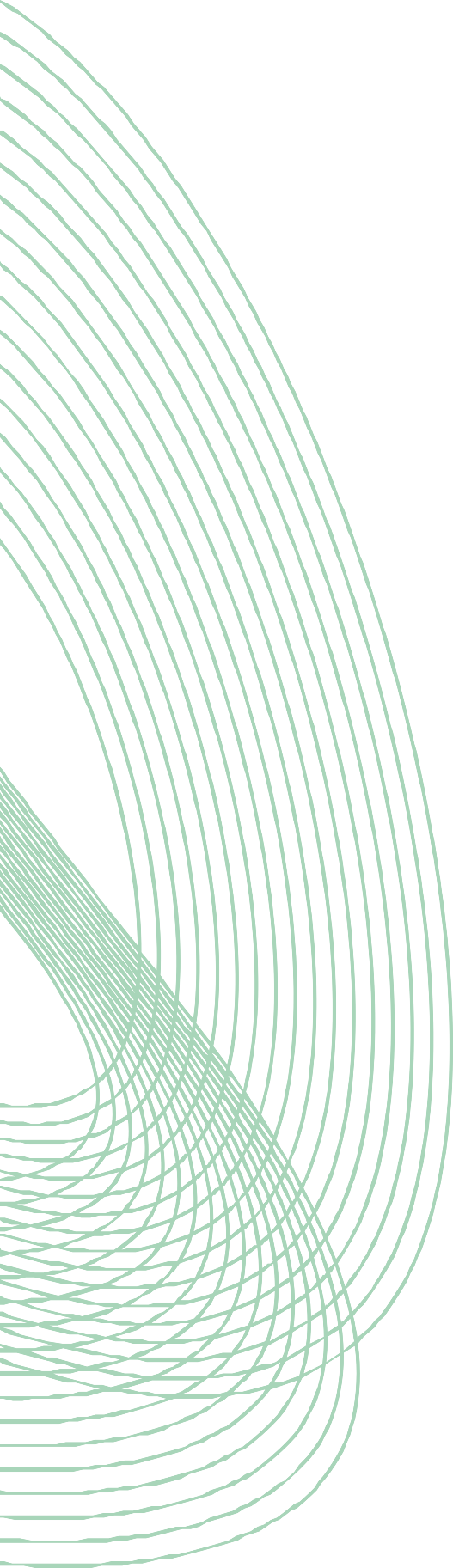
**UNIVERSIDAD NACIONAL**

**DEL COMAHUE**

**JUNIO 2023**

**Proyecto de Beca PPU - Poder Judicial Neuquen**

Web Service Rest sobre plataforma XROAD

**INTEGRANTES:**

Mariano Agustin Vergara Flores

Maria Laura Pino

**AÑO 2023**

1. **INTRODUCCIÓN**

*Introducción* **3**

1. **TECNOLOGIAS UTILIZADAS**

*Tecnologias* **4**

1. **INTRODUCCION A X-ROAD**

*¿Que es X-Road?* **5**

*Vision General* **5**

*Ecosistema X-Road* **6**

*Modelo Organizacional* **8**

*Arquitectura* **9**

## INTRODUCCION A DOCKER

*¿Que es Docker?* **13**

*¿Que beneficios tiene usar Docker?* **14**

## ESTRUCTURA DE LA APLICACION

*Estructura* **15**

**ÍNDICE**

## VERIFICAR FUNCIONAMIENTO DE LOS WEB SERVICES

*Verificar Funcionamiento* **16**

## AGREGAR UN NUEVO WEB SERVICE

*Agregar nuevo web service* **20**

B E C A P P U P O D E R J U D I C I A L N E U Q U E N

B E C A P P U P O D E R J U D I C I A L N E U Q U E N

1. **DESPLIEGUE DE LA APLICACION**

Introducción **23**

1. DOCKER COMPOSE

¿Que es Docker Compose? **25**

1. ANEXO. CONSULTAS

[Rubricas 28](#_TOC_250000)

Socios 29

Antecedentes **30**

Inscripcion y Sede **31**

1. ANEXO. DOCS API

Rubricas 32

Socios 40

Antecedentes **44**

**ÍNDICE**

Inscripcion y Sede **50**

# I N T R O D U C C I O N

El presente trabajo es un informe sobre la arquitectura, herramientas y tecnologias que se usaron para desarrollar e implementar los servicios WEB requeridos por el Poder Judicial de Neuquen

Tecnologias Usadas Java 17

Framework Spring Boot 3 Maven



MariaDB (version mayor a la 5) Swagger 3

IDE Utilizado (Intellij IDEA Community Edition)



# A C C E S O A L P R O Y E C T O

[*https://github.com/marianvf10/Poder\_Judicialv2*](https://github.com/marianvf10/Poder_Judicialv2)

# T E C N O L O G I A S U T I L I Z A D A S



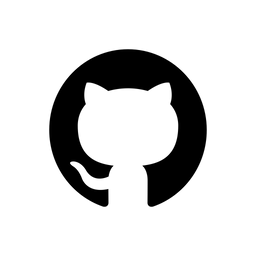
Se utilizo swagger para documentar cada uno de los servicios web

La base de datos utilizada para las pruebas fue brindada por el poder judicial

Postman fue utilizado para hacer las pruebas en etapas de desarrollo y produccion

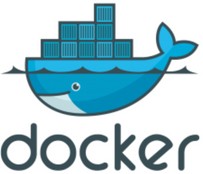
 

Se utilizo spring boot para desarrollar los servicios web



Se uso la plataforma GitHub para almacenar el repositorio de los servicios

Se utilizo X-ROAD para el intercambio de informacion entre el Centro Pymes y el Poder Judicial



Docker se utilizo para hacer el despliegue de la aplicacion

X-Road® es un software de código abierto y una solución de ecosistema que proporciona un intercambio de datos unificado y seguro entre las organizaciones.

La idea básica de X-Road es que los miembros de un ecosistema intercambien datos a través de puntos de acceso (Servidores de Seguridad - Security Servers) que implementan las mismas especificaciones técnicas.

X-Road es un bien público digital verificado por la Alianza de Bienes Públicos Digitales, y se publica bajo la licencia de código abierto del MIT y está disponible de forma gratuita.

# 2 . V i s i o n G e n e r a l

X-Road es una capa de intercambio de datos distribuida administrada centralmente entre sistemas de información que proporciona una forma estandarizada y segura de producir y consumir servicios.

X-Road implementa un conjunto de características estándar para apoyar y facilitar el intercambio de datos y garantiza la confidencialidad, integridad e interoperabilidad entre las partes del intercambio de datos:

gestión de direcciones (address management) enrutamiento de mensajes (message routing)

administración de permisos de acceso (access rights management)

autenticación a nivel de organización (organization-level authentication) autenticación a nivel de máquina (machine-level authentication)

cifrado a nivel de transporte (transport-level encryption) sellado de tiempo (time-stamping)

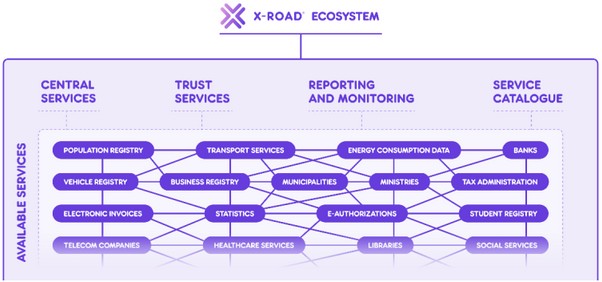


firma digital de mensajes (digital signature of messages) Registro (logging)

manejo de errores (error handling)



Un ecosistema X-Road es una comunidad de organizaciones que utilizan la misma instancia del software X-Road para producir y consumir servicios. El propietario del ecosistema, el Operador de X-Road, controla quién puede unirse al Ecosistema, y el propietario define las regulaciones y prácticas que el ecosistema debe seguir



# 2 . 2 R e d d e C o n f i a n z a ( T r u s t e d N e t w o r k )

Incluso si el software X-Road es de código abierto, unirse a un ecosistema X- Road requiere pasar por un proceso de incorporación. Durante el proceso, la identidad de cada organización y punto de acceso técnico se verifica mediante certificados emitidos por una entidad de certificación (CA) de confianza. Las identidades se mantienen de forma centralizada, pero todos los datos se intercambian directamente entre un consumidor de servicios y un proveedor de servicios.

El enrutamiento de mensajes se basa en identificadores de nivel de servicio y organización que X Road asigna a las ubicaciones de red físicas de los servicios. Toda la evidencia relacionada con el intercambio de datos es almacenada localmente por las partes del intercambio de datos, y ningún tercero tiene acceso a los datos. El sellado de tiempo y la firma digital juntos garantizan el no repudio de los datos enviados a través de X-Road. Los registros proporcionados por X- Road se pueden utilizar en un procedimiento judicial como prueba.

X-Road implementa un framework de autorización que se utiliza para administrar los permisos de acceso a los servicios. La administración de los permisos de acceso se basa en los identificadores de nivel de servicio y organización.

La idea clave de X-Road es que cada proveedor de servicios es propietario de sus datos y es responsable de administrar los permisos de acceso de sus servicios. En otras palabras, la publicación del servicio en X-Road no significa que el servicio sea accesible automáticamente para todas las organizaciones miembros de X- Road. Por lo general, los permisos de acceso se otorgan a nivel del sistema de información: un proveedor de servicios otorga a un sistema de información específico acceso a un servicio.

# 2 . 4 M o n i t o r e o e i n f o r m e s

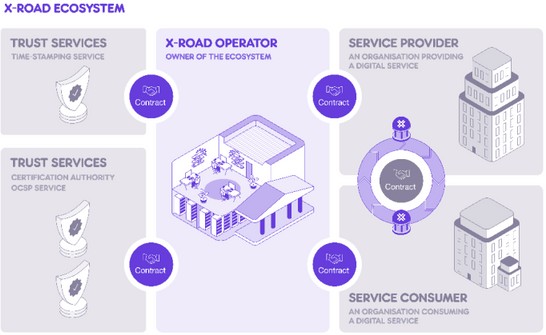
X-Road proporciona capacidades de monitoreo e informes que se pueden usar para recopilar datos de informes operativos e información de monitoreo técnico del ecosistema. La información se puede utilizar para medir el uso de servicios individuales, comprender dependencias y relaciones entre diferentes sistemas y servicios de información, monitorear el estado del servicio, monitorear las versiones de software X-Road utilizadas, etc. Cada organización miembro de X- Road puede acceder a sus propios datos, mientras que el operador de X-Road puede acceder a todos los datos de los miembros.

# 2 . 5 I n t e r c a m b i o t r a n s f r o n t e r i z o d e d a t o s

X-Road proporciona soporte integrado para el intercambio de datos transfronterizos a través de la federación, lo que significa unir dos ecosistemas de X-Road. Los miembros de los ecosistemas federados pueden publicar y consumir servicios entre sí como si fueran miembros del mismo ecosistema.

Es posible crear conexiones de federación con varios ecosistemas, pero no se admiten relaciones de federación transitivas. Un ecosistema no tiene una relación de federación con otro ecosistema con el que no esté directamente federado.

El ecosistema de X-Road consiste en un Operador de X-Road, organizaciones Miembros y Proveedor(es) de Servicios de Confianza



# 3 . 1 O p e r a d o r d e X - R o a d

Como propietario del ecosistema X-Road, el Operador es responsable de todos los aspectos de las operaciones. Las responsabilidades incluyen definir regulaciones y prácticas, aceptar nuevos miembros, brindar soporte a los Miembros y operar los componentes centrales del software X-Road.

# 3 . 2 M i e m b r o s d e X - R o a d

Los Miembros de X-Road son organizaciones que se han unido al ecosistema y producen y/o consmen servicios con y de otros Miembros. Una organización miembro puede ser un proveedor de servicios, un consumidor de servicios o ambos. Las organizaciones pueden convertirse en miembros de un ecosistema completando el proceso de incorporación definido por el Operador. Además, los miembros deben tener acceso al componente técnico que se requiere para intercambiar mensajes a través de X-Road, el servidor de seguridad.

# d e C o n f i a n z a

Para que un ecosistema X-Road funcione se requieren dos tipos de servicios de confianza: 1) autoridad de sellado de tiempo (TSA) y 2) autoridad de certificación (CA). Los proveedores de servicios de confianza son organizaciones que brindan estos servicios. Los proveedores de servicios de confianza pueden ser terceros comerciales, o los servicios también pueden ser proporcionados y mantenidos por el Operador de X-Road.

# 4 . A r q u i t e c t u r a

Técnicamente, el ecosistema de X-Road consiste en Servicios Centrales, Servidores de Seguridad, Sistemas de Información, TSA y CA

# 4 . 1 S e r v i c i o s C e n t r a l e s

Los servicios centrales consisten en el servidor central y el proxy de configuración. El Servidor Central contiene el registro de los miembros de X- Road y sus servidores de seguridad. Además, el servidor central contiene la directiva de seguridad de la instancia de X-Road que incluye una lista de entidades de certificación de confianza, una lista de autoridades de sellado de tiempo de confianza y parámetros de configuración. Tanto el registro de miembros como la directiva de seguridad se ponen a disposición de los servidores de seguridad a través del protocolo HTTP. Este conjunto distribuido de datos forma la configuración global que los servidores de seguridad utilizan para mediar los mensajes enviados a través de X-Road. El Operador de X-Road es responsable de operar el Servidor Central.

El Proxy de configuración es un componente opcional que se puede utilizar como proxy para publicar la configuración global en Servidores de seguridad para su descarga. El proxy de configuración primero descarga la configuración global del servidor central y, a continuación, la distribuye de forma segura. El proxy de configuración se puede utilizar para aumentar la disponibilidad del sistema mediante la creación de un origen de configuración adicional y reducir la carga en el servidor central. El operador de X-Road es responsable de operar el proxy de configuración

El Servidor de Seguridad es el punto de entrada a X-Road, y es necesario tanto para producir como para consumir servicios a través de X-Road. El servidor de seguridad media las llamadas de servicio y las respuestas de servicio entre los sistemas de información. Éste encapsula los aspectos de seguridad de la infraestructura de X-Road: administración de claves para la firma y autenticación, envío de mensajes a través de un canal seguro, creación del valor de prueba para mensajes con firmas digitales, sellado de tiempo y registro. Para un consumidor de servicios y un sistema de información de proveedor de servicios, el Servidor de Seguridad ofrece un protocolo de mensajes basado en REST y soap. El protocolo es el mismo tanto para el cliente como para el proveedor de servicios, lo que hace que el Servidor de Seguridad sea transparente para las aplicaciones

El Servidor de Seguridad administra dos tipos de claves. Las claves de autenticación se asignan a un servidor de seguridad y se utilizan para establecer canales de comunicación criptográficamente seguros con otros servidores de seguridad. Las claves de firma se asignan a los clientes del Servidor de Seguridad y se utilizan para firmar los mensajes intercambiados. Una entidad de certificación de confianza emite certificados para las claves. Los certificados emitidos por otras entidades de certificación se consideran inválidos.

Para poder mediar mensajes, el Servidor de Seguridad debe tener una copia válida de la configuración global disponible todo el tiempo. El servidor de seguridad descarga la configuración global del servidor central con regularidad y utiliza una copia local mientras procesa los mensajes. El servidor de seguridad permanece operativo siempre que tenga una copia válida de la configuración global disponible localmente. Del mismo modo, la información de validez del certificado se descarga de la entidad de certificación y se almacena en caché localmente. El almacenamiento en caché permite que el Servidor de Seguridad funcione incluso cuando los orígenes de datos de configuración no están disponibles.

El Servidor Seguridad tiene un equilibrador de carga interno del lado del cliente y también admite el equilibrio de carga externo. El equilibrador de carga del lado del cliente es una característica integrada y proporciona alta disponibilidad. En su lugar, el equilibrio de carga externo proporciona alta disponibilidad y escalabilidad desde el punto de vista del rendimiento

El Sistema de Información produce y/o consume servicios a través de X-Road y es propiedad de un miembro de X-Road. X-Road admite el consumo y la producción de servicios REST y SOAP. Sin embargo, X-Road no proporciona conversiones automáticas entre diferentes tipos de mensajes y servicios.

Para un sistema de información al consumidor de servicios, el servidor de seguridad actúa como un punto de entrada a todos los servicios de X-Road. El consumidor puede descubrir a los miembros registrados de X-Road y sus servicios disponibles mediante el protocolo de metadatos de X-Road

Un sistema de información del proveedor de servicios implementa un servicio REST y/o SOAP y lo pone a disposición a través de X-Road. Los servicios REST existentes no requieren ningún cambio, ya que se pueden publicar tal cual. En su lugar, los servicios SOAP deben implementar el protocolo de mensajes X-Road para SOAP. Las descripciones de servicio de los servicios REST se definen mediante la especificación OpenAPI3 y las descripciones de servicio de los servicios SOAP se definen mediante WSDL. Los consumidores de servicios pueden descargar descripciones de servicios mediante el protocolo de metadatos X-Road

# 4 . 4 A u t o r i d a d d e S e l l a d o d e T i e m p o

Todos los mensajes enviados a través de X-Road tienen un selladode tiempo y son registrados por el servidor de seguridad. El propósito del sellado de tiempo es certificar la existencia de elementos de datos en un momento determinado. La TSA proporciona un servicio de sellado de tiempo que el servidor de seguridad utiliza para marcar la hora de todas las solicitudes/respuestas entrantes/salientes. Solo se pueden utilizar los TSA de confianza definidos en el Servidor Central.

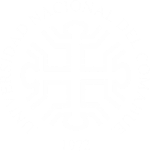
La autoridad de sellado de tiempo debe implementar el protocolo de sellado de tiempo admitido por X-Road. X-Road utiliza el sellado de tiempo por lotes, lo que reduce la carga del servicio de sellado de tiempo. La carga no depende del número de mensajes intercambiados a través de X-Road. En su lugar, depende del número de servidores de seguridad en el Ecosistema

# C e r t i f i c a c i o n

La Autoridad de Certificación (AC) emite certificados a los servidores de seguridad (certificados de autenticación) y a las organizaciones miembros de X- Road (certificados de firma). Los certificados de autenticación se utilizan para proteger la conexión entre dos servidores de seguridad. Los certificados de firma se utilizan para firmar digitalmente los mensajes enviados por los miembros de X-Road. Solo se pueden usar certificados emitidos por entidades de certificación de confianza definidas en el servidor central.

El servidor de seguridad comprueba la validez de los certificados de firma y autenticación a través del Protocolo de estado de certificados en línea (OCSP). Cada servidor de seguridad es responsable de consultar la información de validez de sus certificados y, a continuación, compartir la información con otros servidores de seguridad como parte del proceso de intercambio de mensajes. Solo los servidores de seguridad con certificados de firma y autenticación válidos pueden intercambiar mensajes con otros servidores de seguridad.

La idea detrás de Docker es crear contenedores ligeros y portables para las aplicaciones software que puedan ejecutarse en cualquier máquina con Docker instalado, independientemente del sistema operativo que la máquina tenga por debajo, facilitando así también los despliegues.



B E C A P P U

Universidad nacional del comahue

**¿ Q u é e s D o c k e r ?**

Este concepto ya es antiguo, y viene de Linux, pero para hacer una analogía con el mundo real, vendría a ser como esos contenedores que suelen llevar los barcos de mercancías, que contienen distintos productos.

Es algo auto contenido en sí, que se puede llevar de un lado a otro de forma independiente, es portable.

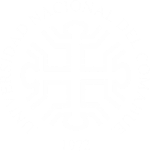
Ahora, volviendo al software, para que podamos acceder como usuarios normales a una aplicación, dicha aplicación software necesita estar ejecutándose en una máquina, en un ordenador. Pero además, dependiendo del tipo de aplicación, dicho ordenador también necesita tener instaladas una serie de cosas para que la aplicación se ejecute correctamente: cierta versión de Java instalado, un servidor de aplicaciones (p.e tomcat, que es el software que realmente estará ejecutando mi aplicación y haciendo que pueda interactuar con ella).

Docker, permite meter en un contenedor (“una caja”, algo auto contenido, cerrado) todas aquellas cosas que mi aplicación necesita para ser ejecutada (Java, Maven, Tomcat…) y la propia aplicación. De esta manera, se puede llevar ese contenedor a cualquier máquina que tenga instalado Docker y ejecutar la aplicación sin tener que hacer nada más, ni preocuparse de qué versiones de software tiene instalada esa máquina, de si tiene los elementos necesarios para que funcione mi aplicación , de si son compatibles…etc.

La aplicación se ejecuta desde el contenedor de Docker, y dentro de él estarán todas las librerías y cosas que necesita dicha aplicación para funcionar correctamente.

.

# D o c k e r ?



B E C A P P U

Universidad nacional del comahue

**¿ Q u é b e n e f i c i o s t i e n e u s a r**

Docker es una herramienta diseñada para beneficiar tanto a desarrolladores, testers, como administradores de sistemas, en relación a las máquinas, a los entornos en sí donde se ejecutan las aplicaciones software, los procesos de despliegue, etc.

En el caso de los desarrolladores, el uso de Docker hace que puedan centrarse en desarrollar su código sin preocuparse de si dicho código funcionará en la máquina en la que se ejecutará.

Por ejemplo, sin utilizar Docker un posible escenario podría ser el siguiente (hay otras formas de solucionar este escenario, pero por poner un ejemplo claro):

* Pepe tiene en su ordenador instalado Java 8, y está programando una funcionalidad específica de la aplicación con algo que solo está disponible en esa versión de Java.
* José tiene instalado en su máquina Java 7, porque está en otro proyecto trabajando sobre otro código, pero Pepe quiere que José ejecute el código de su aplicación en su máquina. O José se instala Java 8, o la aplicación en su máquina fallará.

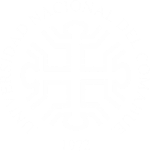
Este escenario desaparece con Docker. Para ejecutar la aplicación, Pepe se crea un contenedor de Docker con la aplicación, la versión 8 de Java y el resto de recursos necesarios, y se lo pasa a José.

José, teniendo Docker instalado en su ordenador, puede ejecutar la aplicación a través del contenedor, sin tener que instalar nada más.

Por eso Docker también es muy bueno para el testing, para tener entornos de pruebas. Por un lado, es muy sencillo crear y borrar un contenedor, además de que son muy ligeros, por lo que podemos ejecutar varios contenedores en una misma máquina (donde dicho contenedor tendría el entorno de nuestra aplicación: base de datos, servidor, librerías…). Por otro, un mismo contenedor funcionará en cualquier máquina Linux: un portátil, el ordenador de tu casa, máquinas alojadas en Amazon, tu propio servidor…

Esto además beneficia a la parte de sistemas, ya como los contenedores son más ligeros que las máquinas virtuales, se reduce el número de máquinas necesarias para tener un entorno.

Para el desarrollo de los servicios utilizamos SpringBoot, una tecnologia de Spring.



B E C A P P U

Universidad nacional del comahue

**E S T R U C T U R A D E L P R O Y E C T O**

Con Spring Boot nos facilitamos las configuraciones como las dependencias de conexion con la base de datos, despliegue de servidor, entre otras.

El patron utilizado para organizar la estructura del proyecto se divide en distintas capas, lo cual nos permite separar la logica de negocios (entidades, servicios, repositorio) de los controladores que manejan las peticiones HTTP especificas.

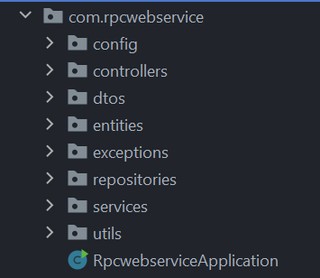
Tambien se utilizaron DTOs (Data Transfer Objects) debido a que nos facilitan mostrar la informacion requerida por cada uno de los servicios web.

El patrón DTO tiene como finalidad de crear un objeto plano (POJO) con una serie de atributos que puedan ser enviados o recuperados del servidor en una sola invocación, de tal forma que un DTO puede contener información de múltiples fuentes o tablas y concentrarlas en una única clase simple.

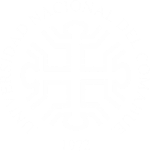
El paquete config se utiliza para almacenar la configuracion de OpenAPI que se utiliza para la documentacion de los servicios.

En el paquete de exceptions se almacenan distintos tipos de excepciones particulares para cada situacion

Por ultimo, el paquete utils es para utilidades, como por ejemplo, se utiliza un validador de cuit que se utiliza en el controlador a la hora de chequear que el formato y tipo de cuit sea el correcto



# l o s W e b S e r v i c e s



B E C A P P U

Universidad nacional del comahue

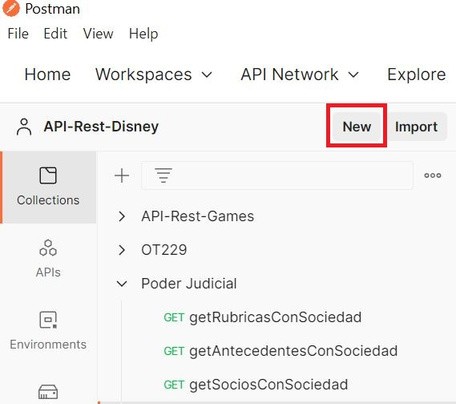
**V e r i f i c a r F u n c i o n a m i e n t o d e**

Para el verificar el correcto funcionamiento de cada uno de los web services utilizamos POSTMAN.

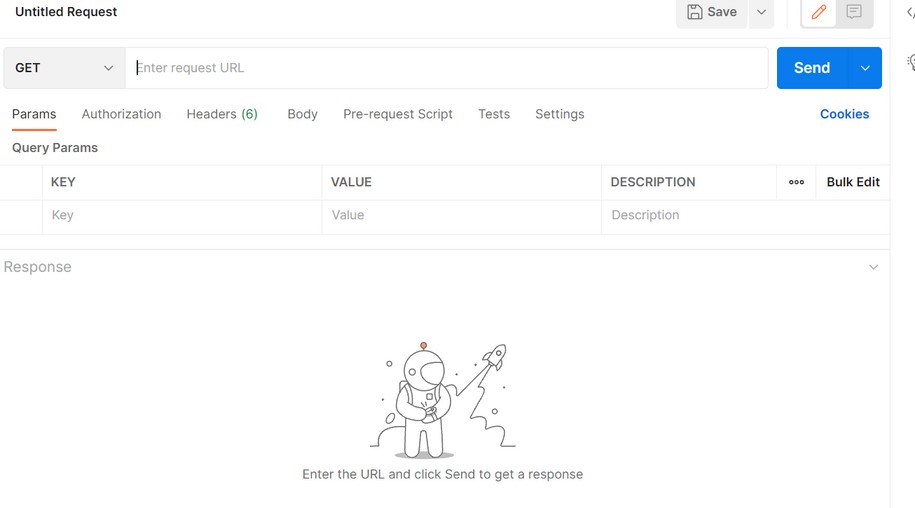
Voy a mostrar un ejemplo de como verificar el funcionamiento del servicio web de rubricas (en el anexo de los consultas se encuentra la descripcion).

Primero que nada debemos ejecutar nuestra aplicacion y luego debemos ir a postman.

Una vez abierto POSTMAN debemos hacer click en donde dice "New" y seleccionar HTTP Request



Luego de hacer click, se nos abre la siguiente ventana

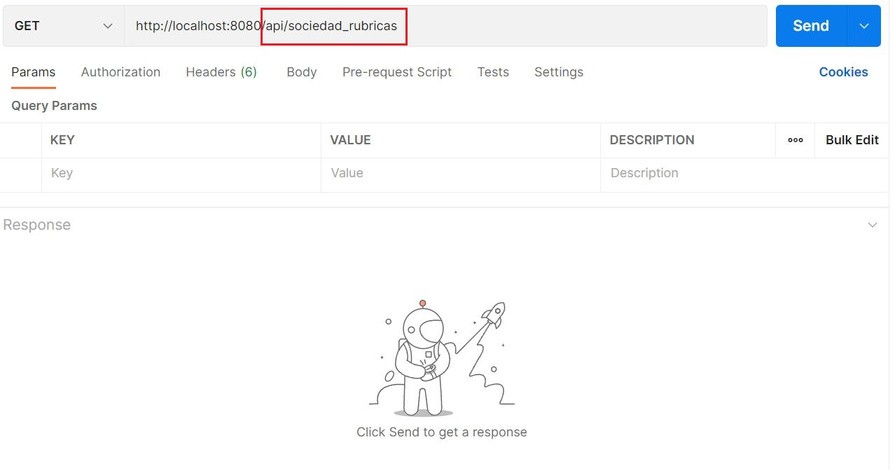


Donde dice GET lo que hacemos es seleccionar el tipo de peticion HTTPS que estamos haciendo, como los web service que construí son todos de consulta, todos van a utilizar GET.

Del lado derecho de donde esta GET, se ubica la url de la aplicacion del ambiente de desarrollo, por lo general cuando trabamos con spring boot la url de desarrollo es localhost y el puerto por defecto es 8080. Ejemplo: http://localhost:8080

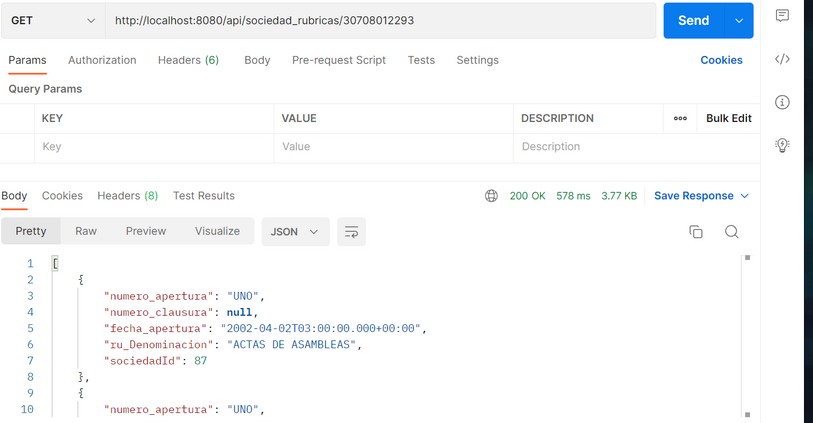
En la seccion de Response es donde muestra el resultado de esa peticion.

Luego del puerto, va el path del servicio web que estamos queriendo verificar, en el ejemplo de rubricas yo defini el siguiente path:



Por ultimo, luego del path lo que ubico es el parametro de busqueda que viene definido por las consultas en el documento de la especificacion. En este caso el parametro del busqueda es el CUIT. Por lo cual, debemos ubicar el "CUIT" sin guiones y puntos seguido del path para que el cuit se recupere a traves del mismo.

Con esto esto dicho, el resultado de esa consulta seria el siguiente:



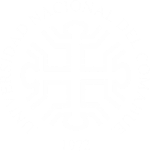
El resultado se muestra en formato JSON en la seccion de Response. El codigo 200 OK es un codigo de HTTP que indica que la solicitud se completo exitosamente.

En el caso de que se ingrese un cuit que no exista en la base de datos se devuelve un codigo 404 que indica que la entidad que se estaba buscando no se encuentra en la base de datos

Si se llegara a ingresar un cuit invalido, el codigo de respuesta es un 400 que significa que el servidor no puede procesar la peticion debido a un error del cliente, en este caso por el cuit invalido.

Estos codigos de error se utilizan para que la API Rest nos brinde informacion apropiada para cada peticion

# s e r v i c e ?



B E C A P P U

Universidad nacional del comahue

**¿ C ó m o**

**a g r e g a r**

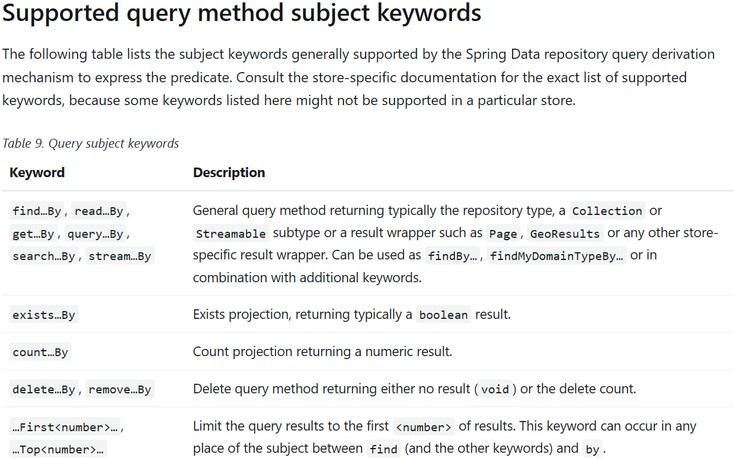
**u n**

**n u e v o**

**w e b**

Primero que nada se debe definir la consulta y en base a esta, lo que hacemos es definir una clase DTO para mostrar solo los campos que la consulta solicita. Luego de esto, lo que hacemos es definir en el repositorio de la entidad correspondiente la consulta deseada.

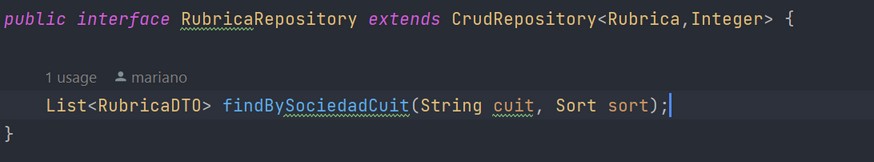
Hay varias maneras de recuperar informacion de la base de datos con spring, una de ellas es a traves de los metodos de consulta.



Referencia: https://docs.spring.io/spring- data/jpa/docs/current/reference/html/#repositories.query-methods.details

Una vez que tenemos la consulta lo que hacemos es elegir el tipo de entidad que va a retornar la misma, si definimos que la consulta retorna un tipo de dato DTO, tenemos que tener sumo cuidado a la hora de definir los atributos en el DTO, es decir, el DTO debe tener los mismos atributos que define la consulta, para que el mapeo entre la entidad retornada y el DTO

sea exitoso



Luego de esto, lo que hacemos es crear el servicio e inyectamos el repositorio para poder llamarlo y recuperar la consulta.

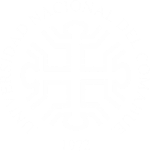


Por ultimo lo que hacemos es crear el controlador para poder manejarlas peticiones HTTP.



Con la notacion @GetMapping indicamos que se espera una peticion de tipo GET, luego escribimos la url que queremos definir para esa peticion, por ultimo en el encabezado indicamos que el parametro de busqueda lo recuperamos del path (notacion @PathVariable), que en este caso es el CUIT

# a p l i c a c i o n ?



B E C A P P U

Universidad nacional del comahue

**¿ C ó m o h a c e r e l d e s p l i e g u e d e l a**

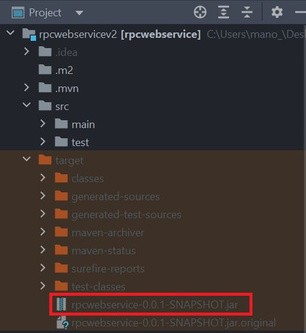
Primero que nada debemos instalar y configurar Docker Desktop en nuestra maquina.

<https://docs.docker.com/engine/install/>

Una vez que docker este instalado y ejecutandose en nuestra maquina, debemos crear el ejecutable de nuestra aplicacion. Para hacer esto abrimos la terminal en el IDE que estemos utilizando y escribimos el siguiente comando



Luego de ejecutar este comando se va a crear una carpeta denominada "target" en el directorio de nuestro proyecto, dentro de esta carpeta se va a crear el ejecutable .jar de nuestra aplicacion.



Con este archivo ya podemos crear el Dockerfile que utiliza docker para leer y ejecutar nuestra aplicacion.

Lo que hacemos es crear un nuevo archivo denominado Dockerfile en el directorio del proyecto y escribimos los comandos para crear la imagen de nuestra aplicacion



1. **FROM openjdk:17.** Este comando indica la imagen base que se utilizará para construir la nueva imagen. En este caso, se está utilizando la imagen openjdk:17, que es una imagen oficial de OpenJDK versión 17. Esto significa que la imagen de Docker resultante tendrá instalado Java Development Kit (JDK) 17.
2. **ARG JAR\_FILE=target/\*.jar.** Este comando define una variable llamada JAR\_FILE y le asigna el valor target/\*.jar. La variable JAR\_FILE se utiliza posteriormente para copiar el archivo JAR de la aplicación en la imagen de Docker. En este caso, se espera que exista un archivo JAR en la carpeta target del contexto de construcción.
3. **COPY ${JAR\_FILE} app.jar.** Aquí se copia el archivo JAR de la aplicación en la imagen de Docker. Se utiliza la variable JAR\_FILE definida anteriormente para especificar la ruta del archivo JAR en el contexto de construcción. El archivo JAR se copiará en la ubicación /app.jar dentro de la imagen de Docker**.**
4. **ENTRYPOINT ["java","-jar","/app.jar"].** Este comando establece el punto de entrada (entrypoint) para la imagen de Docker. Cuando se ejecute un contenedor basado en esta imagen, se ejecutará el comando java -jar

/app.jar. Esto significa que el contenedor ejecutará el archivo JAR de la aplicación utilizando el intérprete de Java

# D o c k e r C o m p o s e

Es una herramienta para definir y ejecutar aplicaciones Docker multicontenedor que permite simplificar el uso de Docker a partir de archivos YAML, de está forma es mas sencillo crear contendores que se relacionen entre sí, conectarlos, habilitar puertos, volumenes, etc. Nos permite lanzar un solo comando para crear e iniciar todos los servicios desde su configuración(YAML), esto significa que puedes crear diferentes contenedores y al mismo tiempo diferentes servicios en cada contenedor, integrarlos a un volumen común e iniciarlos y/o apagarlos, etc. Este es un componente fundamental para poder construir aplicaciones y microservicios.

Docker-Compose funciona en todos los entornos: production, staging, development, testing, así como flujos de trabajo basados en Continuous Integration(CI).

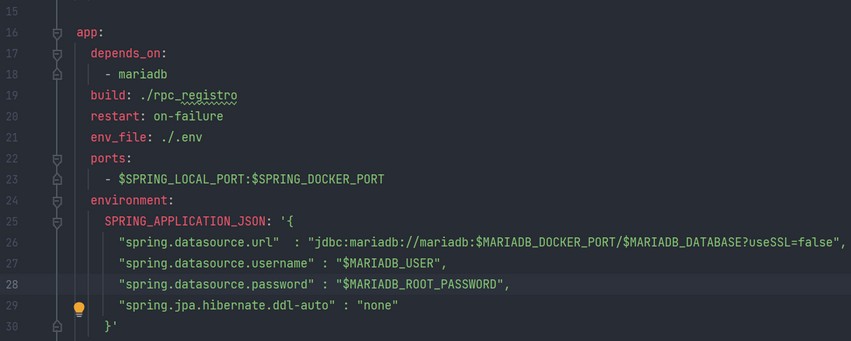
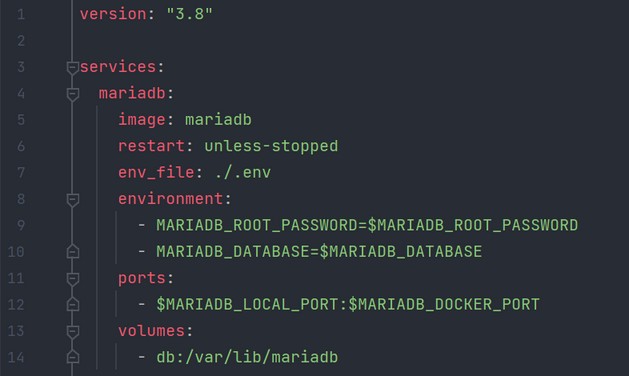
La idea de utilizar Docker Compose es poder facilitar la creacion y configuracion de cada contenedor, es decir el contenedor que contiene la aplicacion y el contenedor que tiene la base de datos para que se conecten de forma automatica.

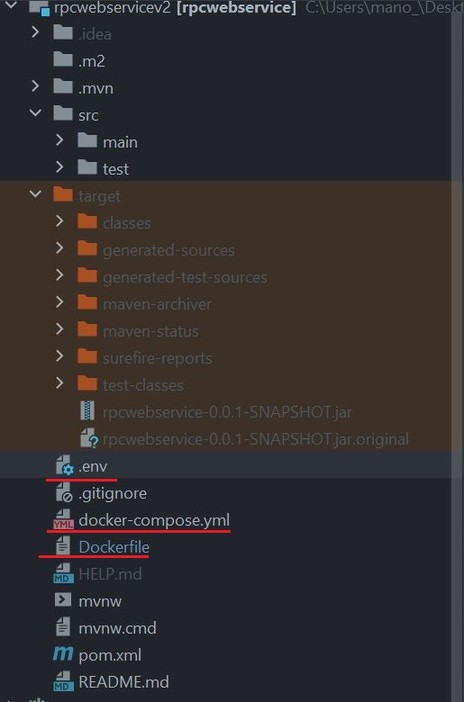
# ¿ C o m o u s a r D o c k e r C o m p o s e ?

Lo que tenemos es crear un archivo de configuracion con la extension .yml En el mismo debemos escribir la configuracion de nuestros contenedores que serian la imagen que se va utilizar para cada uno de los contenedores, los puertos que se van a utilizar, los volumenes que se van a utilizar y las variables de entorno a utilizar.

Por otro lado, se necesita crear un archivo .env para almacenar las variables de entorno correspondientes, y luego llamar a este archivo desde el docker-compose.yml

Ejemplo





Una vez que tenemos estos 3 archivos creados, podemos proceder a ejecutar el comando para correr nuestra aplicacion en el entorno de docker.

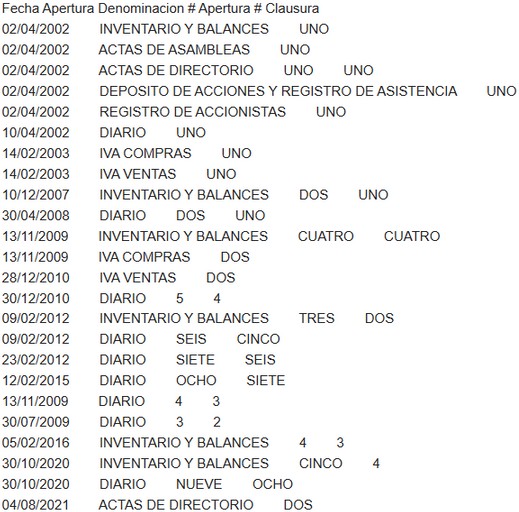
Abrimos la terminal y ejecutamos el siguiente comando



# C o n s u l t a R u b r i c a s

select distinct r.id\_sociedad, r.apertura as Fecha\_apertura, r.denominacion as Ru\_Denominacion,r.numero\_apertura, r.numero\_clausura from rpc\_registro.sociedades s left join rpc\_registro.rubricas r on s.id = r.id\_sociedad where s.cuit = 30708012293

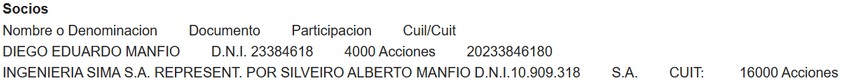
Resultado:



# C o n s u l t a S o c i o s

select s.iddenominacion,tp.nombre as Participacion,pj.denominacion deno, p2.nombre,p2.apellido,p2.numero\_documento,p2.sexo from rpc\_registro.sociedades s left join rpc\_registro.socios s2 on s.id = s2.id\_sociedad left JOIN rpc\_registro.personas p2 on s2.id\_persona = p2.id left JOIN rpc\_registro.personas\_juridicas pj on s2.id\_persona\_juridica = pj.id or s2.id\_sociedad = pj.id\_sociedad left JOIN rpc\_registro.tipos\_participacion tp on s2.id\_tipo\_participacion = tp.id WHERE s2.activo = 1 AND where s.cuit = 30708012293

Resultado:



# C o n s u l t a A n t e c e d e n t e s

SELECT

an.fecha,

descripcion,

tan.nombre TipoAntecedente,

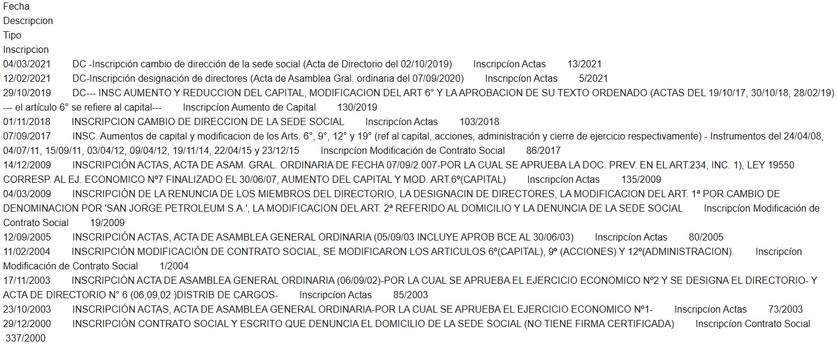
concat(i.numero,'/',i.tomo\_anio) Inscripcion FROM

rpc\_registro.antecedentes an inner join

rpc\_registro.tipos\_antecedentes tan on an.id\_tipo\_antecedente = tan.id inner JOIN

rpc\_registro.inscripciones i on i.id = an.id\_inscripcion inner JOIN rpc\_registro.sociedades s on s.id = an.id\_sociedad

WHERE s.cuit = '30708012293'

order by fecha Resultado:

# C o n s u l t a I n s c r i p c i o n y S e d e

select distinct s.denominacion, s.cuit, ts.siglas as TipoSociedad, s.fecha\_instrumento,

MIN(i.fecha) as Fecha\_Inscripcion, i.numero,

i.folio\_desde, i.folio\_hasta, i.tomo\_numero, i.tomo\_anio,

ss.calle, ss.numero,ss.otros as observaciones, l.nombre as Localidad, p.nombre as Provincias

from

rpc\_registro.sociedades s left join

rpc\_registro.expedientes e on s.id = e.id\_sociedad inner JOIN

rpc\_registro.inscripciones i on e.id = i.id\_expediente inner JOIN rpc\_registro.tipos\_sociedad ts on s.id\_tipo\_sociedad = ts.id

inner JOIN

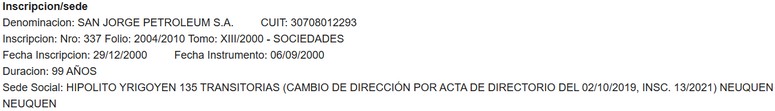
rpc\_registro.sede\_social ss on s.id = ss.id\_sociedad inner JOIN rpc\_registro.localidades l on ss.id\_localidad = l.id inner JOIN rpc\_registro.provincias p on l.id\_provincia = p.id

WHERE

s.cuit = '30708012293'

and ss.validado = 1

order by s.id, s.denominacion asc Resultado:



# D o c u m e n t a c i o n W e b S e r v i c e d e

# R u b r i c a s

## S e r v i c i o W e b : “ / s o c i e d a d \_ r u b r i c a s ” e n X - R o a d

**U r l d e l a f u e n t e A u t e n t i c a**

**h t t p s : / / x r o a d . j u s n e u q u e n . g o v . a r / r 1 / O P T I C / G O B / G O B 0 0 0 1 4 / G P - P O D E R J U D I C I A L /**

**Entorno**

Test

Producci on

**Metodo**

GET

GET

**Service Code**

desa

prod

**Punto Acceso**

api/sociedad\_rubr icas

api/sociedad\_rubr icas

**Ejemplo**

https:// <url xroad local>/

r1/OPTIC/GOB/GOB00014/GP- PODERJUDICIAL/desa

/sociedad\_rubricas

https:// <url xroad local>/ r1/OPTIC/GOB/GOB00014/GP- PODERJUDICIAL/prod

/sociedad\_rubricas

El cuit se debe enviar a traves de la URL. Luego del punto de acesso va el cuit, sin guiones ni puntos.

Formato

String

Ejemplo

30708012293

# R e s p o n s e

|  |  |
| --- | --- |
| **Status** | **Metodo** |
| 200 | [  {  "numero\_apertura": "UNO", "numero\_clausura": null,  "ru\_Denominacion": "ACTAS DE ASAMBLEAS", "fecha\_apertura": "2002-04-02T03:00:00.000+00:00",  "sociedadId": 87  },  {  "numero\_apertura": "UNO", "numero\_clausura": "UNO", "ru\_Denominacion": "ACTAS DE DIRECTORIO",  "fecha\_apertura": "2002-04-02T03:00:00.000+00:00",  "sociedadId": 87  },  {  "numero\_apertura": "UNO", "numero\_clausura": null,  "ru\_Denominacion": "DEPOSITO DE ACCIONES Y REGISTRO DE ASISTENCIA",  "fecha\_apertura": "2002-04-02T03:00:00.000+00:00",  "sociedadId": 87  },  {  "numero\_apertura": "UNO", "numero\_clausura": null,  "ru\_Denominacion": "INVENTARIO Y BALANCES", "fecha\_apertura": "2002-04-02T03:00:00.000+00:00",  "sociedadId": 87  }, |

{

"numero\_apertura": "UNO", "numero\_clausura": null,

"ru\_Denominacion": "REGISTRO DE ACCIONISTAS", "fecha\_apertura": "2002-04-02T03:00:00.000+00:00",

"sociedadId": 87

},

{

"numero\_apertura": "UNO", "numero\_clausura": null, "ru\_Denominacion": "DIARIO",

"fecha\_apertura": "2002-04-10T03:00:00.000+00:00",

"sociedadId": 87

},

{

"numero\_apertura": "UNO", "numero\_clausura": null, "ru\_Denominacion": "IVA COMPRAS",

"fecha\_apertura": "2003-02-14T03:00:00.000+00:00",

"sociedadId": 87

},

{

"numero\_apertura": "UNO", "numero\_clausura": null, "ru\_Denominacion": "IVA VENTAS",

"fecha\_apertura": "2003-02-14T03:00:00.000+00:00",

"sociedadId": 87

},

{

"numero\_apertura": "DOS", "numero\_clausura": "UNO",

"ru\_Denominacion": "INVENTARIO Y BALANCES", "fecha\_apertura": "2007-12-10T03:00:00.000+00:00",

"sociedadId": 87

},

{

{

"numero\_apertura": "3",

"numero\_clausura": "2", "ru\_Denominacion": "DIARIO",

"fecha\_apertura": "2009-07-30T03:00:00.000+00:00",

"sociedadId": 87

},

{

"numero\_apertura": "4",

"numero\_clausura": "3", "ru\_Denominacion": "DIARIO",

"fecha\_apertura": "2009-11-13T03:00:00.000+00:00",

"sociedadId": 87

},

{

"numero\_apertura": "CUATRO", "numero\_clausura": "CUATRO", "ru\_Denominacion": "INVENTARIO Y BALANCES",

"fecha\_apertura": "2009-11-13T03:00:00.000+00:00",

"sociedadId": 87

},

{

"numero\_apertura": "DOS", "numero\_clausura": null, "ru\_Denominacion": "IVA COMPRAS",

"fecha\_apertura": "2009-11-13T03:00:00.000+00:00",

"sociedadId": 87

},

{

"numero\_apertura": "DOS", "numero\_clausura": null, "ru\_Denominacion": "IVA VENTAS",

"fecha\_apertura": "2010-12-28T03:00:00.000+00:00",

"sociedadId": 87

},

{

"numero\_apertura": "5",

"numero\_clausura": "4", "ru\_Denominacion": "DIARIO",

"fecha\_apertura": "2010-12-30T03:00:00.000+00:00",

"sociedadId": 87

},

{

"numero\_apertura": "SEIS", "numero\_clausura": "CINCO", "ru\_Denominacion": "DIARIO",

"fecha\_apertura": "2012-02-09T03:00:00.000+00:00",

"sociedadId": 87

},

{

"numero\_apertura": "TRES", "numero\_clausura": "DOS",

"ru\_Denominacion": "INVENTARIO Y BALANCES", "fecha\_apertura": "2012-02-09T03:00:00.000+00:00",

"sociedadId": 87

},

{

"numero\_apertura": "SIETE", "numero\_clausura": "SEIS", "ru\_Denominacion": "DIARIO",

"fecha\_apertura": "2012-02-23T03:00:00.000+00:00",

"sociedadId": 87

},

{

"numero\_apertura": "OCHO", "numero\_clausura": "SIETE", "ru\_Denominacion": "DIARIO",

"fecha\_apertura": "2015-02-12T03:00:00.000+00:00",

"sociedadId": 87

},

{

"numero\_apertura": "4",

"numero\_clausura": "3",

"ru\_Denominacion": "INVENTARIO Y BALANCES", "fecha\_apertura": "2016-02-05T03:00:00.000+00:00",

"sociedadId": 87

},

{

"numero\_apertura": "NUEVE", "numero\_clausura": "OCHO", "ru\_Denominacion": "DIARIO",

"fecha\_apertura": "2020-10-30T03:00:00.000+00:00",

"sociedadId": 87

},

{

"numero\_apertura": "CINCO", "numero\_clausura": "4",

"ru\_Denominacion": "INVENTARIO Y BALANCES", "fecha\_apertura": "2020-10-30T03:00:00.000+00:00",

"sociedadId": 87

},

{

"numero\_apertura": "DOS", "numero\_clausura": null,

"ru\_Denominacion": "ACTAS DE DIRECTORIO", "fecha\_apertura": "2021-08-04T03:00:00.000+00:00",

"sociedadId": 87

}

]

404

400

500

{

"statusCode": 404,

"timestamp": "2023-05-12T17:52:26.791+00:00",

"message": "No se encontro sociedad con el cuit ingresado",

"description": "uri=/api/sociedad\_rubricas/11111111111"

}

{

"statusCode": 400,

"timestamp": "2023-05-12T17:53:44.213+00:00",

"message": "Formato de Cuit Invalido", "description": "uri=/api/sociedad\_rubricas/3012fsd2"

}

{

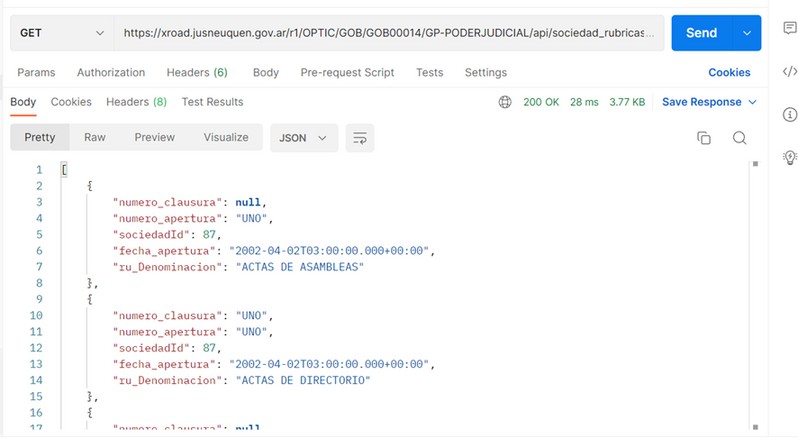
"status": "error", "result": { "error\_id": "500",

"error\_msg": "Server Internal Error"

}

}

# E j e m p l o U s a n d o P o s t m a n



**D o c u m e n t a c i o n W e b S e r v i c e d e S o c i o s**

## S e r v i c i o W e b : “ / s o c i e d a d \_ s o c i o s ” e n X - R o a d

**U r l d e l a f u e n t e A u t e n t i c a**

**h t t p s : / / x r o a d . j u s n e u q u e n . g o v . a r / r 1 / O P T I C / G O B / G O B 0 0 0 1 4 / G P - P O D E R J U D I C I A L /**

**Entorno**

Test

Producci on

**Metodo**

GET

GET

**Service Code**

desa

prod

El cuit se debe enviar a traves de la URL. Luego del punto de acesso va el cuit, sin guiones ni puntos.

**Punto Acceso**

api/sociedad\_soci os

api/sociedad\_soci os

**Ejemplo**

https:// <url xroad local>/

r1/OPTIC/GOB/GOB00014/GP- PODERJUDICIAL/desa

/sociedad\_socios

https:// <url xroad local>/ r1/OPTIC/GOB/GOB00014/GP- PODERJUDICIAL/prod

/sociedad\_socios

Formato

Ejemplo

String

30708012293

**R e s p o n s e**

|  |  |
| --- | --- |
| **Status** | **Metodo** |
| 200 | [  {  "nombre": "DIEGO EDUARDO",  "apellido": "MANFIO", "sexo": null,  "denominacion": "SAN JORGE PETROLEUM S.A. ", "participacion": "Acciones",  "sociedadId": 87,  "numeroDocumento": "23384618", "deno": ""  },  {  "nombre": "",  "apellido": "",  "sexo": "",  "denominacion": "SAN JORGE PETROLEUM S.A. ", "participacion": "Acciones",  "sociedadId": 87, "numeroDocumento": "",  "deno": "INGENIERIA SIMA S.A. REPRESENT. POR SILVEIRO ALBERTO MANFIO D.N.I.10.909.318"  }  ] |
| 404 | {  "status": "error", "result": { "error\_id": "404",  "error\_msg": "Cuit ingresado no corresponde a un cuit existente"  } |

{

"status": "error", "result": {

400

"error\_id": "400",

"error\_msg": "Cuit ingresado no es valido, tiene un formato incorrecto"

}

}

500

{

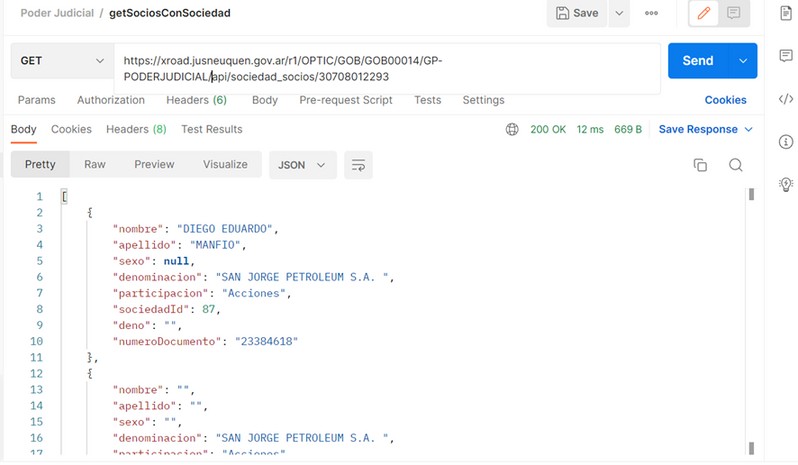
"status": "error", "result": { "error\_id": "500",

"error\_msg": "Server Internal Error"

}

}

**E j e m p l o u s a n d o P o s t m a n**



**D o c u m e n t a c i o n W e b S e r v i c e d e A n t e c e d e n t e s**

## S e r v i c i o W e b : “ / s o c i e d a d \_ a n t e c e d e n t e s ” e n X - R o a d

**U r l d e l a f u e n t e A u t e n t i c a**

**h t t p s : / / x r o a d . j u s n e u q u e n . g o v . a r / r 1 / O P T I C / G O B / G O B 0 0 0 1 4 / G P - P O D E R J U D I C I A L /**

**Entorno**

Test

Producci on

**Metodo**

GET

GET

**Service Code**

desa

prod

El cuit se debe enviar a traves de la URL. Luego del punto de acesso va el cuit, sin guiones ni puntos.

**Punto Acceso**

api/sociedad\_ant ecedentes

api/sociedad\_ant ecedentes

**Ejemplo**

https:// <url xroad local>/

r1/OPTIC/GOB/GOB00014/GP- PODERJUDICIAL/desa

/sociedad\_antecedentes

https:// <url xroad local>/ r1/OPTIC/GOB/GOB00014/GP- PODERJUDICIAL/prod

/sociedad\_antecedentes

Formato

Ejemplo

String

30708012293

# R e s p o n s e

**Status Metodo**

200

[

{

"inscripcion": "337/2000",

"descripcion": "INSCRIPCIÓN CONTRATO SOCIAL Y ESCRITO QUE DENUNCIA EL DOMICILIO DE LA SEDE SOCIAL (NO TIENE FIRMA CERTIFICADA)",

"fecha": "2000-12-29 00:00:00.0",

"tipoAntecedente": "Inscripcíon Contrato Social"

},

{

"inscripcion": "73/2003",

"descripcion": "INSCRIPCIÓN ACTAS, ACTA DE ASAMBLEA GENERAL ORDINARIA-POR LA CUAL SE APRUEBA EL EJERCICIO ECONOMICO Nº1-",

"fecha": "2003-10-23 00:00:00.0",

"tipoAntecedente": "Inscripcíon Actas"

},

{

"inscripcion": "85/2003",

"descripcion": "INSCRIPCIÓN ACTA DE ASAMBLEA GENERAL ORDINARIA (06/09/02)-POR LA CUAL SE APRUEBA EL EJERCICIO ECONOMICO Nº2 Y SE DESIGNA EL DIRECTORIO- Y ACTA DE DIRECTORIO N° 6 (06,09,02 )DISTRIB DE CARGOS-",

"fecha": "2003-11-17 00:00:00.0",

"tipoAntecedente": "Inscripcíon Actas"

},

# R e s p o n s e

{

"inscripcion": "1/2004",

"descripcion": "INSCRIPCIÓN MODIFICACIÓN DE CONTRATO SOCIAL, SE MODIFICARON LOS ARTICULOS 6º(CAPITAL), 9º (ACCIONES) Y 12º(ADMINISTRACION).",

"fecha": "2004-02-11 00:00:00.0",

"tipoAntecedente": "Inscripcíon Modificación de Contrato Social"

},

{

"inscripcion": "80/2005",

"descripcion": "INSCRIPCIÓN ACTAS, ACTA DE ASAMBLEA GENERAL ORDINARIA (05/09/03 INCLUYE APROB BCE AL 30/06/03)",

"fecha": "2005-09-12 00:00:00.0",

"tipoAntecedente": "Inscripcíon Actas"

},

{

"inscripcion": "19/2009",

"descripcion": "INSCRIPCIÓN DE LA RENUNCIA DE LOS MIEMBROS DEL DIRECTORIO, LA DESIGNACIN DE DIRECTORES, LA MODIFICACION DEL ART. 1ª POR CAMBIO DE DENOMINACION POR 'SAN JORGE PETROLEUM S.A.', LA MODIFICACION DEL ART. 2ª REFERIDO AL DOMICILIO Y LA DENUNCIA DE LA SEDE SOCIAL",

"fecha": "2009-03-04 00:00:00.0",

"tipoAntecedente": "Inscripcíon Modificación de Contrato Social"

},

{

"inscripcion": "135/2009",

"descripcion": "INSCRIPCIÓN ACTAS, ACTA DE ASAM. GRAL. ORDINARIA DE FECHA 07/09/2.007-POR LA CUAL SE APRUEBA LA DOC. PREV. EN EL ART.234, INC. 1), LEY 19550 CORRESP. AL EJ.ECONOMICO Nº7 FINALIZADO EL 30/06/07, AUMENTO DEL CAPITAL Y MOD. ART.6º(CAPITAL)",

"fecha": "2009-12-14 00:00:00.0",

"tipoAntecedente": "Inscripcíon Actas"

},

# R e s p o n s e

{

"inscripcion": "86/2017",

"descripcion": "INSC. Aumentos de capital y modificacion de los Arts. 6°, 9°, 12° y 19° (ref al capital, acciones, administración y cierre de ejercicio respectivamente) - Instrumentos del 24/04/08, 04/07/11, 15/09/11, 03/04/12, 09/04/12, 19/11/14, 22/04/15 y 23/12/15", "fecha": "2017-09-07 00:00:00.0",

"tipoAntecedente": "Inscripcíon Modificación de Contrato Social"

},

{

"inscripcion": "103/2018",

"descripcion": "INSCRIPCION CAMBIO DE DIRECCION DE LA SEDE SOCIAL",

"fecha": "2018-11-01 00:00:00.0",

"tipoAntecedente": "Inscripcíon Actas"

},

{

"inscripcion": "130/2019",

"descripcion": "DC--- INSC AUMENTO Y REDUCCION DEL CAPITAL, MODIFICACION DEL ART 6° Y LA APROBACION DE SU TEXTO ORDENADO (ACTAS DEL 19/10/17, 30/10/18, 28/02/19) --- el

artículo 6° se refiere al capital---", "fecha": "2019-10-29 00:00:00.0",

"tipoAntecedente": "Inscripcíon Aumento de Capital"

},

{

"inscripcion": "5/2021",

"descripcion": "DC-Inscripción designación de directores (Acta de Asamblea Gral. ordinaria del 07/09/2020)",

"fecha": "2021-02-12 00:00:00.0",

"tipoAntecedente": "Inscripcíon Actas"

},

{

"inscripcion": "13/2021",

"descripcion": "DC -Inscripción cambio de dirección de la sede social (Acta de Directorio del 02/10/2019)",

"fecha": "2021-03-04 00:00:00.0",

"tipoAntecedente": "Inscripcíon Actas"

}]

**R e s p o n s e**

{

404

"statusCode": 400,

"timestamp": "2023-05-12T18:28:15.647+00:00",

"message": "Formato de Cuit Invalido", "description":

"uri=/api/sociedad\_antecedentes/30%207080122%20931"

}

{

400

"statusCode": 404,

"timestamp": "2023-05-12T18:28:50.478+00:00",

"message": "No se encontro sociedad con el cuit ingresado", "description": "uri=/api/sociedad\_antecedentes/11321321321"

}

500

{

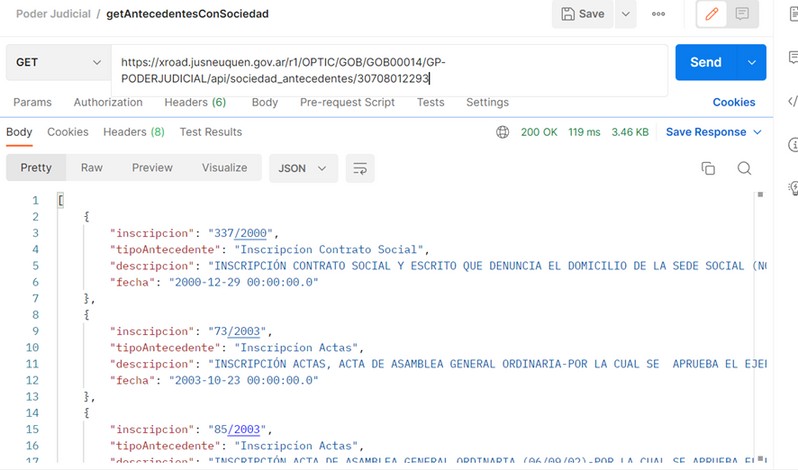
"status": "error", "result": { "error\_id": "500",

"error\_msg": "Server Internal Error"

}

}

**E j e m p l o u s a n d o P o s t m a n**



**D o c u m e n t a c i o n W e b S e r v i c e d e I n s c r i p c i o n y S e d e**

## S e r v i c i o W e b : “ / s o c i e d a d \_ i n s c r i p c i o n \_ s e d e s o c i a l ” e n X - R o a d

**U r l d e l a f u e n t e A u t e n t i c a**

**h t t p s : / / x r o a d . j u s n e u q u e n . g o v . a r / r 1 / O P T I C / G O B / G O B 0 0 0 1 4 / G P - P O D E R J U D I C I A L /**

**Entorno**

Test

Producci on

**Metodo**

GET

GET

**Service Code**

desa

prod

**Punto Acceso**

api/sociedad\_insc ripcion\_sedesocia l

api/sociedad\_insc ripcion\_sedesocia l

El cuit se debe enviar a traves de la URL. Luego del punto de acesso va el cuit, sin guiones ni puntos.

**Ejemplo**

https:// <url xroad local>/

r1/OPTIC/GOB/GOB00014/GP- PODERJUDICIAL/desa

/sociedad\_inscripcion\_sedesoci al

https:// <url xroad local>/ r1/OPTIC/GOB/GOB00014/GP- PODERJUDICIAL/prod

/sociedad\_inscripcion\_sedesoci al

Formato

Ejemplo

String

30708012293

**R e s p o n s e**

**Status**

**Metodo**

200

{

"fechaInscripcion": "2000-12-29T03:00:00.000+00:00",

"numero": 337,

"folio\_desde": 2004,

"folio\_hasta": 2010, "tomo\_numero": "XIII", "tomo\_anio": "2000",

"denominacion": "SAN JORGE PETROLEUM S.A. ", "cuit": "30708012293",

"fecha\_instrumento": "2000-09- 06T03:00:00.000+00:00",

"siglas": "S.A.", "duracion": 99, "sedesSocialesInfo": [

{

"calle": "HIPOLITO YRIGOYEN",

"numeroSedeSocial": "135",

"observaciones": "TRANSITORIAS (CAMBIO DE DIRECCIÓN POR ACTA DE DIRECTORIO DEL 02/10/2019, INSC. 13/2021)",

"localidad": "NEUQUEN", "provincia": "NEUQUEN"

}

]

}

**R e s p o n s e**

404

{

"status": "error", "result": { "error\_id": "404",

"error\_msg": "No se encontro sociedad con el cuit ingresado” "

}

}

400

{

"status": "error", "result": { "error\_id": "400",

"error\_msg": "Formato de cuit invalido"

}

}

500

{

"status": "error", "result": { "error\_id": "500",

"error\_msg": "Server Internal Error"

}

}

**E j e m p l o u s a n d o P o s t m a n**

