



Manual API ACI DescargaSAT

Control de Versiones

Nombre	No. Versión	Modificaciones	Fecha
Daniel Jesús Hernandez Francisco	1.0	Creación de documento y contenido	27-02-2019

Control de Revisiones

Nombre	No. Versión	Modificaciones	Fecha
Gloria Minerva González Hernández	1.0	Observaciones de redacción y ortografía.	05/03/2019

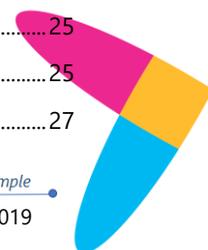
Control de Autorizaciones

Nombre	No. Versión	Modificaciones	Fecha

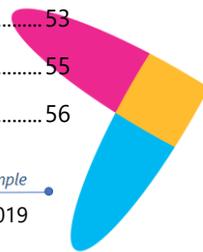


Contenido

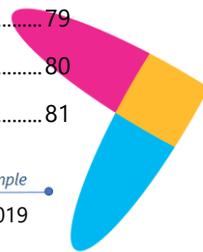
1.	Generalidades.....	7
1.1	Introducción.....	7
1.2	Propósito.....	7
1.3	Alcance.....	7
1.4	Requerimientos Mínimos.....	7
1.5	Términos y Definiciones.....	7
2.	Requerimientos.....	8
3.	Convenciones del API.....	8
3.1	Response general del API.....	8
3.2	Parámetros.....	9
3.2.1	Consideraciones especiales para métodos que incluyen parámetros en el URL PATH.....	9
3.2.2	Notacion de tipos de datos.....	10
4.	Autenticación.....	10
5.	URL de producción.....	11
6.	CURL.....	12
6.1.1	Ejemplo genérico de consumo en Python.....	14
7.	SDK.....	15
7.1	SDK de JAVA.....	15
7.1.1	URL de producción SDK Java.....	15
7.2	SDK de PHP.....	15
7.2.1	URL Producción SDK PHP.....	16
7.3	SDK .NET.....	16
7.3.1	Agregar DLL al proyecto.....	17
7.3.2	URL Producción SDK .NET.....	18
8.	Métodos Relacionados A Descargas.....	18
8.1	Crear una Descarga.....	18
8.1.1	Request.....	19
8.1.2	Response.....	20
8.1.3	Ejemplo con SDK de JAVA.....	21
8.1.4	Ejemplo con SDK de PHP.....	22
8.1.5	Ejemplo con SDK de .NET.....	24
8.2	Consultar Descarga Por UUID.....	25
8.2.1	Request.....	25
8.2.2	Response.....	25
8.2.3	Ejemplo SDK JAVA.....	27



8.2.4	Ejemplo con SDK de PHP	27
8.2.5	Ejemplo con SDK de .NET.....	28
8.3	Reactivar Descarga por UUID.....	29
8.3.1	Request	30
8.3.2	Response	30
8.3.3	Ejemplo SDK de JAVA.....	31
8.3.4	Ejemplo con SDK de PHP	31
8.3.5	Ejemplo con SDK de .NET.....	32
8.4	Obtener descargas diarias.....	33
8.4.1	Request	33
8.4.2	Response	35
8.4.3	Tipo de dato DataDescargaProgramada	35
8.4.4	Ejemplo SDK de JAVA.....	37
8.4.5	Ejemplo con SDK de PHP	38
8.4.6	Ejemplo con SDK de .NET.....	40
9.	Métodos relacionados CFDI.....	40
9.1	Obtener información CFDI Por UUID	41
9.1.1	Request	41
9.1.2	Response	41
9.1.3	Ejemplo SDK de JAVA.....	42
9.1.4	Ejemplo con SDK de PHP	43
9.1.5	Ejemplo con SDK de .NET.....	44
9.2	Descargar CFDI por UUID.....	45
9.2.1	Request	45
9.2.2	Response	45
9.2.3	Ejemplo SDK de JAVA.....	46
9.2.4	Ejemplo con SDK de PHP	46
9.2.5	Ejemplo con SDK de .NET.....	47
9.3	Consultar CFDI.....	48
9.3.1	Request	48
9.3.2	Request	49
9.3.3	Response	49
9.3.4	Ejemplo SDK de JAVA.....	52
9.3.5	Ejemplo con SDK de PHP	53
9.3.6	Ejemplo con SDK de .NET.....	55
9.4	Descarga de CFDI por consulta.....	56



- 9.4.1 Request 56
- 9.4.2 Response 57
- 9.4.3 Ejemplo SDK de JAVA 58
- 9.4.4 Ejemplo SDK de PHP 59
- 9.4.5 Ejemplo con SDK de .NET..... 60
- 9.5 Existe CFDI en CFDI resguardos 61
 - 9.5.1 Request 62
 - 9.5.2 Response 62
 - 9.5.3 Ejemplo SDK de JAVA 62
 - 9.5.4 Ejemplo SDK de PHP 63
 - 9.5.5 Ejemplo con SDK de .NET..... 64
- 9.6 Desbloqueo CFDI 65
 - 9.6.1 Request 65
 - 9.6.2 Response 66
 - 9.6.3 Ejemplo SDK de JAVA 67
 - 9.6.4 Ejemplo SDK de PHP 68
 - 9.6.5 Ejemplo SDK de .NET 68
- 10. Métodos relacionados a la EFIRMA..... 69
 - 10.1 Subir EFIRMA..... 69
 - 10.1.1 Request 69
 - 10.1.2 Response 70
 - 10.1.3 Ejemplo con SDK de JAVA 71
 - 10.1.4 Ejemplo con SDK de PHP 72
 - 10.1.5 Ejemplo con SDK de .NET..... 73
- 11. Métodos relacionados a la gestión de descargas 74
 - 11.1 Consultar configuración 74
 - 11.1.1 Request 74
 - 11.1.2 Response 75
 - 11.1.3 Ejemplo con SDK de JAVA 75
 - 11.1.4 Ejemplo con SDK de PHP 77
 - 11.1.5 Ejemplo con SDK .NET 77
 - 11.2 Configurar horas de espera..... 78
 - 11.2.1 Request 78
 - 11.2.2 Response 79
 - 11.2.3 Ejemplo con SDK de JAVA 80
 - 11.2.4 Ejemplo con SDK de PHP 81



11.2.5	Ejemplo con SDK .NET	82
11.3	Configurar método de descarga.....	82
11.3.1	Request	83
11.3.2	Response	83
11.3.3	Ejemplo con SDK de JAVA	84
11.3.4	Ejemplo con SDK de PHP	85
11.3.5	Ejemplo con SDK .NET	86
12.	Tipos de datos.....	86
12.1.1	Tipo CFDI.....	86
12.1.2	Tipo Configuracion (response).....	87
13.	Código de error	88
14.	Información de Contacto con SIFEI	90



1. Generalidades

1.1 Introducción

API ACI DescargaSAT es un servicio Web basado en una arquitectura REST que tiene como objetivo proveer las operaciones más importantes relacionadas a la descarga de CFDI y su gestión. Este documento describe el servicio así como los datos requeridos para consumirlo, además incluye ejemplos para su consumo por medio de los SDK provistos por Sifei.

1.2 Propósito

Proporcionar a los programadores e integradores de software una guía precisa para el consumo de servicios de ACI DescargaSAT.

1.3 Alcance

Este manual Técnico está dirigido a desarrolladores e ingenieros en software que requieran consumir los servicios de ACI DescargaSAT. El documento se limita a proporcionar la descripción sobre cada *endpoint* del API ACI DescargaSAT incluyendo la solicitud y respuesta; para su uso es necesario contar con los conocimientos técnicos requeridos. Para mayor información se recomienda leer el manual de Usuario de ACI DescargaSAT.

1.4 Requerimientos Mínimos

-  Cliente HTTP
-  Token Valido
-  Estar registrado en ACI Descarga SAT

1.5 Términos y Definiciones

Tabla 1. Términos y definiciones

Acrónimo	Definición
CFDI	Comprobante Fiscal Digital por Internet
NDA	<i>Non-Disclosure Agreement</i> , documento de Acuerdo de No Divulgación
PAC	Proveedor Autorizado de Certificación
PCCFDI	Proveedor de Certificación de Comprobantes Fiscales Digitales por Internet (antes PAC)
PCECFDI	Proveedor de Certificación de Expedición de Comprobantes Fiscales Digitales por Internet, que aplica a los contribuyentes que expiden CFDI al ser Adquirientes de Bienes y Servicios, y Sector Primario (simplificado "Sector Primario")
SAT	Servicio de Administración Tributaria
SIFEI	Solución Integral de Facturación Electrónica e Informática SIFEI S. A. de C. V.
SLA	<i>Service Level Agreement</i> , Acuerdo de Niveles de Servicios
JSON	Acrónimo de <i>JavaScript Object Notation</i> , es un formato de texto ligero para el intercambio de datos
REST	La Transferencia de Estado Representacional (en inglés <i>Representational State Transfer</i>) o REST es un estilo de arquitectura de software para describir cualquier interfaz entre sistemas que utilice directamente HTTP para obtener datos o indicar la ejecución de operaciones sobre los

Acrónimo	Definición
	datos, en cualquier formato (XML, JSON, etc.) sin las abstracciones adicionales de los protocolos basados en patrones de intercambio de mensajes, como por ejemplo SOAP.
RESTFUL	RESTful hace referencia a un servicio Web que implementa la arquitectura REST
SOAP	<i>Simple Object Access Protocol</i> , es un protocolo estándar que define cómo dos objetos en diferentes procesos se comunican por medio de intercambio de datos XML.
Token	Un <i>token</i> (token de seguridad, token de autenticación o token criptográfico) es una clave criptográfica que se le otorga a un usuario autorizado de un servicio para facilitar el proceso de autenticación.
Endpoint	Es un punto final de un canal comunicación representado por una URL de un servidor o servicio. Cada endpoint se encarga de proveer acceso y operaciones a recursos específicos, los cuales varían en función del método HTTP solicitado.

2. Requerimientos.

Para consumir y hacer uso de todas las funcionalidades del API DescargaSAT se debe contar con la **EFIRMA** del RFC asociado a los CFDI, la cual debe ser válida (vigente) ante el SAT. Así mismo, debe contar con el **token** de autenticación del RFC de los CFDI a obtener (ver apartado 4), la cual es provista por medio del portal y enviada a tu correo.

3. Convenciones del API

El formato para las peticiones como respuestas es JSON. Si bien este servicio está basado en REST, no se implementan los métodos HTTP: PUT, DELETE y PATCH con el fin de facilitar su implementación, es decir, todas las peticiones son hechas vía GET o POST. Así mismo para la documentación no se hace distinción entre *endpoint* y método de WS, ya que formalmente un *endpoint* puede tener varias operaciones según el método HTTP.

3.1 Response general del API

La respuesta JSON sigue la estructura general que muestra la Tabla 2:

Tabla 2. Estructura de respuesta JSON que ACI DescargaSAT ofrece.

Código	Tipo	Definición
Status	string	Indica el estado de la solicitud, sus valores posibles son: "success", "fail", "error".
Data	Any	Contiene el resultado de la operación. Si se trata de consultas, este campo contendrá el resultado de tu búsqueda. Para otro tipo de operaciones el valor de este campo es opcional, para mayor detalle se debe referir a la respectiva sección Response de cada "método".
Code	String	Si bien a través del campo status podemos saber si una solicitud ejecutó la operación deseada, o bien si fue aceptada y validada, en el campo <i>code</i> se devuelve un código específico para cada tipo de operación, en general este campo contiene el código de error cuando status es diferente a success.

Código	Tipo	Definición
Message	String	Cuando exista un error en tu petición, junto con CODE, este campo contendrá el error verbal, es decir, una descripción del error.

La Tabla 3 presenta la estructura genérica de respuesta.

Tabla 3. Ejemplo de respuesta.

```
{
  "status": "success",
  "data": "",
  "message": "",
  "code": ""
}
```

Con el objetivo de facilitar la deserialización de las respuestas JSON —sobre todo en lenguajes tipados como JAVA o C#—, todos los campos son obligatorios en las respuestas que ofrece la WEB API ACI DescargaSAT, de modo que se pueda implementar fácilmente el deserializador de preferencia, para el caso de JAVA se recomienda GSON o Jackson.

3.2 Parámetros

Los parámetros cuando sean peticiones de tipo GET serán vía URL PATH (ver Tabla 4), para POST puede ser vía URL cuando se trate de un UUID en caso contrario serán por **body**. Cuando se envíe un **body**, los parámetros siempre deberán estar en formato JSON (ver la respectiva documentación de cada ENDPOINT (método) para mayor información).

Tabla 4. Parámetros vía URL PATH.

Código	Tipo	Descripción
{parametro}	string	Cuando una URL incluya al final de su PATH corchetes o @, significará que requiere ser sustituido por el valor necesario (ver documentación específica del método), esto significa que las URL deben cumplir con el patrón señalado dentro de cada método, en caso de no cumplir (MATCH) con la expresión entonces no se considera como un método y se arrojará un error 404.

Nota: si no se coloca un valor en el URL PATH, será arrojado un error método no encontrado (404), debido a que el método en cuestión esperará una URL con el formato indicado.

3.2.1 Consideraciones especiales para métodos que incluyen parámetros en el URL PATH.

Es importante no omitir valores donde se incluyen parámetros en el PATH de la URL, ya que aun si esta correcta la URL (**URLBASE+URL_RELATIVA**) si falta el parámetro entonces (**URLBASE+URL_RELATIVA+PARAMETRO**) el server devolverá una respuesta de método no encontrado, ejemplo:



El método para consultar un CFDI por su UUID es de tipo **GET** y requiere un parámetro en su **PATH**, tal y como se muestra en la Tabla 5. La url se conforma URL_ABSOLUTA +PARAMETRO_UUID para mayor detalle ver **Obtener información CFDI Por UUID**.

Tabla 5. ejemplo de método con parámetro en su URL PATH.

(URLBASE+URL_RELATIVA+PARAMETRO)
https://descargasat.sifei.com.mx/api/v2/descargasatsifei/cfdi/query/E23DF780-96DC-4AF3-80AF-XXXXXXXXXXXX
<pre>{ "status": "error", "data": null, "message": "Error de autenticaci\u00f3n, falta header Authorization", "code": "1003" }</pre>

Si llega a omitir el parámetro, la URL no **cumpliría** con el patrón (pattern) correspondiente al método. Por tanto, no se encontraría el método y una respuesta como la mostrada en la Tabla 6:

Tabla 6. Caso cuando se omite un parámetro el el PATH de la URL.

(URLBASE+URL_RELATIVA)
https://descargasat.sifei.com.mx/api/v2/descargasatsifei/cfdi/query/
<pre>{ "status": "error", "code": 404, "data": null, "message": "Metodo-Endpoint no encontrado. Asegurate de que el path y metodo HTTP sean correctos." }</pre>

3.2.2 Notacion de tipos de datos.

En general, cuando se especifique los datos para un método, cuando incluyan el símbolo "?" después de un tipo de dato, significa que su valor puede ser null (*nullable*), incluso cuando su tipo de dato sea primitivo, un ejemplo es int?, que en Java equivale al tipo de dato Integer (*Boxed primitives*), y no al tipo de dato int (que no soporta null y su valor por defecto es 0), en C# equivale precisamente a "int?", en otros lenguajes no tipados no debe existir mayor detalle.

4. Autenticación



Para el consumo de todos los *endpoint* de ACI DescargaSAT es necesario que cuente con un token válido, el cual va en la cabecera HTTP tal y como se muestra en la siguiente Tabla 7. Este token va en la cabecera Authorization.

Tabla 7. Ejemplo de colocación de token en cabecera.

```
POST /api/v2/descargasatsifei/cfdi/download HTTP/1.1
Authorization: qpJHAYQLsdas89e1MI3j21301230912j3121
Content-Type: application/json; charset=UTF-8
Content-Length: 122
.....
```

NOTA: Todas las peticiones deben llevar el token en esta cabecera, de lo contrario no será posible consumir el servicio.

El token se puede obtener desde la versión WEB de ACI DescargaSAT, para ello debe ir a la sección configuración y dar clic en el botón avanzado, tras ello se desplegará la sección que se muestra en la Figura 1, posteriormente dé clic en el botón "Enviar Token por Email" y listo.



Figura 1. Activación de token

5. URL de producción

La URL de producción es la siguiente: descargasat.sifei.com.mx, para consumir cada método se deberá de concatenar esta URL con la proporcionada en cada método, es decir, las URL presentadas en las secciones son relativas a esta URL base. Es necesario asegurarse de usar el protocolo HTTPS, es decir, la URL de producción base la cual es la siguiente:

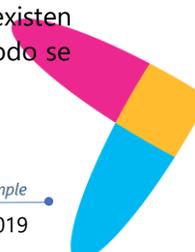
- <https://descargasat.sifei.com.mx>

Por ejemplo, si la URL para consultar CFDI es

"/api/v2/descargasatsifei/DescargaProgramadas/query", entonces la URL absoluta es:

- <https://descargasat.sifei.com.mx/api/v2/descargasatsifei/DescargaProgramadas/query>

Es importante que la URL y el método HTTP **coincidan**, en caso contrario se devolverá un error 404, en el ejemplo anterior la petición debe ser POST (en su respectiva sección se indica el método). Así mismo existen métodos cuyos parámetros están incluidos en la URL, en la respectiva documentación de cada método se describe esta información.



Nota: La URL de producción viene incluida dentro los SDK, favor de referirse a la respectiva sección de cada método que viene incluida en este documento.

6. CURL

Se pueden consumir todos los servicios vía CURL, para ello se debe agregar la bandera de cabecera "-H" seguido del valor "Authorization: **token**", esto es obligatorio ya que para consumir los métodos se espera un token válido el cual es tomado dentro de esa cabecera, la Tabla 8 incluye un ejemplo de consumo:

Tabla 8. Ejemplo de consumo con CURL (método de consultar CFDI por UUID).

Comando curl:
<pre>curl -H "Authorization: qpJHAYQLsdas89e1MI3j21301230912j3121" https://descargasat.sifei.com.mx/api/v2/descargasatsifei/cfdi/query/E23DF780 -96DC-4AF3-ZZZZ-XXXXXXXXXXXX -X GET</pre>
Petición:
<pre>GET /api/v2/descargasatsifei/cfdi/query/dsadsa HTTP/1.1 Host: descargasat.sifei.com.mx:443 User-Agent: curl/7.58.0 Accept: */* Authorization: qpJHAYQLsdas89e1MI3j21301230912j3121</pre>

Si se desea consumir un servicio POST que espere un JSON con la bandera "-d" seguido del JSON, además de la cabecera 'Content-Type', a continuación se ejemplifica como consumir el método para consultar las descargas (Tabla 9):

Tabla 9. Ejemplo consumiendo servicio que recibe un JSON .

Comando curl:
<pre>curl -H "Authorization: qpJHAYQLsdas89e1MI3j21301230912j3121" -H 'Content- Type: application/json' -d '{"estadoDescarga":""}' https://descargasat.sifei.com.mx/api/v2/descargasatsifei/DescargaProgramadas /query -X POST</pre>
Petición:
<pre>POST /api/v2/descargasatsifei/DescargaProgramadas/query HTTP/1.1 Host: descargasat.sifei.com.mx:443 User-Agent: curl/7.58.0 Accept: */* Authorization: qpJHAYQLsdas89e1MI3j21301230912j3121 Content-Type: application/json Content-Length: 22 {"estadoDescarga":""}</pre>

Nota: si se ejecuta este comando desde una terminal Windows, se deben escapar las comillas ya que este sistema operativo no soporta comillas simples tal y como se muestra en la Tabla 10.



Tabla 10. Invocación incorrecta de CURL desde terminal Windows, usando comillas simples tanto para la cabecera CONTENT_TYPE como para el JSON.

Comando CURL sin escapar las comillas , CURL no interpretara la cadena JSON e ignorara el content-type, cambiándolo a x-www-form-urlencoded y por tanto no será procesado por el API:

```
curl -H "Authorization: qpJHAYQLsdas89e1MI3j21301230912j3121" -H 'Content-Type: application/json' -d '{"estadoDescarga": "*"}' https://descargasat.sifei.com.mx/api/v2/descargasatsifei/DescargaProgramadas/query -X POST
```

Petición errónea debido a que no se escaparon las comillas:

```
POST /api/v2/descargasatsifei/DescargaProgramadas/query HTTP/1.1
Host: descargasat.sifei.com.mx:443
User-Agent: curl/7.55.1
Accept: */*
Authorization: qpJHAYQLsdas89e1MI3j21301230912j3121
Content-Length: 20
Content-Type: application/x-www-form-urlencoded

'{estadoDescarga:*}'
```

Forma correcta desde Windows, ver Tabla 11:

Tabla 11 Correcta invocación de CURL desde terminal Windows.

Comando CURL escapado y usando comillas en vez de comillas simples en Windows.

```
curl -H "Authorization: qpJHAYQLsdas89e1MI3j21301230912j3121" -H "Content-Type: application/json" -d "{\"estadoDescarga\": \"*\"}" https://descargasat.sifei.com.mx/api/v2/descargasatsifei/DescargaProgramadas/query
```

Petición correcta

```
POST /api/v2/descargasatsifei/DescargaProgramadas/query HTTP/1.1
Host: descargasat.sifei.com.mx:443
User-Agent: curl/7.55.1
Accept: */*
Authorization: qpJHAYQLsdas89e1MI3j21301230912j3121
Content-Length: 20
Content-Type: application/x-www-form-urlencoded

'{estadoDescarga:*}'
```



6.1.1 Ejemplo genérico de consumo en Python

La Tabla 12 presenta un ejemplo de consumo de API desde Python, a partir de este ejemplo se puede consumir todos los métodos.

Tabla 12 Ejemplo genérico de consumo del API desde python.

```
import requests
#Ejemplo de consumo.
# URL
url = https://descargasat.sifei.com.mx
/api/v2/descargasatsifei/DescargaProgramadas/query'
# cabecera donde va el token
headers ={'Authorization': 'qpJHAYQLsdas89e1MI3j21301230912j3121'}
# el body JSON, para este caso es el body de pide consultar descargas.
payload = {'estadoDescarga': '*'} # todos los estados de descarga
response = requests.post(url,headers=headers,json=payload)
#se compara si todo fue a nivel HTTP:
if response.status_code==200 :
print ("Petición correcta")
#se deserializa a un diccionario
body=response.json()
#ahora a nivel de API se comprueba el estado de la petición.
if body['status']=='success' :
# success entonces data tendrá la información deseada.
print (body['data'])
else:
print ("Error durante la operación, obteniendo información del error:")
#si no es success, con code y message se puede averiguar el porque
print (body['status'])
print (body['message'])
print (body['code'])
else :
print ("Error al realizar petición,código HTTP(no confundir con el del API): ")
print (response.status_code)
```



7. SDK

ACI DescargaSAT ofrece los SDK para el consumo del API el para los siguientes 3 lenguajes:

1. JAVA (.jar)
2. PHP (archivos .php)
3. C# (.dll)

Estos SDK facilitan enormemente el consumo del API y ahorran tiempo de desarrollo a las implementaciones.

7.1 SDK de JAVA

Se requiere como mínimo Java 7, mientras que 8 es el ideal. Se incluye un JAR el cual debe ser importado en el proyecto, los pasos varían según el entorno de desarrollo. La Tabla 13 representa la ficha técnica del SDK JAVA.

Tabla 13. Ficha Técnica SDK JAVA.

Versión mínima	JAVA	7+
Recursos ofrecidos		Descripción
mx.com.sifei.descargasat.ws.sdk-1.0-SNAPSHOT-uber.jar		Biblioteca java que es el SDK, debe ser importada en el proyecto para su uso.
Clase con ejemplo del uso del SDK		Clase con ejemplos de todos los métodos ofrecidos por el SDK

7.1.1 URL de producción SDK Java

Para configurar a producción el SDK basta con establecer la URL por medio de su setter, la constante ya se incluye dentro de la propia definición de la clase `DescargaSATSDK` ver Tabla 14:

Tabla 14. Ejemplo de URL de producción del SDK de JAVA.

```
DescargaSATSDK retro = new DescargaSATSDK();
retro.setUrl(DescargaSATSDK.BASE_PRODUCTION_URL);

retro.setToken(token);
```

7.2 SDK de PHP.

Se requiere como mínimo PHP 5.6. Se requiere tener habilitada la extensión CURL para ejecutar las peticiones. La Tabla 15 presenta la ficha técnica del SDK PHP.



Tabla 15. Ficha Técnica SDK PHP

Versión mínima	PHP	5.6 o superior(compatible con PHP 7.0 , 7.1 +)
Recursos ofrecidos		Descripción
Archivos PHP		SDK que permite y facilita el consumo del API.
Clase con ejemplo del uso del SDK (./Test/App.php)		Clase con ejemplos de todos los métodos ofrecidos por el SDK.
Clase loader		Archivo loader que debe ser incluido en los puntos de ejecución principal para facilitar la importación de clases PHP basado en el espacio de nombre (Namespace).
Dependencias Adicionales		Descripción
N/A		N/A

7.2.1 URL Producción SDK PHP

Para establecer a producción el SDK basta con establecer la URL por medio de su setter, la constante ya se incluye dentro de la propia definición de la clase `DescargaSATSDK` (ver Tabla 16).

Tabla 16. Estableciendo a producción SDK de PHP

```
$sdk = new DescargaSATSDK();
//establecemos el token
$sdk->setToken(self::$token);
$sdk->setURL(DescargaSATSDK::$BASE_PRODUCTION_URL);
```

7.3 SDK .NET

Es importante entender que el SDK hace uso interno de la clase `HttpRequest` propia de .NET, por tanto, se deben esperar todas las excepciones devueltas por este método. La versión mínima requerida es .Net 2.0 por lo que es altamente compatible, aun así, se aconseja utilizar versiones superiores.

Se incluye la DLL:

- DHF.Sifei.DescargaSAT.Ws.SDK.dll

Esta DLL contiene el código necesario para consumir el API de ACI DescargaSAT y debe ser agregada como referencia. Adicionalmente se necesitan 2 dependencias que pueden instalarse mediante el administrador de paquetes NuGet:

1. Newtonsoft.Json
2. DotNetZip



A continuación, en Tabla 17 se muestra la información técnica del SDK .NET:

Tabla 17. Ficha Técnica SDK .NET

Versión mínima	Net	2.0+ (se recomienda usar versiones superiores)
Recursos ofrecidos		Descripción
DHF.Sifei.DescargaSAT.Ws.SDK.dll		SDK que permite y facilita en consumo del API.
Clase con ejemplo del uso del SDK		Clase con ejemplos de todos los métodos ofrecidos por el SDK.
Dependencias Adicionales		Descripción
Newtonsoft.Json		Biblioteca para serializar y deserializar las peticiones JSON.
DotNetZip		Biblioteca para la gestión de archivos ZIP

7.3.1 Agregar DLL al proyecto.

Para incluirla se debe importar la DLL como una referencia en el proyecto .NET tal y como se muestra en la Figura 2:

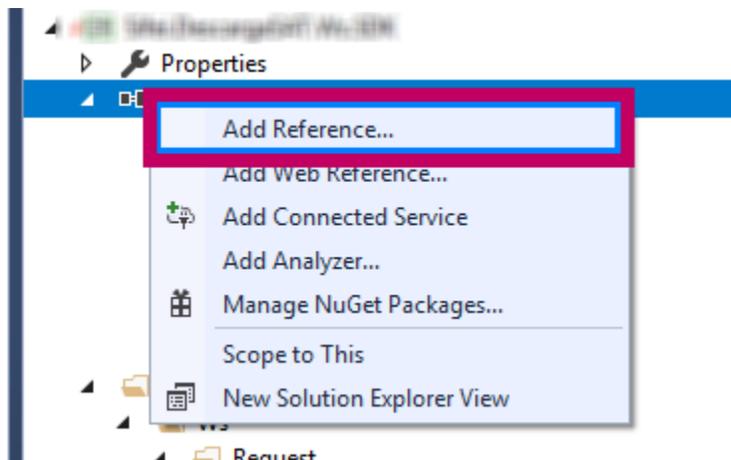


Figura 2. Agregando referencia

Tras ello se desplegará el explorador, se debe seleccionar la DLL e importar tal y como se ejemplifica en la Figura 3:



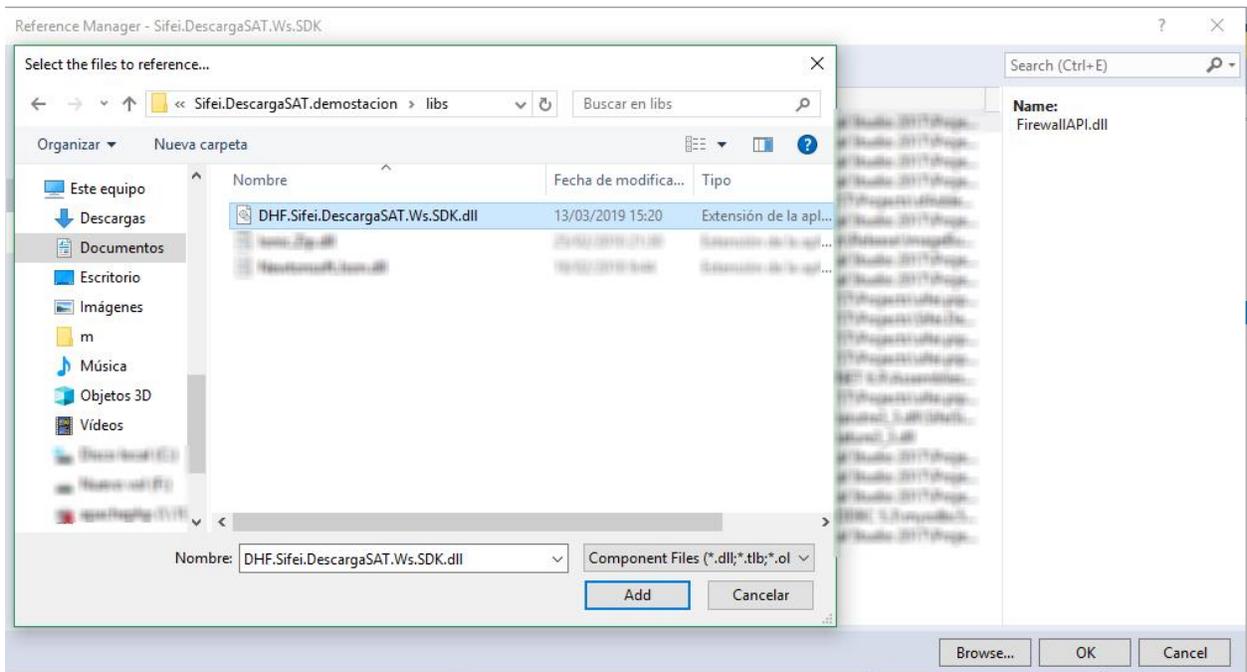


Figura 3 . Agregar DLL a proyecto

Para establecer a producción el SDK solo se requiere establecer la URL por medio de su propiedad (BaseURL), la constante "URL_BASE_PRODUCTION" ya se incluye dentro de la propia definición de la clase DescargaSATSDK.

7.3.2 URL Producción SDK .NET

En la siguiente Tabla 18 se muestra como establecer la URL de producción:

Tabla 18. Línea para volver a producción SDK .NET.

```
var sdk = new DescargaSATSDK();
sdk.Token = token;
//se setea la URL de produccion
sdk.BaseURL = DescargaSATSDK.URL_BASE_PRODUCTION;
```

8. Métodos Relacionados A Descargas

8.1 Crear una Descarga

Para crear la primera descarga de CFDI se debe consumir la siguiente URL como se muestra en la siguiente Tabla 19:



Tabla 19. URL para la creación de descarga.

URL	MÉTODO	Descripción
/api/v2/descargasatsifei/DescargaProgramada	POST	Permite crear una descarga(programar una descarga)

ACI DescargaSAT divide a nivel interno las descargas por días, es decir, si se envía una solicitud de un año se crearan 365 descargas. Por esta misma razón este método devuelve el arreglo de UUID de descargas creadas que deben ser verificadas por medio del método de consulta de descarga. Otro punto que se debe considerar es que ACI DescargaSAT evita las descargas duplicadas, por lo que si se intenta crear una descarga que colisione parcialmente con otra previa se devolverá un arreglo con los UUID de las ya existentes. Si la colisión es total es decir, fecha inicial y final son idénticas, este método devolverá un error.

8.1.1 Request

A continuación en la Tabla 20 se listan los campos esperado por el servicio:

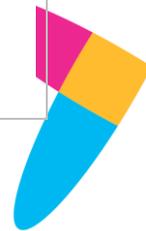
Tabla 20. Campos esperados en el servicio para crear una Descarga.

Parámetro	Tipo	Requerido	Descripción
FechaInicial	datetime	SI	Fecha inicial, formato "YYYY-MM-DD hh:mm:ss"
FechaFinal	datetime	SI	Fecha Final formato "YYYY-MM-DD hh:mm:ss"
TipoDeDescarga	enum	SI	Tipo: Para este momento es solo CFDI

Ejemplo de petición (Tabla 21):

Tabla 21. Ejemplos de petición

<pre> POST /api/v2/descargasatsifei/DescargaProgramada HTTP/1.1 Host: descargasat.sifei.com.mx User-Agent: curl/7.58.0 Authorization: qpJHAYQLsdas89e1MI3j21301230912j3121 Content-Type: application/json Content-Length: 104 { "FechaInicial": "2019-02-28 10:55:35", "FechaFinal": "2019-03-01 10:55:35", "TipoDeDescarga": "CFDI" } </pre>
<pre> POST /api/v2/descargasatsifei/DescargaProgramada HTTP/1.1 Host: descargasat.sifei.com.mx User-Agent: curl/7.58.0 Authorization: qpJHAYQLsdas89e1MI3j21301230912j3121 Content-Type: application/json </pre>



```
Content-Length: 104
{
  "FechaInicial": "2018-02-28T10:55:35",
  "FechaFinal": "2018-03-01 10:55:35",
  "TipoDeDescarga": "CFDI"
}
```

Nota: A pesar de que se incluye el tiempo (horas minutos y segundos, estos no son tomados en cuenta debido a que ACI DescargaSAT tomará ese rango de fecha y generará descargas con días completos.

8.1.2 Response

La respuesta de este método se ve en la Tabla 22.

Tabla 22. Estructura de respuesta.

Campo	Tipo	Requerido	Descripción
status	enum	SI	Estatus de la petición
data	object	SI	Objeto de respuesta. Ver ejemplo de response
message	string	No	Tipo: Para este momento es solo CFDI
code	int	NO	

Data Response

En Tabla 23 se muestra los campos que incluye data:

Tabla 23. Estructura tipo data.

Campo	Tipo	Descripción
resultado	string	Descripción de la operación.
errores	Array[string]	Arreglo de errores
repetidos	Array[string]	Arreglo de UUID ya existentes (ACI DescargaSAT evita la duplicidad de descargas)
exitosos	Array[string]	UUID creados.

En la Tabla 24 se presenta un ejemplo de visualización de la respuesta al crear una descarga:

Tabla 24 Ejemplos del response

```
{
  "status": "success",
  "data": {
    "resultado": "Se program\u00f3 la descarga en d\u00edas de CFDI'S, numero de dias programados 3,numero de repetidos 0",
    "errores": [
    ],
  },
}
```



```

"exitosos":[
"20190228.00.00.00-2019022823.59.59C",
"20190301.00.00.00-2019030123.59.59C",
"20190228.10.41.15-2019030110.41.15C"
],
"repetidos":[

],
"diasTotalesDeDescarga":2,
"totalDiasNuevosProgramados":2,
"diasyprogramados":0,
"tipoDescargaName":"CFDI'S"
},
"message":"",
"code":""
}
    
```

Respuesta en caso de repetida

```

{
  "status": "fail",
  "data": null,
  "message": "Ya existe una descarga programada activa con las mismas
características, uuid: 20180228.10.55.35-2018030110.55.35C",
  "code": "2101"
}
    
```

8.1.3 Ejemplo con SDK de JAVA

A continuación se muestra el ejemplo de código en la Tabla 25:

Tabla 25. Ejemplo con SDK de JAVA para la creación de una descarga

```

//objeto tipo descarga que espera el WS.
DescargaProgramadaRequest descargaProgramadaRequest = new
DescargaProgramadaRequest();
//se establece la fecha inicial
descargaProgramadaRequest.setFechaInicial(new Date());
Date datefinal = new Date();
Calendar cd = Calendar.getInstance();
cd.add(Calendar.DATE, 1);
datefinal = cd.getTime();
//se establece la fecha final
descargaProgramadaRequest.setFechaFinal(datefinal);
    
```



```

descargaProgramadaRequest.setTipodeDescargaDescarga(TipoDescarga.CFDI); //establ
es el tipo
//se inicializa el SDK
DescargaSATSDK sdk = new DescargaSATSDK();
//se establece el token que es necesario para consumir
sdk.setToken(token);

try {
    //ahora se consume el servicio
    Response<WsResponseMaster<DataResponseDescargaProgramada>>
response=sdk.ProgramarDescarga(descargaProgramadaRequest);
//con el metodo isSuccessful se verificar si la peticion http fue correcta
if(response.isSuccessful()) {
    //si a nivel http fue correcto, entonces se deserializa el body:
    WsResponseMaster<DataResponseDescargaProgramada> body=response.body();
    //en el body verificamos que con isSuccess que la OPERACION fue correcta.
    if(body.isSuccess()) {
        //si status es success en el body entonces operar con data
        DataResponseDescargaProgramada data= body.getData();

    }else {
        //algo fue mal un codigo de error
        System.err.println("Codigo:"+body.getCode());
        System.err.println("Error:"+body.getMessage());
        //... haz lo que necesites
    }

}else { //codigo diferente de 200 a 300.

}

}catch(Exception e) {
    e.printStackTrace();
}
    
```

8.1.4 Ejemplo con SDK de PHP

Ejemplo de uso de SDK de PHP en Tabla 26:

Tabla 26. Ejemplo de consumo con el SDK de PHP para la creación de una descarga

```

$descarga= new DescargaProgramadaRequest();
#notar que no es necesario indicar el tiempo.
$descarga->setFechaInicial( new \DateTime("2018-12-01"));
$descarga->setFechaFinal( new \DateTime("2018-12-07"));
    
```



```

$sdk= new DescargaSATSDK();
//establecemos el token
$sdk->setToken(self::$token);
try{
$response=$sdk->ProgramarDescarga($descarga);
if($response->isSuccessful()){
$body=$response->getBody();
if($body->isStatusSuccess()){
//en este caso data es un arreglo compuestos de 4 propiedades o elementos
$data=$body->getData();
#mensaje-descripcion de la operacion (string)
echo $data->resultado;

exitosos: arreglo de uuid creados, estos pueden guardarse para su posterior
*verificacion(consumir el servicio de consula detalle para saber si su estado ya
esta en completado)
*/
    print_r($data->exitosos);
#repetidos: arreglo de uuid de descargas previamente realizadas
print_r($data->repetidos);
#diasTotalesDeDescarga: cantidad de dias entre la fecha inicial y final

echo "diasTotalesDeDescarga:". $data->diasTotalesDeDescarga. "\n";
#totalDiasNuevosProgramados: cantidad de dias nuevos(efectivamente añadidos)
echo "totalDiasNuevosProgramados:". $data->totalDiasNuevosProgramados. "\n";
#diasyaprogramados : cantidad de dias ya programados

echo "diasyaprogramados:". $data->diasyaprogramados. "\n";
#tipoDescargaName
echo "tipoDescargaName:". $data->tipoDescargaName. "\n";
}else{
    //consumo invalido o descarga ya existente, codigo y mensaje del API
    echo "codigo: ". $body->getCode(). "\n";
    echo "message: ". $body->getMessage(). "\n";
}

}else{
    echo $response->getRawBody();
}
}catch(Exception $e){
    print_r($e);
}
    
```

8.1.5 Ejemplo con SDK de .NET

Ejemplo de uso de SDK de .NET en Tabla 27:

Tabla 27. Ejemplo con el SDK de .Net para la creación de una descarga.

```

var sdk = new DescargaSATSDK();
sdk.Token = token;

var paramerters = new DescargaProgramadaRequest();
paramerters.FechaInicial = new DateTime(2018, 12, 1);
paramerters.FechaFinal = new DateTime(2018, 12, 5);
//nota: las descargas de CFDI son divididas siempre por DIAS
//debido a esto al crearse una descarga el servicio devuelve un uuid por cada
descarga-dia, es decir, si se envia un rango de 30 dias, el ws retornara el uuid
de las descargas diarias. Esto permite evitar descargar mas de una vez un mismo
dia.

var wsResponse = sdk.ProgramarDescarga(paramerters);

if (wsResponse.IsStatusSuccess()) {
    var data = wsResponse.Data;
    Console.WriteLine(data.Resultado);
    Console.WriteLine(data.Exitosos);
    //descargas efectivamente creadas
    foreach (var uuidgenerado in data.Exitosos) {
        Console.WriteLine("Se creo descarga con uuid: {0} , apartir de este comento
se debera consumir el servicio de consulta descarga para saber si ya fue
completado", uuidgenerado);
    }
    //descargas diarias ya existentes:
    foreach (var uuidYaExistente in data.Repetidos) {
        Console.WriteLine("Ya existe una descarga con el uuid:{0}, por lo que no se
volvera a descargar");
    }
    //lista de errores (si es que hubo)
    foreach (var error in data.Errores) {

        Console.WriteLine("Error :{0}", error);
    }
}
else {
    //puede deberse a:
    /**
    * Que el rango de la descarga ya existe(descarga duplicada)
    * Para mas detalle ver tabla de errores.
  
```



```

*
*/

    Console.WriteLine(string.Format("Codigo de error:{0}", wsResponse.Code));
    Console.WriteLine(string.Format("Mensaje de error:{0}", wsResponse.Message));

}
Console.ReadLine();
    
```

8.2 Consultar Descarga Por UUID

Este método no necesita un *body*, el parámetro es el UUID de la Descarga y forma parte de la URL.

8.2.1 Request

A continuación se muestra la información de la solicitud a enviar en la Tabla 28:

Tabla 28. URL para la consulta de una descarga

URL	MÉTODO	Descripción
/api/v2/descargasatsifei/DescargaProgramada/{uuid}	@GET	Permite obtener el detalle de una descarga, esto incluye el estado, fechas, y uuid entre más.
URL parámetros (deben sustituirse al final de la URL)		
Parámetro	Tipo	Descripción
@uuid	string	UUID de la descarga

Se presenta un ejemplo de petición para la consultar una descarga en la Tabla 29:

Tabla 29. Petición consultar descarga

```

GET /api/v2/descargasatsifei/DescargaProgramada/20181016.00.00.00-2018101623.59.59 HTTP/1.1
Authorization: qpJHAYQLsdas89e1MI3j21301230912j3121
Host: descargasat.sifei.com.mx
Connection: Keep-Alive
Accept-Encoding: gzip
User-Agent: okhttp/3.12.0
    
```

8.2.2 Response

En este caso data es de tipo DescargaProgramadaDetalle, tal y como se muestra en la Tabla 30:



Tabla 30. Indicando el tipo de dato del atributo data.

Data campos	Tipo	Descripción
Data	DescargaProgramadaDetalle	Objeto de tipo Descarga detalle.

8.2.2.1 Tipo DescargaProgramadaDetalle

El tipo DescargaProgramadaDetalle esta compuesto por lo campos listados que se muestran en la Tabla 31:

Tabla 31. Descripción de los campos del detalle de la descargaProgramada.

Data campos	Tipo	Descripción
<code>uuid</code>	Object	Objeto de tipo Descarga detalle.
<code>fechaInicial</code>	datetime	Fecha inicial de la descarga
<code>fechaFinal</code>	datetime	Fecha final de la descarga
<code>fechaOrigenProgramacion</code>	datetime	Fecha que se creó la descarga
<code>cfdiDescargados</code>	int	Numero de CFDI descargados
<code>ultimaEjecucion</code>	datetime	Fecha en que se ejecutó.
<code>metodoDescarga</code>	int	Indica si fue por portal u oficial.
<code>numeroCfdiEmitidos</code>	Int	Numero de CFDI Emitidos
<code>numeroCfdiRecibidos</code>	int	Numero de CFDI Recibidos
<code>tipoDeDescarga</code>	int	Tipo descarga

En la Tabla 32 se muestra un ejemplo de respuesta a la petición.

Tabla 32 .Ejemplo de respuesta

```
{
  "status": "success",
  "data": {
    "uuid": "20181016.00.00.00-2018101623.59.59",
    "fechaInicial": "2018-10-16 00:00:00",
    "fechaFinal": "2018-10-16 23:59:59",
    "estado": "EN PROCESO",
    "fechaOrigenProgramacion": "2018-10-30 19:03:05",
    "cfdiDescargados": null,
    "ultimaEjecucion": "2018-10-30 07:10:27",
    "metodoDescarga": 2,
    "metodoDescargaString": "Oficial",
    "tipoDeDescarga": 1
  },
  "message": "",
  "code": ""
}
```



8.2.3 Ejemplo SDK JAVA

Ejemplo de consumo mediante SDK de Java se muestra en la Tabla 33:

Tabla 33. Ejemplo de consumo con el SDK de JAVA para la consulta de una descarga por UUID

```
//instanciacion de SKD y asignacion de token
DescargaSATSDK sdk = new DescargaSATSDK();
sdk.setToken(token);
try {

//invocando-consumiendo servicio
    Response<WsResponseMaster<DataDescargaProgramadaDetail>> response =
    sdk.obtenerDetalleDescarga("20181016.00.00.00-2018101623.59.59");
//se verifica la operacion,si es correcta con body() se deserializa la respuesta:
if(response.isSuccessful() ) {
WsResponseMaster<DataDescargaProgramadaDetail> des= response.body();
//verificamos el estado de la operacion y si es success hacemos lo que necesite
if (des.isStatusSuccess()) {
System.out.println("ok");
DataDescargaProgramadaDetail detalle = des.getData();
System.out.println(detalle.getEstado());//estado de la descarga.
}else {
System.err.println(des.getMessage());
}
}
} catch (IOException e) {
e.printStackTrace();
} catch (Exception e) {
e.printStackTrace();
}
```

8.2.4 Ejemplo con SDK de PHP

En la Tabla 34 se ejemplifica el consumo del servicio mediante el uso del SDK de PHP:

Tabla 34. Ejemplo de consumo con el SDK de PHP para la consulta de una descarga por UUID.

```
$sdk= new DescargaSATSDK();
//establecemos el token
$sdk->setToken(self::$token);

try{
#le asignamos un uuid de la descarga a consultar.
$response=$sdk->ConsultarDescarga(self::$descargaUUID);
```



```

if($response->isSuccessful()){
$body=$response->getBody();
#si es succes entonces se encontro y se puede obtener por getData
if($body->isStatusSuccess()){
$data=$body->getData();
print_r($data);
}else{ #error
if($body->isCodeNoEncontrado()){
    echo "No existe la descarga con el UUID";
    echo "codigo: ".$body->getCode()."\n";
    echo "message: ".$body->getMessage()."\n";
}else{#otro error:
//consumo invalido, codigo y mensaje del API
    echo "codigo: ".$body->getCode()."\n";
    echo "message: ".$body->getMessage()."\n";
}
}
}else{
#imprimomos el error inesperado del servidor.
echo $response->getRawBody();
}

}catch(Exception $e){
#en caso de un error de red, de dns, de puerto van, asi mismo reglas de validacion
del SDK
#sin embargo una vez que validez tu logica las excepciones orginanidas por la
validacion del SDK dejaran de ocurrir.
echo "Exception:\n";
print_r($e);
}
    
```

8.2.5 Ejemplo con SDK de .NET

En la Tabla 35 se ejemplifica el consumo mediante el uso del SDK:

Tabla 35. Ejemplo con el SDK de .Net para la consulta de una descarga por UUID.

```

var sdk = new DescargaSATSDK();
sdk.Token = token;
WsResponseMaster<DataDescargaProgramadaDetail> wsResponse =
sdk.ObtenerDetalleDescarga(descargaUUID);
if (wsResponse.IsStatusSuccess()) {
    DataDescargaProgramadaDetail data = wsResponse.Data;
    Console.WriteLine(data.Uuid);
    Console.WriteLine(data.FechaInicial);
}
    
```



```

Console.WriteLine(data.FechaFinal);
Console.WriteLine(data.Estado);
Console.WriteLine(data.UltimaEjecucion);
Console.WriteLine(data.FechaOrigenProgramacion);
if (data.isEstadoCompletado()) {
//solicita la descarga de los cfdi en ese rango
Console.WriteLine("Esta completo, solicitar CFDIs");
}
else if (data.isEstadoEnProceso()) {
//hay que volver a consultar en X tiempo (se recoeminda cada hora o mas
Console.WriteLine("En proceso");
}
else if (data.isEstadoERROR()) {
//hubo un error
Console.WriteLine("En Error");
}
else if (data.isEstadoEsperando()) {
//aun no ha sido atendida
Console.WriteLine("Esperando");
}
else if (data.isEstadoBloqueado()) {
//ya no tienes mas folios en tu paquete
Console.WriteLine("bloqueado, necesita reactivar paquete");
}
}
else {
if (wsResponse.isNotFound()) {
Console.WriteLine(string.Format("No se contro:{0}", wsResponse.Code));

}
else { //otro tipo de error:
Console.WriteLine(string.Format("Codigo de error:{0}", wsResponse.Code));
Console.WriteLine(string.Format("Mensaje de error:{0}",
wsResponse.Message));
}
}
Console.Write(wsResponse);
Console.ReadLine();

```

8.3 Reactivar Descarga por UUID

Es posible que una descarga falle, cuando esto ocurre el estado de la descarga cambiará a ERROR, así mismo si se llega a terminar el folio del paquete, las descargas que aún no han sido procesadas adquirirán un



estado de bloqueada, por lo que este método permitirá reactivar esta descarga para volver a intentar Descargar tus CFDI.

8.3.1 Request

En la Tabla 36 se muestra la información del método para reactivar descargas por UUID:

Tabla 36. URL para la reactivación de una descarga.

URL	MÉTODO	Descripción
/api/v2/descargasatsifei/DescargaProgramada/{uuid}	@POST	Permite reactivar una Descarga con estado de ERROR, BLOQUEADA, INCOMPLETA. Si la descarga no tiene uno de estos estados, se devolverá un status fail y además de un código de error "2102", ver <i>Código de error</i> En caso de no existir la descarga se devolver un not found(404).
URL parámetros (deben sustituirse al final de la URL)		
Parámetro	Tipo	Descripción
@uuid	string	UUID de la descarga

En la Tabla 37 se muestra en ejemplo de la petición:

Tabla 37 Ejemplo de petición

```
POST /api/v2/descargasatsifei/DescargaProgramada/20150101.00.00.00-2015010123.59.59C HTTP/1.1
Authorization: qpJHAYQLsdas89e1MI3j21301230912j3121
Content-Length: 0
Host: descargasat.sifei.com.mx
```

8.3.2 Response

En esta respuesta el tipo de dato de data es string, tal y como se muestra en la Tabla 38:

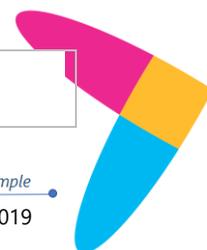
Tabla 38. Tipo de dato de data en la respuesta del método de reactivación.

Data campos	Tipo	Descripción
Data	string	En caso exitoso. Mensaje descriptivo de la reactivación de la descarga.

A continuación en la Tabla 39 se muestra un ejemplo de respuesta de método de reactivación de descarga por UUID.

Tabla 39. Ejemplo respuesta método de reactivación de descarga por UUID.

```
{
  "status": "success",
```



```
"data": "Se reactivo la descarga programada con UUID:20150101.00.00.00-2015010123.59.59C",
"message": "",
"code": ""
}
```

Nota: se recomienda consumir el servicio de consulta de descarga si se quiere corroborar los cambios de estado.

8.3.3 Ejemplo SDK de JAVA

La Tabla 40 incluye un ejemplo de consumo utilizando el SDK de Java.

Tabla 40. Ejemplo de consumo con el SDK de JAVA para la reactivación de descarga por UUID.

```
DescargaSATSDK sdk = new DescargaSATSDK();
//habilitar log para testing:
sdk.setLog(true);
sdk.setToken(token);
try {
//se consume el ws pasando el UUID de la descarga a reactivar
Response<WsResponseMaster<String>> response=
sdk.ReactivarDescarga("20150101.00.00.00-2015010123.59.59C");
if(response.isSuccessful()) {
WsResponseMaster<String> body=response.body();
if(body.isStatusSuccess()) {
//en este caso es un mensaje de confirmacion
String data=body.getData();
//haz lo que necesites hacer ...
}else {
System.err.println(response.body().getMessage());
}
}

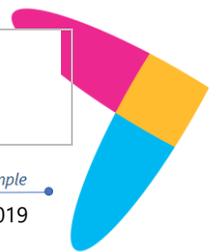
}catch(Exception e) {
e.printStackTrace();
}
```

8.3.4 Ejemplo con SDK de PHP

La Tabla 41 incluye un ejemplo de consumo utilizando el SDK de PHP para reactivar una descarga.

Tabla 41. Ejemplo de consumo con el SDK de PHP para la reactivación de descarga por UUID.

```
$sdk= new DescargaSATSDK();
//establecemos el token
$sdk->setToken(self::$token);
```



```

try{
#le asignamos un uuid de la descarga a consultar.
$response=$sdk->ReactivarDescarga(self::$descargaUUID);
if($response->isSuccessful()){
    $body=$response->getBody();
    #si es succes entonces se reactivo
    if($body->isStatusSuccess()){
        #data en este caso solo es un string conteniendo un mensaje de ok.
        $data=$body->getData();
        echo($data);
    }else{ #error
        if($body->isCodeNoEncontrado()){
            echo "No existe la descarga con el UUID para ser reactivada";
            echo "codigo: ".$body->getCode()."\n";
            echo "message: ".$body->getMessage()."\n";
        }else{#otro error:
            //consumo invalido, codigo y mensaje del API
            echo "codigo: ".$body->getCode()."\n";
            echo "message: ".$body->getMessage()."\n";
        }
    }
}else{
    #imprimomos el error inesperado del servidor.
    echo "ERROR:\n";
    echo $response->getRawBody();
}
}catch(Exception $e){
#en caso de un error de red, de dns, de puerto van, asi mismo reglas de
validacion del SDK
#sin embargo una vez que validez tu logica las excepciones orginadas por la
validacion del SDK dejaran de ocurrir.
echo "Exception:\n";
print_r($e);
}
    
```

8.3.5 Ejemplo con SDK de .NET

La siguiente Tabla 42 incluye un ejemplo de consumo utilizando el SDK de .Net para la reactivación de una descarga por medio de su UUID:

Tabla 42. Ejemplo con el SDK de .Net para la reactivación de descarga por UUID

```

var sdk = new DescargaSATSDK();
sdk.Token = token;
    
```

```

WsResponseMaster<string> wsResponse = sdk.ReactivarDescarga(descargaUUID);
if (wsResponse.IsStatusSuccess()) {
    //data en este caso solo es un string con el mensaje de exito, mientras
    estatus sea success significa que se reactivo
    Console.WriteLine(wsResponse.Data);
}
else {
    Console.WriteLine(string.Format("Codigo de error:{0}", wsResponse.Code));
    Console.WriteLine(string.Format("Mensaje de error:{0}",
wsResponse.Message));
}
Console.ReadLine();
    
```

8.4 Obtener descargas diarias

8.4.1 Request

La URL relativa y el método se describen en la Tabla 43 y los parámetros en formato JSON que espera recibir el método se describe en la Tabla 44.

Tabla 43. URL para la obtención de descargas diarias.

URL	METODO	Descripción
/api/v2/descargasatsifei/DescargaProgramadas/query	@POST	Permite consultar los UUID de todas las descargas que son diarias (con tiempo en 00:00:00 y 23:59:59).

Tabla 44. Parametros de consumo.

Parámetro	Tipo	Requerido	Descripción
estadoDescarga	string	SI	Estado de la descarga Programada ,
fechaInicial	Datetime?	Opcional	Fecha inicial, este valor de consulta se compara con respecto al valor de fecha inicial de la descarga, si se establece, se consultarán las descargas cuya fecha inicial sea mayor a este (inicio de rango, límite inferior de fecha). "YYYY-MM-DD hh:mm:ss","YYYY-MM-DDThh:mm:ss"
fechaFinal	Datetime?	Opcional	Fecha final, este valor de consulta se compara con respecto al valor fecha final de la descarga, si se establece, se consultarán las descargas cuya fecha final sea menor a este parámetro (fin de rango, límite superior de fecha). "YYYY-MM-DD hh:mm:ss","YYYY-MM-DDThh:mm:ss"
tipoDescarga	Int?	Opcional	Tipo de recursos solicitado en la descarga. 1= CFDI, 2 = META
metodoDescarga	Int?	Opcional	Método actual de la descarga, 1= Portal, 2= WS oficial.

La siguiente Tabla 45 incluye un ejemplo de la solicitud al servicio de consulta.



Tabla 45 Ejemplos de petición para la consulta de descargas.

<pre> POST /api/v2/descargasatsifei/DescargaProgramadas/query HTTP/1.1 Authorization: qpJHAYQLsdas89e1MI3j21301230912j3121 Content-Type: application/json; charset=UTF-8 Content-Length: 31 Host: descargasat.sifei.com.mx Connection: Keep-Alive Accept-Encoding: gzip User-Agent: okhttp/3.12.0 { "estadoDescarga": "COMPLETADO" } </pre>
<pre> POST /api/v2/descargasatsifei/DescargaProgramadas/query HTTP/1.1 Host: descargasat.sifei.com.mx User-Agent: curl/7.58.0 Authorization: qpJHAYQLsdas89e1MI3j21301230912j3121 Content-Type: application/json Content-Length: 59 { "estadoDescarga": "*", "fechaInicial": "2017-01-05T12:00:00" } </pre>
<pre> POST /api/v2/descargasatsifei/DescargaProgramadas/query HTTP/1.1 Host: descargasat.sifei.com.mx User-Agent: curl/7.58.0 Authorization: qpJHAYQLsdas89e1MI3j21301230912j3121 Content-Type: application/json Content-Length: 78 { "estadoDescarga": "*", "fechaFinal": "2017-01-05 12:00:00", "metodoDescarga": "2" } </pre>
<pre> POST /api/v2/descargasatsifei/DescargaProgramadas/query HTTP/1.1 Host: descargasat.sifei.com.mx User-Agent: curl/7.58.0 Authorization: qpJHAYQLsdas89e1MI3j21301230912j3121 Content-Type: application/json Content-Length: 76 { "estadoDescarga": "*", </pre>



```

"fechaFinal": "2017-01-05 12:00:00",
"tipoDescarga": "1"
}

```

```

POST /api/v2/descargasatsifei/DescargaProgramadas/query HTTP/1.1
Authorization: qpJHAYQLsdas89e1MI3j21301230912j3121
Content-Type: application/json;charset=utf-8
Host: descargasat.sifei.com.mx
Content-Length: 152
Connection: Keep-Alive
{
  "estadoDescarga": "COMPLETADO",
  "tipoDescarga": 1,
  "metodoDescarga": null,
  "fechaInicial": "2018-12-01T00:00:00",
  "fechaFinal": null
}

```

8.4.2 Response

Data Response es un arreglo (colección) que contiene en cada uno de sus elementos la siguiente estructura (ver Tabla 46).

Tabla 46. Data es una colección de tipo DadaDescargaProgramada.

Data campos	Tipo	Descripción
Data	Array de tipo DataDescargaProgramada	Arreglo de tipo DataDescargaProgramada.

8.4.3 Tipo de dato DataDescargaProgramada

Este tipo de dato se compone de las propiedades listadas en la Tabla 47.

Tabla 47. Descripción del tipo de dato devuelto por data.

Data campos	Tipo	Descripción
uuid	string	UUID de la descarga.
fechaInicial	datetime	Fecha Inicial de la descarga
fechaFinal	datetime	Fecha final de la descarga
tipoDescarga	Integer(enum)	Enum con valores 1=CFDI, se deja abierto a que otros valores enteros se incluyan en el dominio.

Data campos	Tipo	Descripción
estado	string	Estado de la descarga, tipo string, valores posibles: "COMPLETADO", "ERROR", "ESPERANDO" "EN PROCESO", "INCOMPLETA", "BLOQUEADA". Para más información sobre cada estado favor de referirse al Manual de Usuario de ACI DescargaSAT ya que son los mismos estados.
metodoDescarga	int	Método de la descarga , 1=Portal, 2=oficial,ws

A continuación se muestra un ejemplo de respuesta que devuelve el servicio (ver Tabla 48), como se puede observar, data es un arreglo el cual incluye elementos cuyas propiedades corresponden a la superior Tabla 47.

Tabla 48. Ejemplo respuesta para la obtencion de descargas diarias

Respuesta cuando la petición es correcta. <pre>{ "status": "success", "data": [{ "uuid": "20180120.00.00.00-2018012023.59.59", "fechaInicial": "2018-01-20 00:00:00", "fechaFinal": "2018-01-20 23:59:59", "tipoDescarga": 1, "estado": "COMPLETADO" }, { "uuid": "20180122.00.00.00-2018012223.59.59", "fechaInicial": "2018-01-22 00:00:00", "fechaFinal": "2018-01-22 23:59:59", "tipoDescarga": 1, "estado": "COMPLETADO" }], "message": "", "code": "" }</pre>
Respuesta cuando Fecha (en este caso inicial) es invalida. <pre>{ "status": "fail", "data": null, "message": "Fecha inicial Invalida", }</pre>



<pre>"code": "2301" }</pre>
Respuesta cuando Tipo de descarga es inválida.
<pre>{ "status": "fail", "data": null, "message": "Tipo de descarga es invalido.", "code": "2301" }</pre>
Respuesta cuando Metodo de descarga es inválido.
<pre>{ "status": "fail", "data": null, "message": "Metodo de descarga invalido: dac.", "code": "2301" }</pre>

8.4.4 Ejemplo SDK de JAVA

La Tabla 49 incluye un ejemplo del uso del SDK de Java para el método de consulta de descargas.

Tabla 49. Ejemplo de consumo con el SDK de JAVA.

```
DescargaSATSDK retro = new DescargaSATSDK();
//retro.setUrl(DescargaSATSDK.BASE_PRODUCTION_URL);

retro.setToken(token);
try {
  ParametersDescargasDiarias para= new ParametersDescargasDiarias();
  Calendar c=Calendar.getInstance();
  c.set(2018,12,1);
  para.fechaInicial= c.getTime();
  para.tipoDescarga=TIPO_DESCARGA.CFDI;
  //para.estadoDescarga=DataDescargaProgramada.ESTADO_COMPLETADO;
  Response<WsResponseMaster<List<DataDescargaProgramada>>> response =
retro.obtenerDescargasDiarias(para);
  if(response.isSuccessful()) {
    //deserializacion de body
    WsResponseMaster<List<DataDescargaProgramada>> body = response.body();
    //verificacion de estado
    if (body.isStatusSuccess()) {
      List<DataDescargaProgramada> lst = body.getData();
      //se recorre la coleccion de descargas
      for (DataDescargaProgramada d : lst) {
```



```

    System.out.println(d.getUuid());
    System.out.println(d.getTipoDescarga());
  }
  }else {
System.err.println(body.getCode()+body.getMessage());
  }
  }else {
    System.err.println("Error http code:" + response.code());
    //obtenemos el body del error
    ResponseBody dc = response.errorBody();
    String error= Utils.parseErrorBody(dc);
    System.err.println(error);
  }
} catch (IOException e) {
    e.printStackTrace();
// POSIBLES EXEPCIONES:

    //javax.net.ssl.SSLHandshakeException:
sun.security.validator.ValidatorException: PKIX path building failed:
sun.security.provider.certpath.SunCertPathBuilderException: unable to find valid
certification path to requested target
    // javax.net.ssl.SSLException

    //java.net.SocketTimeoutException: Read timed out
}

```

8.4.5 Ejemplo con SDK de PHP

La Tabla 50 incluye un ejemplo del uso del SDK de PHP para el método de consulta de descargas.

Tabla 50. Ejemplo de consumo con el SDK de PHP.

```

$sdk = new DescargaSATSDK();
//establecemos el token
$sdk->setToken(self::$token);
try {
#le asignamos un uuid de la descarga a consultar.

$parameterDescargaDiarias = [
    'estadoDescarga' => DescargaProgramada::ESTADO_ALL,
    #'fechaInicial' => '2019-01-01 00:00:01',
    'fechaFinal' => '2017-01-05 12:00:00',
    'estadoDescarga' => DescargaProgramada::PROCESADA_POR_PORTAL,
];

```



```

$response = $sdk->ConsultarLasDescargasDiarias($parameterDescargaDiarias);
if ($response->isSuccessful()) {
    $body = $response->getBody();
    #si es succes entonces todo fue bien.
    if ($body->isStatusSuccess()) {
        /*
        *data en este caso data puede ser null si no hay descargas(en caso de ser
        nueva) o contiene un array con
        *los datos basicos de las descargas
        */
        $data = $body->getData();
        if (null == $data) {
            #aun no hay descargas
        } else {
            #print_r($data);
            foreach ($data as $descarga) {
                echo "uuid {$descarga->uuid}\n";
                if ($descarga->isEstadoCompletado()) {
                    echo "Completado\n";
                }
            }
        } else { #error
            //consumo invalido, codigo y mensaje del API
            echo "codigo: " . $body->getCode() . "\n";
            echo "message: " . $body->getMessage() . "\n";
        }
    } else {
        #imprimomos el error inesperado del servidor.
        echo "ERROR:\n";
        echo $response->getRawBody();
    }
} catch (Exception $e) {
    #en caso de un error de red, de dns, de puerto van, asi mismo reglas de
    validacion del SDK
    #sin embargo una vez que validez tu logica las excepciones orginanidas por la
    validacion del SDK dejaran de ocurrir.
    echo "Exception:\n";
    print_r($e);
}

```

8.4.6 Ejemplo con SDK de .NET

La Tabla 51 incluye un ejemplo del uso del SDK en .NET para el método de consulta de descargas.

Tabla 51. Ejemplo con el SDK de .Net para la obtención de descargas.

```

var sdk = new DescargaSATSDK();
sdk.Token = token;
//se setea la URL de produccion
// sdk.BaseURL = DescargaSATSDK.URL_BASE_PRODUCTION;
var parameters = new ParametersDescargasDiarias();
//se puede filtrar por estado de la descarga.
parameters.estadoDescarga = ESTADO.COMPLETADO;
parameters.tipoDescarga = TIPO_DESCARGA.CFDI;
parameters.fechaInicial = new DateTime(2018,12,01);

WsResponseMaster<List<DataDescargaProgramada>> wsResponse =
sdk.ConsultarLasDescargasDiarias(parameters);
if (wsResponse.IsSuccess()) {
  //obtenermos el array de descargas:
  List<DataDescargaProgramada> lst = wsResponse.Data;

  foreach (var descarga in lst) {
    Console.WriteLine(descarga.Uuid);
    Console.WriteLine(descarga.FechaInicial);
    Console.WriteLine(descarga.FechaFinal);
    Console.WriteLine(descarga.Estado);
    Console.WriteLine(descarga.TipoDescarga);

    if (descarga.isEstadoCompletado()) {
      //lo que necesites...
    }
  }
}
else {
  Console.WriteLine(string.Format("Codigo de error:{0}", wsResponse.Code));
  Console.WriteLine(string.Format("Mensaje de error:{0}", wsResponse.Message));
}
Console.ReadLine();

```

9. Métodos relacionados CFDI

Una vez que se han descargado los CFDI a la bodega de CFDI de ACI DescargaSAT, se podrán realizar una serie de operaciones sobre ellos. A continuación se listan las operaciones de gestión y consulta de CFDI.



9.1 Obtener información CFDI Por UUID

Permite consultar la información de un CFDI por su UUID.

9.1.1 Request

Para el método de obtención de CFDI por UUID se debe consumir en la forma descrita por la Tabla 52.

Tabla 52 URL para obtener la información de un CFDI por medio de su UUID.

URL	METODO	Descripción
/api/v2/descargasatsifei/cfdi/query @uuid_cfdi	@GET	Permite consultar un CFDI por su UUID.
URL parámetros (deben sustituirse al final de la URL)		
Parámetro	Tipo	Descripción
@uuid_cfdi	string	UUID del CFDI

A continuación en la Tabla 53 se lista un ejemplo de solicitud al servicio, donde se muestra el método, url y el UUID.

Tabla 53. Ejemplo de petición para consulta de CFDI por UUID.

```
GET /api/v2/descargasatsifei/cfdi/query/E23DF780-96DC-4AF3-80AF-XXXXXXXXXX
HTTP/1.1
Authorization: qpJHAYQLsdas89e1MI3j21301230912j3121
Host: descargasat.sifei.com.mx
Connection: Keep-Alive
Accept-Encoding: gzip
User-Agent: okhttp/3.12.0
```

9.1.2 Response

En esta respuesta (Tabla 54) data es un arreglo de tipo "CFDI" (ver Tipo CFDI).

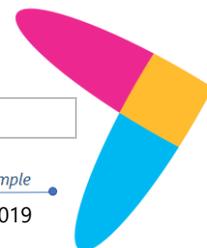
Tabla 54. Descripción de data el cual es un arreglo de tipo CFDI.

Campo	Tipo	Descripción
Data	Array de tipo CFDI	Array de tipo CFDI.

En el ejemplo de la Tabla 55 se muestra que "data" un arreglo (colección) compuesto por elementos de tipo CFDI (ver referencia: Tipo CFDI).

Tabla 55. Ejemplo respuesta para obtener un CFDI

```
{
```



```

    "status": "success",
    "data": {
      "uuid": "XXXXXXXX-96DC-XXXX-80AF-XXXXXX",
      "rfcEmisor": "AAA010101AAA",
      "rfcReceptor": "AAA010101AAA",
      "fechaDescarga": "2019-02-19 17:12:14",
      "total": "$568.40",
      "fechaEmision": "2019-01-02 19:33:15",
      "pacquecertifico": "AAA010101AAA",
      "fechaCertificacion": "2019-01-02 19:47:04",
      "solicitadaMetodo": "DESC",
      "estado": 1,
      "efecto": "I",
      "disponibleXML": true,
      "disponibleMETA": false
    },
    "message": "",
    "code": ""
  }

```

9.1.3 Ejemplo SDK de JAVA

La Tabla 56 ejemplifica el consumo del servicio mediante el SDK de Java.

Tabla 56. Ejemplo de consumo con el SDK de JAVA para la obtención de información de CFDI por UUID

```

DescargaSATSDK sdk = new DescargaSATSDK();
sdk.setLog(true);
sdk.setToken(token);
try {
  Response<WsResponseMaster<CFDIModelAPI>> response=
  sdk.ObtenerCFDIporUUID("XXXXX-96DC-4AF3-80AF-XXXXX");
  if(response.isSuccessful()){
    //deserializa body
    WsResponseMaster<CFDIModelAPI> body=response.body();
    if(body.isStatusSuccess()) {
      //todo bien, hago lo que necesite con el CFDI
      CFDIModelAPI cfdiInfo=body.getData();
      System.out.println(cfdiInfo.uuid);
    }
  }else{
    //error de conexion, porpias de la peticion
  }
}catch(Exception e) {
  e.printStackTrace();
}

```



9.1.4 Ejemplo con SDK de PHP

La Tabla 57 ejemplifica el consumo del servicio mediante el SDK de PHP para consulta de CFDI por UUID.

Tabla 57. Ejemplo de consumo con el SDK de PHP para la obtención de información de CFDI por UUID.

```

$sdk= new DescargaSATSDK();
//establecemos el token
$sdk->setToken(self::$token);
try{
#le asignamos un uuid de la descarga a consultar.
$response=$sdk->ObtenerCFDIPorUUID(self::$CFDIUUID);
if($response->isSuccessful()){
    $body=$response->getBody();
    #si es succes entonces todo fue bien.
    if($body->isStatusSuccess()){
        #en este caso data es la info de un CFDI
        $data=$body->getData();
        print_r($data);

    }else{ #error
if($body->isCodeNoEncontrado()){
    echo "No existe CFDI con el UUID ";
    echo "codigo: ".$body->getCode()."\n";
    echo "message: ".$body->getMessage()."\n";
}else{#otro error:
//consumo invalido, codigo y mensaje del API
    echo "codigo: ".$body->getCode()."\n";
    echo "message: ".$body->getMessage()."\n";
}
}
}else{
    #imprimomos el error inesperado del servidor.
    echo "ERROR:\n";
    echo $response->getRawBody();
}
}catch(Exception $e){
#en caso de un error de red, de dns, de puerto van, asi mismo reglas de validacion
del SDK
#sin embargo una vez que validez tu logica las excepciones orginanidas por la
validacion del SDK dejaran de ocurrir.
echo "Exception:\n";
print_r($e);
}
    
```



9.1.5 Ejemplo con SDK de .NET

La Tabla 58 ejemplifica el consumo del servicio mediante el SDK de PHP para consulta de CFDI por UUID.

Tabla 58. Ejemplo con el SDK de .Net para la obtención de información de CFDI por UUID.

```

var cli = new DescargaSATSDK();
//se establece el TOKEN
cli.Token = token;
//se establece el uuid del CFDI, ws ResponseMaster contendra la respuesta y se
parametriza el data de tipo CFDIModelAPI(ver referencia)
WsResponseMaster<CFDIModelAPI> wsResponse = cli.ObtenerCFDIPorUUID(cfdiUUID);
if (wsResponse.IsStatusSuccess()) {
    CFDIModelAPI cfdi = wsResponse.Data;

    Console.WriteLine(string.Format("UUID:{0}", cfdi.Uuid));
    Console.WriteLine(string.Format("FechaEmision:{0}", cfdi.FechaEmision));
    Console.WriteLine(string.Format("RfcEmisor:{0}", cfdi.RfcEmisor));
    Console.WriteLine(string.Format("RfcReceptor:{0}", cfdi.RfcReceptor));
    Console.WriteLine(string.Format("Total:{0}", cfdi.Total));
    Console.WriteLine(string.Format("Efecto:{0}", cfdi.Efecto));
    Console.WriteLine(string.Format("FechaCertificacion:{0}",
cfdi.FechaCertificacion));
    Console.WriteLine(string.Format("Pacquecertifico:{0}",
cfdi.Pacquecertifico));

    if (cfdi.EsSolicitadoPorXML() && cfdi.DisponibleXML) { //si es true y no en
cancelado se puede descargar.
        //invocar al metodo de descarga CFDI, o indicar en tu bd que esta
registrado,etc:
        // .....
    }
    else if (cfdi.EsSolicitadoPorXML() && cfdi.DisponibleXML == false) {
        //significa que el CFDI no ha sido descontado de tu paquete, por lo cual
deberas comprar un paquete para poder descargarlo

    }
    else if (cfdi.EsSolicitadoPorMeta()) {
        //solo se tiene el meta del CFDI(es decir el resumen de datos), se debe
solicitar la descarga de CFDI.(ver manual para entender mejor)

    }
}
else {
    if (wsResponse.IsNotFound()) {
        //no existe el cfdi.
    }
}
    
```



```

Console.WriteLine("No existe el CFDI");
}
else {
    ///otro tipo de error:
}
}
Console.ReadLine();
    
```

9.2 Descargar CFDI por UUID.

9.2.1 Request

Para este servicio será necesario colocar el UUID del CFDI solicitado en la URL tal y como se muestra en Tabla 59.

Tabla 59 URL para la descarga de un CFDI por su UUID.

URL	METODO	Descripción
/api/v2/descargasatsifei/cfdi/query @uuid_cfdi	@GET	Permite descargar un CFDI por su UUID.
URL parámetros (deben sustituirse al final de la URL)		
Parámetro	Tipo	Descripción
@uuid_cfdi	string	UUID del CFDI

A continuación se incluye un ejemplo de solicitud para la descarga de un CFDI por su UUID (ver Tabla 60):

Tabla 60 Ejemplo de petición

```

GET /api/v2/descargasatsifei/cfdi/download/E23DF780-96DC-4AF3-80AF-XXXXXXXXXXXX HTTP/1.1
Authorization: qpJHAYQLsdas89e1MI3j21301230912j3121
Host: descargasat.sifei.com.mx
Connection: Keep-Alive
Accept-Encoding: gzip
    
```

9.2.2 Response

En esta respuesta data es un string el cual contiene el XML codificado en base 64 (por lo cual debe ser decodificado), ver Tabla 61.

Tabla 61 Data de respuesta de de descarga de XML por uuid.

Campo	Tipo	Descripción
Data	string	XML codificado en base64



En la Tabla 62 se muestra un ejemplo de respuesta, se ha omitido el valor real de data para su fácil representación:

Tabla 62 Ejemplo respuesta de descarga de XML, data contiene una cadena codificada en base64.

```
{
  "status": "success",
  "data": "xmlcodificadoenbase64",
  "message": "",
  "code": ""
}
```

De esta respuesta el atributo data contiene el XML codificado en base64.

9.2.3 Ejemplo SDK de JAVA

A continuación se incluye un ejemplo utilizando el SDK de java para descargar el XML (ver Tabla 63):

Tabla 63. Ejemplo de consumo con el SDK de JAVA.

```
DescargaSATSDK retro = new DescargaSATSDK();
retro.setLog(true);
retro.setToken(token);
try {
  Response<WsResponseMaster<String>> response= retro.descargarCFDI(uuidCFDI);
  if(response.isSuccessful()){
    WsResponseMaster<String> body= response.body();
    if(body.isStatusSuccess()) {
      //todo bien, hago lo que necesite con el CFDI
      String XMLB64=body.getData();
      //escribir a archivo, enviar por correo, insertar en BD, etc
    }
  }else{
    //error de conexion, propias de la peticion
  }
}catch(Exception e) {
  e.printStackTrace();
}
```

9.2.4 Ejemplo con SDK de PHP

A continuación se incluye un ejemplo utilizando el SDK de PHP para descargar el XML (ver Tabla 64):

Tabla 64. Ejemplo de consumo con el SDK de PHP.

```
$sdk = new DescargaSATSDK();
//establecemos el token
```



```

$sdk->setToken(self::$token);
try {
#le asignamos un uuid de la descarga a consultar.
$response = $sdk->DescargarCFDIPorUUID(self::$CFDIUUID);
if ($response->isSuccessful()) {
    $body = $response->getBody();
    #si es succes entonces todo fue bien.
    if ($body->isStatusSuccess()) {
        #en este caso data es el XML codificado en base64
        $data = $body->getData();
        echo $data;
        $xml = base64_decode($data);
        echo $xml;

    } else { #error
        if ($body->isCodeNoEncontrado()) {
            echo "No existe CFDI con el UUID ";
            echo "codigo: " . $body->getCode() . "\n";
            echo "message: " . $body->getMessage() . "\n";
        } else { #otro error:
            //consumo invalido, codigo y mensaje del API
            echo "codigo: " . $body->getCode() . "\n";
            echo "message: " . $body->getMessage() . "\n";
        }
    }
} else {
    #imprimomos el error inesperado del servidor.
    echo "ERROR:\n";
    echo $response->getRawBody();
}
} catch (Exception $e) {
#en caso de un error de red, de dns, de puerto van, asi mismo reglas de validacion
del SDK
#sin embargo una vez que validez tu logica las excepciones orginandas por la
validacion del SDK dejaran de ocurrir.
echo "Exception:\n";
// Couldn't connect to server
print_r($e);
}
    
```

9.2.5 Ejemplo con SDK de .NET

A continuación se incluye un ejemplo utilizando el SDK de .NET para descargar el XML (verTabla 65):



Tabla 65. Ejemplo con el SDK de .Net para la descarga de CFDI por UUID.

```

var sdk = new DescargaSATSDK();
sdk.Token = token;
//se le pasa el uuid del CFDI a descargar. se parametriza data a tipo string
WsResponseMaster<string> wsResponse = sdk.DescargarCFDIPorUUID(cfdiUUID);
if (wsResponse.IsStatusSuccess()) {
    //data contiene el xml en base64
    var xml = wsResponse.Data;
    //se decodifica el XML para obtener los bytes y escribir:
    byte[] xmlBytes = Convert.FromBase64String(xml);
    //a manera de ejemplificacion se muestra el ejemplo para escribirlo:
    try {
        using (var fs = new FileStream(Path.Combine(@"F:\salidaTest\", cfdiUUID +
        ".xml"), FileMode.Create, FileAccess.Write)) {
            fs.Write(xmlBytes, 0, xmlBytes.Length);
        }
    }
    catch (Exception e) {
        Console.WriteLine(e.Message);
    }
}
else {
    if (wsResponse.isNotFound()) {
        //no se encontro el XML
    }
    else {
        //otro error:
    }
}
}

```

9.3 Consultar CFDI

El método Consultar CFDI permite obtener un resumen de todos los CFDI que cumplan un criterio de búsqueda.

9.3.1 Request.

La solicitud debe cumplir con la estructura de la Tabla 66:

Tabla 66 URL para la consulta (búsqueda) de CFDI.

URL	METODO	Descripción
/api/v2/descargasatsifei/cfdi/query	@POST	Este método permite descargar todos los CFDI que cumplan con el filtrado indicado.



A continuación en la Tabla 67 se describen los campos a enviar en la petición para la consulta de CFDI.

Tabla 67 Descripción de los campos a enviar en la petición

Data campos	Tipo	Requerido	Descripción
fechaInicial	Datetime	opcional	Fecha Inicial (inicio del rango)
fechaFinal	datetime	opcional	Fecha final (fin del rango)
rfcEmisor	string	opcional	RFC del receptor del CFDI
rfcReceptor	string	opcional	RFC del emisor del CFDI
uuidCFDI	string	opcional	UUID específico, si se usa este campo se ignoran los demás
efectoCFDI	string	opcional	Clave efecto
limit	int	requerido	Limit (número máximo de resultados)
offset	int	requerido	Desplazamiento de resultados (número de resultados que serán "saltados", ejemplo: Si se tiene 100 resultados y solo que quieren obtener los últimos 10(del 91 al 100) entonces el offset seria valor 91 y limit 10.

9.3.2 Request

En seguida en la Tabla 68 se incluye un ejemplo de la petición:

Tabla 68. Ejemplo de petición para la consulta de CFDI.

```
{
  "fechaFinal":null,
  "fechaInicial":"2017-12-01 02:15:35",
  "rfcEmisor":"AAA010101AAA",
  "rfcReceptor":null,
  "uuidCFDI":null,
  "efectoCFDI":"I",
  "limit":3,
  "offset":0
}
```

9.3.3 Response

El response incluye 2 campos especiales: 1) count que es el length del arreglo y 2) total, que es el número total de registros que entran en la consulta, la descripción de estos campos esta en la Tabla 69:



Tabla 69. Descripción de data para servicio de Consulta de CFDI.

Campo	Tipo	Descripción
total	string	Total de resultados encontrados.
count	int	Cantidad de resultados devueltos en petición. Este valor es el número de elementos devueltos y contenidos en el parámetro CFDI
cfdi	Array de tipo Tipo CFDI.	Arreglo de CFDI. Ver referencia: Tipo CFDI.

En el ejemplo de la Tabla 70 se muestra como data se compone de los 3 campos, total, count y cfdi, donde este último es una colección.

Tabla 70. Ejemplo respuesta Consulta de CFDI

```
{
  "status": "success",
  "data": {
    "total": "180",
    "count": 5,
    "cfdi": [
      {
        "uuid": "X-E67F-48E6-ADEF-XXXX",
        "rfcEmisor": "AAAAAA",
        "rfcReceptor": "AAAAAAA",
        "fechaDescarga": "2018-10-01 16:37:29",
        "total": "276.23",
        "fechaEmision": "2017-12-30 12:06:06",
        "pacquecertifico": "AAA010101AAA",
        "fechaCertificacion": "2017-12-30 12:04:47",
        "solicitadaMetodo": "BOTH",
        "estado": 1,
        "efecto": "I",
        "disponibleXML": true,
        "disponibleMETA": false
      },
      {
        "uuid": "7702B849-25F0-441D-AC93-XXXXXXXX",
        "rfcEmisor": "AAA010101AAA",
        "rfcReceptor": "AAA010101AAA",
        "fechaDescarga": "2018-10-01 16:37:28",
        "total": "276.23",
        "fechaEmision": "2017-12-30 12:04:50",
        "pacquecertifico": "AAA010101AAA",
        "fechaCertificacion": "2017-12-30 12:03:31",
        "solicitadaMetodo": "BOTH",
        "estado": 1,

```



```

"efecto": "I",
"disponibleXML": true,
"disponibleMETA": false
  },
  {
"uuid": "608DAFBA-F4FA-4A54-B17D-XXXXXX",
"rfcEmisor": "AAA010101AAA",
"rfcReceptor": "AAA010101AAA",
"fechaDescarga": "2018-10-01 16:37:29",
"total": "561.17",
"fechaEmision": "2017-12-30 12:03:56",
"pacquecertifico": "AAA010101AAA",
"fechaCertificacion": "2017-12-30 12:02:38",
"solicitadaMetodo": "BOTH",
"estado": 1,
"efecto": "I",
"disponibleXML": true,
"disponibleMETA": false
  },
  {
"uuid": "ADFC60CE-7338-4C3F-90BD-XXXXX",
"rfcEmisor": "AAA010101AAA ",
"rfcReceptor": "AAA010101AAA",
"fechaDescarga": "2018-10-01 16:37:27",
"total": "992.00",
"fechaEmision": "2017-12-30 12:02:18",
"pacquecertifico": "AAA010101AAA",
"fechaCertificacion": "2017-12-30 12:00:59",
"solicitadaMetodo": "BOTH",
"estado": 1,
"efecto": "I",
"disponibleXML": true,
"disponibleMETA": false
  },
  {
"uuid": "60010045-0D28-41DE-AB25-XXXXX",
"rfcEmisor": "AAA010101AAA ",
"rfcReceptor": "AAA010101AAA",
"fechaDescarga": "2018-10-01 16:37:27",
"total": "1,113.63",
"fechaEmision": "2017-12-30 11:59:59",
"pacquecertifico": "AAA010101AAA",
"fechaCertificacion": "2017-12-30 11:58:40",
"solicitadaMetodo": "BOTH",
"estado": 1,

```



```
"efecto": "I",
"disponibleXML": true,
"disponibleMETA": false
}
],
},
"message": "",
"code": ""
}
```

9.3.4 Ejemplo SDK de JAVA

Ejemplo de búsqueda de CFDi utilizando SDK de Java en Tabla 71.

Tabla 71. Ejemplo de consumo con el SDK de JAVA para la búsqueda de CFDI resguardados.

```
DescargaSATSDK sdk = new DescargaSATSDK();
sdk.setLog(true);
//token
sdk.setToken(token);
try {
//se instancia el objeto que permite establecer el criterio de busqueda
ParametersCFDIQuery params= new ParametersCFDIQuery();
//efecto es el mismo que de clave TipoCFDI:
params.setEfectoCFDI("I");
//si se quiere filtrar por RFC EMISOR
params.setRfcEmisor(rfcEmisor);
//numero de resultados que deseamos obtener(como maximo)
params.setLimit(3);
//
Calendar fechaInicial = Calendar.getInstance();
fechaInicial.add(Calendar.MONTH, -15);
//fecha inicial de busqueda.
params.setFechaInicial(fechaInicial.getTime());

//se setea el objeto query y se consume el ws
Response<WsResponseMaster<DataResponseCFDIConsulta>> response =
sdk.ConsultarCFDIDatos(params);
if (response.isSuccessful()) {// everything is ok
WsResponseMaster<DataResponseCFDIConsulta> body =response.body();
//cheamos el estaus
if (body.isStatusSuccess()) {
//obtenemos el data
DataResponseCFDIConsulta data=body.getData();
//este objeto alberga varios campos, el de mayor interes es la lista de CFDI
que consultamos
```



```

//asi que se extrae via getter y se recorre:
List<CFDIModelAPI> lstCFDI= data.getCfdi();
for(CFDIModelAPI cfdi : lstCFDI){
    System.out.println(cfdi.uuid);
}
//otros campos:
//total de cfdi existentes que cumplen el criterio:
System.out.println(data.getTotal());
//cantidad de cfdi devuelvtos en la peticion(delimitado por el limit de la
peticion)
System.out.println(data.getCount());

    }else {
System.err.println(response.body().getMessage());
}
} else {
// si algo fue mal en la respuesta y deseas ver lo que devuelve el server en la
peticion:
    System.err.println("Error http code:" + response.code());
//obtenemos el body
    ResponseBody dc = response.errorBody();
//se obtiene el stream stream y se convierte a String
    InputStream inputStream = dc.byteStream();
    String body = convert(inputStream, Charset.forName("utf-8"));
//imprimmos el error:
    System.err.println(body);
}

} catch (Exception e) {
e.printStackTrace();
}
    
```

9.3.5 Ejemplo con SDK de PHP

Ejemplo de búsqueda de CFDi utilizando SDK de PHP en Tabla 72:

Tabla 72. Ejemplo de consumo con el SDK de PHP para la búsqueda de CFDI resguardados.

```

$sdk = new DescargaSATSDK();
#token
$sdk->setToken(self::$token);
#se crea el objeto de consulta:
$params = new ParametersCFDIQuery();
#formato valido de fecha(EL SDK se encarga de enviar la fecha en el formato
correcto)
$params->setFechaFinal(new DateTime("2019-01-01"));
    
```



```

$params->setRfcReceptor($rfcReceptor);
#Indica el numero maximo de resultados a devolver
$params->setLimit(5);
#offset indica a partir de que elemento se iniciara la busqueda de CFDI
$params->setOffset(10);
#optionalmente le pondemos indicar que sean de tipo nomina
$params->setEfectoCFDI("N");

try {
#le asignamos un uuid de la descarga a consultar.
$response = $sdk->BuscarCFDI($params);
if ($response->isSuccessful()) {
    $body = $response->getBody();
    #si es succes entonces todo fue bien.
    if ($body->isStatusSuccess()) {
        # data es un objeto de tipo DataResponseCFDIConsulta
        $data = $body->getData();
        #count contiene el numero de elementos devueltos, el length del array de CFDI
        echo "Numero de cfdi devueltos: " . $data->count . "\n";
        # total es el numero total de CFDi que se encontraron, es un campo informativo
        echo "Total de resultados encontrados: " . $data->total . "\n";
        #si el arreglo es mayor a 0 entonces lo recorremos
        if ($data->count > 0) {
            foreach ($data->cfdi as $cfdi) {
                #cada elemento del arreglo es una clase CFDIModelAPI
                echo "-----\n";
                echo "uuid: " . $cfdi->uuid . "\n";
                echo "rfcEmisor: " . $cfdi->rfcEmisor . "\n";
                echo "rfcReceptor: " . $cfdi->rfcReceptor . "\n";
                echo "fechaDescarga: " . $cfdi->fechaDescarga . "\n";
                echo "total: " . $cfdi->total . "\n";
                echo "fechaEmision: " . $cfdi->fechaEmision . "\n";
                echo "pacquercertifico: " . $cfdi->pacquercertifico . "\n";
                echo "fechaCertificacion: " . $cfdi->fechaCertificacion . "\n";
                echo "estado: " . $cfdi->estado . "\n";
                echo "efecto: " . $cfdi->efecto . "\n";
            }
        }
    } else { #error
        if ($body->isCodeNoEncontrado()) {
            echo "No existe CFDI con el UUID ";
        }
        echo "codigo: " . $body->getCode() . "\n";
        echo "message: " . $body->getMessage() . "\n";
    }
}

```

```

    } else { #otro error:
    //consumo invalido, codigo y mensaje del API
        echo "codigo: " . $body->getCode() . "\n";
        echo "message: " . $body->getMessage() . "\n";
    }
}
} else {
    #imprimomos el error inesperado del servidor.
    echo "ERROR:\n";
    echo $response->getRawBody();
}
} catch (Exception $e) {
#en caso de un error de red, de dns, de puerto van, asi mismo reglas de validacion
del SDK
#sin embargo una vez que validez tu logica las excepciones orginandas por la
validacion del SDK dejaran de ocurrir.
echo "Exception:\n";
// Couldn't connect to server
print_r($e);
}
    
```

9.3.6 Ejemplo con SDK de .NET

Ejemplo de búsqueda de CFDi utilizando SDK en .NET en Tabla 73.

Tabla 73. Ejemplo con el SDK de .Net para la búsqueda de CFDI resguardados.

```

var sdk = new DescargaSATSDK();
sdk.Token =token;
var paramsN = new ParametersCFDIQuery();
//se puede establecer el efecto del CFDI
paramsN.EfectoCFDI = "I";
//se establece la fecha inicial
paramsN.FechaInicial = DateTime.ParseExact("2019-01-01", "yyyy-MM-dd",
System.Globalization.CultureInfo.InvariantCulture);
//se establece numero maximo de resultados a devolver:
paramsN.Limit = 3;
//se setean los parametros:
WsResponseMaster<DataResponseCFDIConsulta> wsResponse =
sdk.ConsultarCFDIDatos(paramsN);
//se verifica el status de la peticion.
if (wsResponse.IsStatusSuccess()) {
    //si todo fue bien entonces:
    Console.WriteLine(wsResponse.Data.cfdi);
    if (wsResponse.Data.cfdi.Count > 0) {
    
```



```

//hubo resultados
foreach (var cfdi in wsResponse.Data.cfdi) {
    Console.WriteLine(cfdi.Uuid);
    Console.WriteLine(cfdi.RfcEmisor);
    Console.WriteLine(cfdi.RfcReceptor);
}
}
else { //error:
    Console.WriteLine(string.Format("Codigo de error:{0}", wsResponse.Code));
    Console.WriteLine(string.Format("Mensaje de error:{0}",
wsResponse.Message));
}
Console.ReadLine();
    
```

9.4 Descarga de CFDI por consulta.

Este *endpoint* permite descargar los CFDI resguardados por un rango de fecha, estos son devueltos en un ZIP codificado en base64 en el campo data.

9.4.1 Request

El método y URL que forman la petición para la descarga se describe en la siguiente Tabla 74:

Tabla 74. URL para la descarga de CFDI.

URL	METODO	Descripción
/api/v2/descargasatsifei/cfdi/download	@POST	Este método permite descargar todos los CFDI que cumplan con el criterio de búsqueda.

Así mismo los datos a enviar en la petición se describen en la siguiente Tabla 75:

Tabla 75. Campos esperados por el servicio.

Data campos	tipo		Descripción
fechaInicial	Datetime	requerido	Fecha Inicial (inicio del rango)
fechaFinal	datetime	requerido	Fecha final (fin del rango)
emitidoRecibido	int	requerido	Indica si es recibido , emitido
efectoCFDI	string	opcional	Clave de efecto de CFDI: I,E,N,P,T



Ejemplo JSON de petición en la siguiente Tabla 76:

Tabla 76. Ejemplo de petición.

```
{
  "emitidoRecibido": -1,
  "efectoCFDI": "I",
  "fechaFinal": "2019-04-01 03:08:24",
  "fechaInicial": "2018-12-01 03:08:24"
}
```

9.4.2 Response

La respuesta contiene en el campo data el archivo ZIP codificado en base 64, ver Tabla 77:

Tabla 77. Estructura de respuesta.

Campo	Tipo	Descripción
Data	string	Zip codificado en base 64, el cual contiene los XML.

A continuación se muestra un ejemplo de respuesta en la Tabla 78 :

Tabla 78. Ejemplo respuesta, donde data en una cadena en base64 del archivo ZIP con los XML.

```
{
  "status": "success",
  "data": "zipEnBase64",
  "message": "",
  "code": ""
}
```

En caso de que una petición no genere una descarga, se devolverá la siguiente respuesta que se presenta en la Tabla 79:

Tabla 79. Ejemplo respuesta en caso de error.

```
{
  "status": "fail",
  "data": null,
  "message": "No hay CFDIS para esta descarga ",
  "code": "2204"
}
```



```
}

```

9.4.3 Ejemplo SDK de JAVA

El ejemplo para la descarga de CFDI utilizando el SDK de java se encuentra en la siguiente Tabla 80:

Tabla 80. Ejemplo de consumo con el SDK de JAVA.

```
//se crea instancia del SDK
DescargaSATSDK sdk = new DescargaSATSDK();
sdk.setToken(token);
try {
//se creamos objeto que nos permitira establecer el criterio de CFDI a Descargar
ParametersDownloadCFDI cfdirequest = new ParametersDownloadCFDI();
//objetos fecha inicial y final
Calendar fechaInicial = Calendar.getInstance();
fechaInicial.add(Calendar.MONTH, -3);
Calendar fechaFinal = Calendar.getInstance();

fechaFinal.add(Calendar.MONDAY, 1);
cfdirequest.setFechaInicial(fechaInicial.getTime());
cfdirequest.setFechaFinal(fechaFinal.getTime());

//se indica si deseamos que sean los Emitidos o recibidos o ambos (consultar
Enumeracion )
cfdirequest.setEmitidoRecibido(EmitidoRecibido.Todos);
//opcionalemnte le podemos indicar el efecto
cfdirequest.setEfectoCFDI("I");

Response<WsResponseMaster<String>> response = sdk.DescargarCFDIS(cfdirequest);
System.out.println("Peiticon realizada");
//verficamos su fue exitosa, es exitosa si fue recibida por el server, entendida
y acepta
if(response.isSuccessful()) {
System.out.println("Exitosa(HTTP OK)");
//deserializamos la el body y se verifica que todo este bien
WsResponseMaster<String> result = response.body();
//verficamos a NIVEL DE API si la peticion fue correcta
if(result.isStatusSuccess()) {
/*se obtiene data, que para este metodo es de tipo String y que
contiene el archivo zip en base64, por ello se decodifica y se obtiene el arreglo
de bytes
*/
byte[] bytesZip = Base64.decodeBase64(result.getData());
//desde este punto se puede hacer lo que se desee, para ejemplificar lo
escribiremos a un archivo:
File file = new File("F:/tmp", "zip.zip");

```

```

FileOutputStream outputStream = new FileOutputStream(file);
//escribimos los bytes y ya terminado se cierra el outputStream.
outputStream.write(bytesZip, 0, bytesZip.length);
outputStream.close();

}else {
System.err.println(result.getMessage());
}
}else {
System.out.println("error");
}
}
}
catch (IOException e) {
e.printStackTrace();
} catch (Exception e) {
e.printStackTrace();
}
}
    
```

9.4.4 Ejemplo SDK de PHP

El ejemplo para la descarga de CFDI utilizando el SDK de java se encuentra en la siguiente Tabla 81:

Tabla 81. Ejemplo de consumo con el SDK de PHP.

```

$sdk = new DescargaSATSDK();
#token
$sdk->setToken(self::$token);
#se crea el objeto de consulta:
$params = new ParametersDownloadCFDI();
#fecha inicial es requerida
$params->setFechaInicial(new DateTime("2018-01-01"));

#fecha final es requerida en este metodo.formato valido de fecha(EL SDK se encarga
de enviar la fecha en el formato correcto)
$params->setFechaFinal(new DateTime("2019-01-03"));
#indicamos que sean emitidos por nuestro rfc
$params->setOrigenEmitidos();
#Indica el numero maximo de resultados a devolver

#optionalmente le pondemos indicar que sean de tipo nomina
$params->setEfectoCFDI("N");

try {
#le asignamos un uuid de la descarga a consultar.
    
```



```

$response = $sdk->DescargarCFDIConsulta($params);
if ($response->isSuccessful()) {
    $body = $response->getBody();
    #si es succes entonces todo fue bien.
    if ($body->isStatusSuccess()) {
        # data es un objeto de tipo DataResponseCFDIConsulta
        $data = $body->getData();
        echo $data;
        file_put_contents('archivo.zip', base64_decode($data));
    } else { #error
        if ($body->isCodeNoEncontrado()) {
            echo "No existen CFDI con en la consulta dada ";
            echo "codigo: " . $body->getCode() . "\n";
            echo "message: " . $body->getMessage() . "\n";
        } else { #otro error:
            //consumo invalido, codigo y mensaje del API
            echo "codigo: " . $body->getCode() . "\n";
            echo "message: " . $body->getMessage() . "\n";
        }
    }
} else {
    #imprimomos el error inesperado del servidor.
    echo "ERROR:\n";
    echo $response->getRawBody();
}
} catch (Exception $e) {
    #en caso de un error de red, de dns, de puerto van, asi mismo reglas de validacion
    del SDK
    #sin embargo una vez que validez tu logica las excepciones orginadas por la
    validacion del SDK dejaran de ocurrir.
    echo "Exception:\n";
    // Couldn't connect to server
    print_r($e);
}
    
```

9.4.5 Ejemplo con SDK de .NET

El ejemplo para la descarga de CFDI utilizando el SDK de java se encuentra en la siguiente Tabla 82:

Tabla 82. Ejemplo de Descarga de CFDI que ya están resguardados en ACI DescargaSAT con el SDK de .Net.

```

var sdk = new DescargaSATSDK();
sdk.Token = token;
    
```

```

//se instancia objeto que permite que establecer los criterios de seleccion de
CFDI
var parameters = new ParametersDownloadCFDI();
//se establece el efecto
parameters.EfectoCFDI = "N";
parameters.FechaInicial = new DateTime(2018, 01, 01); //requerido
parameters.FechaFinal = new DateTime(2018, 02, 01); //requerido

try {
    //se establecen los paramtros , en este caso data sera un string que
    contiene el archivo zip en b64
    WsResponseMaster<string> wsResponse = sdk.DescargarCFDIConsulta(parameters);
    if (wsResponse.IsStatusSuccess()) { //verifica el campos status sea success
        string zipXML = wsResponse.Data;
        //se decodifica para obtener los bytes.
        byte[] zipBytes = Convert.FromBase64String(zipXML);
        //haz lo que se necesite. por ejemplo extraer a un dir. el el SDK se incluye
        la clase Utils:
        int numeroXmlExtraidos=Utils.ExtraerArchivosZipAFolder(zipBytes,
@"F:\salidaTest");
        Console.WriteLine(string.Format("Se han extraido {0}
xmls", numeroXmlExtraidos));

    }
    else {
        Console.WriteLine(string.Format("Codigo de error:{0}", wsResponse.Code));
        Console.WriteLine(string.Format("Mensaje de error:{0}",
wsResponse.Message));
    }
}
catch (System.Net.WebException e) {
    Console.WriteLine(e);
}
catch (Exception e) {
    Console.WriteLine(e);
}
Console.ReadLine();
    
```

9.5 Existe CFDI en CFDI resguardos

Este método recibe un UUID de CFDI e indica si está o no resguardado en ACI DescargaSAT.



9.5.1 Request

EL método y URL para la verficiacion de CFDI en ACI DescargaSAT se describen en la Tabla 83:

Tabla 83. URL para la verificación de CFDI resguardado.

URL	METODO	Descripción
/api/v2/descargasatsifei/cfdi/exist/ @uuid_cfdi	@GET	Este método permite comprobar si existe el CFDI en ACI DescargaSAT.
URL parámetros (deben sustituirse al final de la URL)		
Parámetro	Tipo	Descripción
@uuid_cfdi	string	UUID del CFDI

A continuación se muestra un ejemplo de una petición en la Tabla 84:

Tabla 84. Ejemplo de petición.

```
GET /api/v2/descargasatsifei/cfdi/exist/1EE48BE4-203B-4452-8CCC-XXXXXXXXXX
HTTP/1.1
Authorization: qpJHAYQLsdas89e1MI3j21301230912j3121
Host: descargasat.sifei.com.mx
Connection: Keep-Alive
Accept-Encoding: gzip
User-Agent: okhttp/3.12.0
```

9.5.2 Response

En este caso la respuesta en su campo data indica si CFDI está resguardado.

Tabla 85. Descripción de data de response.

Campo	Tipo	Descripción
Data	bool	Indica si existe o no en ACI DescargaSAT.

Tabla 86 Ejemplo respuesta

```
{
  "status": "success",
  "data": true,
  "message": "",
  "code": ""
}
```

9.5.3 Ejemplo SDK de JAVA

El ejemplo para la verficiacion de existencia de CFDI en ACI DescargaSAT utilizando el SDK de Java en la Tabla 87:



Tabla 87. Ejemplo de consumo con el SDK de JAVA para la verificación de existencia de CFDI.

```

DescargaSATSDK sdk = new DescargaSATSDK();
sdk.setLog(true);
sdk.setToken(token);
try {
//se le pasa el UUID que se desea verificar si existe en ACI DescargaSAT
Response<WsResponseMaster<Boolean>> response = sdk.ExisteCFDI(uuidCFDI);
//verificamos si la peticion fue recibida:
if (response.isSuccessful()) {
//deserializamos el JSON y obtenemos el body
WsResponseMaster<Boolean> body=response.body();
//a nivel de API se verifica que todo fue bien
if (body.isStatusSuccess()) {
//para este servicio data es un booleano, true indica que esta resguardo,false
indica que no lo esta.
if (body.getData() == true) {
    System.out.println("Existe el cfdi");
} else {
    System.out.println("No existe el CFDI");
}
} else {
System.err.println(body.getMessage());
}
} else { //algo fue mal en la peticion
    System.err.println("Error http code:" + response.code());
    //obtenemos el body del error
    ResponseBody dc = response.errorBody();
    InputStream inputStream = dc.byteStream();
    //tomamos el input stream y lo escribimos a un String
    String body = convert(inputStream, Charset.forName("utf-8"));
    System.err.println(body);
}

} catch (Exception e) {
// TODO Auto-generated catch block
e.printStackTrace();
}

```

9.5.4 Ejemplo SDK de PHP

El ejemplo para la verificación de existencia de CFDI en ACI DescargaSAT utilizando el SDK de Java en la Tabla 88:



Tabla 88. Ejemplo de consumo con el SDK de PHP para la verificación de existencia de CFDI.

```

$sdk = new DescargaSATSDK();
//establecemos el token
$sdk->setToken(self::$token);
//le pasamos el UUID y ejecutamos el consumo del WS
$response = $sdk->ExisteCFDI(self::$CFDIUUID);
//verificamos si la operacion HTTP fue correcta
if ($response->isSuccessful()) {
//con get Body se mapea la respuesta a un objeto WsResponse
$body = $response->getBody();
//se verifica asi a nivel de API todo fue correcto
if ($body->isStatusSuccess()) {
//en esta operacion Data es un booleano.
$existe = $body->getData();
if ($existe === true) {
echo "Existe CFDI";
} else if ($existe === false) {
echo "no existe";
} else {
echo "otra opcion";
}
} else {
//consumo invalido
echo "codigo: " . $body->getCode() . "\n";
echo "message: " . $body->getMessage() . "\n";
}
} else {
echo $response->getRawBody();
}
    
```

9.5.5 Ejemplo con SDK de .NET

El ejemplo para la verificación de existencia de CFDI en ACI DescargaSAT utilizando el SDK de Java en la siguiente Tabla 89:

Tabla 89. Ejemplo de verificación de existencia de CFDI con SDK .NET

```

var sdk = new DescargaSATSDK();
//se establece el token
sdk.Token = token;
//se establece el UUID del CFDI.
WsResponseMaster<bool> wsResponse=sdk.ExisteCFDI(cfdiUUID);
if (wsResponse.IsStatusSuccess()) {
    
```



```

//para este "metodo", data es un booleano que indica true si existe o false
en caso contrario
if (wsResponse.Data) {
    Console.WriteLine(string.Format("El CFDI con uuid :{0} esta resguardo en ACI
DescargaSAT",cfdiUUID));
    //si se sabe que esta resguardado entonces se puede consultar la informacion
de ese CFDI o descargarlo con los otros metodos:
}
else {
    Console.WriteLine(string.Format("El CFDI con uuid :{0} No esta resguardo en
ACI DescargaSAT", cfdiUUID));
}
}
else {
    //chechar los errores a nivel de api(ojo para otros errores como timeout,
son excepciones ):
    Console.WriteLine(string.Format("Codigo de error:{0}", wsResponse.Code));
    Console.WriteLine(string.Format("Mensaje de error:{0}",
wsResponse.Message));
}
}
}

```

9.6 Desbloqueo CFDI

Este método permite desbloquear tanto CFDI como META basado en el parámetro del cuerpo de la solicitud. Para que sea efectivo deberán existir elementos **bloqueados** del tipo indicado en la solicitud así como contar con un paquete con folios **disponibles** para ese **tipo**. De igual forma, el número máximo de elementos a desbloquear es de 2000 por petición, por lo que si se tienen más elementos bloqueados se deberá de enviar otra solicitud. Los CFDI que estén bloqueado no serán consultados en otros métodos por lo que se debe utilizar este método.

9.6.1 Request

EL método y URL para el desbloqueo de elementos en ACI DescargaSAT se describen en la Tabla 90:

Tabla 90. URL y body de endpoint desbloquear CFDI.

URL	METODO	Descripción	
/api/v2/descargasatsifei/cfdi/DesbloquearCFDIAPI	@POST	Este método permite desbloquear tanto meta como CFDI.	
Body JSON			
body	Tipo	Requerido	Descripción
TipoDesbloqueo	string	si	Solo puede ser "CFDI" o "META", e indica el tipo de elementos a desbloquear.



A continuación se muestra un ejemplo de una petición en la Tabla 91:

Tabla 91. Ejemplo de petición de desbloqueo de CFDI.

```
POST /api/v2/descargasatsifei/cfdi/DesbloquearCFDIAPI HTTP/1.1
Authorization: qpJHAYQLsdas89e1MI3j21301230912j3121
Content-Type: application/json; charset=UTF-8
Content-Length: 25
Host: descargasat.sifei.com.mx
Connection: Keep-Alive
Accept-Encoding: gzip
User-Agent: okhttp/3.12.0
{
  "TipoDesbloqueo": "CFDI"
}
```

9.6.2 Response

En este caso la respuesta en su campo data indica el número de CFDI desbloqueados.

Tabla 92. Descripción de data de response.

Campo	Tipo	Descripción
data	Int	Número de CFDI desbloqueados. En caso de error será null, por lo que si se utiliza un lenguaje tipado como java o c# deberá elegirse un tipo entero nutable (nullable).

La Tabla 93 demuestra el ejemplo de petición exitosa en la primera parte y fallo en la segunda.

Tabla 93. Ejemplo respuesta , la primera es afirmativa, la segunda es cuando no hay CFDI bloqueados.

Response exitoso <pre>{ "status": "success", "data": 2000, "message": "", "code": "" }</pre>
Response con error. <pre>{ "status": "fail",</pre>

```

    "data": null,
    "message": "No hay CFDI bloqueados",
    "code": "2400"
  }

```

9.6.3 Ejemplo SDK de JAVA

Ejemplo de desbloqueo de CFDI utilizando el SDK en la Tabla 94:

Tabla 94. Ejemplo de desbloqueo de CFDI utilizando el SDK de JAVA.

```

// Objeto de tipo ParameterDesbloquearItems que espera el servicio.
ParameterDesbloquearItems params = new ParameterDesbloquearItems();
// se estable el tipo de elememnto a desbloquear desde el enumerado.
params.tipoDesbloqueo = TipoDesbloqueo.META;
// se instancia el SDK.
DescargaSATSDK sdk = new DescargaSATSDK();
// se establece el token que es necesario para consumir
sdk.setToken(token);
try {
    // ahora se consume el servicio
    Response<WsResponseMaster<Integer>> response = sdk.DesbloquearItems(params);
    // con el metodo isSuccessfull se verficar si la peticion http fue correcta
    if (response.isSuccessfull()) {
        // si a nivel http fue correcto, entonces se deserializa el body:
        WsResponseMaster<Integer> body = response.body();
        // en el body verificamos que con isSuccess que la OPERACION fue
        correcta.
        if (body.isSuccess()) {
            // si status es success en el body entonces entonces data contiene el numero
            de
            // elementos desbloqueados.
            Integer data = body.getData();
            System.out.println("El numero de elementos desbloqueados fue de:" + data);
        } else {
            // algo fue mal un codigo de error, en este caso devolvera un error
            // si no habia elementos a desbloquear o si no se tiene un paquete de donde
            // descontar los folios.
            System.err.println("Codigo:" + body.getCode());
            System.err.println("Error:" + body.getMessage());
            // ... haz lo que necesites
        }
    } else { // codigo http diferente de 200 a 300.
        System.err.println("Error http code:" + response.code());
        // obtenemos el body del error
    }
}

```



```

ResponseBody dc = response.errorBody();
String error = Utils.parseErrorBody(dc);
System.err.println(error);
    }

} catch (Exception e) {
    e.printStackTrace();
}
    
```

9.6.4 Ejemplo SDK de PHP

Ejemplo de desbloqueo de CFDI utilizando el SDK en la Tabla 95:

Tabla 95. Ejemplo de desbloqueo de CFDI utilizando el SDK de PHP.

```

//instanciamos el SDK
$sdk = new DescargaSATSDK();
//establecemos el token
$sdk->setToken(self::$token);
#unico campo que es obligatorio, sino un error de parametros sera devuelto.
$params=[
    'TipoDesbloqueo'=> DescargaSATSDK::DESBLOQUEO_CFDI
];
$response = $sdk->DesbloquearCFDI($params);
if($response->isSuccessful()){
$body = $response->getBody();

if ($body->isStatusSuccess()) {

    $numeroItemsDesbloqueados = $body->getData();
    echo "\n Se desbloquearon: {$numeroItemsDesbloqueados}";
} else {
    //consumo invalido
    echo "\ncodigo: " . $body->getCode() . "\n";
    echo "message: " . $body->getMessage() . "\n";
}
}else{
echo $response->getRawBody();
}
    
```

9.6.5 Ejemplo SDK de .NET

Ejemplo de desbloqueo de CFDI utilizando el SDK en la Tabla 96:



Tabla 96. Ejemplo de desbloqueo de CFDI utilizando el SDK de .Net.

```

var sdk = new DescargaSATSDK();
//se establece el token
sdk.Token = token;

//se instancia el tipo de objeto esperado por el WS
var pa = new ParameterDesbloquarItems();
//en este caso TipoDesbloqueo solo puede ser META o CFDI
pa.TipoDesbloqueo = TIPO_DESBLOQUEO.CFDI;

//se invoca al metodo.
WsResponseMaster<int?> wsResponse = sdk.DesbloquarCFDI(pa);
//se verifica si todo fue bien:
if (wsResponse.IsStatusSuccess()) {
// en este caso data es un int, el cual indica el numero de elementos
desbloqueados.
//nota, el numero maximo de elementos por desbloqueo se hace de 2000.
    int? numeroDeelementos = wsResponse.Data;
    Console.WriteLine(string.Format("El numero de elementos desbloqueados fue de
:{0}", numeroDeelementos));
}
else {
    //chechar los errores a nivel de api(ojo para otros errores como timeout,
son excepciones ):
    Console.WriteLine(string.Format("Codigo de error:{0}", wsResponse.Code));
    Console.WriteLine(string.Format("Mensaje de error:{0}",
wsResponse.Message));
}
Console.ReadLine();
  
```

10. Métodos relacionados a la EFIRMA

10.1 Subir EFIRMA

Por medio de este método es posible subir la EFIRMA (la cual es necesaria para descargar los CFDI) a ACI DescargaSAT.

10.1.1 Request

En la Tabla 97 se muestra la URL de descarga de CFDI.



Tabla 97. URL para la descarga de CFDI.

URL	METODO	Descripción
/api/v2/descargasatsifei/certificado	@POST	Este método permite subir la eFirma correspondiente al RFC de la cuenta.

Datos a enviar en la petición se describe en la Tabla 98:

Tabla 98. Parametros JSON a enviar en el método Subir EFIRMA.

Data campos	tipo		Descripción
key	string	requerido	Llave codificada en base64
pass	string	requerido	Contraseña de la Efirma
cert	string	requerido	EFIRMA codificada en base64

Ejemplo de la petición para subir la Efirma se muestra en la Tabla 99

Tabla 99. Ejemplo de petición (se han omitido los valores).

```

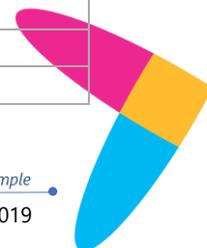
POST /api/v2/descargasatsifei/certificado HTTP/1.1
Authorization: qpJHAYQLsdas89e1MI3j21301230912j3121
Content-Type: application/json; charset=UTF-8
Content-Length: 4146
Host: descargasat.sifei.com.mx
Connection: Keep-Alive
Accept-Encoding: gzip
User-Agent: okhttp/3.12.0
{"cert":"","key":"","pass":""}
    
```

10.1.2 Response

Para este método cuando el "status" es "success" data contiene un objeto JSON con los campos descritos en la Tabla 100.

Tabla 100. Descripción de data para servicio de subir EFIRMA.

Campo	Tipo	Descripción
rfc	string	RFC que contiene la EFIRMA.
noSerie	string	Número de serie de la EFIRMA
validFrom	Datetime	Fecha de inicio de vigencia incluida en la EFIRMA
validTo	Datetime	Fecha de expiración incluida en la EFIRMA



En la Tabla 101 se muestra un ejemplo de respuesta cuando status contiene el valor "success" (recordar que con cualquier otro valor en "status" "data" será null)

Tabla 101. Ejemplo de respuesta cuando status contiene el valor "success".

```
{
  "status": "success",
  "data": {
    "rfc": "AAAAAAAAAAAA",
    "noSerie": "00005255152515",
    "validFrom": "2016-05-01 18:16:23",
    "validTo": "2021-05-01 18:17:03"
  },
  "message": "",
  "code": ""
}
```

10.1.3 Ejemplo con SDK de JAVA

El ejemplo para subir la EFirma a ACI DescargaSAT utilizando el SDK de Java se muestra en la Tabla 102:

Tabla 102. Ejemplo de subir eFirma con SDK de JAVA.

```
//Objeto de tipo SubirEFirma que espera el servicio, los 3 campos son
obligatorios
ParameterSubirEFirma params = new ParameterSubirEFirma();
//se establece la fecha inicial

//se lee el certificado y se convierte a base64
params.cert= Utils.readFileAndGetBase64(CERT_PATH);
//se lee el key y se convierte a base64
params.key= Utils.readFileAndGetBase64(KEY_PATH);
//se establece el pass del key
params.pass=KEY_PASS;
//se inicializa el SDK
DescargaSATSDK sdk = new DescargaSATSDK();
//se establece el token que es necesario para consumir
sdk.setToken(token);

try {
  //ahora se consume el servicio
  Response<WsResponseMaster<DataEFirmaResponse>>
  response=sdk.SubirEFirma(params);
  //con el metodo isSuccessful se verificar si la peticion http fue correcta
  if(response.isSuccessful()) {
    //si a nivel http fue correcto, entonces se deserializa el body:
```



```

WsResponseMaster<DataEFirmaResponse> body=response.body();
//en el body verificamos que con isSuccess que la OPERACION fue
correcta.
if(body.isSuccess()) {
//si status es success en el body entonces operar con data
DataEFirmaResponse data= body.getData();
System.out.println(data.getNoSerie());
System.out.println(data.getRfc());
System.out.println(data.getValidFrom());
System.out.println(data.getValidTo());

}else {
//algo fue mal un codigo de error
System.err.println("Codigo:"+body.getCode());
System.err.println("Error:"+body.getMessage());
//... haz lo que necesites
}

}else {//codigo http diferente de 200 a 300.
System.err.println("Error http code:" + response.code());
//obtenemos el body del error
ResponseBody dc = response.errorBody();
String error= Utils.parseErrorBody(dc);
System.err.println(error);
}

}catch(Exception e) {
e.printStackTrace();
}

```

10.1.4 Ejemplo con SDK de PHP

El ejemplo para subir la EFirma a ACI DescargaSAT utilizando el SDK de PHP se muestra en la Tabla 103:

Tabla 103. Ejemplo de subir eFirma con SDK de PHP.

```

//instanciamos el SDK
$sdk = new DescargaSATSDK();
//establecemos el token
$sdk->setToken(self::$token);
$params=[];
$params['cert']=base64_encode(file_get_contents(self::$CERT_PATH));
$params['key']=base64_encode(file_get_contents(self::$KEY_PATH));

```



```

$params['pass']=self::$KEY_PASS;

//le pasamos el UUID y ejecutamos el consumo del WS
$response = $sdk->SubirEFirma($params);
//verificamos si la operacion HTTP fue correcta
if ($response->isSuccessful()) {
//con get Body se mapea la respuesta a un objeto WsResponse
$body = $response->getBody();
//se verifica así a nivel de API todo fue correcto
if ($body->isStatusSuccess()) {
    //en esta operacion Data es un objeto compuesto por las propiedades:
    /**
    * rfc
    * noSerie
    * validFrom
    * validTo
    */
    $certificado = $body->getData();

} else {
    //consumo invalido
    echo "codigo: " . $body->getCode() . "\n";
    echo "message: " . $body->getMessage() . "\n";
}
} else {
echo $response->getRawBody();
}

```

10.1.5 Ejemplo con SDK de .NET

A continuación la Tabla 104 muestra el ejemplo de como subir la EFIRMA usando el SDK de .NET:

Tabla 104. Ejemplo de subir eFirma con SDK .NET.

```

var sdk = new DescargaSATSDK();
//se establece el token
sdk.Token = token;
//se instancia el tipo de objeto esperado por el WS
var pa = new ParameterSubirEFirma();

//se leen los archivos y se codifican en base64.
pa.Cert = Utils.ReadFileAndEncodeToBase64(@"F:\Certs\test\efirma.cer");
pa.Key = Utils.ReadFileAndEncodeToBase64(@"F:\Certs\test\key.key");
//pass de la llave

```



```

pa.KeyPass = "sadsdas";

WsResponseMaster<DataEFirmaResponse> wsResponse = sdk.SubirEFirma(pa);
//se verifica si todo fue bien:
if (wsResponse.IsStatusSuccess()) {

    var infoCer = wsResponse.Data;
    Console.WriteLine("Numero de serie:"+infoCer.NoSerie);
    Console.WriteLine("RFC:"+infoCer.RFC);
    Console.WriteLine("Valido desde: "+infoCer.ValidFrom);
    Console.WriteLine("Valido hasta: "+infoCer.ValidTo);
}
else {
    //checar los errores a nivel de api(ojo para otros errores como timeout,
    son excepciones ):
    Console.WriteLine(string.Format("Codigo de error:{0}", wsResponse.Code));
    Console.WriteLine(string.Format("Mensaje de error:{0}",
wsResponse.Message));
}
Console.ReadLine();

```

11. Métodos relacionados a la gestión de descargas

Estos métodos permiten gestionar desde el API la forma en que las descargas son gestionadas por ACI Descarga SAT, esto permite un mayor control sobre el método de descarga CFDI y tiempos de espera en que una descarga será atendida.

11.1 Consultar configuración

Este método permite consultar el método de descarga actual del RFC así como las horas de espera configuradas.

11.1.1 Request

La petición debe seguir la especificación indicada en la Tabla 105:

Tabla 105. Especificación de endpoint

URL	MÉTODO	Descripción
/api/v2/descargasatsifei/configuracion/query	@POST	Este método permite consultar la configuración de la gestión de tus descargas.
Body JSON		
body	Tipo	Requerido Descripción



URL	MÉTODO	Descripción
N/A	N/A	N/A

Ejemplo de solicitud en Tabla 106 en donde se observa que a pesar de ser post no lleva un body:

Tabla 106. Ejemplo solicitud

Request
<pre>POST /api/v2/descargasatsifei/configuracion/query HTTP/1.1 Authorization: qpJHAYQLsdas89e1MI3j21301230912j3121 Content-Length: 0 Host: 127.0.0.1:80 Connection: Keep-Alive Accept-Encoding: gzip User-Agent: okhttp/3.12.0</pre>

11.1.2 Response

En esta respuesta data es de tipo Configuracion 12.1.2.

Ejemplo de respuesta en Tabla 107:

Tabla 107. Ejemplo response Raw

Ejemplo Response body
<pre>{ "status": "success", "data": { "metodoDescarga": "WS", "horasEsperaWs": 72, "horasEsperaPortal": 72, "mensaje": null }, "message": "", "code": "" }</pre>

11.1.3 Ejemplo con SDK de JAVA

Ejemplo de invocación de método consultar configuración con el SDK en Tabla 108:



Tabla 108. Ejemplo consulta configuración de descargas con SDK de JAVA

```

//se instancia el SDK.
DescargaSATSDK sdk = new DescargaSATSDK();
//se establece el token que es necesario para consumir
sdk.setToken(token);

try {
//ahora se consume el servicio
Response<WsResponseMaster<ResponseConsultarConfiguracion>>
response=sdk.consultarConfiguracion();
//con el metodo isSuccessful se verificar si la peticion http fue correcta
if(response.isSuccessful()) {
//si a nivel http fue correcto, entonces se deserializa el body:
WsResponseMaster<ResponseConsultarConfiguracion> body=response.body();
//en el body verificamos que con isSuccess que la OPERACION fue correcta.
if(body.isSuccess()) {
//si status es success en el body entonces entonces data contiene un objeto tipo
ResponseConsultarConfiguracion.
ResponseConsultarConfiguracion data= body.getData();
System.out.println("Horas en espera Portal:"+data.HorasEsperaPortal);
System.out.println("Horas en espera WS:"+data.HorasEsperaPortal);
System.out.println("Metodo Descarga:"+data.HorasEsperaPortal);

}else {
//algo fue mal un codigo de error, en este caso devolvera un error
//si no habia elementos a desbloquear o si no se tiene un paquete de donde
descontar los folios.
System.err.println("Codigo:"+body.getCode());
System.err.println("Error:"+body.getMessage());
//... haz lo que necesites
}

}else {//codigo http diferente de 200 a 300.
System.err.println("Error http code:" + response.code());
//obtenemos el body del error
ResponseBody dc = response.errorBody();
String error= Utils.parseErrorBody(dc);
System.err.println(error);
}

}catch(Exception e) {
e.printStackTrace();
}

```



11.1.4 Ejemplo con SDK de PHP

Ejemplo de invocación de método consultar configuración con el SDK en Tabla 109:

Tabla 109 . Ejemplo consulta configuración de descargas con SDK de PHP

```
//instanciamos el SDK
$sdk = new DescargaSATSDK();
//establecemos el token
$sdk->setToken(self::$token);

$response = $sdk->consultarConfiguracion();
if ($response->isSuccessful()) {
    $body = $response->getBody();
    #si fue success significa que se acepto el cambio de horas de espera.
    if ($body->isStatusSuccess()) {
        #data en este caso es un objeto con 3 campos: con un mensaje de
        confirmacion.
        $dataConfiguracion = $body->getData();
        echo "\nMetodo descarga: " . $dataConfiguracion['metodoDescarga'] . "\n";
        echo "Horas espera ws: " . $dataConfiguracion['horasEsperaWs'] . "\n";
        echo "Horas espera Portal: " . $dataConfiguracion['horasEsperaPortal'] .
        "\n";
    } else {
        //consumo invalido
        echo "\ncodigo: " . $body->getCode() . "\n";
        echo "message: " . $body->getMessage() . "\n";
    }
} else {
    echo $response->getRawBody();
}
```

11.1.5 Ejemplo con SDK .NET

Ejemplo de invocación de método consultar configuración con el SDK en Tabla 110:

Tabla 110. Ejemplo consulta configuración de descargas con SDK de .NET

```
var sdk = new DescargaSATSDK();
//se establece el token
sdk.Token = token;
//se invoca al metodo.
WsResponseMaster<ResponseConsultarConfiguracion> wsResponse =
sdk.ConsultarConfiguracion();
//se verifica si todo fue bien:
```



```

if (wsResponse.IsSuccess()) {
    // en este caso data es un objeto tipo ResponseConsultarConfiguracion.
    ResponseConsultarConfiguracion config = wsResponse.Data;
    Console.WriteLine(string.Format("Horas espera portal :{0}",
config.HorasEsperaPortal));
    Console.WriteLine(string.Format("Horas espera ws :{0}",
config.HorasEsperaWs));
    Console.WriteLine(string.Format("Metodo descarga :{0}",
config.MetodoDescarga));
}
else {
    //chechar los errores a nivel de api(ojo para otros errores como timeout,
son excepciones ):
    Console.WriteLine(string.Format("Codigo de error:{0}", wsResponse.Code));
    Console.WriteLine(string.Format("Mensaje de error:{0}",
wsResponse.Message));
}
Console.ReadLine();
  
```

11.2 Configurar horas de espera

Este método permite configurar las horas de espera para el método WS como por portal, es necesario indicar ambos valores, esto es útil si se quiere asegurar que se descargarán todos los CFDI emitidos con fecha posterior a las 72 horas, por lo que deberá enviarse este valor (72) horas.

11.2.1 Request

Tabla 111. .Especificación de endpoint

URL	METODO	Descripción	
/api/v2/descargasatsifei/configuracion/horasEspera	@POST	Este endpoint permite configurar las horas de espera.	
Body JSON			
body	Tipo	Requerido	Descripción
horasEsperaWs	int	SI	Número de horas de espera cuando el método es WS. No debe ser menor a 24 hrs ni mayor a 90.
horasEsperaPortal	int	SI	Número de horas de espera cuando el método es Portal. No debe ser menor a 24 hrs ni mayor a 90.



Ejemplo

Tabla 112. Ejemplo de petición para establecer las horas de espera.

```
Request
POST /api/v2/descargasatsifei/configuracion/horasEspera HTTP/1.1
Authorization: qpJHAYQLsdas89e1MI3j21301230912j3121
Content-Type: application/json; charset=UTF-8
Content-Length: 65
Host: descargasat.sifei.com.mx
Connection: Keep-Alive
Accept-Encoding: gzip
User-Agent: okhttp/3.12.0
{
  "horasEsperaWs": 72,
  "horasEsperaPortal": 72
}
```

11.2.2 Response

En esta respuesta data es de tipo [Configuracion](#) 12.1.2.

Ejemplo de respuesta en la Tabla 113:

Tabla 113. Ejemplos de respuesta del método cambiar horas de espera.

```
Response de ejemplo en caso de éxito.
{
  "status": "success",
  "data": {
    "metodoDescarga": "WS",
    "horasEsperaWs": 72,
    "horasEsperaPortal": 72,
    "mensaje": "Se ha actualizado las horas de espera para el RFC:SIF0403229F9"
  },
  "message": "",
  "code": ""
}

Response en caso de fallo.
{
  "status": "fail",
  "data": null,
  "message": "Horas ws debe entre 24 horas y 90",
}
```

```
"code": "2301"
}
```

11.2.3 Ejemplo con SDK de JAVA

Ejemplo de invocación de método establecer horas espera con el SDK en Tabla 114:

Tabla 114. Ejemplo de cambio de horas de espera utilizando el SDK de JAVA

```
//se instancia el SDK.
DescargaSATSDK sdk = new DescargaSATSDK();
//se establece el token que es necesario para consumir
sdk.setToken(token);

ParametersEstablecerConfiguracion para = new ParametersEstablecerConfiguracion();
//se establecen los 2 valores obligatorios
para.HorasEsperaPortal=72;
para.HorasEsperaWs=72;
try {
//ahora se consume el servicio
Response<WsResponseMaster<ResponseConsultarConfiguracion>>
response=sdk.EstablecerHorasEspera(para);
//con el metodo isSuccessful se verificar si la peticion http fue correcta
if(response.isSuccessful()) {
//si a nivel http fue correcto, entonces se deserializa el body:
WsResponseMaster<ResponseConsultarConfiguracion> body=response.body();
//en el body verificamos que con isSuccess que la OPERACION fue correcta.
if(body.isSuccess()) {
//si status es success en el body entonces entonces data contiene un objeto tipo
ResponseConsultarConfiguracion.
ResponseConsultarConfiguracion data= body.getData();
System.out.println("Horas en espera Portal:"+data.HorasEsperaPortal);
System.out.println("Horas en espera WS:"+data.HorasEsperaPortal);
System.out.println("Metodo Descarga:"+data.MetodoDescarga);
System.out.println("Metodo Descarga:"+data.Mensaje);

}else {
//algo fue mal un codigo de error, en este caso devolvera un error
//si no habia elementos a desbloquear o si no se tiene un paquete de donde
descontar los folios.
System.err.println("Codigo:"+body.getCode());
System.err.println("Error:"+body.getMessage());
//... haz lo que necesites
}

}else {//codigo http diferente de 200 a 300.
```



```

System.err.println("Error http code:" + response.code());
//obtenemos el body del error
ResponseBody dc = response.errorBody();
String error= Utils.parseErrorBody(dc);
System.err.println(error);
}

}catch(Exception e) {
e.printStackTrace();
}
    
```

11.2.4 Ejemplo con SDK de PHP

Ejemplo de invocación de método establecer horas espera con el SDK en Tabla 115:

Tabla 115. Ejemplo de cambio de horas de espera utilizando el SDK de PHP.

```

//instanciamos el SDK
$sdk = new DescargaSATSDK();
//establecemos el token
$sdk->setToken(self::$token);

#Ambos campos son obligatorios, sino un error de parametros sera devuelto.
$params = [
'horasEsperaWs' => "1",
'horasEsperaPortal' => '50',
];
$response = $sdk->EstablecerHorasEspera($params);
if ($response->isSuccessful()) {
$body = $response->getBody();
#si fue success significa que se acepto el cambio de horas de espera.
if ($body->isStatusSuccess()) {
    $dataConfiguracion = $body->getData();
    echo "\nMetodo descarga: " . $dataConfiguracion['metodoDescarga'] . "\n";
    echo "Horas espera ws: " . $dataConfiguracion['horasEsperaWs'] . "\n";
    echo "Horas espera Portal: " . $dataConfiguracion['horasEsperaPortal'] . "\n";
    echo "mensaje: " . $dataConfiguracion['mensaje'] . "\n";
} else {
    //consumo invalido
    echo "\ncodigo: " . $body->getCode() . "\n";
    echo "message: " . $body->getMessage() . "\n";
}
} else {
echo $response->getRawBody();
}
    
```



11.2.5 Ejemplo con SDK .NET

Ejemplo de invocación de método establecer horas espera con el SDK en Tabla 116:

Tabla 116. Ejemplo de cambio de horas de espera utilizando el SDK en .NET.

```

var sdk = new DescargaSATSDK();
//se establece el token
sdk.Token = token;
ParametersEstablecerConfiguracion param = new ParametersEstablecerConfiguracion();
//se establece horas por portal y ws, ambos son obligatorios.
param.HorasEsperaPortal = 60;
param.HorasEsperaWs = 72;
//se invoca al metodo.
WsResponseMaster<ResponseConsultarConfiguracion> wsResponse =
sdk.EstablecerHorasEspera(param);
//se verifica si todo fue bien:
if (wsResponse.IsStatusSuccess()) {
    // en este caso data es un objeto tipo ResponseConsultarConfiguracion.

    ResponseConsultarConfiguracion config = wsResponse.Data;
    Console.WriteLine(string.Format("Horas espera portal :{0}",
config.HorasEsperaPortal));
    Console.WriteLine(string.Format("Horas espera ws :{0}", config.HorasEsperaWs));
    Console.WriteLine(string.Format("Metodo descarga :{0}", config.MetodoDescarga));
    Console.WriteLine(string.Format("Mensaje :{0}", config.Mensaje));
}
else {
    //chechar los errores a nivel de api(ojo para otros errores como timeout, son
    excepciones ):
    Console.WriteLine(string.Format("Codigo de error:{0}", wsResponse.Code));
    Console.WriteLine(string.Format("Mensaje de error:{0}", wsResponse.Message));
}
Console.ReadLine();
}
  
```

11.3 Configurar método de descarga

Este método permite cambiar el método de descarga entre WS y por portal, de acuerdo a las necesidades propias que se puedan presentar, siendo "WS" el método oficial dado por el SAT y "portal" un método no oficial y propenso a cambios. Se da la opción de cambiar entre estos 2 métodos, sin embargo, en caso de un cambio en el mecanismo del método "portal" del SAT, la configuración puede cambiar a método WS sin previo aviso, esto con el objetivo de atender las descargas.



11.3.1 Request

Tabla 117. Especificación de endpoint configurar método de descarga.

URL	METODO	Descripción	
<code>/api/v2/descargasatsifei/configuracion/metodoDescarga</code>	@POST	Este endpoint permite configurar el método de descarga.	
Body JSON			
body	Tipo	Requerido	Descripción
<code>metodoDescarga</code>	string	SI	Método de descarga, valores posibles "WS", "PORTAL".

Ejemplo

Tabla 118. Ejemplo de petición para método cambiar método de descarga.

Request
<pre>POST /api/v2/descargasatsifei/configuracion/metodoDescarga HTTP/1.1 Host: descargasat.sifei.com.mx Accept: */* Authorization: qpJHAYQLsdas89e1MI3j21301230912j3121 User-Agent: sdkphp/1.1 Content-Length: 23 Content-Type: application/json { "metodoDescarga": "WS" }</pre>

11.3.2 Response

En esta respuesta data es de tipo Configuracion 12.1.2.

Ejemplo de response:

Tabla 119. Ejemplo de respuesta del método cambiar método descarga.

Response
<pre>{ "status": "success", "data": { "metodoDescarga": "WS", "horasEsperaWs": 72, "horasEsperaPortal": 72, "mensaje": "Actualizacion de metodo descarga exitoso" }, }</pre>



```
"message": "",
"code": ""
}
```

11.3.3 Ejemplo con SDK de JAVA

Ejemplo de invocación de método establecer método de descarga con el SDK en Tabla 120:

Tabla 120. Ejemplo de cambio de método descarga utilizando el SDK de JAVA.

```
//se instancia el SDK.
DescargaSATSDK sdk = new DescargaSATSDK();
//se establece el token que es necesario para consumir
sdk.setToken(token);
ParametersEstablecerConfiguracion para = new ParametersEstablecerConfiguracion();
para.MetodoDescarga=MetodoDescarga.WS;
try {
//ahora se consume el servicio
Response<WsResponseMaster<ResponseConsultarConfiguracion>>
response=sdk.EstablecerMetodoDescarga(para);
//con el metodo isSuccessfull se verificar si la peticion http fue correcta
if(response.isSuccessfull()) {
//si a nivel http fue correcto, entonces se deserializa el body:
WsResponseMaster<ResponseConsultarConfiguracion> body=response.body();
//en el body verificamos que con isSuccess que la OPERACION fue correcta.
if(body.isSuccess()) {
//si status es success en el body entonces entonces data contiene una consulta con
la configuración actual
ResponseConsultarConfiguracion data= body.getData();
System.out.println("Horas en espera Portal:"+data.HorasEsperaPortal);
System.out.println("Horas en espera WS:"+data.HorasEsperaPortal);
System.out.println("Metodo Descarga:"+data.MetodoDescarga);
System.out.println("Metodo Descarga:"+data.Mensaje);

}else {
//algo fue mal un codigo de error, en este caso devolvera un error
//si no habia elementos a desbloquear o si no se tiene un paquete de donde
descontar los folios.
System.err.println("Codigo:"+body.getCode());
System.err.println("Error:"+body.getMessage());
//... haz lo que necesites
}

}else {//codigo http diferente de 200 a 300.
System.err.println("Error http code:" + response.code());
```



```

//obtenemos el body del error
ResponseBody dc = response.errorBody();
String error= Utils.parseErrorBody(dc);
System.err.println(error);
}

}catch(Exception e) {
e.printStackTrace();
}
    
```

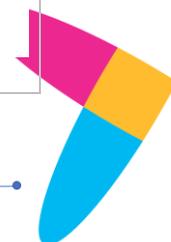
11.3.4 Ejemplo con SDK de PHP

Ejemplo de invocación de método establecer método de descarga con el SDK en Tabla 121:

Tabla 121. Ejemplo de cambio de método descarga utilizando el SDK de PHP.

```

//instanciamos el SDK
$sdk = new DescargaSATSDK();
//establecemos el token
$sdk->setToken(self::$token);
$params = [
'metodoDescarga' => DescargaSATSDK::METODO_DESCARGA_WS,
# 'metodoDescarga'=>"1d" # erroneo
];
$response = $sdk->EstablecerMetodoDescarga($params);
if ($response->isSuccessful()) {
$body = $response->getBody();
#si fue success significa que se acepto el cambio de horas de espera.
if ($body->isStatusSuccess()) {
#data en este caso es un objeto con 4 campos
$dataConfiguracion = $body->getData();
echo "\nMetodo descarga: " . $dataConfiguracion['metodoDescarga'] . "\n";
echo "Horas espera ws: " . $dataConfiguracion['horasEsperaWs'] . "\n";
echo "Horas espera Portal: " . $dataConfiguracion['horasEsperaPortal'] . "\n";
echo "mensaje: " . $dataConfiguracion['mensaje'] . "\n";
} else {
//consumo invalido, no se actualizo.
echo "\ncodigo: " . $body->getCode() . "\n";
echo "message: " . $body->getMessage() . "\n";
}
} else {
echo $response->getRawBody();
}
    
```



11.3.5 Ejemplo con SDK .NET

Ejemplo de invocación de método establecer método de descarga con el SDK en Tabla 122:

Tabla 122. Ejemplo de cambio de método descarga utilizando el SDK en .NET.

```

var sdk = new DescargaSATSDK();
//se establece el token
sdk.Token = token;
ParametersEstablecerConfiguracion param = new ParametersEstablecerConfiguracion();
param.MetodoDescarga = ParametersEstablecerConfiguracion.METODO_PORTAL;
//se invoca al metodo.
WsResponseMaster<ResponseConsultarConfiguracion> wsResponse =
sdk.EstablecerMetodoDescarga(param);
//se verifica si todo fue bien:
if (wsResponse.IsStatusSuccess()) {
    // en este caso data es un objeto tipo ResponseConsultarConfiguracion.

    ResponseConsultarConfiguracion config = wsResponse.Data;
    Console.WriteLine(string.Format("Horas espera portal :{0}",
config.HorasEsperaPortal));
    Console.WriteLine(string.Format("Horas espera ws :{0}", config.HorasEsperaWs));
    Console.WriteLine(string.Format("Metodo descarga :{0}", config.MetodoDescarga));
    Console.WriteLine(string.Format("Mensaje :{0}", config.Mensaje));
}
else {
    //chechar los errores a nivel de api(ojo para otros errores como timeout, son
    excepciones ):
    Console.WriteLine(string.Format("Codigo de error:{0}", wsResponse.Code));
    Console.WriteLine(string.Format("Mensaje de error:{0}", wsResponse.Message));
}
Console.ReadLine();
  
```

12. Tipos de datos

Cada método del API expone un tipo de dato que esta descrito en su propia sección, sin embargo, existen tipos de datos que se utilizan en más de un método. En este apartado se describen cada uno de estos datos.

12.1.1 Tipo CFDI.

Este es el tipo devuelto por el WS para representar un CFDI, donde todos los SDK tienen una clase que modela este objeto.

En la Tabla 123 se describen los campos del CFDI:



Tabla 123. Descripción de Campos CFDI.

Data campos		Descripción
uuid	string	UUID del CFDI
rfcEmisor	string	RFC Emisor del RFC
rfcReceptor	string	RFC Receptor del RFC
tipoDescarga	string	<i>Tipo de descarga (CFDI, META)</i>
fechaDescarga	datetime	Fecha que se descargó el CFDI.
total	int	Total del CFDI.
fechaEmision	datetime	Fecha de emisión del CFDI.
pacquecertifico	string	RFC del PAC que certifico.
fechaCertificacion	datetime	Fecha de certificación de CFDI.
solicitadaMetodo	string	Método por el cual se descargó el CFDI.
estado	string	Estado del CFDI: 0 = Cancelado, 1 = vigente. Este campo tiene el valor al momento en que se obtuvo el CFDI. Por lo cual no deberá ser 100% fiable.
efecto	string	Efecto del CFDI (catalogo): I= Ingreso. E=Egreso, T= Traslado, P= Pago, N=Nomina.
disponibleXML	boolean	Indica si se descontó de Paquete, en caso de true, se podrá descargar el XML, en caso de false indica que o no se ha descargado el XML o que no se ha descontado de tu paquete, es decir, el CFDI está bloqueado, se deberá conseguir un nuevo paquete y desbloquear el CFDI.
disponibleMETA	boolean	Indica si se descontó del paquete, en caso de true se podrá consultar.

12.1.2 Tipo Configuración (response)

Este es el tipo de respuesta tanto de la consulta de configuración como el de la respuesta de los métodos de actualización método descarga y horas de espera.

Tabla 124. Descripción de campos de tipo respuesta.

Campo	Tipo	Descripción
metodoDescarga	string	Método de descarga, valores posibles "WS", "PORTAL".
horasEsperaWS	Int	Número de horas a esperar para que una petición con fecha pasada sea procesada. Es decir, número de horas pasadas desde la fecha actual, ejemplo si se establece 72, deberá haber transcurrido 72 horas desde hora y fecha actual. FechaActual-fechaFinalDeDescargaProgramada > 72 horas.

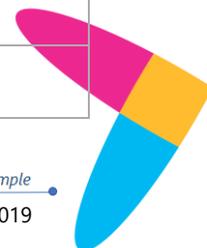
Campo	Tipo	Descripción
horasEsperaPort al	int	Número de horas a esperar para que una petición con fecha pasada sea procesada. Es decir, numero de horas pasadas desde la fecha actual. FechaActual-fechaFinalDeDescargaProgramada>72 horas.
mensaje	string	Mensaje adicional de confirmación para métodos de actualización.

13. Código de error

A continuación en la Tabla 125 se listan los códigos de error que el API devuelve.

Tabla 125 Codigos de error y su respectiva descripcion del API.

Código	Descripción
400	Bad Request
402	Payment Required
404	Not Found
500	Error Interno, intenta más tarde
501	No implemented
1001	TOKEN invalido, no existe o ha expirado
1003	Error de autenticación, falta header Authorization
1004	Token vacío
2000	Archivo inválido, no está correctamente codificado
2001	El archivo sobrepasa el limite
2100	Error genérico, operación inválida.
2101	Descarga Repetida.
2102	Descarga no pudo Reactivarse (por ejemplo: estado no es apto para ser reactivado, debe ser ERROR, BLOQUEADA, INCOMPLETA).
2110	Archivo inválido RAR
2120	Zip inválido
2200	Número de CFDI máximo excedido



Código	Descripción
2201	CFDI no encontrado
2202	El CFDI no tiene XML
2203	CFDI no descargable (por ejemplo solo se tenga el meta)
2205	CFDI bloqueado (cuando no se ha descontado el CFDI de paquete se mantendrá resguardado pero no podrás consultarlo hasta obtener un nuevo paquete)
2300	Petición incompleta, falta uno o más campos
2301	Error en formato o tipo de uno o más campos
2204	No hay que descargar
2300	No tienes EFirma
2301	Error al subir EFirma (certificado inválido, formato inválido, contraseña no corresponde), certificado expirado,
2302	Certificado no corresponde al RFC.
2304	EFirma Revocada o Caduca. Cuando se usa tu Efirma y el SAT deniega la operación por certificado Revocado o Caduco.
2305	EFirma Inválida. Cuando se usa tu Efirma y el SAT deniega la operación por EFirma inválida.
2400	No hay CFDI o META a desbloquear
2401	No hay paquete disponible para CFDI o Meta a desbloquear.



14. Información de Contacto con SIFEI

CENTRO DE SOPORTE TÉCNICO SIFEI

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

Dirección electrónica

[Centro de Soporte Técnico SIFEI](#)

TELÉFONOS DE CONTACTO

Orizaba, Ver.	+52 (272) 726 6999
CDMX	(55) 4624 0146
Puebla, Pue.	(222) 620 0239
con 10 líneas	

ATENCIÓN A INCIDENTES

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

Correo Electrónico

helpdesk@sifei.com.mx

HORARIO DE ATENCIÓN

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

Lunes a viernes

De 09:00 a 19:00 hrs.

PÁGINAS OFICIALES DE SIFEI

Síto web	http://www.sifei.com.mx/
Facebook	http://www.facebook.com/SIFEIMexico
Twitter	http://twitter.com/SIFEIMexico
YouTube	http://www.youtube.com/SIFEIMexico
LinkedIn	http://www.linkedin.com/company/SIFEIMexico

UBICACIÓN DE OFICINA MATRIZ

Primera Privada de Oriente 17 No. 32
Col. Centro, Orizaba, Veracruz, México
CP 94300

<Fin del Documento>

