Investigación y Desarrollo

Banco Central de Costa Rica

INVOCADOR PERSONA JURÍDICA

Contenido

[Glosario de términos a utilizar 2](#_Toc83970161)

[Introducción 3](#_Toc83970162)

[OAuth2 4](#_Toc83970163)

[Roles 4](#_Toc83970164)

[Dueño del recurso 4](#_Toc83970165)

[Cliente 4](#_Toc83970166)

[Servidor de autorización 4](#_Toc83970167)

[Servidor de recursos 4](#_Toc83970168)

[Diagrama de secuencia para obtener un token de acceso 5](#_Toc83970169)

[Arquitectura simplificada del servicio 6](#_Toc83970170)

[Arquitectura extendida del servicio 7](#_Toc83970171)

[Usuario Interno 7](#_Toc83970172)

[Archivo de configuración 7](#_Toc83970173)

[Suscriptor 7](#_Toc83970174)

[Certificado Cliente 7](#_Toc83970175)

[Invocador Persona Jurídica 8](#_Toc83970176)

[Gestión de una consulta al servicio externo 8](#_Toc83970177)

[Parametrización del Invocador 10](#_Toc83970178)

[ConnectionStrings 10](#_Toc83970179)

[Redis 10](#_Toc83970180)

[DistributedCacheEntryOptions 10](#_Toc83970181)

[Suscriptor 10](#_Toc83970182)

[Suscriptores 10](#_Toc83970183)

[Parámetros del Suscriptor 10](#_Toc83970184)

[ApiEndpoints 11](#_Toc83970185)

[Prerrequisitos de instalación 12](#_Toc83970186)

[Tecnología del componente 13](#_Toc83970187)

[Instalación del componente (Invocador Persona Jurídica) 14](#_Toc83970188)

[Anexos 15](#_Toc83970189)

# Glosario de términos a utilizar

**Autorizador de Persona Jurídica**: es el componente que implementa el estándar OAuth2[[1]](#footnote-1), y permite emitir tokens de acceso en el formato JWT, para ser usados por los usuarios para autenticarse ante un servicio REST.

**Usuario interno**: usuario que requiere la información, puede ser un servicio o una aplicación de terceros.

**Suscriptor**: representa el nombre del servicio al que pertenece la configuración para obtener un token de acceso desde el autorizador.

**Invocador Persona Jurídica**: componente que actúa como un proxy para obtener la información privilegiada desde el exterior a la organización.

**Certificado de agente electrónico**: certificado digital que permite identificar de manera inequívoca al usuario del servicio.

**DMZ**: es una red aislada que se encuentra dentro de la red interna de la organización. En ella se encuentran ubicados exclusivamente todos los recursos de la empresa que deben ser accesibles desde Internet.

**Servicio Externo**: es el servicio que provee la información privilegiada y requiere un token de acceso para acceder a la misma. Este servicio pertenece al Banco Central de Costa Rica.

**JWT (JSON Web Token)**: es un estándar qué está dentro del documento RFC 7519.

Está compuesto por tres partes -encabezado, contenido y firma- concatenadas utilizando puntos.

En el mismo se define un mecanismo para poder propagar entre dos partes, y de forma segura, la identidad de un determinado usuario, además con una serie de claims o privilegios.

Estos privilegios están codificados en objetos de tipo JSON, que se incrustan dentro de del payload o cuerpo de un mensaje que va firmado digitalmente.

**Claim**: es un par ordenado nombre: valor, que permite construir la identidad del usuario.

# Introducción

La necesidad de compartir información como medio para el desarrollo y la actualización constante nos hace buscar cada día formas de llevar a cabo este cometido de manera eficaz, eficiente y segura.

Así como los medios de comunicación masiva, y las redes sociales permiten acceder a información pública, también existen medios que nos permiten acceder a información privilegiada. Con este propósito existen estándares para el intercambio seguro de información.

En el caso de la información restringida se hace imperativo el uso de herramientas que permitan identificar quiénes tienen acceso a la misma, así como definir los medios por los cuales será accedida dicha información.

Con el propósito de cumplir con estos requisitos nace la implementación del estándar OAuth2, para identificar los usuarios de los servicios RESTFul que expone el Banco Central de Costa Rica. Dicha implementación contempla tanto usuarios físicos (personas), como jurídicos (empresas o instituciones), quienes son usuarios de los servicios que la institución ofrece.

La implementación que se expone en el presente documento corresponde a un servicio que facilita la gestión de la identidad del cliente ante un servicio RESTFul expuesto por el Banco Central de Costa Rica, utilizando un certificado digital de agente electrónico como identidad.

# 

# OAuth2

OAuth 2.0 es un estándar abierto para la autenticación en servicios RESTFul, que nos permite compartir información entre aplicaciones sin tener que compartir los credenciales.

## Roles

Dentro de OAuth 2.0 encontramos diferentes roles que van a participar en el proceso:

* Dueño del recurso (Owner).
* Cliente (Client).
* Servidor de recursos protegidos (Resource Server).
* Servidor de autorización (Authorization Server).

### Dueño del recurso

El propietario del recurso es el usuario que da autorización a una determinada aplicación para acceder a su cuenta y poder hacer algunas cosas en su nombre.

Se le llama dueño del recurso porque, si bien el servicio que consulta no es suyo, los datos que maneja sí lo son.

### Cliente

El cliente sería la aplicación que desea acceder a esa cuenta de usuario. Antes de que pueda hacerlo debe ser autorizada por el usuario, y dicha autorización debe ser validada por el servicio.

### Servidor de autorización

El servidor de autorización es el responsable de gestionar las peticiones de autorización. Verifica la identidad de los usuarios y emite un token de acceso a la aplicación cliente.

### Servidor de recursos

El servidor de recursos será el servicio propiamente, es decir, el servidor que aloja el recurso protegido al cual queremos acceder.

## 

### Diagrama de secuencia para obtener un token de acceso

Figura 1: Diagrama de secuencia para obtener un token usando un certificado de agente electrónico.

Utilizando un certificado digital de agente electrónico, el servicio cliente se identifica ante el autorizador para obtener un token de acceso al RESTFul API.

Lo más costoso en tiempo de la gestión es realizar la autenticación con el autorizador con el certificado y lo que el componente Invocador busca es establecer una arquitectura de referencia para persistir el token por el tiempo de la vigencia de este y reutilizarlo la veces que se requiera.

El estándar OAuth2, define estos mecanismos, los cuales se han implementado permitiendo la identificación de usuarios mediante certificados digitales; en este caso, certificados digitales de agente electrónico.

# Arquitectura simplificada del servicio

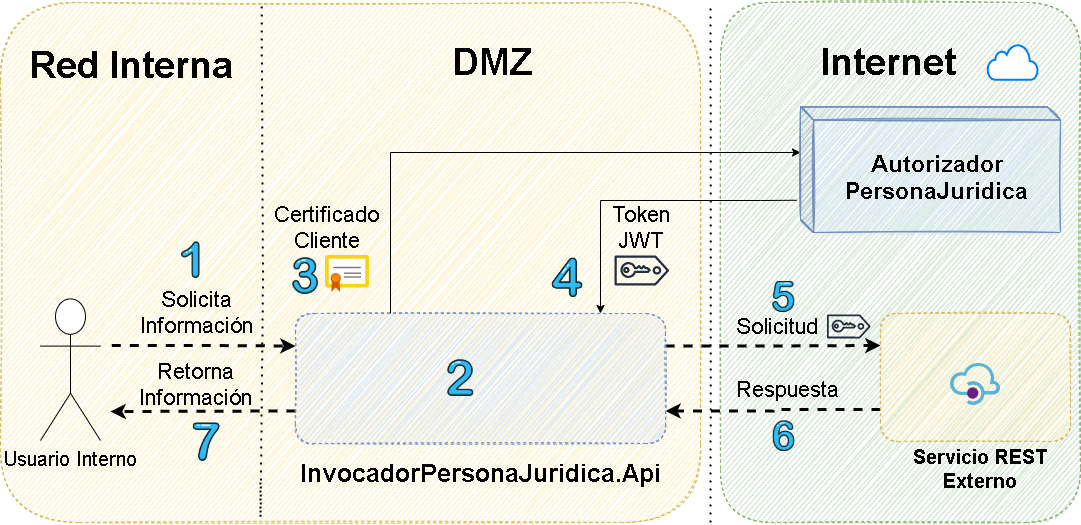


Figura 2: Arquitectura simplificada del invocador de persona jurídica, los números indican el orden en el que se ejecuta el proceso.

“El InvocadorPersonaJuridica.Api” es el componente encargado de abstraer la funcionalidad de comunicación para el usuario, con el servicio externo que requiere autenticación mediante un token de acceso JWT.

Su función es gestionar ante el autorizador de persona jurídica un token de acceso para el certificado de persona jurídica de la entidad a la que pertenece, almacenarlo en un caché y adjuntarlo en cada consulta que se realice hacia el servicio externo. En caso de contar con un token vigente en el caché, se utilizará ese mismo para identificar la consulta ante el servicio.

Estaría instalado en un servidor, o grupo de servidores capaces de comunicarse con el exterior de la infraestructura de red de la organización.

# Arquitectura extendida del servicio

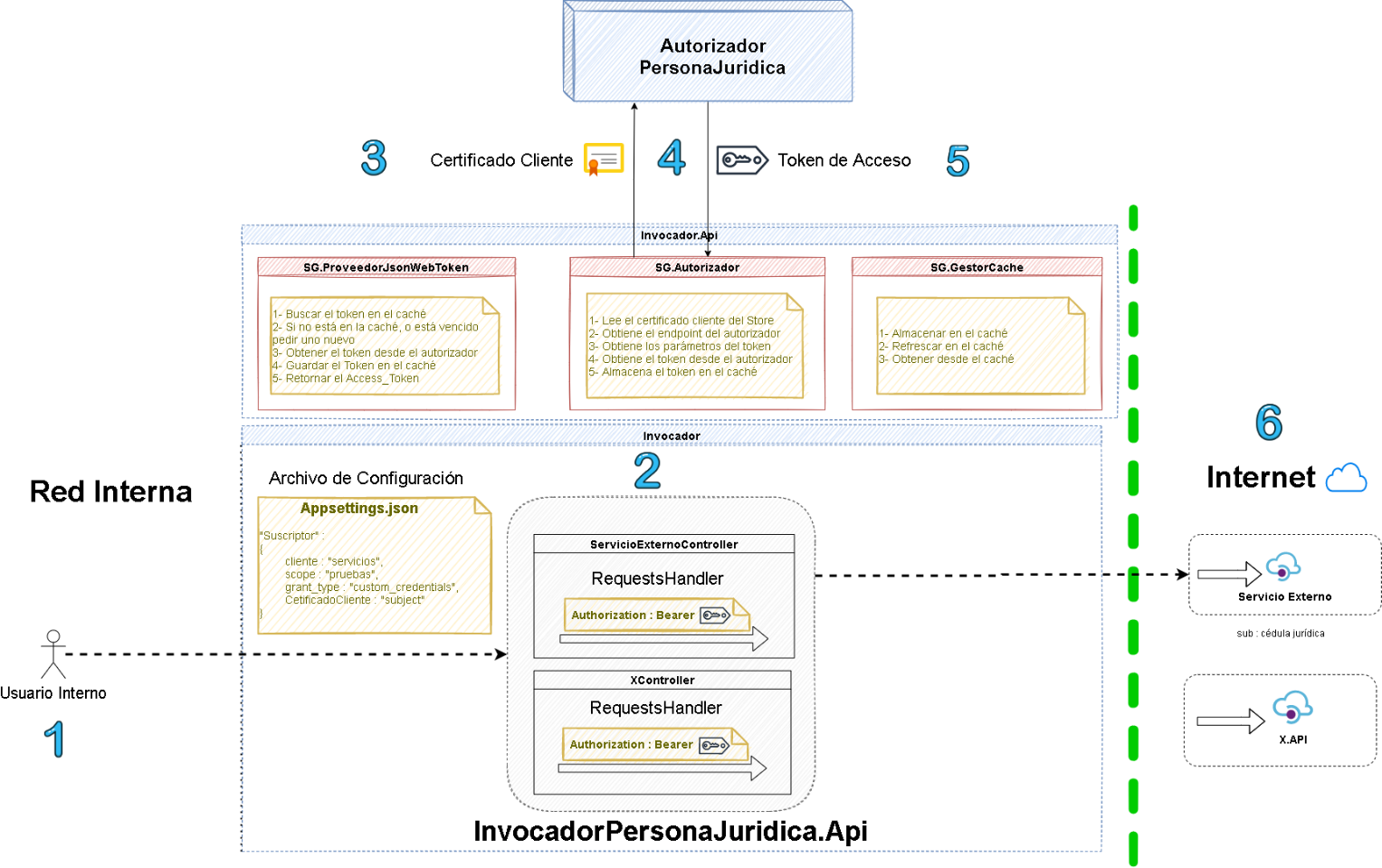


Figura 3: Arquitectura extendida del invocador de persona jurídica, el diagrama completo, lo puede encontrar en los anexos.

## Usuario Interno

Se identifica como usuario interno a cualquier persona, servicio o software de terceros que sea capaz de ejecutar una consulta mediante el protocolo HTTPS, desde el interior de la organización. Claro está, habiendo cumplido con los requisitos establecidos por la organización para el acceso a la información del servicio externo.

## Archivo de configuración

appsettings.json contiene las configuraciones para la puesta en producción.

appsettins.development.json contiene las configuraciones para el desarrollo.

## Suscriptor

Contiene los parámetros para obtener un token de acceso desde el autorizador de persona jurídica, para el servicio externo que será consultado.

### Certificado Cliente

Certificado digital de agente electrónico del cliente, o la entidad que consultará el servicio externo.

## Invocador Persona Jurídica

Es una solución web que expone un controlador homólogo al que expone el servicio externo, con la salvedad que la seguridad y el acceso es controlado por la organización. Por omisión este permitirá el acceso de modo irrestricto.

Además, los verbos expuestos concordarán con los que expone el servicio externo, así como sus parámetros.

### Gestión de una consulta al servicio externo

A continuación, se detallan los pasos ejecutados para obtener información desde un servicio externo que requiere acceso privilegiado. Los números corresponden con los números indicados en la figura 3.

1. La solicitud genera dentro de la red interna de la organización, puede ser iniciada por un usuario físico, o por un servicio, quien consulta una ruta específica del InvocadorPersonaJuridica.Api.

Nota: El mecanismo de seguridad que la organización implemente a nivel interno para tener acceso a dicha información escapa al alcance de este documento. Queda a discreción de la organización dicha implementación, para que se permita usar el mismo servicio (el mismo invocador), para consultar diferentes servicios externos, usando los privilegios correspondientes por cada servicio a consultar.

Un ejemplo de la consulta sería:

https://servidorDeMiOrganizacion/invocador/elControlador/?suscriptor=elSuscriptor&api-version=1

1. Cuando la solitud llega al controlador indicado en la solicitud, este contiene la lógica para ejecutar una consulta externa y obtener la información solicitada, así como los parámetros que esta requiere.
   1. Lo primero que se hace es solicitar un token al componente encargado de proveerlo, y adjuntarlo a la solicitud externa.
   2. Luego se retorna al usuario la información solicitada.
2. El proveedor del token obtiene del caché el token que existe registrado para **elSuscriptor** indicado en la solicitud, y lo retorna al componente que lo solicitó.
   1. Se verifica si existe otro procedimiento obteniendo un token. Si es así se debe esperar a que dicho procedimiento finalice.
   2. Se intenta obtener un token desde el caché, para **elSuscriptor** indicado en la solicitud, si el token está vigente, se retorna al componente que lo solicitó.
   3. Si no existe un token registrado; por ser la primera vez, o si el token se encuentra expirado, se debe indicar al proceso que se gestionará un nuevo token, para que, al llegar otra solicitud, esta se espere a que el proceso finalice, así como se indicó en el punto 3.a. Al finalizar el proceso para obtener el token, se libera el bloqueo para que las solicitudes que se encuentran en cola puedan obtener el token.
3. Procedimiento para obtener el token de acceso desde el autorizador de persona jurídica.
   1. Obtener desde el archivo de configuración los parámetros correspondientes a **elSuscriptor** indicado en la solicitud.
   2. Obtener desde el archivo de configuración el *endpoint* correspondiente para obtener el token de acceso desde el autorizador de persona jurídica.
   3. Se construye la solicitud hacia el autorizador de persona jurídica, con los parámetros obtenidos en el punto a.
   4. Se obtiene el token de acceso, y se almacena en el caché, usando el componente de gestión del caché.
4. El componente de gestión del caché es el responsable de persistir en el caché el token de acceso obtenido desde el autorizador de persona jurídica, así como de proveerlo cuando se lo solicitan.

Autorizador Persona Jurídica

Es la implementación del Banco Central de Costa Rica del estándar OAuth2, para la emisión de tokens de acceso, para certificados digitales de personas jurídicas.

Dicha implementación acepta peticiones según el estándar indicado en el documento [RFC 6749, sección 4.4.2](https://datatracker.ietf.org/doc/html/rfc6749#section-4.4.2). Respondiendo según la [sección 4.4.3](https://datatracker.ietf.org/doc/html/rfc6749#section-4.4.3), del mismo RFC.

Servicio externo

Es un servicio RESTFul que se expone para ser accedido desde la red pública, utilizando seguridad mediante token de acceso JWT.

# Parametrización del Invocador

## ConnectionStrings

### Redis

Nombre del clúster con que se accede al caché distribuido de Redis.

*Ejemplo:*

"ConnectionStrings": {

"Redis": "localhost"

}

## DistributedCacheEntryOptions

Parámetros de vigencia del caché.

*Ejemplo:*

"DistributedCacheEntryOptions": {

"AbsoluteExpirationRelativeToNow": "01:00:00",

"SlidingExpiration": "00:10:00"

}

## Suscriptor

Indica el suscriptor por omisión, cuando no se especifica en la solicitud al invocador. Suponiendo que solo se consulta un servicio externo, y no hace falta especificar de dónde se tomarán los parámetros para obtener

*Ejemplo:*

"Suscriptor": "Pruebas"

## Suscriptores

Corresponde a una sección en el archivo de configuración, en la que se registra el o los posibles suscriptores que tenga el invocador, cada suscriptor corresponde a un servicio externo al que se le consulta información, dado que cada servicio externo necesita un token de acceso específico, el invocador debe estar en capacidad de gestionar cada uno de ellos por separado, y tenerlo disponible para el momento de ser requerido.

El nombre del suscriptor puede corresponder con el nombre del servicio que se consulta así al almacenar el token de acceso en el caché, no se reemplazará con uno existente de otro servicio, ya que, el suscriptor se usa para formar la llave del token en el caché.

### Parámetros del Suscriptor

*client\_id*: identificador del cliente registrado en el autorizador de persona jurídica.

*scope*: nombre del servicio para el que se obtendrá el token de acceso

*grant\_type*: mecanismo de autenticación del usuario. En este caso será *custom\_credentials*.

*CertificadoCliente*: es el subject del certificado de agente electrónico que usará la organización para autenticarse ante el autorizador de persona jurídica.

*Ejemplo en la siguiente página.*

*Ejemplo:*

"Suscriptores": {

"Pruebas": {

"client\_id": "servicios",

"scope": "pruebas",

"grant\_type": "custom\_credentials",

"CertificadoCliente": "CN=COSIM 2019 (AGENTE ELECTRONICO), O=PERSONA JURIDICA, C=CR, SERIALNUMBER=CPJ-3-105-586122"

}

}

## ApiEndpoints

Son la colección de *endpoints* a los que el invocador puede tener acceso. Se incluye el enpoint de acceso al autorizador de persona jurídica.

*Ejemplo:*

"ApiEndpoints": [

{

"Nombre": "AutorizadorPersonaJuridica",

"Ruta": "https://dev.auth.bccr.fi.cr/personajuridica/connect/token?api-version=1",

"TimeOutEnSegundos": "120"

},

{

"Nombre": "Pruebas",

"Ruta": "https://dev.auth.bccr.fi.cr/Dummy.PersonaJuridica.Api/Usuario?api-version=1",

"TimeOutEnSegundos": "120"

}

],

# Prerrequisitos de instalación

1. Certificado de agente electrónico instalado en el almacenamiento local, al cual tenga acceso el proceso del Invocador.
2. Redis, como gestor de caché distribuido.
3. Runtime Net Core 3.1

# Tecnología del componente

El invocador persona jurídica, está desarrollado utilizando .Net Core 3.1

Utiliza Redis como gestor de caché distribuido.

# Instalación del componente (Invocador Persona Jurídica)

La aplicación es un website, así que debe ser publicado como tal según la infraestructura de la organización. Estableciendo los parámetros según se indicó en los apartados anteriores.

# Anexos

1. Diagrama de la arquitectura, se debe importar en la siguiente aplicación <https://app.diagrams.net/>
   1. File -> Import.
   2. Y selecciona el archivo correspondiente.



1. Repositorio público del proyecto InvocadorPersonaJuridica.Api

<https://github.com/bccr-oss/invocador-persona-juridica.git>

**Nota**: este es un repositorio no oficial del Banco Central de Costa Rica.

1. Ejemplo de consulta al Invocador de Persona Jurídica, para pruebas.

<https://dev.auth.bccr.fi.cr/pruebas/connect/token?api-version=1>

Con ese código podemos hacer una petición de tipo POST con esta estructura, adjuntando un certificado digital de agente electrónico para identificar al servicio:

POST /token HTTP/1.1

Host: dev.auth.bccr.fi.cr

client\_id:servicios

scope:pruebas

grant\_type:custom\_credentials

Si todo es correcto, se obtendría la respuesta válida, que sería de este tipo:

HTTP/1.1 200 OK

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

Cache-Control: no-store

Pragma: no-caché

{

"access\_token":"eyJhbGciOiJSUzI1NiI…",

"expires\_in": 3600,

"token\_type": "Bearer",

"scope": "pruebas"

}

1. Certificado de agente electrónico para pruebas.
   1. 
   2. 
   3. Certificado Cliente
2. Endpoint del autorizador de persona jurídica, ambiente producción.

<https://oauth2.bccr.fi.cr/personajuridica/connect/token?version=1.0>

Requiere un certificado de agente electrónico válido, para obtener un token de acceso, así como de los parámetros correspondientes que serían:

* 1. El *clientid*.
  2. El *scope*.
  3. El *grant\_type*.
  4. El subject del certificado de agente electrónico.

1. OAuth2: https://datatracker.ietf.org/doc/html/rfc6749 [↑](#footnote-ref-1)