# ESPECIFICACIONES TÉCNICAS PARA SERVICIOS WEB

Servicio para Generar la solicitud de pago, mediante el requerimiento de anticipo.

**Historial de Revisiones**

|  |  |  |
| --- | --- | --- |
| Fecha | Versión | Descripción |
| 09/12/2022 | 1.0 | El presente Documento expone los alcances para la generación del servicio web “Servicio para Generar la solicitud de pago, mediante el requerimiento de anticipo.” necesarios para la integración con SIREGOB. |
| 27/01/2023 | 1.0.1 | Ajustes al Documento |

**Documento de Entendimiento**

1. **Introducción**

Actualmente se está desarrollando una Plataforma para la distribución de los Recursos a los municipios en donde se requiere la sincronización con el sistema SIREGOB, para la Generación de una solicitud de pago desde la plataforma de distribución.

* 1. **Gerencia o Departamento Solicitante**

Coordinación de Gestión de la Información

* 1. **Propósito**

Generar la solicitud de pago del requerimiento de anticipo desde la plataforma “Distribución de recursos” y sincronizar las operaciones que el usuario esté realizando a la plataforma SIREGOB, mediante un Servicio REST.

* 1. **Definiciones, Acrónimos, y Abreviaturas**

N/A: No aplica.

U. Resp: Unidad Responsable

1. **Descripción del Requerimiento**

Publicar Servicio Rest/API que será consumido por un cliente para generar la solicitud de pago desde el requerimiento de anticipo, determinado por ciertos parámetros que serán enviados al consumir el servicio.

El requerimiento de anticipo se deberá generar con los datos que se le enviaran como parámetros, por ejemplo.

* U. Resp
* Usuario
* Clave de operación
* Tipo de Pago
* Clasificación de órdenes de pago
* Agrupar por proveedor
* Pago con desglose (Pago por beneficiario)
  1. **Protocolo Servicio Web**

se propone utilizar el protocolo de transferencia de datos REST (Representational State Transfer)

* 1. **Método de Servicio**

Para el uso del servicio debe ser mediante el método POST, ya que es el más recomendado para la trasferencia de información

* 1. **Formato de Salida**

Formato: JSON

Tipo MIME: application/json

* 1. **Parámetros de entrada**

A continuación, se describen los parámetros de entrada que debe considerar el servicio

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CAMPO** | **TIPO** | **OBLIGATORIO** | **DESCRIPCION** | **EJEMPLO** |
| USUARIO | VARCHAR(20) | SI | Describe el usuario con el que se está realizando la operación | VCARDENAZ |
| U.Resp | VARCHAR(4) | SI | Describe la unidad responsable que tiene relacionada el usuario | 0405 |
| Clave de operación | VARCHAR(20) | SI | Describe de que tipo es el requerimiento de anticipo | 68665 |
| Tipo de Pago | Varchar(10) | SI | Describe el tipo de pago de la solicitud | AN |
| Clasificación de orden de pago | Varchar(10) | SI | Describe la clasificación de la solicitud de pago | OPMP |
| Agrupar por Proveedor | Boolean | NO | Describe si la solicitud de pago se agrupara con proveedores | True / false |
| Pago con desglose/pago por beneficiario | Boolean | No | Describe si la solicitud de pago se va a distribuir entre los beneficiarios | True / false |

Ejemplo de datos a enviar

[{

USUARIO:'VCARDENAZ',

U.Resp:0405,

Clave de operación:69521,

Tipo de Pago:'PA',

Clasificación de orden de pago:'OPMP',

Agrupar por Proveedor:false,

Pago con desglose/pago por beneficiario:false

}]

\*\*Las operaciones deben se ser registrar como si el usuario ingresara directamente al sistema a hacerlas, se deben contemplar los cálculos en caso de requerirse.

* 1. **Parámetros de Salida**

A continuación, la Salida estándar del Servicio Web,

Siempre regresando una estructura de la siguiente forma:

**ESTATUS**: debe regresar el número de estatus de la petición http tomando el siguiente criterio

200: si la petición fue un éxito

400: si existe error en conexión con el cliente

500: si existe error en conexión con el servidor

550: si existe un error que impida completar la operación, (falta de permisos, falta de datos, error en la operación)

**MENSAJE**: debe regresar el mensaje de la solicitud, por ejemplo, si hubo un error, si fue un éxito, o si existe algún mensaje que no permitió completar la solicitud.

**DATA**: debe regresar la clave de la orden de pago registrada y el estatus de la operación (INGRESANDO OPERACION) en caso de error debe regresar el número y mensaje de error.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CAMPO | TIPO | OBLIGATORIO | DESCRIPCION | EJEMPLO |
| Estatus | Numérico | Si | Indica el estatus de la solicitud | 200: la acción se realizó correctamente (Succesful)  300: se ha tenido que tomar una acción adiconal(Redirección)  400: (Errores del cliente)  500: (Errores del servidor)  550: (falta de permisos, falta de datos, error en la operación) |
| Mensaje | Varchar(2000) | Si | Indica el mensaje de respuesta de la solicitud | “Solicitud Exitosa” /  “Usuario no tiene permisos” /  “Error …”  Etc.. |
| Data | Json Object | SI | Indica la Respuesta de la petición , esta puede ser una lista de objetos en formato json | Lista de Objeto, con la respuesta de la consulta  [{  Numero de orden de pago:69413,  Estatus:”INGRESANDO OPERACION”  }]  O en caso de error  [{  Numero de error:56685,  Mensaje de error: ”No existe …”  }] |

1. **Limitaciones**

Se debe definir la forma de autenticación para poder consumir el servicio

1. **Glosario**

**Parámetros**: Los parámetros de una función son los valores que esta recibe por parte del código que la llama. Pueden ser tipos simples u objetos.

**Protocolo**: En redes, un protocolo de comunicaciones o protocolo de red es la especificación de una serie de reglas para un tipo particular de comunicación. La red Internet se basa en el modelo de referencia TCP/IP (Transmission Control Protocol/Internet Protocol) que toma su nombre de los dos principales protocolos que regulan la comunicación a través de esta red.

**Proxies**: Un proxy (representante) es un agente o sustituto autorizado para actuar en nombre de otra máquina que lo autoriza a hacerlo y puede utilizarse en el contexto de servidor proxy, que hace de intermediario en las peticiones de recursos que realiza un cliente (A) a otro servidor (C).

**REST:** Representational State Transfer - Arquitectura que, haciendo uso del protocolo HTTP, proporciona una API que utiliza cada uno de sus métodos (GET, POST, PUT, DELETE) para poder realizar diferentes operaciones entre la aplicación que ofrece el servicio web y el cliente.

**Servicio Web:** Conjunto de aplicaciones o de tecnologías con capacidad para interoperar en la Web. Estas aplicaciones o tecnologías intercambian datos entre sí con el objetivo de ofrecer unos servicios. Los proveedores ofrecen sus servicios como procedimientos remotos y los usuarios solicitan un servicio llamando a estos procedimientos a través de la Web.

**SOAP**: Protocolo Simple de Acceso a Objetos, el cual es un protocolo basado en XML, que permite la interacción entre varios dispositivos y que tiene la capacidad de transmitir información compleja.