </E> </RSAPK>

Módulo <M>: Indica el valor del módulo de la llave. Este valor es la codificación en Base64 del arreglo de bytes en orden Big-Endian (el byte más significativo es el elemento 0 del arreglo) que contiene el valor entero sin signo (unsigned integer) del módulo. Exponente <E> : Indica el valor del exponente de la llave. Este valor es la codificación en Base64 del arreglo de bytes en orden Big-Endian (el byte más significativo es el elemento 0 del arreglo) que contiene el valor entero sin signo (unsigned integer) del exponente.

```
<RSAPK>
  <M>AMPa7mxz8ysTRazehr5/Oiau98/ ... lku7y2twwndl/142ds54aWjqd </M>
  <E>A2.../B</E>
</RSAPK>
```

Figura A.1: Ejemplo de “Llave Pública del contribuyente” para una llave RSA

h) Identificación llave pública del SII (<IDK>): Identificación de la llave pública del SII que permite verificar la firma del SII sobre el CAF (<FRMA>). Se trata de un identificador de la llave y no de la llave.

Ejemplo: Si se estuviese:

- Autorizando al contribuyente RUT 11.111.111-1, con razón social “Ejemplo S.A.”
- el rango de folios desde el 50 al 101
- para el tipo de DTE factura (definida como “33” por SII)
- con fecha 10 de Junio del 2002
- donde al firmante autorizado que solicitó los folios se le generó una llave pública RSA en la solicitud de autorización y
- la llave pública del SII que verifica su firma sobre el CAF tiene la identificación 1

La sección Datos quedaría compuesta por la cadena de caracteres mostrada en Figura A.2

```
<DA>
  <RE>11111111-1</RE>
  <RS>Ejemplo S.A.</RS>
  <TD>33</TD>
  <RNG>
    <D>50</D>
    <H>101</H>
  </RNG>
  <FA>2002-06-10</FA>
  <RSAPK>
    <M>AMPa7mxz8ysTRazehr5/Oiau98/ ... lku7y2twwndl/142ds54aWjqd </M>
    <E>A2.../B</E>
  </RSAPK>
  <IDK> 1 </IDK>
</DA>
```

Figura A.2: Ejemplo de Datos

Sección Firma

La sección Firma (<FRMA>) corresponde a la firma digital del SII sobre Datos (<DA>) es decir Firma = FirmaSII(Datos).

Se entiende por firma digital a la aplicación de un algoritmo criptográfico de firma sobre el extracto (digest) calculado a partir de Datos. A la fecha de este documento, el modelo de operación de DTEs soporta dos algoritmos de firmas digitales SHA1+RSA y SHA1+DSA. Dependiendo del algoritmo, el valor de este campo es como sigue a continuación:

a) Algoritmos SHA1 y RSA: firma digital generada con el algoritmo de digest SHA1 y el algoritmo criptográfico de firma RSA según lo definido por OSI Interoperability Workshop. El valor de "Firma" queda definido en este caso de la siguiente forma:

```
<FRMA algoritmo="SHA1withRSA"> </FRMA>
```

Donde:

- SHA1withRSA: Indica que la firma fue generada usando el algoritmo de digest SHA1 y el algoritmo criptográfico de firma RSA. El valor es la cadena de caracteres ASCII "SHA1withRSA".

- Valor Firma (<FRMA>) : Contiene el valor de la firma. Este valor es la codificación en Base64 del valor de la firma usando el formato DER encoded PKCS#1.

```
<FRMA algoritmo="SHA1withRSA">  
F2/hdgF42d4eAw ... iU7=6faRDs6k=  
</FRMA>
```

Figura A.3: Ejemplo de Firma SHA1+RSA

b) Algoritmos SHA1 y DSA: firma digital generada con el algoritmo de digest SHA1 y el algoritmo criptográfico de firma DSA según lo definido por FIPS PUB 186. El valor de "Firma" queda definido en este caso como se muestra en A.6.

```
<FRMA algoritmo="SHA1withDSA"> </FRMA> Donde:
```

- SHA1withDSA: Indica que la firma fue generada usando el algoritmo de digest SHA1 y el algoritmo criptográfico de firma DSA. El valor es la cadena de caracteres ASCII "SHA1withDSA".

- Valor Firma (<FRMA>): Contiene el valor de la firma. Una firma utilizando DSA está compuesta por dos valores r y s. Este valor es la codificación en Base64 del valor de la firma representado por la secuencia ASN.1 de dos valores INTEGER de r y s en ese orden (SEQUENCE ::= r INTEGER, s INTEGER).

```
<FRMA algoritmo="SHA1withDSA">  
MCw7yGfcx451 ... aKhy72bvDw==  
</FRMA>
```

Figura A.4: Ejemplo de Firma SHA1+DSA

Sección Llave Privada

La sección Llave Privada (<RSASK>) corresponde a llave privada asignada por el SII al contribuyente para generar el timbre electrónicos de los DTEs del rango de folios respectivo. La firma generada con la llave privada puede ser validada con la llave pública asociada (<RSAPK>)

La llave privada RSA generada por el SII, se entrega en formato estándar PEM (Privacy Enhanced Mail).

Sección Llave Pública

La sección Llave Pública (<RSAPUBK>) corresponde a la llave pública asignada por el SII al contribuyente, para verificar el timbre electrónico de los DTEs. esta llave también se incluye dentro del CAF y se entrega además en formato estándar PEM (Privacy Enhanced Mail), para facilitar al contribuyente la verificación del par de llaves proporcionadas por el SII, antes de comenzar a utilizarlas.

En el ejemplo de Figura A.5,

Si se estuviese:

- Autorizando al contribuyente con RUT 11.111.111-1, con Razón Social “Ejemplo S.A.”
- el rango de folios desde el 50 al 101
- para el tipo de DTE factura (definida como “33” por SII)
- con fecha 10 de Junio del 2002 • donde al solicitante se le asignó una llave pública RSA
- el identificador de la llave pública del SII que verifica la firma es 3
- el SII firmó la sección datos usando SHA1+RSA y asignó la llave privada RSA respectiva

El código de Autorización de folios quedaría compuesto por la cadena de caracteres mostrada:

```
<AUTORIZACION>
  <CAF version="1.0">
    <DA>
      <RE>11111111-1</RE>
      <RS>Ejemplo S.A.</RS>
      <TD>33</TD>
      <RNG>
        <D>50</D>
        <H>101</H>
      </RNG>
      <FA>2002-06-10</FA>
      <RSAPK>
        <M>AMPa7mxz8ysTRazehr5/Oiau98/ ... lku7y2twwndl/142ds54aWjqd </M>
        <E>A2.../B</E>
      </RSAPK>
      <IDK> 3 </IDK>
    </DA>
    <FRMA algoritmo="SHA1withRSA"> F2/hdgF42d4eAw ... iU7=6faRDs6k= </FRMA>
  </CAF>
  <RSASK>
    :----- BEGIN RSA PRIVATE KEY-----  nShd63c ...      -----END RSA PRIVATE KEY-----
  </RSASK>
  <RSAPUBK>
    :----- BEGIN PUBLIC KEY-----  fs3D21axS ...      -----END PUBLIC KEY-----
  </RSAPUBK>
</AUTORIZACION>
```

Figura A.5: Ejemplo de Autorización de Folios

A 3. Almacenamiento y Uso del CAF

El archivo XML con la información de folios autorizados por el SII debe ser almacenado y resguardado adecuadamente por los contribuyentes autorizados ya que contiene la información que permite asignar folios y generar el timbre electrónico de los documentos tributarios electrónicos que emita.

El CAF debe incluirse en el timbre electrónico de cada documento tributario para permitir que el SII verifique que el número de folio fue efectivamente autorizado por el Servicio. Esta verificación del SII arrojará un resultado negativo si el CAF incluido en el timbre electrónico presenta alguna diferencia con el entregado por el SII, por lo que resulta de vital importancia almacenar y conservar el CAF tal como fue entregado por el Servicio.

Anexo B. Timbre Electrónico del DTE (SII)

B 1. Introducción

Cada documento emitido debe ir timbrado electrónicamente, incluyendo la información representativa del DTE y el código de autorización de folios asociado.

B 2. Generación De Un Timbre

a) El contribuyente debe identificar cuál es el código de autorización de folios correspondiente al folio del DTE que va a emitir. Este CAF debe ser incorporado a la sección Datos del timbre, sin modificación alguna respecto al entregado por el SII.

b) Debe armar la sección Datos del timbre electrónico del DTE con los datos del DTE según la estructura definida en A.2.2 Estructura.

c) Posteriormente, el contribuyente debe firmar digitalmente la sección Datos usando la llave privada entregada por el SII al autorizar los folios y construye la sección Firma.

d) El timbre estará formado por las secciones Datos y Firma según el formato que se definió en la sección anterior A.2.2 Estructura.

Este timbre, se debe incluir en el DTE resultante y en su versión impresa. En el caso de la versión impresa del DTE, el timbre se imprime en un código de barras bidimensional, simbología PDF417.

B 3. Estructura

El Timbre es una cadena de caracteres ASCII que se compone de dos secciones, datos y firma, como se detalla a continuación:

```
<TED version="1.0"> <DD> . . . </DD> <FRMT
algoritmo="SHA1withRSA"> ... </FRMT> </TED>
```

Sección Datos

La sección Datos (<DD>), contiene los datos representativos del DTE: RUT del emisor, tipo del DTE, folio del DTE, fecha de emisión, RUT del receptor, Razon Social del Receptor, Monto Total del DTE, Descripción del primer Item de detalle, el CAF y el TimeStamp (fecha y hora) de generación del timbre, etiquetados de la siguiente forma:

```
<TED version="1.0"> <DD> <RE>... </RE> <TD>... </TD> <F>... </F>
<FE>... </FE> <RR>.... </RR> <RSR>.... </RSR> <MNT>... </MNT> <IT1>....
</IT1> <CAF>... </CAF> <TSTED>... </TSTED> </DD>
```

Donde las etiquetas representan:

a) Versión: es una cadena de caracteres ASCII indicando la versión del timbre. La versión que define este documento es la "1.0".

b) RUT Emisor (<RE>) : es una cadena de caracteres ASCII indicando el RUT del emisor del DTE en formato XXXXXXXX-X. Por ejemplo “11111111-1”, que equivale al contribuyente emisor con RUT 11.111.111-1

c) Tipo DTE (<TD>): es una cadena de caracteres ASCII que representa el tipo de DTE que se está timbrando (Facturas, Guías de Despacho, etc.). Este valor está conforme a la definición de tipos de DTE impuesta por SII, donde, por citar un ejemplo, el tipo de DTE de factura quedaría representado por la cadena de caracteres ASCII “33”.

d) Folio DTE (<F>): es una cadena de caracteres indicando el número de folio en notación decimal entera del DTE que se está timbrando . Por ejemplo, el timbre electrónico del DTE número de folio 2752, contendría en este valor la cadena ASCII “2752”.

e) Fecha (<FE>): es una cadena de caracteres ASCII indicando la fecha de emisión (generación) del DTE en formato AAAA-MM-DD, es decir, los primeros 4 caracteres señalando el año, los dos siguientes señalando el mes y los últimos 2 señalando el día; separados por el carácter “-” (guión). Si la fecha fuese el 29 de Febrero del 2004, el valor quedaría “2004-02-29”.

f) RUT Receptor (<RR>): es una cadena de caracteres ASCII indicando el RUT del receptor del DTE en formato XXXXXXXX-X. Por ejemplo “22222222-2”, que equivale al contribuyente emisor con RUT 22.222.222-2

g) Razón Social del Receptor (<RSR>): representa el nombre de la empresa asociado al Rut Empresa, con un máximo de 40 caracteres.

h) Monto Total del DTE (<MNT>): es una cadena de caracteres ASCII que representa el Monto Total del DTE en pesos chilenos, en notación decimal entera (sin decimales). Por ejemplo, el valor 13 queda representado por “13” y el valor 9 queda representado por “9”.

i) Ítem 1 (<IT1>): es una cadena de caracteres que contiene la descripción del primer ítem del detalle, con un máximo de 40 caracteres.

j) Código de Autorización de Folios (<CAF>): Contiene el código de autorización de folios, que autoriza el DTE, es decir, que el folio del DTE está dentro del rango del código de autorización, que el tipo y emisor coinciden en ambos, y que la llave pública del contribuyente contenida en el código, es la que verifica la sección Firma del timbre electrónico. Para la especificación del código de autorización vea el ANEXO 1 “Código de Autorización de Folios”.

k) TimeStamp del Timbre Electrónico (<TSTED>): Contiene la fecha y hora en que se generó el timbre electrónico en formato AAAA-MM-DDTHH:MI:SS

Si se estuviese generando el timbre electrónico del:

- DTE número 67 • tipo de DTE factura electrónica (definida como “33” por SII)
- del contribuyente emisor RUT 11.111.111-1
- para el Receptor RUT 12345678-5, razón social “Comprador S.A.”

- con fecha 11 de Junio del 2002,
- por un total de 24365 y el primer item del detalle es “Caja de Zapatos”
- el timbre se hubiera generado a las 7 horas con 34 minutos y 15 segundo

La sección Datos quedaría compuesta por la cadena de caracteres mostrada en Figura A.6.

```

<TED version="1.0">
  <DD>
    <RE>11111111-1</RE>
    <TD>33</TD>
    <F>67</F>
    <FE>2002-06-11</FE>
    <RR>12345678-5</RR>
    <RSR>Comprador S.A.</RSR>
    <MNT>24365</MNT>
    <IT1>Caja de Zapatos</IT1>
    <CAF>... </CAF>
    <TSTED>2002-06-11T07:34:15</TSTED>
  </DD>
  ....
</TED>

```

Figura A.6: Ejemplo de Datos en timbre electrónico (ver detalle de CAF en figura A.5)

Sección Firma La sección Firma (<FMRT>) corresponde a la firma digital del contribuyente emisor (con la llave privada generada y entregada por el SII para ese propósito) sobre Datos (<DD>)

Firma = Firma Contribuyente(Datos)

Según se indicó en A.1.2. el SII entregará sólo llaves para algoritmo de firmas digitales SHA1+RSA por lo que el valor de “Firma” queda definido de la siguiente forma:

```
<FRMT algoritmo="SHA1withRSA"> ... </FRMT>
```

Donde:

- SHA1withRSA: Indica que la firma fue generada usando el algoritmo de digest SHA1 y el algoritmo criptográfico de firma RSA. El valor es la cadena de caracteres ASCII “SHA1withRSA”.

- Valor Firma (<FMRT>) : Contiene el valor de la firma. Este valor es la codificación en Base64 del valor de la firma usando el formato DER-encoded PKCS#1.

En Figura A.7 se muestra un ejemplo de “Firma” para el caso SHA1 y RSA.

```

<FRMT algoritmo="SHA1withRSA">
  G3=dhiawT5a4/... =09UjhGfsR7I/
</FRMT>

```

Figura A.7: Ejemplo de Firma SHA1+RSA

B 4. Consideraciones para la Generación y Firma del Timbre Electrónico.

Los valores de claves públicas, firmas, digest y certificados que se incluyen en los XML descritos anteriormente para el CAF y Timbre Electrónico, van codificadas en base 64 (como se define en el RFC 2045 Sección 6.8) y se imprimen a lo más 76 caracteres por línea.

Es importante verificar que las glosas incluidas en el timbre electrónico: Razón Social del Receptor (tag <RSR>) y Nombre del Item 1 (tag <IT1>) incluyan sólo caracteres codificados de acuerdo al estándar ISO-8859-1, de lo contrario el SII podría obtener un resultado negativo al verificar la firma electrónica del timbre y por lo tanto declarar no válido el documento que se esté fiscalizando. El contribuyente debe cuidar que las librerías que utiliza para firmar no realicen transformaciones sobre la codificación de los caracteres (por ejemplo a la codificación UTF-8).

Se debe tener presente que en XML se han predefinido de manera estándar 5 representaciones para caracteres con significado especial dentro de la estructura del XML (predefined entities). Las entidades predefinidas y su representación estándar son:

Carácter Especial	Representación	Alternativa
&	&	&
<	<	<
>	>	>
“	"	"
‘	'	'

Por ejemplo:

La Razón Social Receptor: Empresas A&B Limitada (21 caracteres)
Debe tener la Codificación XML: Empresas A&B Limitada (25 caracteres)

El resto de los caracteres especiales (acentos, ñes, etc.) deben ser codificados de acuerdo a lo especificado en el set de caracteres ISO-8859-1. Si no se respeta esta convención el archivo con el envío de DTEs será rechazado al validar el schema, con el error “Invalid Character”.

Para efecto de calcular el digest de la firma digital del timbre electrónico y de la autorización de folios se eliminan los caracteres de fin de línea y los blancos y/o tab entre TAGS, así como la referencia a NameSpaces. El digest se calcula sobre el string resultante.

En resumen para generar y firmar correctamente un timbre electrónico, se debe tener presente lo siguiente:

- El CAF se incluye tal como fue entregado por el SII, sin ningún tipo de modificaciones.
- Los campos de caracteres incluidos en el TED (razón social receptor y descripción del primer ítem) se deben codificar respetando las entidades predefinidas por XML y el set de caracteres ISO-8859-1.

- La información incluida en el TED debe coincidir con la información de Encabezado y Detalle del DTE.
- La firma del TED se realiza sobre el string resultante de eliminar todos los caracteres que están entre el tag de cierre de un elemento y el tag de inicio del siguiente, sin modificar la información que va entre el tag de inicio y el tag de fin de los elementos terminales.

Es importante respetar estas indicaciones, de lo contrario la verificación del timbre electrónico resultará incorrecta.

Debido a que el Timbre Electrónico se debe imprimir en un código de barras 2D (PDF417) y a las restricciones de espacio asociadas, las firmas que el Timbre Electrónico incluye deben regirse estrictamente por lo especificado en el presente documento y no por el estándar XMLDSIG como es el caso de la firma del DTE y envío de DTE.

B 5. Reglas Para La Generación e Impresión Del Timbre PDF417

El SII ha establecido las siguientes reglas en la generación e impresión del código PDF417

Impresión del timbre

- Impresora láser o inyección tinta con una resolución mínima de 300 DPI
- Color de impresión: Negro
- Quiet Zone: Para evitar que líneas o textos cercanos al código puedan ser interpretados como parte de éste, el código debe tener una Quiet Zone (espacio en blanco) de mínimo 0,25 pulgadas alrededor de cada uno de sus cuatro lados.
- Truncated: Esta opción que permite omitir alguna parte del código, pero que aumenta su sensibilidad al daño, **no debe usarse**.

Generación del Timbre

- Para evitar problemas con los caracteres especiales que pudiera contener el timbre electrónico, al generar el código PDF417 se debe utilizar el modo de codificación binario (Byte Compaction Mode)
- Error Correction Level (ECL): Dada la cantidad de información, se debe utilizar nivel 5.
- X Width (X Dim): Es el ancho del elemento impreso del código, barra o espacio, más angosto y se expresa en mils (milésimas de pulgada). Se debe usar un valor de X Dim mínimo de 6,7 mils. – Row Height (Y Dim): Es la dimensión vertical, expresada en mils, de una fila del código PDF417. Se debe usar una relación (3:1) respecto al valor X Dim.
- Recomendamos ajustar los parámetros para obtener un código de barras impreso de un tamaño máximo de 3 cms de alto x 9 cms de ancho.