

Declaración de Información
Financiera de las
Empresas del Sistema Financiero
Manual de Servicios REST



ÍNDICE

1. Lineamientos generales	4
2. Procedimiento para obtención de credenciales y consumir servicios de autenticación de SUNAT	4
3. Generación de token para el uso de servicios REST	5
4. Generación de llave pública y encriptamiento de archivo ZIP.....	6
5. Modelo de integración de las Empresas del Sistema Financiero y la SUNAT	7
6. Estructura y secuencia de servicios REST	7
6.1. REST consulta padrón	7
6.2. REST validación si existe una declaración	8
6.3. REST validación vencimiento	9
6.4. REST envío de información	10
6.5. REST servicio de envío de archivo TUS.....	14
6.6. REST servicio consulta	15
7. Validación en el receptor de la declaración	17
8. Anexos	17
8.1. Anexo 01_Especificaciones Receptor.xlsx	17
8.2. Anexo 02_Validaciones Receptor.xlsx	17

CUADRO DE CONTROL DE CAMBIOS DEL DOCUMENTO

N.º	Descripción	Fecha	Versión	Responsable	Motivo de cambio
1	Creación del documento	01/09/2021	1.0.0	1U2104	

Transferencia de información financiera

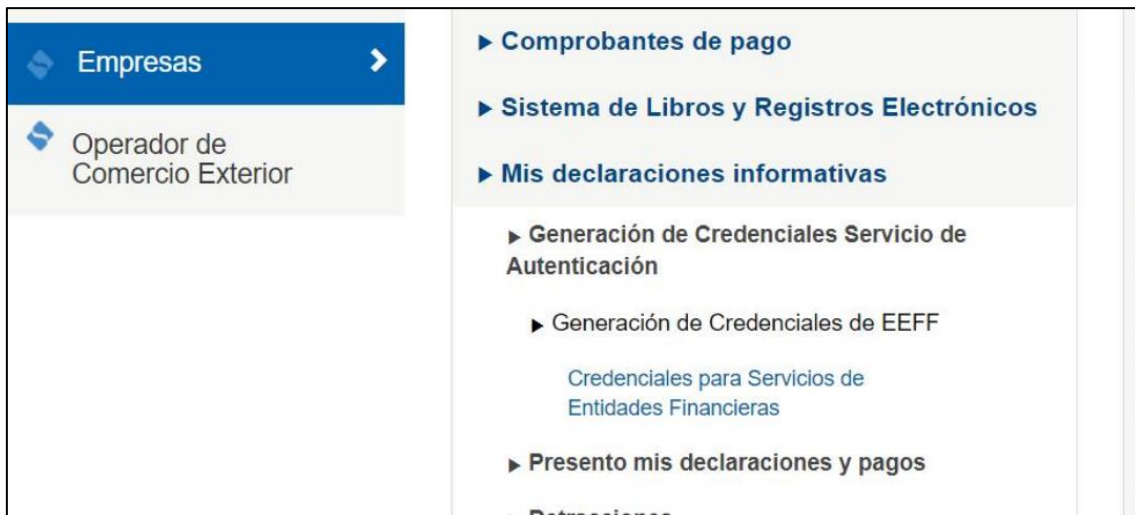
1. Lineamientos generales

Debido a la necesidad de transferencia de información financiera se implementa un conjunto de servicios REST en la SUNAT los cuales están expuestos para el consumo de las empresas del sistema financiero, los cuales están basados en estándar JSON.

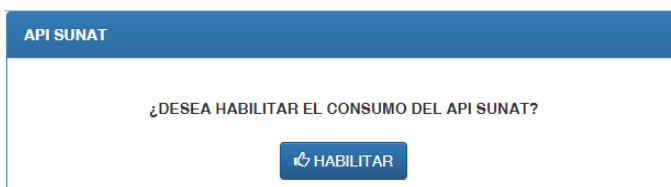
2. Procedimiento para obtención de credenciales y consumir servicios de autenticación de SUNAT

La empresa del sistema financiero obligada a presentar la declaración financiera debe ingresar al **Portal SOL de SUNAT** (<https://e-menu.sunat.gob.pe/cl-ti-itmenu/MenuInternet.htm>) e iniciar sesión con su correspondiente **Clave SOL**.

Ingresar a la opción “EMPRESAS / Mis Declaraciones Informativas / Generación de Credenciales Servicio de Autenticación / Generación de Credenciales EEFF / **Credenciales para Servicios de Entidades Financieras**”¹



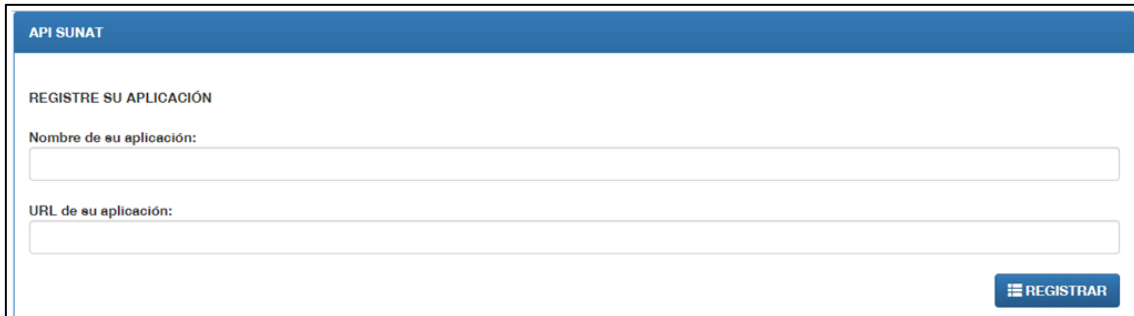
La empresa del sistema financiero debe confirmar que desea habilitar el consumo del API SUNAT.



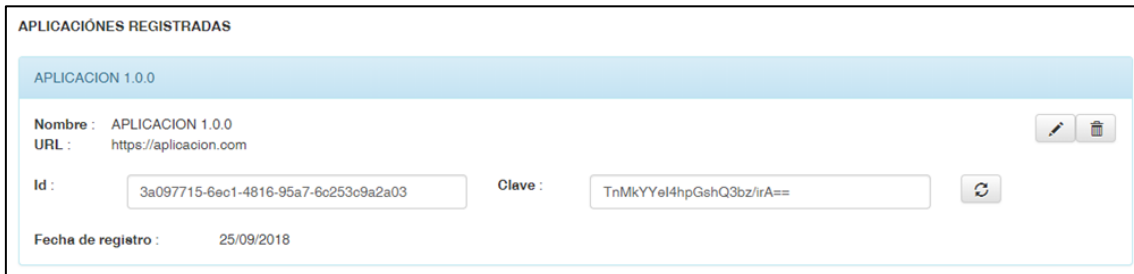
¹ Disponible a partir del 9.9.2021

La empresa del sistema financiero debe ingresar los datos solicitados:

Nombre de la Aplicación	Nombre con el que identifica su sistema ante la SUNAT.
URL de su Aplicación	URL de la aplicación en caso de tener acceso público o en su defecto URL de la página web oficial de la empresa del sistema financiero.



Al presionar el botón “**REGISTRAR**”, el sistema genera las credenciales correspondientes:



La empresa del sistema financiero debe almacenar estos valores para ser utilizado en su Sistema de Información.

3. Generación de token para el uso de servicios REST

Una vez que cuenten con los datos del client_id y client_secret, y además el usuario y clave SOL, podrán generar el token del api-seguridad-sunat con la siguiente url:

https://api-seguridad.sunat.gob.pe/v1/clientessol/{client_id}/oauth2/token

Ejemplo:

<https://api-seguridad.sunat.gob.pe/v1/clientessol/ababababa-9abc-453s-s5s4s4-s4575551/oauth2/token>

Desde postman debe configurar los siguientes datos:

Body: x-www-form-urlencoded

Configurar los siguientes datos:

KEY	VALUE
grant_type	Password
scope	https://api-eeff.sunat.gob.pe
client_id	ababababa-9abc-453s-s5s4s4-s4575551
client_secret	CSCSSKSJDSKSNKSKSSKSJDDN
username	{ RUC } { USUARIO }
password	{ CLAVESOL }

Como respuesta el API seguridad generara el token:

```
{
  "access_token":
  "eyJraWQiOiJhcGkuc3VuYXQuZ29iLnBILmtpZDAwMSIsInR5cCI6IkpXVCIsImFsZyI6IiJTMjU2I
  nfOS7413x6Fa7MCG2P0FHM1jh0sjP6YaHhtxnOQo9aRtHVLpaKQnU8QPXIQNbMyUIE2cMn
  pvlynnL8mugNc1PmMa86HfLKjllimQdAZ2RuEwOyF_tArc8Zm8MgOgnlIZOCttRloOBk4WX
  3x7ivKblUmyszrVJ3NkQdu3P2QibDNTgY8h4X2oOZkMCml5JCivb2dSNhVopuR9oPcLnqd3
  1BZvtJ8Y_p9w",
  "token_type": "JWT",
  "expires_in": 3600
}
```

Con el token generado ya se pueden consumir los servicios REST.

4. Generación de llave pública y encriptamiento de archivo ZIP

La información transferida desde la empresa del sistema financiero debe ser segura por ello es necesario que se envíe en formato ZIP y encriptado.

Para el encriptamiento del archivo ZIP es necesario contar una llave pública que debe ser tramitada ante la SUNAT, una vez obtenida, se debe encriptar el archivo en conjunto con la llave pública con un método RSA 248.

Ejemplo de llave pública: publickey

```
---- BEGIN SSH2 PUBLIC KEY ----
```

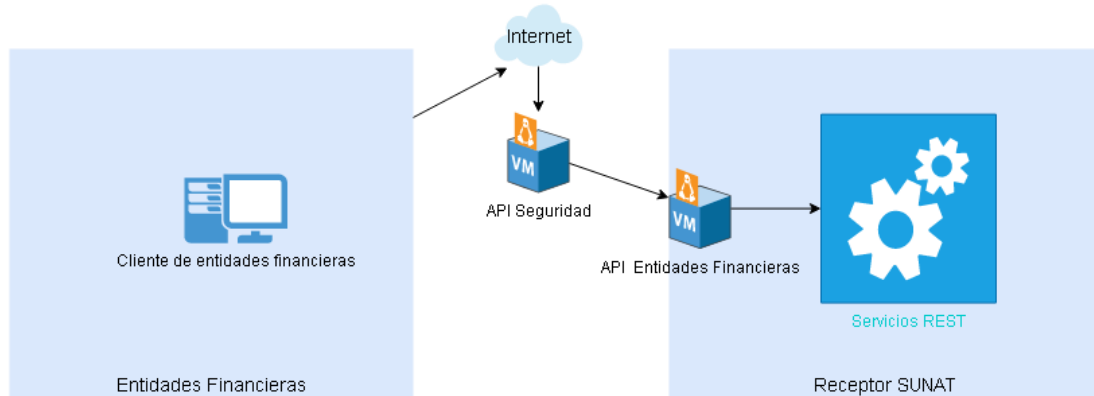
```
Comment: "rsa-key-20210625"
```

```
4ZglUdqho/DmH1EpfYKsh/0tA/ZvlykBrJwAZFQ0nrdudZto2NOH7JAdntd3iBEoQ4kqycuxhO
sLH9RDmQzv70dDNwlsq9y+bjTeEV9BpiURpu56TKycJCIDoOLqAZeZjMXhTJpN1B74wqM06
KvZEyd4Jqq7WtaAzefHRlftf508zfp4CyEYtO7n07L1zxcU/5BOoP8o6xiLyznsI/9LeLIX+nIhXYf/
DAI97Hzhp+xof+w1ijeTnkG
```

```
---- END SSH2 PUBLIC KEY ----
```

5. Modelo de integración de las Empresas del Sistema Financiero y la SUNAT

Modelo de integración de las empresas del sistema financiero con la SUNAT para el servicio seguro:



6. Estructura y secuencia de servicios REST

Para el envío de la información de manera satisfactoria desde la empresa del sistema financiero debe seguir la secuencia de los siguientes pasos:

6.1. REST consulta padrón

Permite validar si la empresa del sistema financiero se encuentra en el padrón de obligados, de no figurar en el padrón, la empresa del sistema financiero debe comunicar a la SUNAT que es un obligado a presentar la declaración de información financiera a fin de ser incluido en el mismo y acceda al servicio REST para presentar la declaración a la SUNAT.

Estructura REST:

Método	GET
Header	numRuc: 20100066603
URL	https://api-eeff.sunat.gob.pe/v1/auditoria/tributaria/consultapadron/{numRuc}
Ejemplo:	https://api-eeff.sunat.gob.pe/v1/auditoria/tributaria/consultapadron/20100066603

Respuesta REST:

```
{
  "sucess": true, -
- (true=se encuentra en padrón, false=no se encuentra en padrón)
  "codError": 0,
  "desError": ""
}
```

El valor “sucess” es true si se encuentra en padrón, de lo contrario es false.

6.2. REST validación si existe una declaración

Permite validar si ya se cuenta con una declaración original o previa, además permite validar si aún existe alguna declaración en estado recepcionado o en curso.

Estructura REST:

Método	GET
Header	
URL	https://api.eeff.sunat.gob.pe/v1/auditoria/tributaria/verificadef/{numRuc}/{tipDecl}/{anioDecl}/{semDecl}
Ejemplo:	https://api.eeff.sunat.gob.pe/v1/auditoria/tributaria/verificadef/20100066603/01/2021/06

- Validación si el tipo de declaración es original (tipDecl = 01)

Respuesta REST:

```
{
  "sucess": true, -- (true = Existe una declaración original para el semestre, false= No existe una declaración original)
  "codError": "0",
  "desError": "",
  "numTicket": ""
}
```

El valor de la variable “sucess” es true si existe una declaración original para el semestre, false si no existe una declaración original.

En un semestre solo puede existir una declaración original.

- Si el tipo de declaración es sustitutoria o rectificatoria (tipDecl = 02)

Respuesta REST:

```
{
  "success":true, -
  - (true = Existe una declaracion original o referencia para el semestre , false= No existe una declaracion original o esta en curso)
  "codError": "0",
  "desError": "",
  "numTicket" : "21a9abd2-216c-418e-846c-0059bfa1aeb6" -
  - (Numero de ticket de referencia)
}
```

Si el valor de la variable “sucess” es true, existe una declaración original o de referencia para el semestre, si es false, no existe una declaración original o está en curso)

El valor de la variable: numTicket es el número de ticket de referencia.

Respuesta REST:

Ejemplo response si esta existe una declaración en curso:

```
{
  "sucess": false,
  "codError": "0",
  "desError": "",
  "numTicket": ""
}
```

6.3. REST validación vencimiento

El servicio valida la fecha de vencimiento con el cual se verifica si se encuentra en la fecha de envío de información y de acuerdo a la fecha fin de vencimiento se asigna el tipo de declaración y nombre del archivo para las sustitutorias y rectificatorias.

Si la fecha de la declaración es anterior o igual a la fecha de vencimiento corresponde hacer una declaración sustitutoria.

Si la fecha de la declaración es posterior a la fecha de vencimiento corresponde realizar una declaración rectificatoria.

Estructura REST:

Método	GET
Header	
URL	https://api-eeff.sunat.gob.pe/v1/auditoria/tributaria/consultavenc/{numRuc}
Ejemplo:	https://api-eeff.sunat.gob.pe/v1/auditoria/tributaria/consultavenc/20100066603

Ejemplo response:

```
{
  "fecInicio": "01/09/2021", -- (Fecha inicio de vencimiento)
  "fecFin": "17/09/2021", -- (Fecha fin de vencimiento)
  "codError": "0",
  "desError": ""
}
```

Para el nombre del archivo en caso sea el tipo declaración 02 se debe crear el archivo ZIP con las iniciales:

Sustitutoria=SUS (Antes del vencimiento)

Rectificatoria=REC (después del vencimiento)

6.4. REST envío de información

Estructura REST:

Método	POST
Header	numRuc:20100066603 codUsuario:MODDATOS
URL	https://api-eeff.sunat.gob.pe/v1/auditoria/tributaria/enviodef
Ejemplo:	https://api-eeff.sunat.gob.pe/v1/auditoria/tributaria/enviodef

Ejemplo Request Original:

```
{
  "numRuc": "20100066603",
  "tipDecl": "01", -- (01 = Original, 02 = Rectificatoria o Sustitutoria)
  "anioDecl": "2021",
  "semDecl": "06", -- 06 (Primer Semestre) y 12 (Segundo Semestre)
  "afiTransInt": "01", -- (01 = Sí; 02 = No)
  "numPeriodos": 3, -- (Valores desde 0 hasta 6)
  "periodos": "202101,202102,202103", -- Conjunto de periodos
  "indProc": "1", -- (1: APP Cliente, 2: Servicio Rest)
  "fecEnvio": "2020-11-16 16:34:08",
  "cantArch": "9",
  "nomArchZip": "20100066603-202106-ORI.zip",
  "desHashZip": "396a3a710629e02ecabab72acf3448e1f9178853d0129a40b6df441970c27b2d",
  "tamArchZip": "100gb"
}
```

Ejemplo Request Sustitucion:

```
{
  "numRuc": "20100066603",
  "tipDecl": "02",
  "anioDecl": "2021",
  "carDecl": "01", --(01: Total, no existe por periodos para SUS)
  "numTicketRef": "9d991c56-bb6f-4c9b-bf2d-493e63058914", -
  - (Declaración a quien se sustituye)
  "semDecl": "01",
  "afiTransInt": "01",
  "numPeriodos": 3,
  "periodos": "202101,202102,202103",
  "indProc": "1",
  "fecEnvio": "2020-11-16 16:34:08",
  "cantArch": "9",
  "nomArchZip": "20100066603-202101-SUS.zip",
  "desHashZip": "e802389c03a99a94eb25094d4b306da41581e584d16edf1d06e577f11d12c891",
  "tamArchZip": "100gb"
}
```

Ejemplo Request Rectificaroria:

```
{
  "numRuc": "20100066603",
  "tipDecl": "02",
  "anioDecl": "2021",
  "carDecl": "01", --(01: Total, 02: Por periodos)
  "numTicketRef": "9d991c56-bb6f-4c9b-bf2d-493e63058914", -
  - (Declaración a quien se rectifica)
  "semDecl": "01",
  "afiTransInt": "01",
  "numPeriodos": 3,
  "periodos": "202101,202102,202103",
  "indProc": "1",
  "fecEnvio": "2020-11-16 16:34:08",
  "cantArch": "9",
  "nomArchZip": "20100066603-202101-REC.zip",
  "desHashZip": "e802389c03a99a94eb25094d4b306da41581e584d16edf1d06e577f1
1d12c891",
  "tamArchZip": "100gb"
}
```

Response: Original

```
{
  "numRuc": "20132218472",
  "numTicket": "bc2bf589-f4b1-4d84-877c-1a1f384abdf3",
  "fecRecepcion": "2020-11-16T16:34:13", --(Guardar en pre-declaración)
  "tipDecl": "01", -
  - (01 = Original, 02 = Rectificatoria o Sustitutoria)
  "anioDecl": "2021",
  "semDecl": "06", -- (06=Primer Semestre y 12= Segundo Semestre)
  "codEstado": "01", -
  - (01:recepionado, 02:procesando, 03: validado ok, 04:validado error)
  "cantArch": "9",
}
```

Response: Sustitutorio o Rectificadorio

```
{
  "numRuc": "20132218472",
  "numTicket": "bc2bf589-f4b1-4d84-877c-1a1f384abdf3",
  "numTicketRef": "bc2bf589-f4b1-4d84-877c-1a1f384abdf3",
  "fecRecepcion": "2020-11-16T16:34:13", --(Guardar en pre declaración)
  "tipDecl": "02", -
  - (01 = Original, 02 = Rectificatoria o Sustitutoria)
  "anioDecl": "2021",
  "semDecl": "06", -- (06=Primer Semestre y 12= Segundo Semestre)
  "codEstado": "01", -
  - (01:recepionado, 02:procesando, 03: validado ok, 04:validado error)
  "cantArch": "9",
}
```

Mensajes de error:

```
{
  "cod": "500",
  "msg": "Internal Server Error - Se presento una condicion inesperada que impidio completar el Request",
  "exc": "java.lang.NullPointerException at ..."
}
{
  "cod": "422",
  "msg": " Los datos de entrada no cumplen con .....",
  "exc": ""
}
```

Cuadro de mensajes de error del receptor:

Cuadro de mensajes de Error	
cod	1504
msg	El contribuyente no se encuentra afiliado al EEFF
cod	1505
msg	El RUC del documento no coincide con el RUC del usuario
cod	1506
msg	No corresponde al tipo de declaración
cod	1507
msg	El usuario no ha enviado ningún archivo adjunto
cod	1508
msg	El nombre del archivo no corresponde al formato (RUC-AÑOSEMESTRE-TIPODECL.zip)
cod	1509
msg	No se tiene registrado ninguna declaración al cual hace referencia (Sus o Rec)
cod	1510
msg	Periodos no corresponden al semestre informado
cod	1511
msg	El hash no corresponde (Archivo manipulado)
cod	1512
msg	No envía campo obligatorio
cod	1513
msg	El número de archivos enviados no coincide con el número de periodos
cod	1514
msg	Ya existe una declaración original
cod	1515
msg	La fecha informada se encuentra fuera de vencimiento
cod	1516
msg	No corresponde realizar una declaración rectificatoria antes de la fecha de vencimiento
cod	1517
msg	No se puede presentar una declaración sustitutoria por periodos.
cod	1518
msg	No corresponde realizar una declaración sustitutoria después de la fecha de vencimiento

✓ **Caso archivo cero**

Se presenta cuando el obligado no tiene información para declarar, sin embargo, debe cumplir con la declaración enviando un archivo vacío.

Ejemplo Request Original:

```
{
  "numRuc": "20100066603",
  "tipDecl": "01",
  "anioDecl": "2021",
  "semDecl": "06",
  "afiTransInt": "01",
  "numPeriodos": 0, -- (Debe especificar el valor 0)
  "periodos": "", --(Debe especificar el vacío el campo)
  "indProc": "2",
  "fecEnvio": "2020-11-16 16:34:08",
  "cantArch": "0", -- (Debe especificar el valor 0)
  "nomArchZip": "20100066603-202106-ORI.zip", --(Debe especificar el
  nombre del archivo vacío)
  "desHashZip": "396a3a710629e02ecabab72acf3448e1f9178853d0129a40b6df4419
  70c27b2d", --(Puede consignar el valor del hash o vacío)
  "tamArchZip": "0gb" -- (Debe especificar el valor "0gb")
}
```

Ejemplo Request Sustitucion:

```
{
  "numRuc": "20100066603",
  "tipDecl": "02",
  "anioDecl": "2021",
  "carDecl": "01",
  "numTicketRef": "9d991c56-bb6f-4c9b-bf2d-493e63058914",
  "semDecl": "01",
  "afiTransInt": "01",
  "numPeriodos": 0, -- (Debe especificar el valor 0)
  "periodos": "", --(Debe especificar el vacío el campo)
  "indProc": "1",
  "fecEnvio": "2020-11-16 16:34:08",
  "cantArch": "0", -- (Debe especificar el valor 0)
  "nomArchZip": "20100066603-202101-SUS.zip", --(Debe especificar el
  nombre del archivo vacío)
  "desHashZip": "e802389c03a99a94eb25094d4b306da41581e584d16edf1d06e577f1
  1d12c891", --(Puede consignar el valor del hash o vacío)
  "tamArchZip": "0gb" -- (Debe especificar el valor "0gb")
}
```

Ejemplo Request Rectificatoria:

```
{
  "numRuc": "20100066603",
  "tipDecl": "02",
  "anioDecl": "2021",
  "carDecl": "01",
  "numTicketRef": "9d991c56-bb6f-4c9b-bf2d-493e63058914",
  "semDecl": "01",
  "afiTransInt": "01",
  "numPeriodos": 0, -- (Debe especificar el valor 0)
  "periodos": " ", --(Debe especificar el vacío el campo)
  "indProc": "1",
  "fecEnvio": "2020-11-16 16:34:08",
  "cantArch": "0", -- (Debe especificar el valor 0)
  "nomArchZip": "20100066603-202101-REC.zip", --(Debe especificar el nombre del archivo vacío)
  "desHashZip": "e802389c03a99a94eb25094d4b306da41581e584d16edf1d06e577f11d12c891", --(Puede consignar el valor del hash o vacío)
  "tamArchZip": "0gb" -- (Debe especificar el valor "0gb")
}
```

6.5. REST servicio de envío de archivo TUS

El servicio TUS permite la transferencia del archivo ZIP usando upload parciales hacia el receptor de SUNAT en el siguiente formato:

El formato del nombre del archivo es la siguiente:

<{RUC_OBLIGADO}><{AÑOSEMESTRE}><{TIPO_DECLARACION}.<{EXTENSION}>

Los campos estarán separados por guion ("-") y la extensión luego de un punto donde:

- {RUC_OBLIGADO}: debe de ser igual al número de RUC que se usó para acceder al sistema.
- {AÑOSEMESTRE}: debe de tener el formato YYYYMM (Año y semestre (06,12)).
- {TIPO_DECLARACION}: debe ser: ORI: Original, SUS: Sustitutoria, REC: Rectificatoria
- {EXTENSION}: debe de ser "zip" o "ZIP"

Ejemplo:

20100047218-202106-ORI.zip

20100047218-202106-SUS.zip

20100047218-202106-REC.zip

Estructura servicios REST TUS:

Método	POST
Header	
URL	https://api-eeff.sunat.gob.pe/v1/auditoria/tributaria/analisis/t/enviodeffile https://api-eeff.sunat.gob.pe/v1/auditoria/tributaria/enviodeffile/{idFile}
Ejemplo:	https://api-eeff.sunat.gob.pe/v1/auditoria/tributaria/enviodeffile/a1a4a2-44a4as1-1a4a4s

6.6. REST servicio consulta

Servicio de consulta que permite verificar el estado de la declaración, el detalle de los errores en caso de rechazo, y que descarga de la constancia de presentación (CDP) o Constancia de Rechazo (CDR) en caso la declaración no se presenta satisfactoriamente.

Estructura REST:

Método	POST
Header	
URL	https://api-eeff.sunat.gob.pe/v1/auditoria/tributaria/consultadef
Ejemplo:	

Consulta por periodo: Devuelve solo el periodo definido en el campo “semDecl”

```
{
  "numRuc": "20100066603",
  "fecDesde": "",
  "fecHasta": "",
  "anioDecl": "2021",
  "semDecl": "06" -- 06 (Primer Semestre) y 12 (Segundo Semestre)
}
```

Consulta por fecha: Devuelve todas las declaraciones enviadas en el rango de fecha “fecDesde” y “fecHasta”

```
{
  "numRuc": "20100066603",
  "fecDesde": "2021-08-08 00:00:00",
  "fecHasta": "2021-08-18 23:59:59",
  "anioDecl": "",
  "semDecl": ""
}
```

Respuesta de la consulta:

```
[
  {
    "desDecl": "202106-SUS", -
    - ( YYYYSS: Año y semestre, SUS= Tipo declaración)
    "numTicket": "bef244c3-51e7-4081-bfe1-c21d2bef870f",
    "fecRecep": "2021-08-19 11:55:08",
    "fecProc": "2021-08-19 11:59:32",
    "codEstado": "03", -
    - (01:recepcionado, 02:procesando, 03: validado ok, 04:validado error)
    "numTotalfile": "3",
    "numfileOK": "3",
    "numfileError": "0",
    "archivoError": [], -
    - (Si existe algun error devuelve el detalle del error)
    "archivoCDP": null,
    "archivoCDR": null
  },
  {
    "desDecl": "202106-SUS",
    "numTicket": "c9b61620-d86c-478a-8a6e-e0a2fd3ea1b2",
    "fecRecep": "2021-08-19 12:04:33",
    "fecProc": null,
    "codEstado": "04",
    "numTotalfile": "3",
    "numfileOK": "2",
    "numfileError": "1",
    "archivoError": [
      {
        "nombreamensajeError": "20100066603-202102-DT_ERROR",
        "archivoErrorString": "Error|Mensaje Error|Linea\n1163|
Titular no tiene cuentas asociadas|[583, 585, 507, 592, 595, 516, 519,
604, 524, 527, 613, 802, 560, 637,905, 908, 910, 916]",
        "archivoErrorBytes": "RXJyb3J8TWVuc2FqZSBFcnJvcnxMaW51Y
QoxMTYzffFRpdHVsnjksIDc3MSwgNzczLCA3NzsIDg4MywgODg2LCA4ODgsIDg5MCwgODkyL
CA40TUsIDg5OCwgODk5LCA5MDEsIDkwMywgOTA1LCA5MDgsIDkxMCwgOTE2XQ=="
      }
    ],
    "archivoCDP": null,
    "archivoCDR": null
  }
]
```

- REST consulta CDP y CDR

Estructura REST:

Método	POST
Header	
URL	https://api-eeff.sunat.gob.pe/v1/auditoria/tributaria/reporte/{numTicket}
Ejemplo:	https://api-eeff.sunat.gob.pe/v1/auditoria/tributaria/reporte/12406aee-ef3b-42a1-b638-fa1128e0179a

Para mayor detalle de las especificaciones de los servicios REST consultar la lista en el Anexo 01_Especificaciones Receptor.xlsx.

7. Validación en el receptor de la declaración

El receptor tiene un conjunto de validaciones para cada tipo de archivo:

Datos del titular: DT, Datos de cuenta: DC, Cuenta Titular: CT

Se inicia por validaciones de estructura, consistencia entre datos y finalmente los datos paramétricos.

Las validaciones se muestran en el Anexo: **8.2. Anexo 02_Validaciones Receptor.xlsx**

8. Anexos

8.1. Anexo 01_Especificaciones Receptor.xlsx



Anexo
01_Especificaciones.

8.2. Anexo 02_Validaciones Receptor.xlsx



Anexo
02_Validaciones Rec