ESQUEMA DE AUTENTICACIÓN DE SISTEMAS EXTERNOS PARA INTEROPERABILIDAD

IDENTIDAD
v.1.3.1

Identificador del Documento:	IDENTIDAD
Nombre del documento:	ESQUEMA DE AUTENTICACIÓN DE SISTEMAS EXTERNOS PARA INTEROPERABILIDAD
Estado del documento:	

Versión	Creación	Descripción Cambio	Autor/es
1.0	19/01/2018	Especificación interoperabilidad.	ERNESTO MEDINA, ANGELA SUAREZ, ARMANDO FRADE CLIMACO LLAMAS
1.1	05/03/2018	Se agregan ejemplos de las tramas de identidad.	ERNESTO MEDINA, ANGELA SUAREZ CLIMACO LLAMAS
1.2	04/04/2018	Soporte encriptación ClientSecret y Password en método login	ERNESTO MEDINA, ANGELA SUAREZ CARLOS DIAZ CLIMACO LLAMAS
1.3	04/15/2018	Alcance de seguridad detallado.	ERNESTO MEDINA CLIMACO LLAMAS
1.3.1	04/25/2018	Alcance de seguridad detallado.	ERNESTO MEDINA CLÍMACO LLAMAS
1.3.2	27/07/2018	Manual usuario registro de aplicaciones	FABIAN HUERFANO CLIMACO LLAMAS

Contenido

1. Introducción

1.1. Objetivo

Esta especificación de autenticación está orientada a la habilitación de la interoperabilidad con los sistemas Muisca de la DIAN. Describe los mecanismos y requerimientos para que un sistema externo (aplicación) pueda obtener un token de acceso para el consumo de servicios.

Se expone también el esquema general de seguridad que aplicará para el consumo de cualquier servicio Muisca desde el año 2017 en adelante, así que debe ser utilizado como referencia para cualquier proceso de interoperabilidad con la DIAN.

1.2. Terminología

Para facilidad del entendimiento de este documento, se establece la siguiente terminología de uso común.

Término / Abreviatura	**Descripción**
TOKEN	También conocido como token de autenticación o token criptográfico es un elemento electrónico que se le da a un usuario autorizado de un servicio computarizado para facilitar el proceso de autenticación.
JWT	JSON Web Token es un estándar abierto (RFC-7519) basado en JSON para crear un token que sirva para enviar datos entre aplicaciones o servicios y garantizar que sean válidos y seguros. El caso más común de uso de los JWT es para manejar la autenticación en aplicaciones móviles o Web.
REST-Based Web Service	Representational State Transfer (REST por sus siglas en ingles), son una forma de proveer interoperabilidad entre sistemas. Los servicios que cumplen con los requerimientos REST permitir a sistemas acceder y manipular representaciones de Recursos Web usando una forma única y predefinada de operaciones sin estado.
JSON	JavaScript Object Notation, es un formato mínimo y legible para estructurar datos. Es utilizado para la transmisión de datos entre aplicaciones web como una alternativa al XML.
CLIENTE	Aplicación externa que quiere hacer interoperabilidad con los servicios de la entidad.
TOKEN ENDPOINT	Para el caso de la DIAN, el token endpoint es sinónimo del servicio de identidad de la organización, quien es el encargado de administrar el ciclo de vida del token.

2. Seguridad

2.1. Generalidades

Para el consumo por un actor externo de cualquier servicio ofrecido por la DIAN, se debe tener en cuenta que:

- Como estándar para el intercambio de mensajes para los servicios se debe utilizar HTTPS conforme al protocolo HTTPS (HTTP + TLS v1.2) con las siguientes características
    - Suites de cifrado: ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-SHA384:ECDHE-RSA-AES256-SHA384:ECDHE-ECDSA-AES128-SHA256:ECDHE-RSA-AES128-SHA256.
    - Tipo de certificado: ECDSA
    - Tamaño mínimo de la llave del certificado de comunicación segura: 2048
    - Algoritmo de firma del certificado de comunicación: sha256WithRSAEncryption, ecdsa-with-SHA256, ecdsa-with-SHA384, ecdsa-with-SHA512

2.2. Autenticación basada en dirección IP.

La DIAN se reserva el derecho de definir cuando un servicio se asegurará por IP. De ser así, solo permitirá el acceso al servicio(s) para aquellas direcciones IP que se encuentren registradas, de tal manera que se garantice que el acceso al servicio sea privado. Sin embargo, esta condición no aplica para los servicios de autenticación, ya que estos son de uso general.

Si la autenticación por IP es requerida, el actor externo está obligado a entregar durante la configuración del servicio a consumir las direcciones IP públicas para poder habiltar el acceso.

2.3. REST-Based Web Service

La información de autenticación va ubicada en el header de la solicitud de cualquier servicio REST. A continuación, se presenta un ejemplo de este:

Authorization: Bearer {Token obtenido en la autenticación del sistema}
ClientId: {ClientId de la aplicación registrada}

Ejemplo:

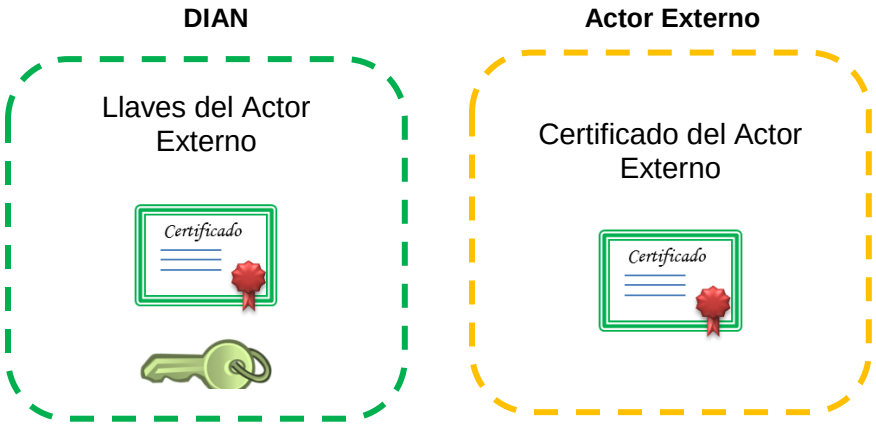
…

Authorization: Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI5MDAwMDAwMDA2MDAwNC0wLTg5MDkwMzkzOCIsImF1ZCI6WyJLVHNsTEg3UmhOeWxHOGZHQjlXM0FScjhmbEFhIl0sImlzcyI6Imh0dHBzOlwvXC9tdWlzY2EuZGlhbi5nb3YuY28iLCJleHAiOjE1MjIyNjEyNTIsImlhdCI6MTUyMjI1NzY1Miwicm9sIjoiaHR0cHM6XC9cL2dvb2dsZS5jb20uY28iLCJqdGkiOiIzZDMzOWFjZC1lMzM1LTRiMGUtYjFlYy01NTgzYzQxY2M5NTcifQ.IX1O_b5kpK_gX4He8JzX-NiYuhOB4sUdGvvH5r8rl2w
ClientId: KTslLH7RhNylG8fGB9W3ARr8flAa
…

2.4. Seguridad a nivel de canal con HTTPS.

El HTTPS garantiza la seguridad del canal entre los host o servidores relacionados, en comparación de la VPN que sólo llega hasta los equipos de borde. Sin embargo, en ocasiones se ponen servidores intermediarios que terminan la conexión TLS como balanceadores de carga, firewalls de aplicación, entre otros.

Para la implementación del HTTPS se requiere un certificado digital por parte de DIAN. Este certificado digital debe ser instalado en el servicio que se expone hacia el actor externo y también debe ser enviado al actor externo para poder autenticar al servidor (DIAN). Éste certificado sólo será recibido por el actor externo si se envía de la persona que acompaña el proceso de implementación.



Se utilizan certificados digitales vigentes, emitidos por entidades certificadoras reconocidas, cuyo propósito permita el cifrado de información. El certificado digital debería tener las siguientes características:

Característica	Valor
Tipo	X509v3
Algoritmo de firma	SHA2
Algoritmo de llave pública	RSA

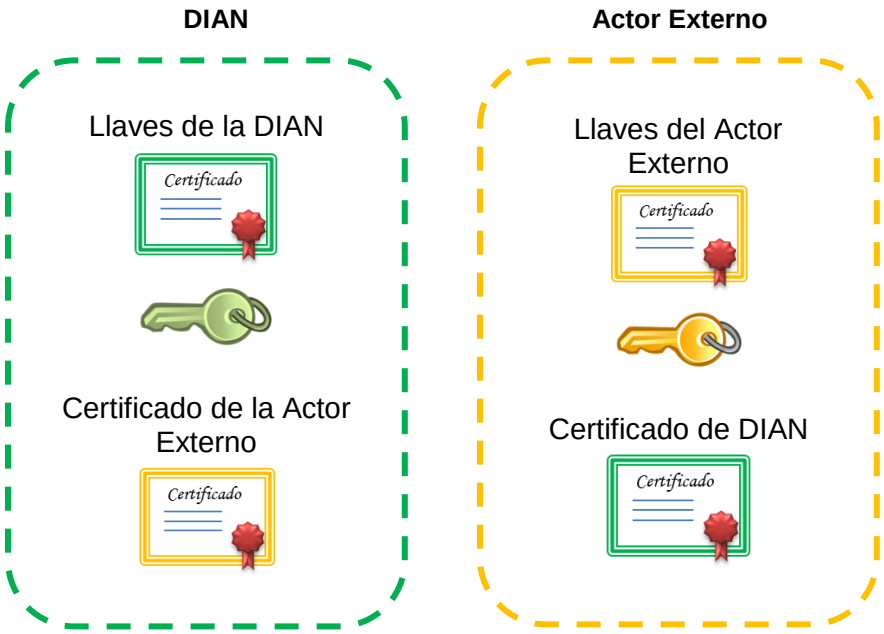
Característica	Valor
Tamaño mínimo de llave pública (bits)	2048
Vigencia	Mínimo un (1) año y máximo tres (3) años
Entidad certificadora	Reconocida a nivel internacional (Ej: VeriSign)
Common Name del certificado (CN)	Razón social de la DIAN. En el actor externo se usa NOMBRE ACTOR.
Atributos	Cifrado (certificado Web)

2.5. Autenticación de cliente en HTTPS.

El TLS en el que se basa el HTTPS tiene una opción donde se puede definir que se requiera autenticación del cliente. Para esto el servidor le solicita al cliente su certificado digital, que es comparada con los certificados válidos y finalmente lo autentica mediante procesos criptográficos que garantizan que el cliente tenga la llave privada asociada.

Esto sólo se puede realizar cuando se tiene establecida la conexión HTTPS adicionalmente el cliente posee instalado un certificado digital para este propósito. El actor externo posee el certificado digital y está en la capacidad de utilizar esta característica.

Para poder usar esta característica, el actor externo debe enviar a la DIAN su certificado digital. Éste será enviado por personal que acompañará la implementación. Los certificados y el intercambio de llaves entre las entidades deben quedan de la siguiente manera para el completo funcionamiento del HTTPS con esta funcionalidad.



2.6. Trazabilidad de transacciones.

Todas las transacciones procesadas por la DIAN deben dejar registro en un log transaccional que permita la trazabilidad de la transacción, grabando la información necesaria para cumplir con la regulación pertinente.

2.7. Disponibilidad del servicio.

Para garantizar la disponibilidad de la solución por parte de la DIAN y actor externo se requiere que la plataforma ofrezca redundancia y sea lo suficientemente robusta para soportar las peticiones de todos los actores externos.

2.8. Acerca de los formatos de comunicación

- Protocolo de aplicación: REST-Based Web Service (API Web),
- Formato de intercambio de datos en el contenido del mensaje: JSON
- Formato de codificación en el contenido del mensaje: UTF-8.
- Formato de datos de tipo Datetime: Los datos de tipo datetime (fecha y hora) se deben enviar usando el estándar ISO 8601, en formato YYYY-MM-DDThh:mm:ss, usando la hora local colombiana (para facilitar los procesos de comparación de fechas). Ejemplo, 2017-07-16T19:20:30.984.

3. Planteamiento general y consideraciones

El esquema de autenticación presenta tres escenarios/operaciones:

OBSERVACIONES

- La obtención de cualquier código de respuesta HTTP diferente a 200 OK en el consumo de servicios de autenticar, renovar o revocar se considera como error.

3.1. Autenticar.

Método HTTP: POST https://api.dian.gov.co/identidad/sts/v1/tokens/login
Método HTTP Pruebas: POST https://apipruebasexternas.dian.gov.co/identidad/sts/v1/tokens/login

Dentro de este contexto, el sistema externo obtendrá un token del sistema de identidad de la DIAN. Aquí se solicita la siguiente información para poder acceder a un token:

Entradas:

Parámetro/Objeto	Tipo Dato	Longitud	Tipo Parámetro	Descripcion
grant-type	String	15	Query	Atributo obligatorio que representa el tipo de operación, siempre será la palabra “password” literal. Ver Ejemplo.
client_id	String	50	Query	Representa la identificación de la aplicación que se conecta.
client_secret*	String	150	Query	Representa el resultado de cifrar el client_secret usando AES-128 conforme al procedimiento que se describe en el apartado 3.1.1. La longitud máxima del campo antes de cifrado es 50.
tipoDocumento	String	3	Query	Identifica el tipo de documento conforme la

				especificación Swagger. Debe ser "US" para la mayoría de escenarios externos.
nroDocumento	String	15	Query	Identificación del usuario
nit	String	15	Query	Identificación de la empresa.
Password*	String	150	Query	Representa el resultado de cifrar el password usando AES-128 conforme al procedimiento que se describe en el apartado 3.1.1. La longitud máxima del campo antes de cifrado es 15

***ESTOS ATRIBUTOS CORRESPONDEN A VALORES CIFRADOS Y DEBEN SER CALCULADOS CONFORME SE DESCRIBE EN EL APARTADO 3.1.1**

Salidas:

Un objeto **DToken** en el body del response con los siguientes atributos:

Atributo	Tipo Dato	Longitud	Descripción
clientId	String	50	Identifica el ClientId al que se le generá el token.
accessToken	String	50	Un identificador del token
idToken	String	500	Es el token JWT
refreshToken	String	50	Token de renovación automática
tokenType	String	15	Siempre será "Bearer"
expireIn	Number	NA	Identifica el tiempo de expiración del token en segundos.

ACLARACIONES

- La duración del token puede variar en el tiempo, producto de cambios en las políticas internas de la entidad. Por lo tanto, siempre se deberá verificar la vigencia de este al momento de la obtención para confirmar la fecha de expiración.
- En un ambiente de producción, donde existirán varios “servidores” clientes de Identidad, cada uno deberá autenticarse de forma independiente. Esto significa que cada “servidor” cliente deberá gestionar su token, así como renovarlo o revocarlo de forma autónoma.
- Solo deberían generarse token que van a ser utilizados. Cuando el token no se requiera más para su uso, deberá revocarse.
- No existen limitaciones para la cantidad de tokens generados por un cliente durante un tiempo particular. Sin embargo, esto no evita que las políticas y reglas en el uso de las APIs detecten los malos usos y generen restricciones al cliente.

3.1.1. Cifrado AES-128

A pesar de tener la opción de cifrado como obligatoria en ambiente producción, será posible en pruebas ejecutar la autenticación sin cifrado. Para tal efecto, solo se requiere informar a la DIAN para habilitar/deshabilitar el cifrado en la prueba. En producción, este proceso se podrá realizar por auto-gestión.

Para el cifrado de los atributos se debe realizar el siguiente procedimiento:

1. Se debe concatenar el contenido a encriptar con la fecha del sistema. (Debe estar sincronizado con la hora colombiana al minuto usando estándar ISO 8601).
   Ejemplo: [ClientSecret]-[2018-08-13T09:30:47] y [Password]-[2018-08-13T09:30:47]
2. Se aplica el algoritmo de encriptación AES-128 CBC al contenido utilizando el EncryptionKey (alfanumérico de 16 caracteres) dado por la DIAN al registrar la aplicación del Banco.
   Ejemplo del EncryptionKey: “68BD795F133F0132”
3. La DIAN descifrará el contenido, comparando que la fecha descifrada no sea menor en 1 minuto ni mayor en 5 minutos de la hora actual. Una vez verificado la hora exitosamente, realizará el proceso de autenticación normal usando el “clientId” y “password” obtenidas del descrifrado para retornar un token válido.

El proceso de encriptación con los atributos involucrados puede ser probado en la siguiente página (sugerida):

http://www.devglan.com/online-tools/aes-encryption-decryption

3.1.2. Ejemplo detallado del procedimiento para el Client Secret

Se tiene un client secret inicial así:

CLIENT SECRET Cadena Inicial:

trttwNZ0OMkHS7CC1_nFx6wlKnca

Definición Fecha:

2018-03-18T19:09:10

Formación cadena:

[trttwNZ0OMkHS7CC1_nFx6wlKnca]-[2018-03-18T19:09:10]

Características para el Cifrado de la cadena:
AES
CBC
128 Key Size

Ejemplo Encryption Key:
6A28CE819A9E001A

Definición Vector Ej:

```
byte[] iv = new byte[256 / 16];
IvParameterSpec ivspec = new IvParameterSpec(iv);
ivspec = new IvParameterSpec(new byte[16]);
```

Cifrado Aplicado

oTr1sm/Yk5lkJVkkRIj6QASsKukRahpvltJoXLes0q2XE9TqmgzZaq8jOF493SP7ZzOCLabdRtIAtW9ZpYf+AQ==

3.1.3. Ejemplo detallado del procedimiento para el Password

Se tiene un password inicial así:

PASSWORD Cadena Inicial:

Prueba2006

Definición Fecha:

2018-03-18T19:09:10

Formación cadena:

[Prueba2006]-[2018-03-18T19:09:10]

Características para el Cifrado de la cadena:
AES
CBC
128 Key Size

Ejemplo Encryption Key:
6A28CE819A9E001A

Definición Vector Ej:

```
byte[] iv = new byte[256 / 16];
IvParameterSpec ivspec = new IvParameterSpec(iv);
ivspec = new IvParameterSpec(new byte[16]);
```

Cifrado Aplicado

x0H+zSi8QK6Hr3SvlEdPr121Ck/XCL0YYzzxvVxl5ln2ez9v+zKwfBDD7fClauRG

3.1.4. Ejemplo de tramas

REQUEST

POST

https://apipruebasexternas.dian.gov.co/identidad/sts/v1/tokens/login?grant_type=password&client_id=wAyXOuElL_w01O8MyUhDLK_Z_Xsa&client_secret=oTr1sm/Yk5lkJVkkRIj6QASsKukRahpvltJoXLes0q2XE9TqmgzZaq8jOF493SP7ZzOCLabdRtIAtW9ZpYf+AQ==&tipoDocumento=US&nroDocumento=800130643&nit=800130643&passw

ord=
oTr1sm/Yk5lkJVkkRIj6QASsKukRahpvltJoXLes0q2XE9TqmgzZaq8jOF493SP7ZzOCLabdRtIAt
W9ZpYf+AQ==

Host: 10.255.5.103:9091

Connection: keep-alive

Content-Length: 0

Origin: chrome-extension://aejoelaoggembcahagimdiliamlcdmfm

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/63.0.3239.84 Safari/537.36

Content-Type: application/json

Accept: */*

Accept-Encoding: gzip, deflate

Accept-Language: en-US,en;q=0.9

RESPONSE

HTTP/1.1 200 OK

X-Powered-By: Undertow/1

Server: WildFly/10

Server: Restlet-Framework/2.3.7

Accept-Ranges: bytes

Date: Thu, 08 Mar 2018 15:53:53 GMT

Connection: keep-alive

Access-Control-Allow-Origin: chrome-extension://aejoelaoggembcahagimdiliamlcdmfm

Vary: Accept-Charset, Accept-Encoding, Accept-Language, Accept

Access-Control-Allow-Credentials: true

Content-Type: application/json;charset=UTF-8

Content-Length: 579

```
{
    accessToken: "18800b30-fa7d-3df4-a4a4-2c2736c42a7c",
    idToken: "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJleHAiOjE1MjAwMjk0NjA
    sInN1YiI6IjkwMDAwMDAwMDYwMDA0LTAtODkwOTAzOTM4IiwiYXVkIjpbIlBfeDhKakFZV
    npFTGdyb0lxdE02N0QydjFRUWEiXSwiaXNzIjoiaHR0cHM6XC9cL211aXNjYS5kaWFuLmd
    vdi5jbyIsImp0aSI6IjUxZWZiMDJiLWNjZDUtNDZlMi1iNjlmLWM5ZDljZTU1YjNmZSIsI
    mlhdCI6MTUyMDAyNTg2MH0.6kRI310oQMSBBb9ZSMaqV-GcdNzKwqz5QJnu9Jmbldg",
    tokenType: "Bearer",
    expireIn: 1101,
    refreshToken: "e161c94b-789e-3c4f-8903-310867440658",
    clientId: "P_x8JjAYVzELgroIqtM67D2v1QQa"
}
```


3.2. Refrescar Token.

Método HTTP: POST https://api.dian.gov.co/identidad/sts/v1/tokens/refresh
Método HTTP Pruebas: POST
https://apipruebasexternas.dian.gov.co/identidad/sts/v1/tokens/refresh

Dentro de este contexto, se busca obtener un nuevo token cuando el token entregado se encuentra próximo a vencer o vencido. Sólo se requiere el envío del refresh token en el Authorization

Entradas:

En el Header de la solicitud deben ir:

Parámetro	Tipo Parámetro	Longitud	Descripción
Authorization	Header	500	Corresponde a la unión de la palabra clave “Bearer” y el valor del parámetro “refreshToken” devuelto en el servicio “Autenticar”.
ClientId	Header	50	Se refiere al ClientId para el cual se generó el token.

Salidas:

Un objeto **DToken** en el body del response con los siguientes atributos:

Atributo	Tipo Dato	Longitud	Descripción
clientId	String	50	Identifica el ClientId al que se le generá el token.
accessToken	String	50	Un identificador del token
idToken	String	500	Es el token JWT
refreshToken	String	50	Token de renovación automática
tokenType	String	15	Siempre será “Bearer”
expireIn	Number	NA	Identifica el tiempo de expiración del token en segundos

Ejemplo de uso:

REQUEST

POST https://apipruebasexternas.dian.gov.co/identidad/sts/v2/tokens/refresh HTTP/1.1

Host: 10.255.5.103:9091

Connection: keep-alive

Content-Length: 0

Authorization: Bearer 64261f2a-f61a-343f-b730-581e11b93b38

ClientId: wAyXOuElL_w01O8MyUhDLK_Z_Xsa

Origin: chrome-extension://aejoelaoggembcahagimdiliamlcdmfm

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/63.0.3239.84 Safari/537.36

Content-Type: application/json

Accept: */*

Accept-Encoding: gzip, deflate

Accept-Language: en-US,en;q=0.9

RESPONSE

HTTP/1.1 200 OK

X-Powered-By: Undertow/1

Server: WildFly/10

Server: Restlet-Framework/2.3.7

Accept-Ranges: bytes

Date: Thu, 08 Mar 2018 16:07:06 GMT

Connection: keep-alive

Access-Control-Allow-Origin: chrome-extension://aejoelaoggembcahagimdiliamlcdmfm

Vary: Accept-Charset, Accept-Encoding, Accept-Language, Accept

Access-Control-Allow-Credentials: true

Content-Type: application/json;charset=UTF-8

Content-Length: 579


```
{
      accessToken: "0e9a0104-b26c-345b-8a04-96348671e1f5",
      idToken: "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI5MDAwMDAwMDA
      2MDAwMy0wLTgwMDEzMDY0MyIsImF1ZCI6WyJ3QXlYT3VFbExfdzAxTzhNeVVoRExLX1pfW
      HNhIl0sImlzcyI6Imh0dHBzOlwvXC9tdWlzY2EuZGlhbi5nb3YuY28iLCJleHAiOjE1MjA
      1Mjg4MjYsImlhdCI6MTUyMDUyNTIyNiwicm9sIjoiaHR0cHM6XC9cL2dvb2dsZS5jb20uY
      28iLCJqdGkiOiJmMTE3NGNlYy1lNjA0LTRkOTMtOTg4Mi00MTIxZDgxMTk0ZTkifQ._FyX
      kGUOYrr4LtCwmWNVjtW9mT0TJezubkykx3PgWos",
      tokenType: "Bearer",
      expireIn: 3600,
      refreshToken: "50be2bea-032d-31ac-963b-d46893d26a3b",
      clientId: " wAyXOuElL_w01O8MyUhDLK_Z_Xsa "
}
```


3.3. Revocar Token.

Método HTTP: POST https://api.dian.gov.co/identidad/sts/v1/tokens/revoke
Método HTTP Pruebas: POST
https://apipruebasexternas.dian.gov.co/identidad/sts/v1/tokens/revoke

Entradas:

En el Header de la solicitud deben ir:

Parámetro	Tipo Parámetro	Longitud	Descripción
Authorization	Header	500	Corresponde a la unión de la palabra clave "Bearer" y el valor del parámetro "idToken" devuelto en el servicio "Autenticar".
ClientId	Header	50	Se refiere al ClientId para el cual se generó el token.

Salidas:

Atributo	Tipo Dato	Longitud	Descripción
message	String	50	Mensaje de éxito de la revocación del token. "success"

Ejemplo de uso:

REQUEST

POST https://apipruebasexternas.dian.gov.co/identidad/sts/v1/tokens/revoke HTTP/1.1

Host: 10.255.5.103:9091

Connection: keep-alive

Content-Length: 0

Authorization: Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI5MDAwMDAwMDA2MDAw
My0wLTgwMDEzMDY0MyIsImF1ZCI6WyJ3QXlYT3VFbExfdzAxTzhNeVVoRExLX1
pfWHNhIl0sImlzcyI6Imh0dHBzOlwvXC9tdWlzY2EuZGlhbi5nb3YuY28iLCJleHAiOjE1
MjA1MjkxNjYsImlhdCI6MTUyMDUyNTU2Niwicm9sIjoiaHR0cHM6XC9cL2dvb2dsZS
5jb20uY28iLCJqdGkiOiI5ODRjZjg0Zi02ZGIzLTRmMmYtYjQ2Mi01YTc3OTE0MGI2Z
TcifQ.UFjN9tJ7JmTFVF6w3l_jvHQ9Q-YbScC1qh_Z-qU_wPU

ClientId: wAyXOuElL_w01O8MyUhDLK_Z_Xsa

Origin: chrome-extension://aejoelaoggembcahagimdiliamlcdmfm

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/63.0.3239.84 Safari/537.36

Accept: */*

Accept-Encoding: gzip, deflate

Accept-Language: en-US,en;q=0.9

RESPONSE

HTTP/1.1 200 OK

X-Powered-By: Undertow/1

Server: WildFly/10

Server: Restlet-Framework/2.3.7

Accept-Ranges: bytes

Date: Thu, 08 Mar 2018 16:13:05 GMT

Connection: keep-alive

Access-Control-Allow-Origin: chrome-extension://aejoelaoggembcahagimdiliamlcdmfm

Vary: Accept-Charset, Accept-Encoding, Accept-Language, Accept

Access-Control-Allow-Credentials: true

Content-Type: application/json;charset=UTF-8

Content-Length: 21

{"message":"success"}

4. Responsabilidades

La integración entre los clientes y los sistemas muisca tiene los siguientes planteamientos base al respecto de los flujos de información y las responsabilidades sobre los mismos.

- El CLIENTE se autentica con el servicio de identidad y obtiene un token válido de la entidad.

- El CLIENTE consume servicios utilizando el token recibido.

- El CLIENTE revoca el certificado una vez termina con las transacciones.

- El CLIENTE puede refrescar el token si está próximo a vencer, usando el servicio de refrescar.

- El CLIENTE es responsable del uso adecuado del token.

- El CLIENTE es responsable de notificar el uso indebido de un token generado cuando pierda control del mismo.

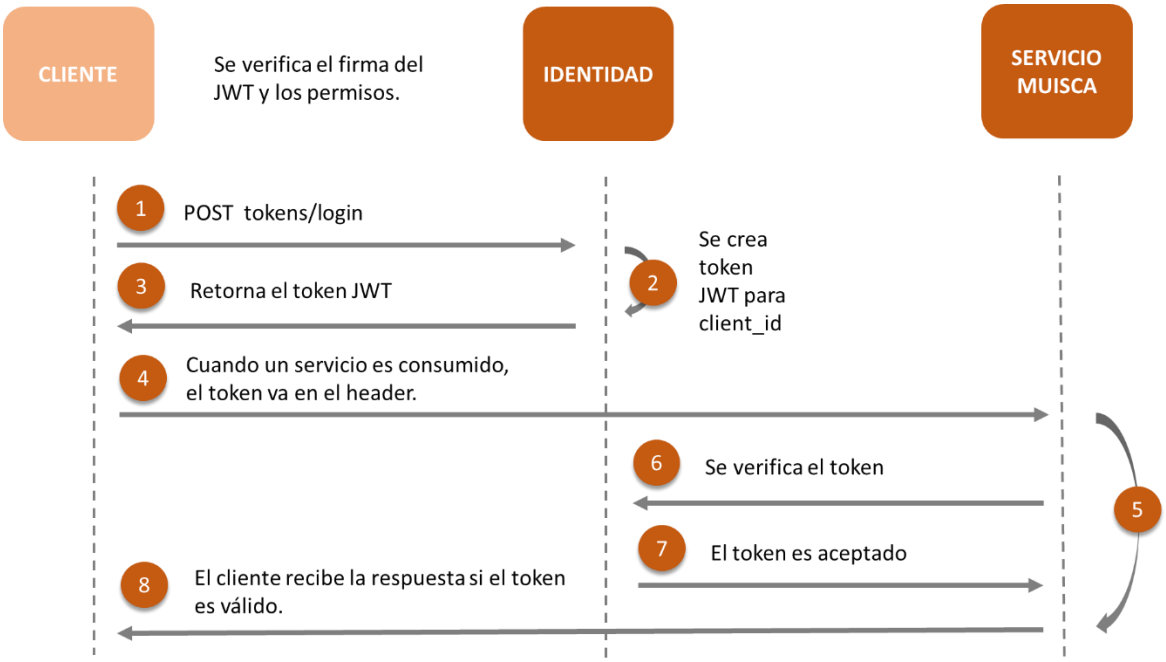
- El CLIENTE es responsable de revocar el token tan pronto no requiera consumir más servicios para evitar que siga vigente.

- La DIAN es responsable de garantizar la disponibilidad de los servicios conforme a los ANS generales de la organización para TI.

5. Registro de aplicaciones y flujos

5.1. Flujo general de consumo de un servicio.

A continuación, se presenta un diagrama de interacción para el consumo de un servicio Muisca:



Este modelo tiene las siguientes restricciones:

- Uso de Tokens basados en el estándar JWT (JSON Web Token) que brindará información sobre fechas de creación, uso y vencimiento de este, entre otros.
- La verificación de los Tokens deberá realizarse contra el Token Endpoint.
- Se limitarán las comunicaciones entrantes conforme a las políticas definidas por la entidad según el usuario mediante el registro de las direcciones IP

origen autorizadas utilizando el Firewall disponible. Esto se dará a conocer para el caso de acuerdos específicos que la entidad realice.

5.2. Confirmación de la recepción

El servicio que recibe la petición envía una respuesta HTTP al cliente. Los posibles códigos de respuesta HTTP generalmente usados en los servicios de la DIAN son:

- 200 – Petición recibida exitosamente. Pendiente de procesamiento.
- 204 – Para cuando no hay contenido (específicamente para peticiones tipo GET)
- 400 – Petición invalida (ErrorResponse - especificación técnica Swagger - es el objeto estándar para especificar la causa del rechazo y deberá ser entregada en el cuerpo de la respuesta).
- 401 – Credencial de autenticación inválida.
- 429 – Demasiadas peticiones. Por favor reintente más tarde.
- 500 – Error interno de servidor.
- 503 – Servicio no disponible. Reintente tarde.

5.3. Registro de aplicaciones Cliente (Web y Móviles)

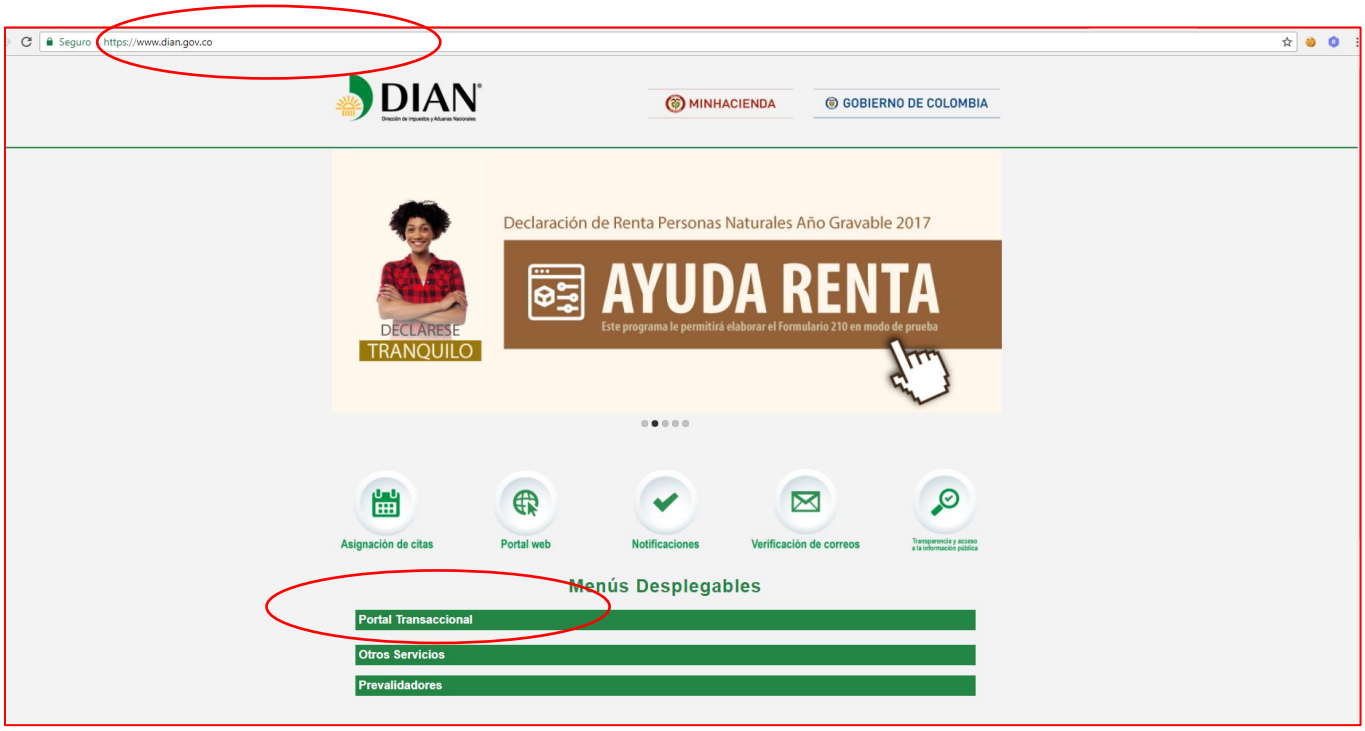
Para que una aplicación externa a la entidad pueda acceder al catálogo de servicios de la DIAN, es necesario registrar la aplicación previamente a través de los servicios de registro existente. Esto permite establecer una identificación a la aplicación y administrar sus características.

Cuando se registra una aplicación cliente, se recibe un **Client ID**. Este identificador es utilizado por la aplicación para identificarse tanto en los servicios a consumir como con los usuarios que deseen autorizar la aplicación. Igualmente, con cada aplicación se entrega un **Client Secret** que será solicitado al momento de la autenticación.

Desde esta función de auto-gestión también podrá cambiar o renovar el EncryptionKey utilizado para el proceso de cifrado AES-128.

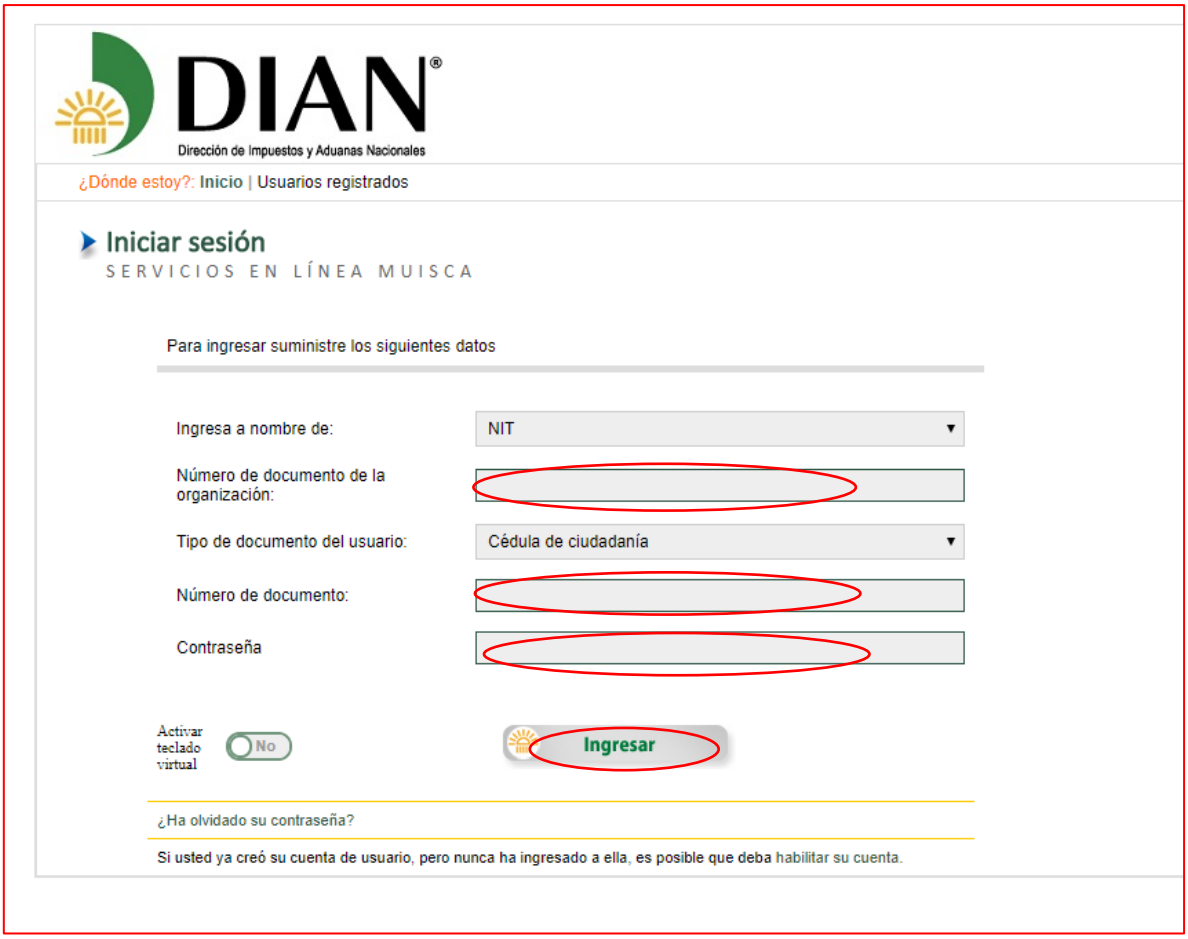
Los pasos para registrar una aplicación son:

1. Ingrese a la plataforma de la DIAN



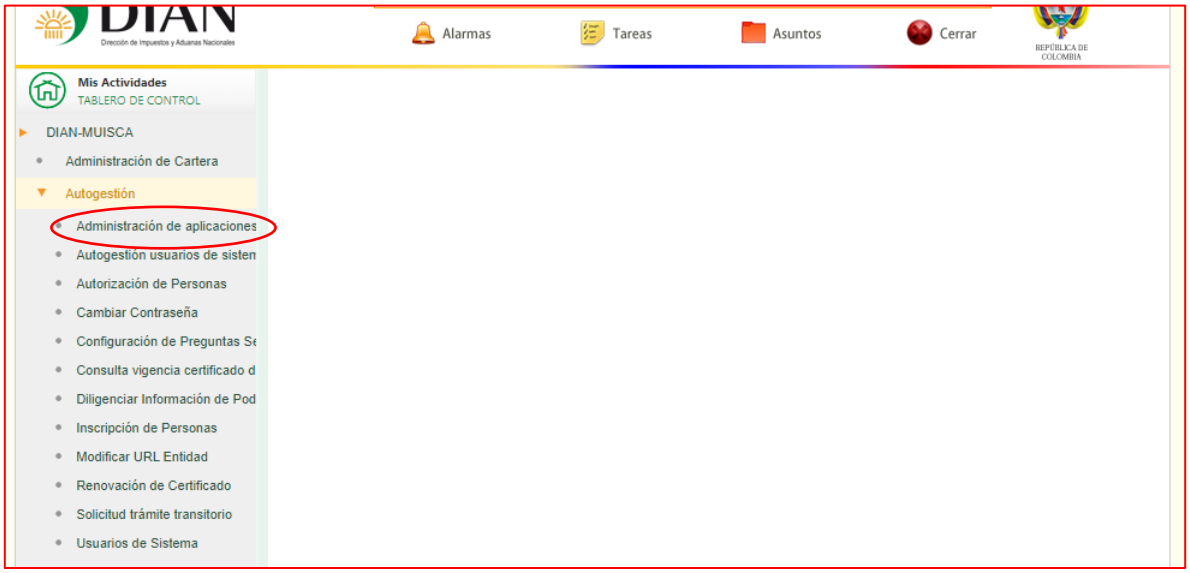
2. Seleccionar usuario registrado.

3. Registrar nit, tipo de documento, número de documento del representante legal principal, representante legal certificado o funcionario con funciones formales que va a realizar la autorización.



4. Localice la opción de autogestión en el menú de servicios.

5. Seleccione “Administrador de Aplicaciones”.



6. Instructivo de registro de aplicaciones

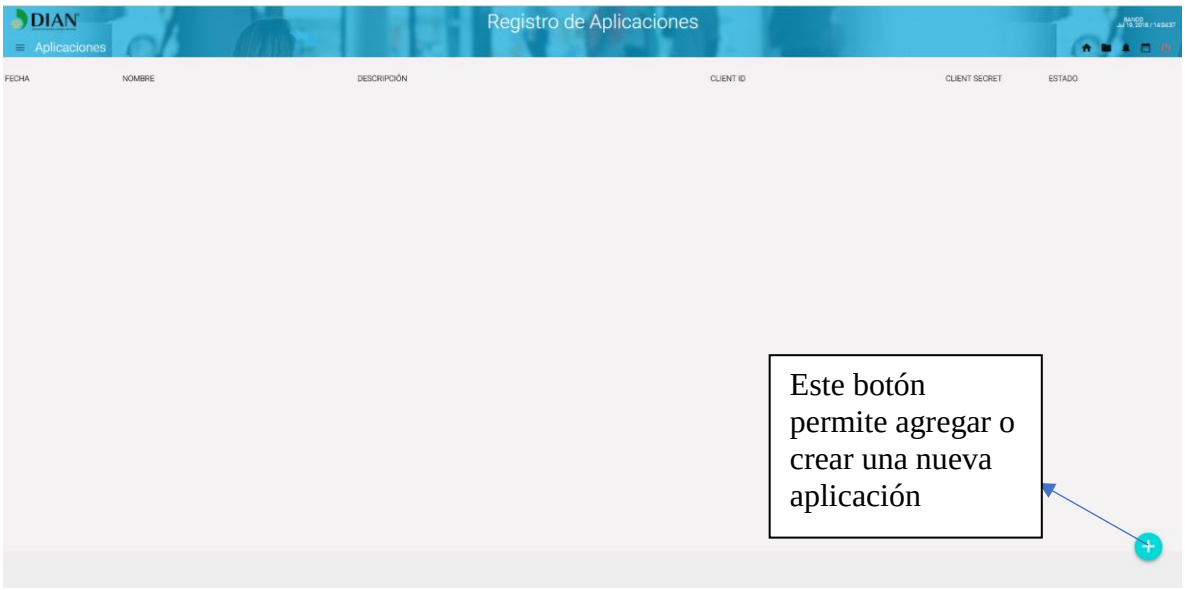
Ya que esta aplicación se encuentra desarrollada en angular a continuación se especifica las versiones y navegadores soportados.

NOTA: (Se recomienda el uso de Chrome, Firefox en sus últimas versiones, los demás navegadores soportan también angular como se puede ver en la tabla, pero pueden presentar problemas de renderización, ocasionando que algunos elementos no se visualicen correctamente):

Chrome	Firefox	Edge	IE	Safari	iOS	Android	IE mobile
latest	latest	14	11	10	10	Marshmallow (6.0)	11
		13	10	9	9	Lollipop (5.0, 5.1)	
			9	8	8	KitKat (4.4)	
				7	7	Jelly Bean (4.1, 4.2, 4.3)	

El presente documento tiene como finalidad enseñarle al usuario el uso correcto de la aplicación para la administración de aplicaciones B2B de las entidades bancarias.

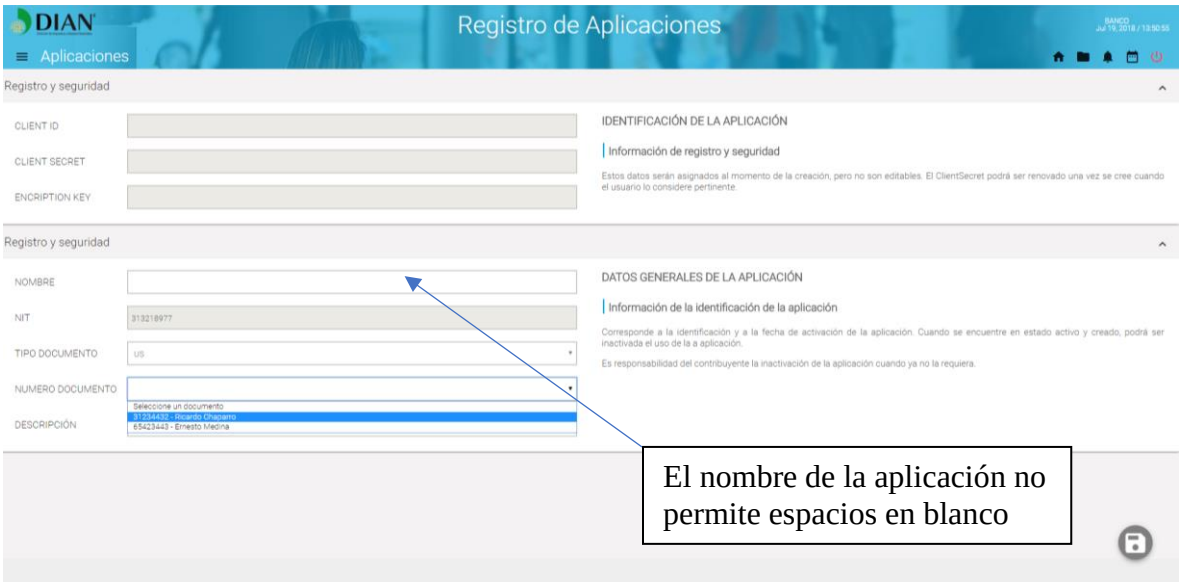
Pantalla lista de aplicaciones: Esta pantalla muestra en una lista las aplicaciones registradas por la entidad y le permite ver el clientSecret y editar cada una de las aplicaciones registradas.



Pantalla registro de aplicación: Esta pantalla permite ingresar los datos necesarios para crear la aplicación, a la cual se le asignaran los datos de seguridad, se debe de tener en cuenta los siguientes puntos.

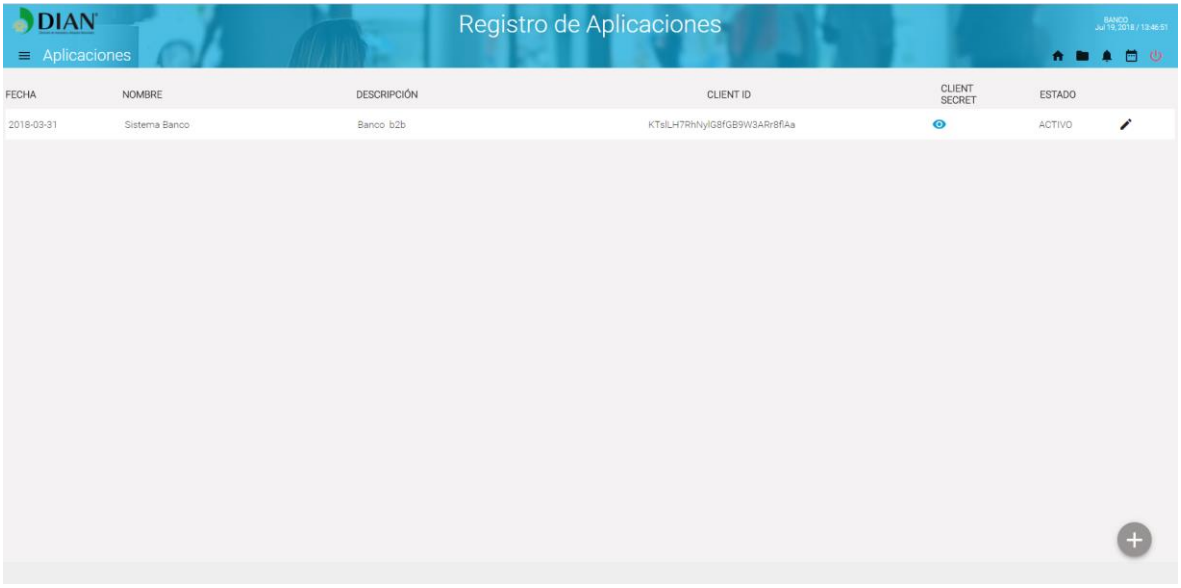
- El Adm. De Aplicaciones se encontrará autorizado para el Representante Legal.
- Cada entidad financiera tendrá una única aplicación B2B registrada.
- Sólo se permitirá actualizar la información de Public Key de la entidad financiera una vez.
- Para el cifrado del login se ha extendido el tiempo hasta 3 minutos antes y después de la hora actual para la verificación.
- Para el cifrado AES-128 se habilitó el Padding K5.

Una vez se hallan llenado todos los campos, el botón que se encuentra en la parte inferior derecha se habilitara y permitirá solicitar la creación del registro, regresando como respuesta el CLIENT ID, CLIENT SECRET y ENCRIPTION KEY que se encuentran en la primera sección de esta pantalla.

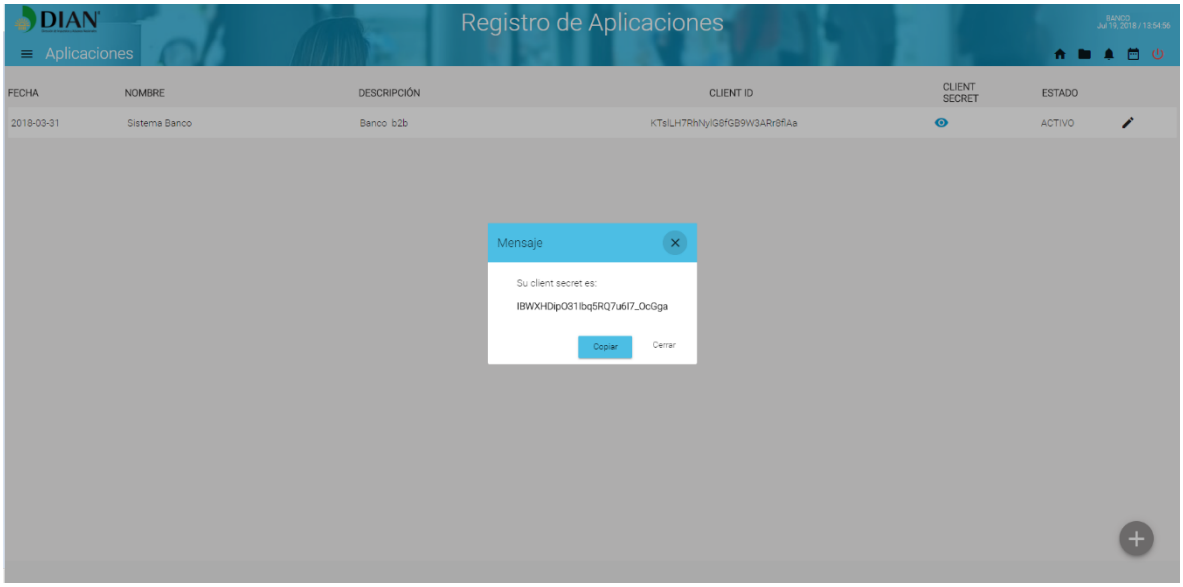


Pantalla lista de aplicaciones: Una vez hecho el registro al regresar a esta pantalla el registro aparecerá, ya que solo se permite el registro de una sola aplicación por el momento el botón de agregar estará deshabilitado.

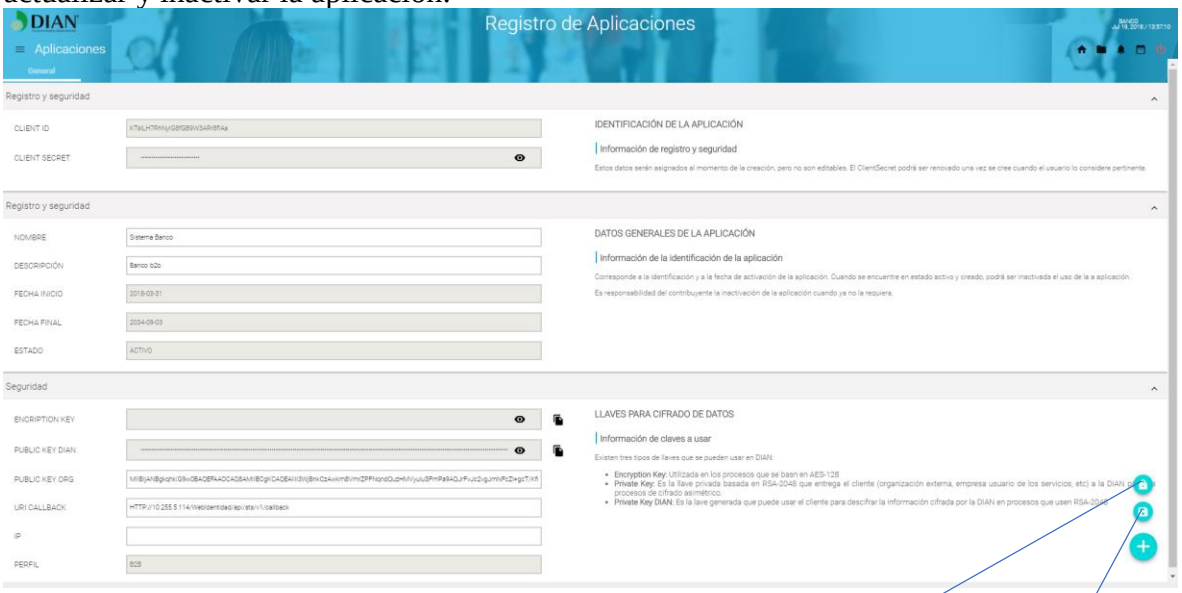
Una vez en esta pantalla podremos editar el registro o ver el clientSecret que se encuentra oculto y solo se puede visualizar en un modal que se lanza al darle click al icono , como se puede ver en la siguiente imagen, o también podemos entrar a editar esta pantalla si le damos click al icono



En este modal de la pantalla de la lista de aplicaciones registradas podemos hacer que se copie el clientSecret en el clipboard, dando click en el botón copiar



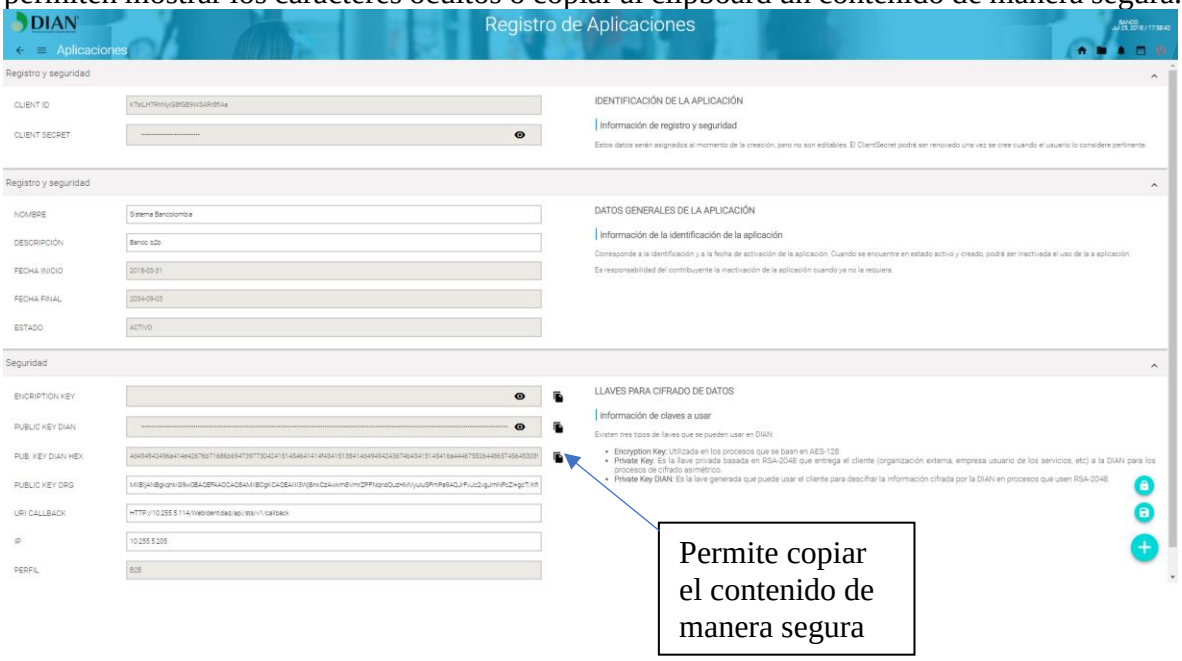
Pantalla edición de aplicaciones: En esta pantalla solo podremos editar algunos de los datos del registro como NOMBRE, DESCRIPCIÓN, PUBLIC KEY ORG, URI CALLBACK e IP, también en la parte inferior derecha podemos ver una sección de botones que nos permite actualizar y inactivar la aplicación.



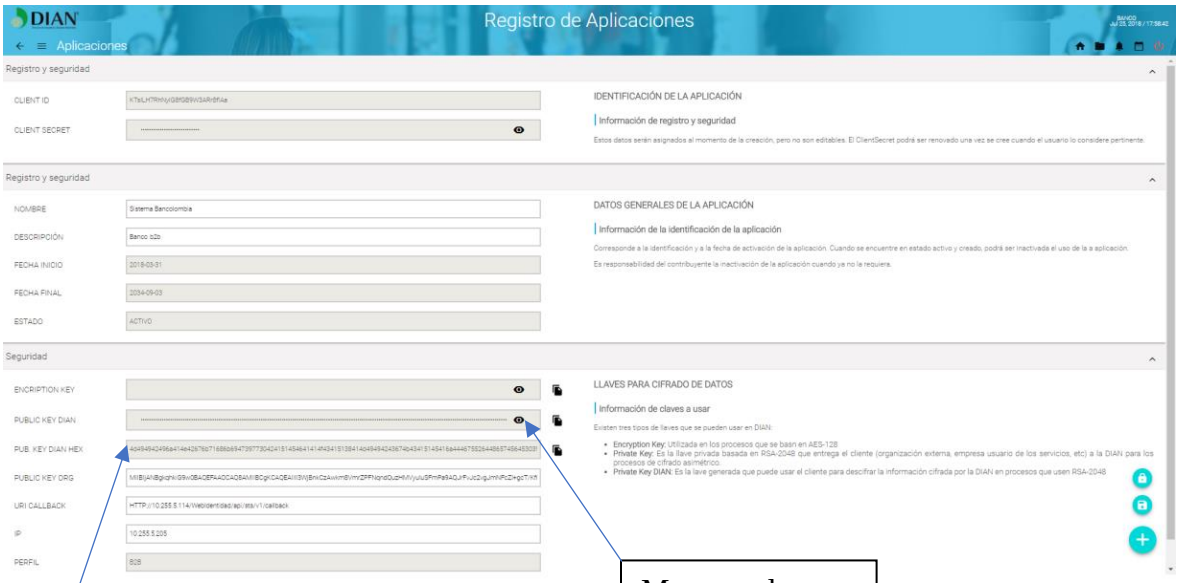
Inactiva la aplicación

Permite actualizar los datos

También en esta pantalla podemos encontrar otras opciones junto a los inputs que nos permiten mostrar los caracteres ocultos o copiar al clipboard un contenido de manera segura.



Permite copiar el contenido de manera segura



Este hexadecimal representa la public key DIAN en

Muestra el contenido oculto

6. Referencias

Los siguientes documentos están referenciados en este documento:

- Este documento referencia la especificación Swagger de autenticación de identidad: API Identidad-EXTERNOS-Swagger20.json

API
Identidad-EXTERNO