API MANAGEMENT

TG3-IMPLEMENTACIÓN DE LAS TECNOLOGÍAS SELECCIONADAS

DESARROLLO CON TECNOLOGÍAS EMERGENTES

Contenido

[1. Autores del trabajo, planificación y entrega 4](#_Toc8070024)

[1.1 Autores 4](#_Toc8070025)

[1.2 Planificación 4](#_Toc8070026)

[1.3 Entrega 4](#_Toc8070027)

[2. Requisitos del prototipo a implementar 4](#_Toc8070028)

[2.1 Requisitos funcionales 4](#_Toc8070029)

[2.2 Otros requisitos 5](#_Toc8070030)

[3. Criterios de comparación en la implementación 6](#_Toc8070031)

[3.1 Criterio 1: Dificultad del proceso de creación y publicación 6](#_Toc8070032)

[3.2 Criterio 2: Horas empleadas en la construcción del prototipo 6](#_Toc8070033)

[3.3 Criterio 3: Recursos necesarios 6](#_Toc8070034)

[3.4 Criterio 4: Configuración del servidor Proxy 6](#_Toc8070035)

[3.5 Criterio 5: Grado de personalización 6](#_Toc8070036)

[3.6 Criterio 6: Calidad del soporte 6](#_Toc8070037)

[3.7 Criterio 7: Calidad de la documentación 6](#_Toc8070038)

[3.8 Criterio 8: Facilidad de uso 7](#_Toc8070039)

[3.9 Criterio 9: Implementación de seguridad y control de concurrencia 7](#_Toc8070040)

[3.10 Criterio 10: Profundidad del análisis 7](#_Toc8070041)

[3.11 Criterio 11: Capacidad de prueba 7](#_Toc8070042)

[4. Proyecto de implementación de un prototipo del sistema utilizando la tecnología Apigee 8](#_Toc8070043)

[4.1 Documentación de diseño 8](#_Toc8070044)

[4.2 Documentación de construcción 10](#_Toc8070045)

[4.3 Documentación de pruebas 20](#_Toc8070046)

[4.4 Documentación de instalación 22](#_Toc8070047)

[4.5 Manual de usuario 23](#_Toc8070048)

[5. Proyecto de implementación de un prototipo del sistema utilizando la tecnología WSO2 25](#_Toc8070049)

[5.1 Documentación de diseño 25](#_Toc8070050)

[5.2 Documentación de construcción 27](#_Toc8070051)

[5.3 Documentación de pruebas 32](#_Toc8070052)

[5.4 Documentación de instalación 33](#_Toc8070053)

[5.5 Manual de usuario 33](#_Toc8070054)

[6. Comparación de las dos implementaciones 37](#_Toc8070055)

[6.1 Evaluación de los criterios en la implementación usando la tecnología Apigee 37](#_Toc8070056)

[6.2 Evaluación de los criterios en la implementación usando la tecnología WSO2 38](#_Toc8070057)

[7. Comparación de la implementación de las tecnologías 40](#_Toc8070058)

[8. Conclusiones 43](#_Toc8070059)

# 1. Autores del trabajo, planificación y entrega

## 1.1 Autores

Grupo T7:

* Laura Guillén Casal 03149840J (Coordinador)
* Francisco Miguel Sáez Bravo 50644226C
* Jesús Gail Bohórquez 09072386J
* Jorge Pérez Campos 09053996T
* Sergio Llanos García 09054884Z

## 1.2 Planificación

## 1.3 Entrega

<https://github.com/lauraguillen/TG3>

# 2. Requisitos del prototipo a implementar

## 2.1 Requisitos funcionales

A continuación, encontramos una tabla con los requisitos funcionales comunes para las dos tecnologías utilizadas. Cabe destacar que la selección de estos requisitos funcionales se encuentra claramente condicionada por las funcionalidades presentes en las versiones de prueba y por lo tanto accesibles para nosotros

| **REQ.** | **DESCRIPCIÓN** |
| --- | --- |
| RF01- Creación/Publicación de una API | Capacidad de crear una API y realizar su posterior implementación |
| RF02- Suscripción a un API | Posibilidad de suscribirse a una API, pudiendo invocarla posteriormente |
| RF03- Políticas de seguridad | Introducción de métodos de autenticación |
| RF04- Edición de la API | Capacidad de modificar y adaptar el código y diseño de la API que hayas creado anteriormente |
| RF05- Control de acceso | Métodos de gestión de acceso concurrente a las APIs |
| RF06- API Store | Portal donde se alojan las APIs y se presentan al usuario para su futura suscripción |
| RF07- Servidor Proxy | Creación de un servidor encargado de recibir las peticiones y procesarlas |
| RF08- Herramientas de Análisis | Conjunto de herramientas que ofrece la plataforma para analizar una API, su rendimiento, funcionamiento, etc. |

## 2.2 Otros requisitos

En la siguiente tabla se muestran los requisitos no funcionales que se tendrán en cuenta para el estudio de ambas tecnologías:

| **REQ.** | **DESCRIPCIÓN** |
| --- | --- |
| R01- Portal de desarrolladores | Apartado de comunicación entre los desarrolladores y miembros de la comunidad |
| R02- Soporte de la compañía | Existencia de un servicio permanente de ayuda a los usuarios por parte de la compañía |
| R03- Documentación | Información suministrada por la compañía para el uso de la plataforma |
| R04- Personalización | Capacidad de customizar tu API |
| R05- Testeabilidad | Posibilidad de probar el código y funcionalidad de la API |

# 3. Criterios de comparación en la implementación

## 3.1 Criterio 1: Dificultad del proceso de creación y publicación

*Nombre del criterio: Dificultad del proceso de creación y publicación.*

*Descripción: Grado de complejidad a la hora de crear y publicar una API.*

*Tipo de valor: Numérico (Escala 1-5).*

## 3.2 Criterio 2: Horas empleadas en la construcción del prototipo

*Nombre del criterio: Horas empleadas en la construcción del prototipo.*

*Descripción: Números de horas utilizadas para llevar a cabo la construcción de los prototipos.*

*Tipo de valor: Numérico (Horas).*

## 3.3 Criterio 3: Recursos necesarios

*Nombre del criterio: Recursos necesarios.*

*Descripción: Número de recursos que deben utilizarse para la utilización de las tecnologías.*

*Tipo de valor: Texto libre.*

## 3.4 Criterio 4: Configuración del servidor Proxy

*Nombre del criterio: Configuración del servidor Proxy.*

*Descripción: Complejidad y rapidez en la puesta en funcionamiento del servidor proxy.*

*Tipo de valor: Numérico (Escala 1-5).*

## 3.5 Criterio 5: Grado de personalización

*Nombre del criterio: Grado de personalización.*

*Descripción: Capacidad para modificar a gusto del usuario la interfaz de la API creada por el mismo.*

*Tipo de valor: Numérico (Escala 1-5).*

## 3.6 Criterio 6: Calidad del soporte

*Nombre del criterio: Calidad del soporte.*

*Descripción: Facilidad de contacto con el soporte de la plataforma y eficiencia en la respuesta de estos.*

*Tipo de valor: Numérico (Escala 1-5).*

## 3.7 Criterio 7: Calidad de la documentación

*Nombre del criterio: Calidad de la documentación.*

*Descripción: Nivel de detalle y claridad de la información que brinda la plataforma.*

*Tipo de valor: Numérico (Escala 1-5).*

## 3.8 Criterio 8: Facilidad de uso

*Nombre del criterio: Facilidad de uso.*

*Descripción: Grado de facilidad de uso y acceso a las distintas funcionalidades de las plataformas.*

*Tipo de valor: Numérico (Escala 1-5).*

## 3.9 Criterio 9: Implementación de seguridad y control de concurrencia

*Nombre del criterio: Implementación de seguridad y control de concurrencia.*

*Descripción: Capacidad de implementar seguridad y control de acceso a la API creada.*

*Tipo de valor: Numérico (Escala 1-5).*

## 3.10 Criterio 10: Profundidad del análisis

*Nombre del criterio: Profundidad del análisis.*

*Descripción: Cantidad y variedad de parámetros que ofrece la plataforma para el análisis de las APIs.*

*Tipo de valor: Numérico (Escala 1-5).*

## 3.11 Criterio 11: Capacidad de prueba

*Nombre del criterio: Capacidad de prueba.*

*Descripción: Posibilidad de realizar un test de prueba del código de la API.*

*Tipo de valor: Booleano (Si/No)*

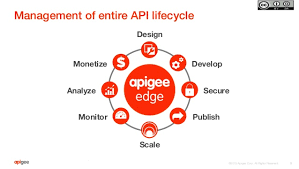
# 4. Proyecto de implementación de un prototipo del sistema utilizando la tecnología Apigee

Partiendo de la base establecida por los requisitos funcionales y no funcionales definidos en el apartado anterior, y en base a las funcionalidades disponibles en las versiones de prueba que se nos han proporcionado, se plantea elaborar una simulación que recoja las características básicas de una plataforma de API Management, haciendo uso en este apartado concreto de la herramienta Apigee.

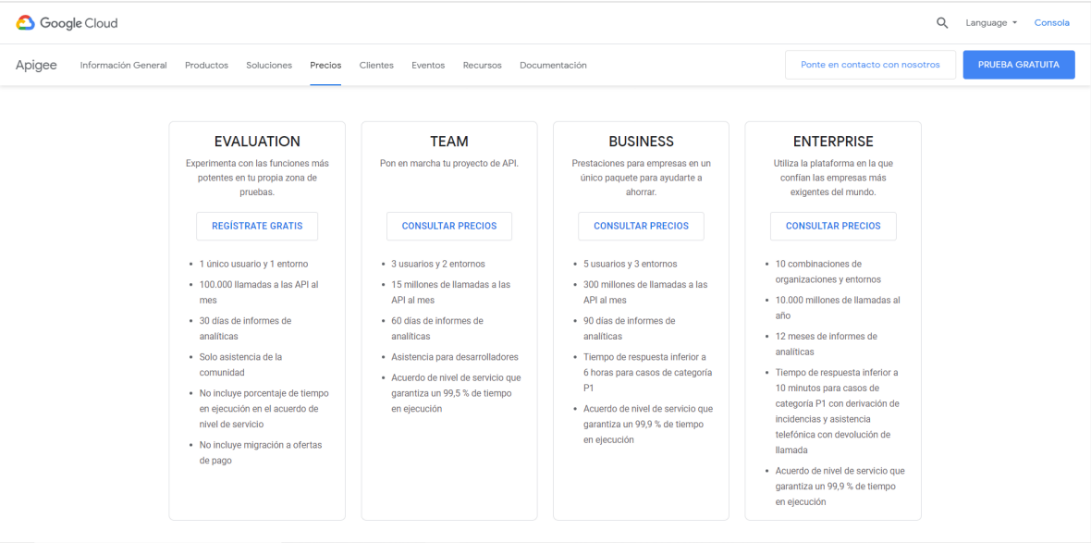
## 4.1 Documentación de diseño

Tal y como se ha mencionado anteriormente, el primer paso a la hora de comenzar con el planteamiento del diseño del prototipo pasa por definir los límites existentes en cuanto a funcionalidades en las herramientas utilizadas.

Se entiende por API Management el proceso de publicar, promocionar y supervisar APIs en un entorno seguro y escalable, siendo por lo tanto un concepto amplio que lleva asociado un número considerable de tareas y funciones; todas ellas gestionadas a través de Apigee Edge, tal y como puede apreciarse en la siguiente imagen:



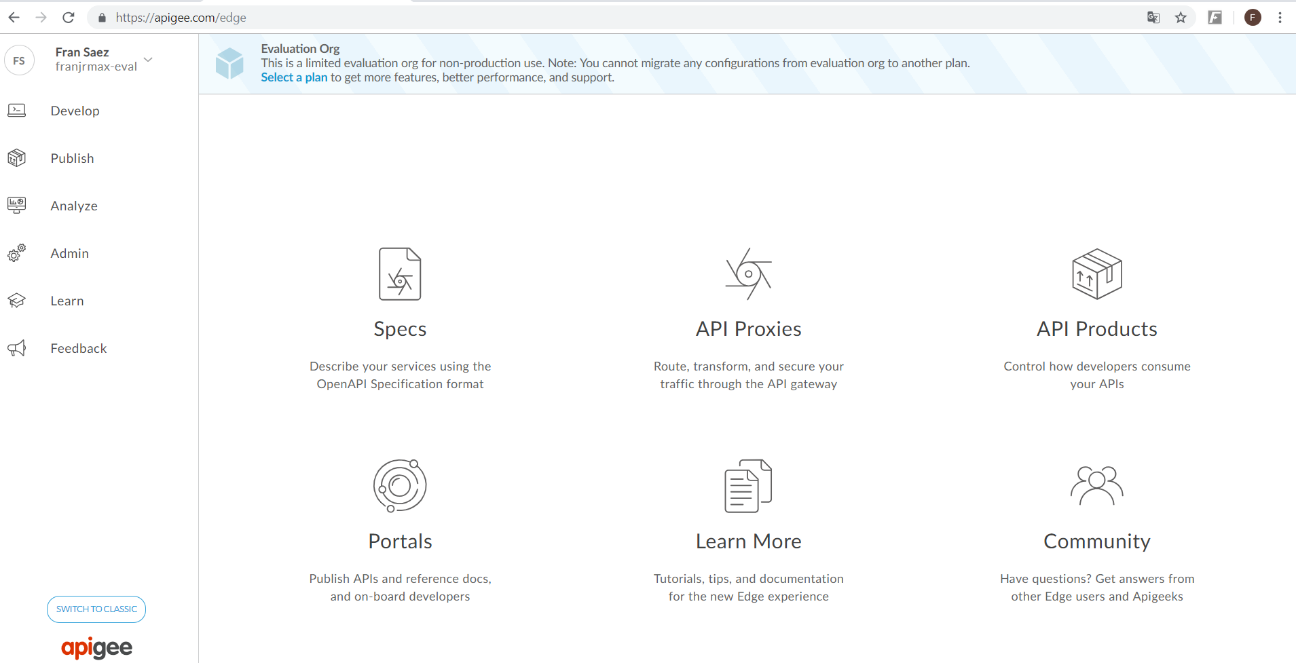
Es en este punto donde es conveniente destacar que la única versión de prueba que Apigee pone a disposición de los usuarios capa algunas de estas funciones, puesto que estarían más destinadas a un uso profesional y lucrativo de la plataforma (como por ejemplo el apartado de monetización de las APIs), siendo por lo tanto indispensable la suscripción a uno de los planes de pago existentes para el acceso a dichas funciones.



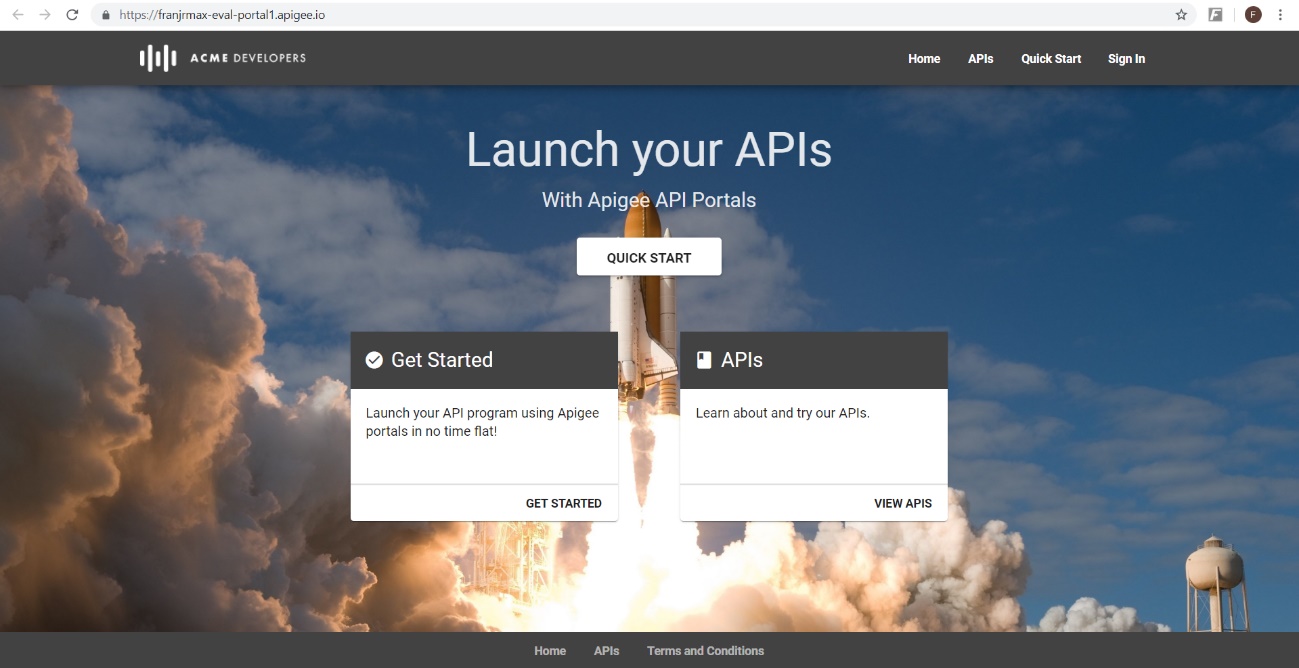
En consecuencia, tras analizar las funcionalidades accesibles en base a la circunstancia descrita en este apartado, se plantea realizar un prototipo que refleje la **creación y publicación de APIs**, **suscripción** a una API, **creación de servidor proxy** e implementación de **políticas de autenticación y control de acceso concurrente**.

Finalmente, en cuanto al diseño de la interfaz de usuario, cabe destacar que en el caso de las tecnologías de API Management seleccionadas no es necesario desarrollar dicho apartado, puesto que todo el trabajo se realiza a través de la plataforma e interfaz proporcionadas por estas. Como elemento diferenciador, Apigee permite personalizar un portal de publicación de APIs.

*Interfaz de Apigee Edge:*



*Interfaz del portal:*

**

## 4.2 Documentación de construcción

En este apartado se describe, paso a paso e incluyendo la máxima cantidad de detalles posible, el proceso de construcción del prototipo definido con anterioridad. Por lo tanto, se considera que este apartado 4.2 engloba gran parte de los contenidos del apartado 4.4, debido principalmente al grado de detalle previamente comentado con el que se expondrá el proceso de construcción del prototipo; por lo que en el apartado 4.4 únicamente se detallarán aspectos muy concretos que pueden servir de utilidad para un usuario externo.

Tras esta introducción previa, es posible comenzar con la descripción de la construcción del prototipo basado en la herramienta Apigee. En primer lugar, es necesario crear una cuenta en Apigee para así poder acceder a la versión de prueba de la plataforma, alojada en la Cloud de Google y a través de la cual se podrán realizar todos los procedimientos que se expondrán posteriormente.

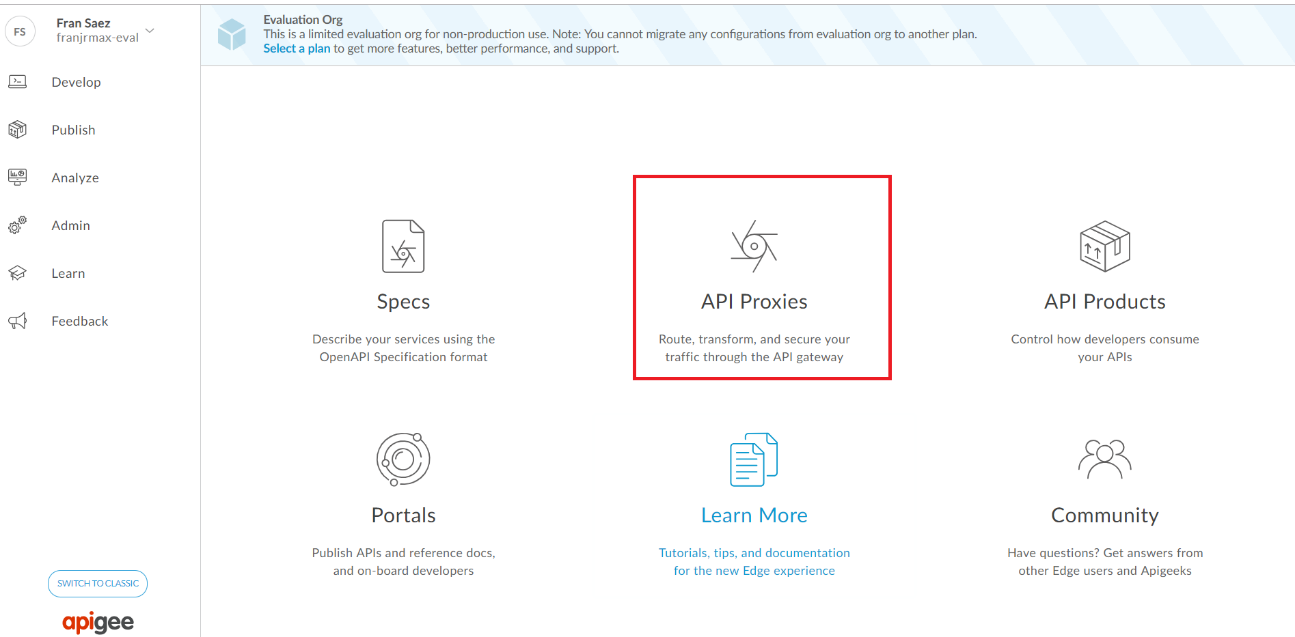
*Acceso al portal de creación de cuenta y solicitud de prueba gratuita. La interfaz de la versión gratuita de la plataforma aparece previamente ilustrada en el apartado 4.1 de este mismo documento:*



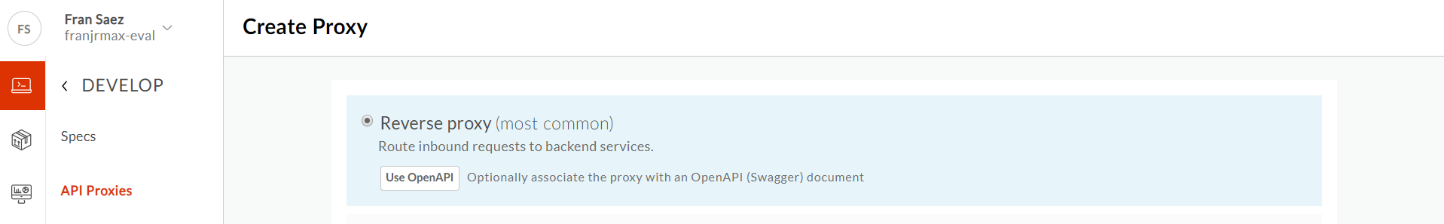
Paralelamente, y debido principalmente al hecho de que no existe experiencia ni conocimientos previos acerca del manejo de las herramientas que son objeto de estudio en este documento en el momento de la realización del mismo, Apigee pone a disposición del usuario el enlace <https://docs.apigee.com/api-platform/get-started/get-started>, el cual supone un notable y constante apoyo durante todo el proceso aquí descrito.

Finalmente, el último de los recursos iniciales que será necesario para llevar a cabo la implementación deseada consiste en la descarga de Curl, una herramienta y librería que actúa directamente sobre la consola de comandos del usuario, y permite la transmisión de datos a través de URLs (descrito en profundidad en el apartado 4.4).

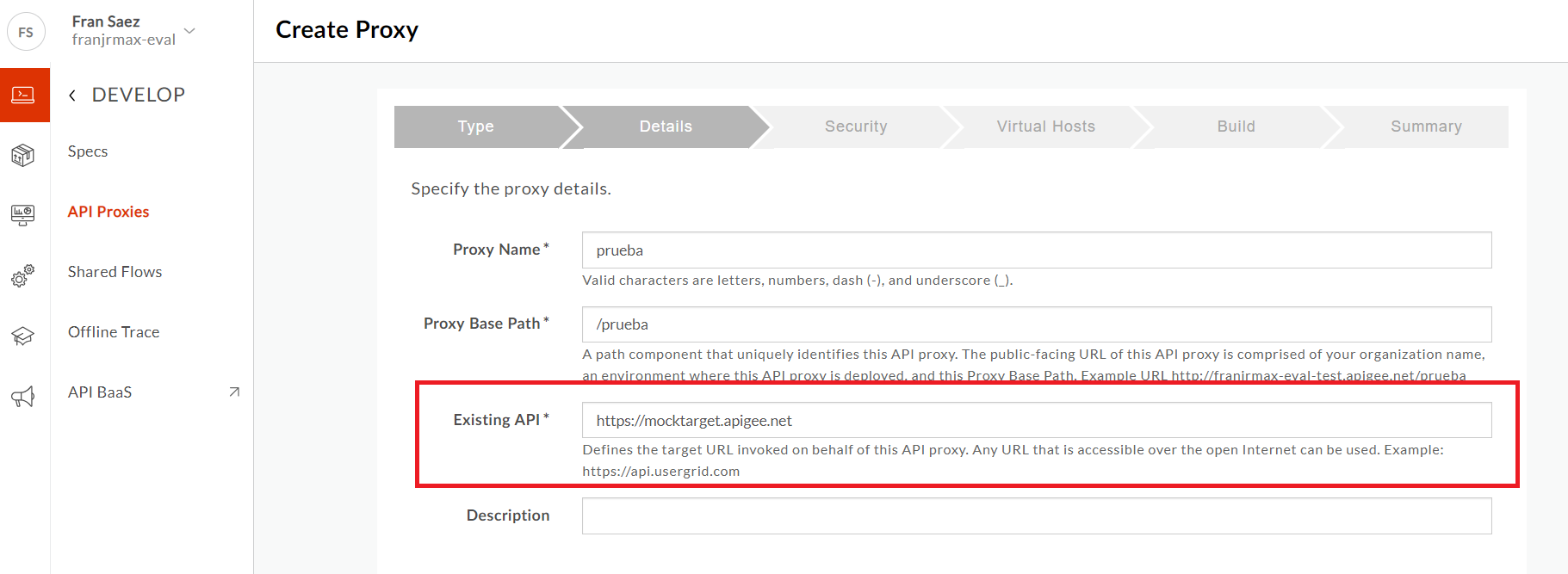
Una vez inicializado el periodo de prueba de Apigee Edge, el primer paso consiste en la **creación de un servidor proxy propio**, el cual será el encargado de recibir y procesar las peticiones de los usuarios en el momento de invocar las diferentes APIs. Para ello, desde el menú principal de Edge se accede al submenú API Proxies y, en la parte superior derecha, se selecciona la opción de crear un nuevo proxy, más concretamente de tipo reverse proxy.



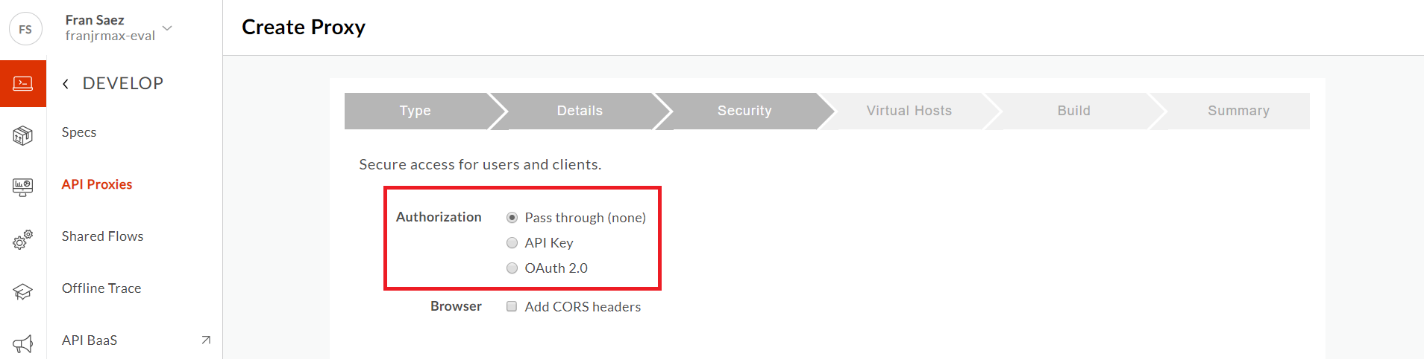




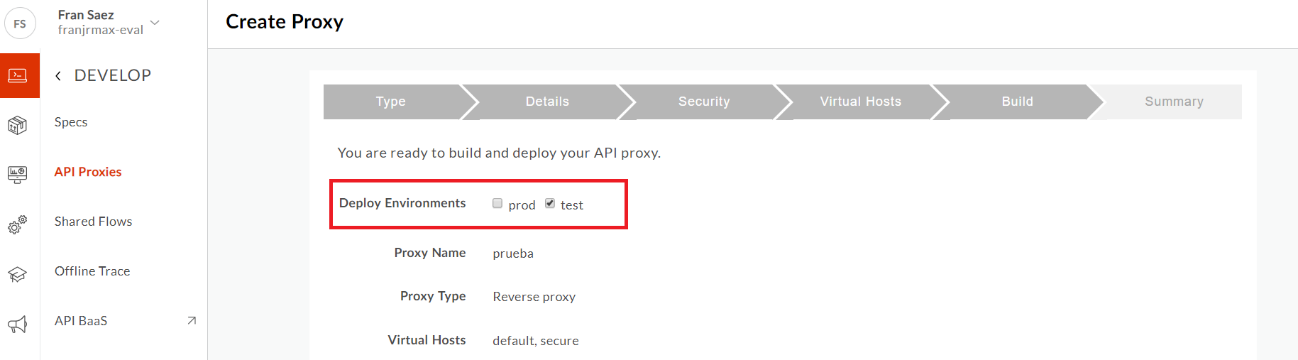
Tras ello, se asigna un nombre concreto al nuevo servidor creado y, como paso fundamental, se vincula la URL de una API existente que será el objetivo de las peticiones al servidor.



A continuación, se determinan los parámetros de seguridad que rigen al servidor. En este punto existen dos variantes interesantes de cara al prototipo definido en el presente documento, puesto que es posible optar por la opción de no definir protocolo de seguridad alguno, por lo que el servidor responderá a todas las peticiones externas de los usuarios; o bien es posible establecer un sistema de seguridad basado en una clave de API secreta, la cual debe acompañar a la petición y permitirá identificar al usuario que desee acceder a dicha aplicación. Puesto que de momento se está tratando la creación de un servidor proxy de manera genérica, se mostrará en primer lugar un servidor sin protocolo de seguridad y, más adelante en este mismo apartado, se retomará esta sección desde este punto incluyendo las acciones asociadas a la implementación de la seguridad.

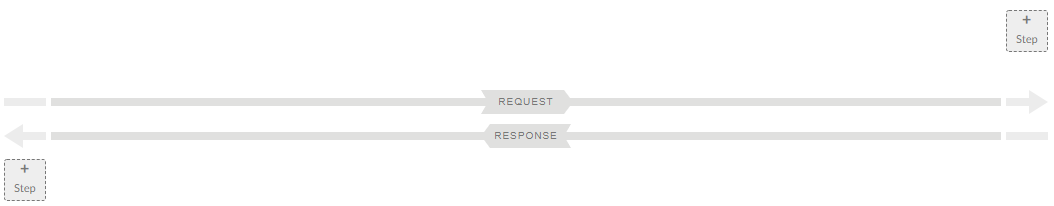


Como última acción, se selecciona el entorno sobre el que se desea construir el servidor, siendo en este caso el entorno de test que permitirá realizar pruebas con el mismo.



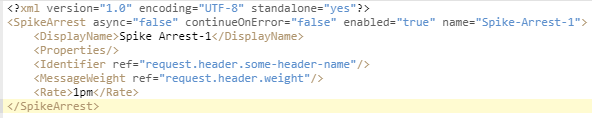
Tras ello finalizaría la implementación del servidor proxy, la cual será posteriormente testeada en el apartado 4.3 haciendo uso de la herramienta Curl.

Para crear una **política de control de acceso** es necesario ir a la pestaña Develop del editor del proxy, dentro de este menú encontraremos un submenú llamado Navigator en el que tendremos que ir al apartado Proxy Endpoints y seleccionar PreFlow. Cuando lo tengamos seleccionado será necesario añadir un paso, este paso es lo que tendrá que realizar el proxy antes de procesar nuestras peticiones.



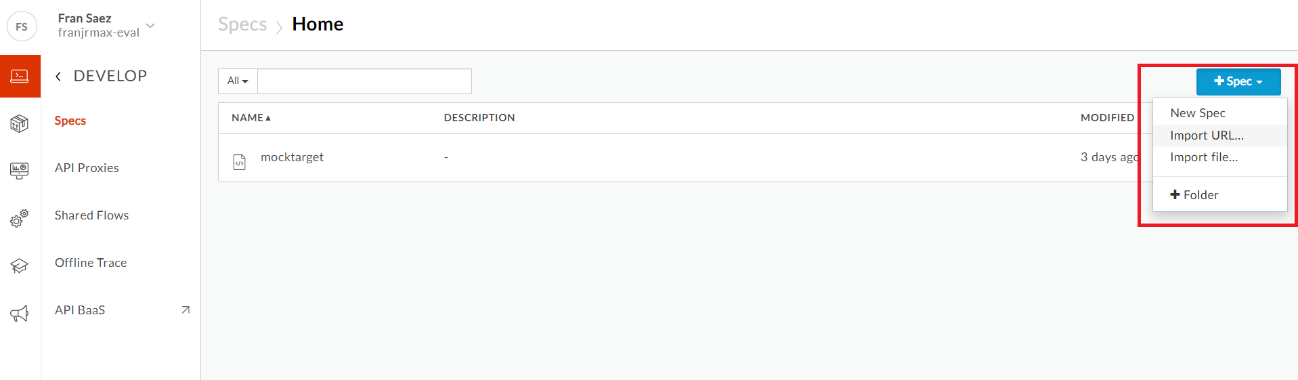
Tenemos que seleccionar el + Step situado en la parte superior derecha, estableciendo así las instrucciones que tiene que realizar el proxy antes de procesar todas las peticiones que le llegan. Una vez clicado se nos mostrará una nueva interfaz en el que tendremos que añadir un Gestor de tráfico, en este caso elegiremos el que se llama Spike Arrest y lo añadimos.

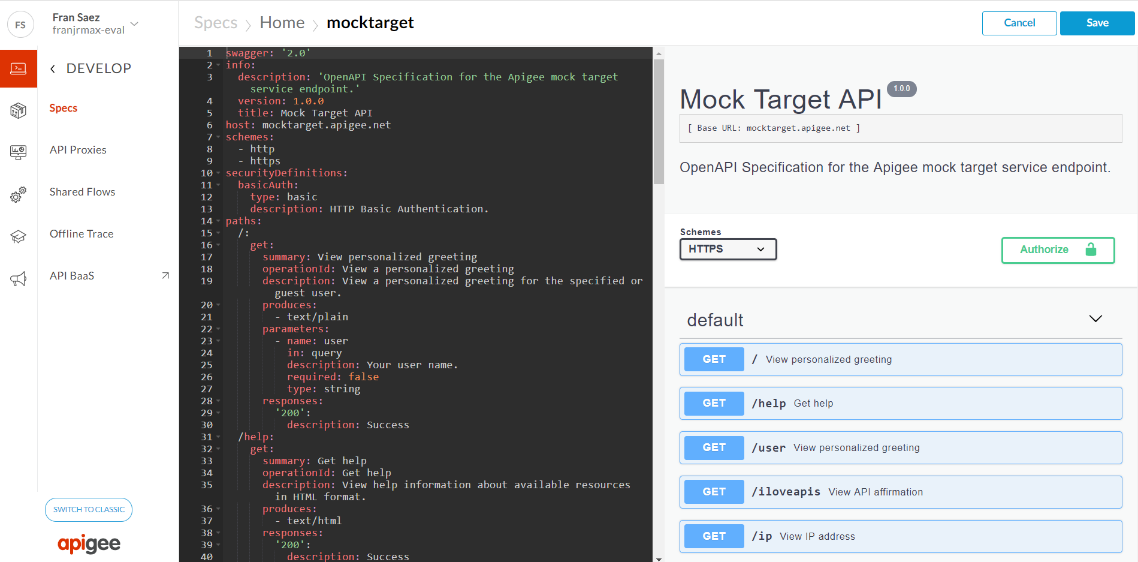
Una vez añadido el Gestor de tráfico tenemos que modificarlo para que cumpla los requisitos que nosotros queramos. Tenemos que volver al submenú Navigator y en el apartado Policies seleccionar Spike Arrest y se nos mostrará el código fuente desde donde podemos modificar cada aspecto.



Como vemos en el código fuente, modificamos el valor que hay entre <Rate> y </Rate> a 1pm, permitiendo sólo dos peticiones por minuto al servidor proxy y guardaremos nuestra configuración.

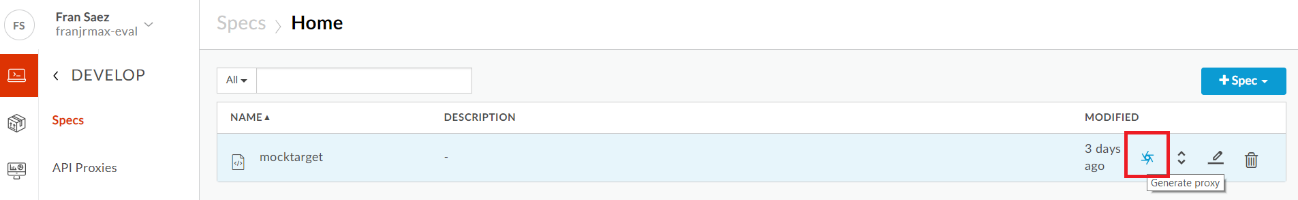
El siguiente objetivo se basará en la **publicación de una determinada API en el portal de la organización**. Para ello, en primer lugar, es necesario importar la especificación de la API deseada, para seguidamente poder construir sobre ella un servidor que procese las peticiones dirigidas hacia la misma; por lo que, desde el apartado Develop > Specs de la plataforma Edge, se importa la especificación de la aplicación a través de la URL de la misma, siendo en este caso concreto una API de prueba proporcionada por Apigee. Tras introducir el nombre y URL solicitados, la especificación aparecería de la siguiente manera:

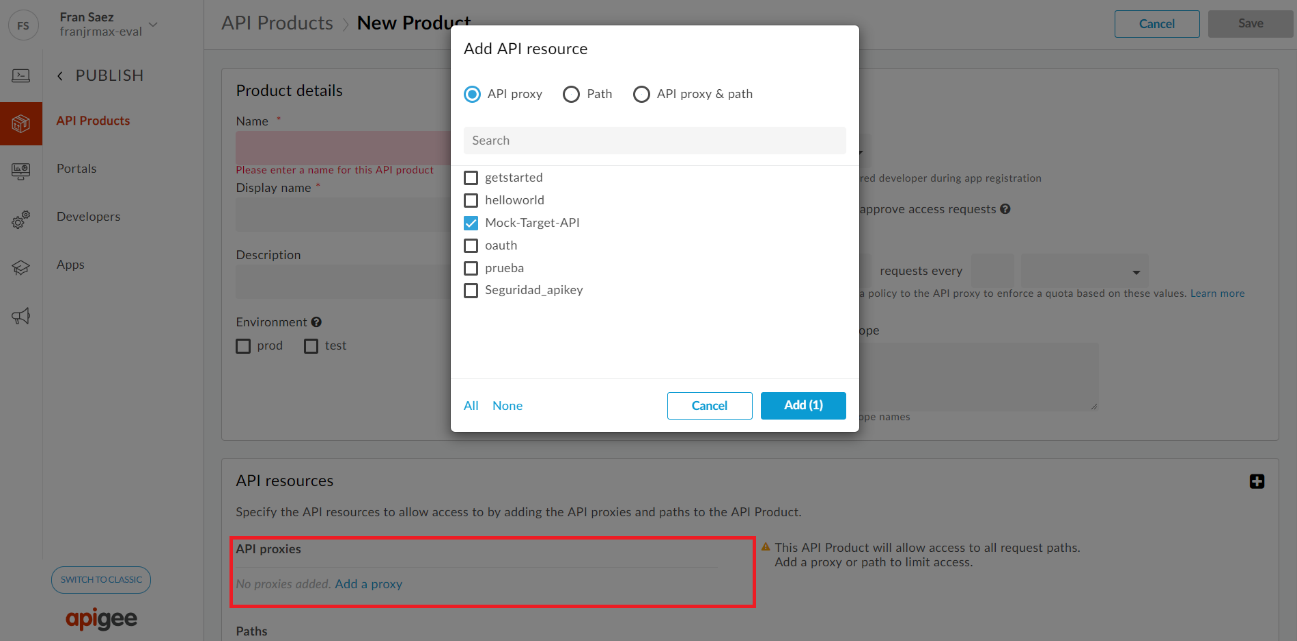




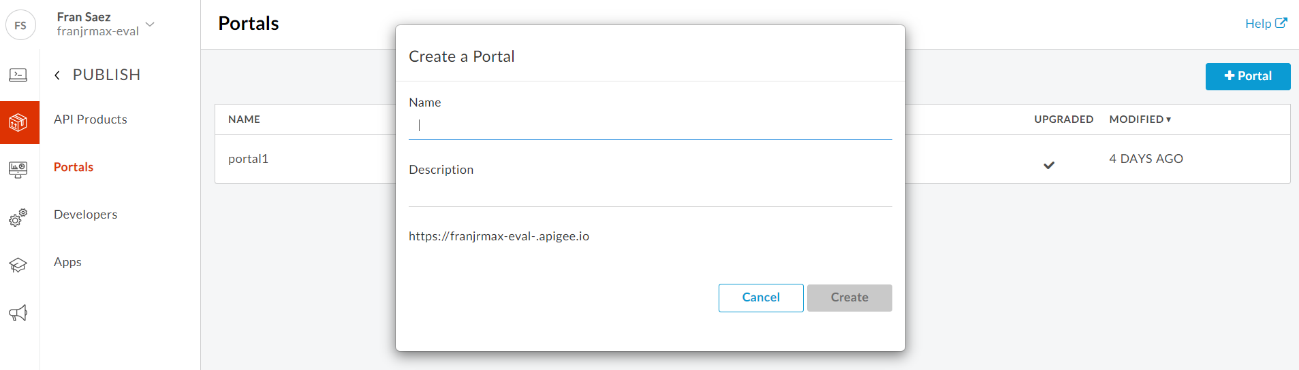
Como siguiente paso, desde el menú de opciones de la especificación recientemente importada se genera el servidor proxy sobre la misma, el cual se configura automáticamente en base al código existente, por lo que el proceso es idéntico al descrito anteriormente relativo a la creación del servidor.

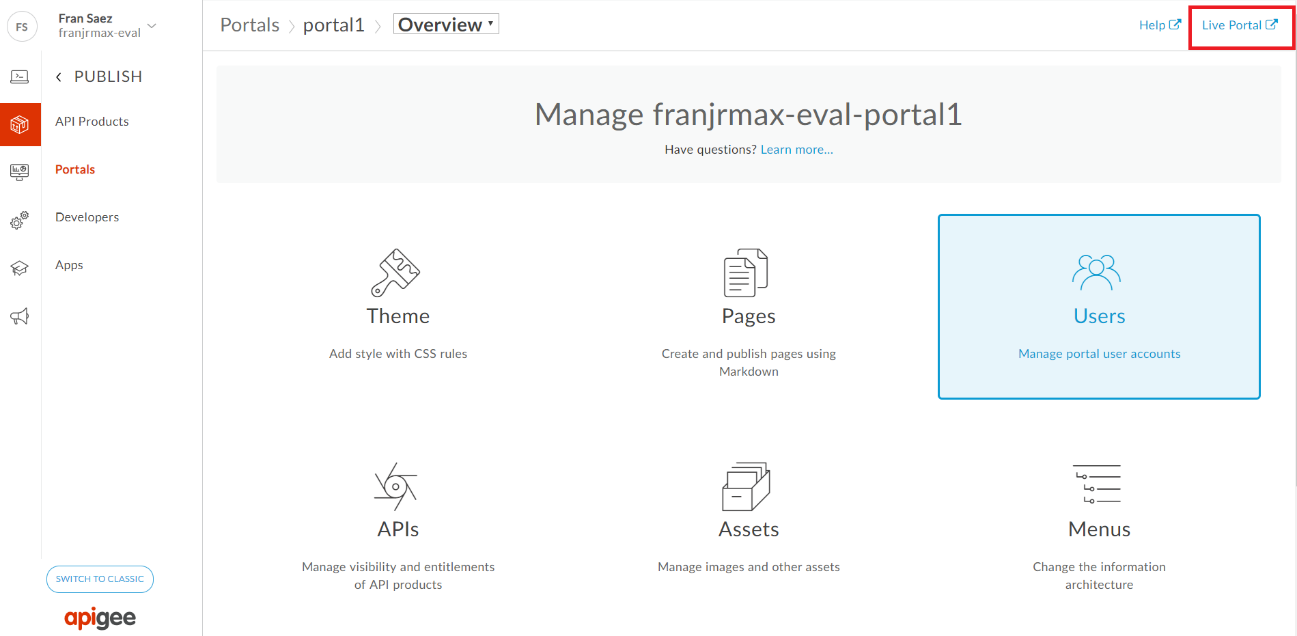
Seguidamente, cabe destacar que lo que finalmente es publicado no es la especificación de la que se dispone hasta el momento, sino que esta ha de estar integrada en un denominado producto API sobre el que se importan los recursos anteriores, acción realizada desde el apartado Publish > API Products. En este apartado, se completan los campos relativos a la definición del producto (incluyendo el acceso público al producto y Key Approval Type en modo automático) y se vincula el proxy constituido en el apartado anterior, finalizando así la importación.

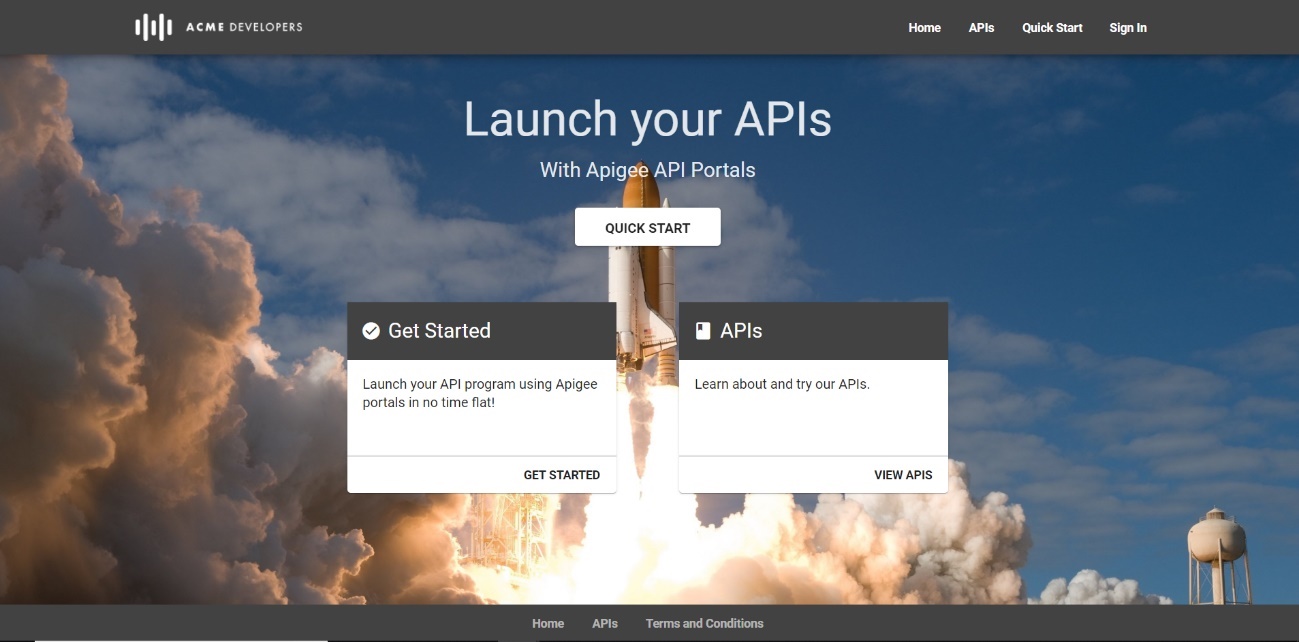




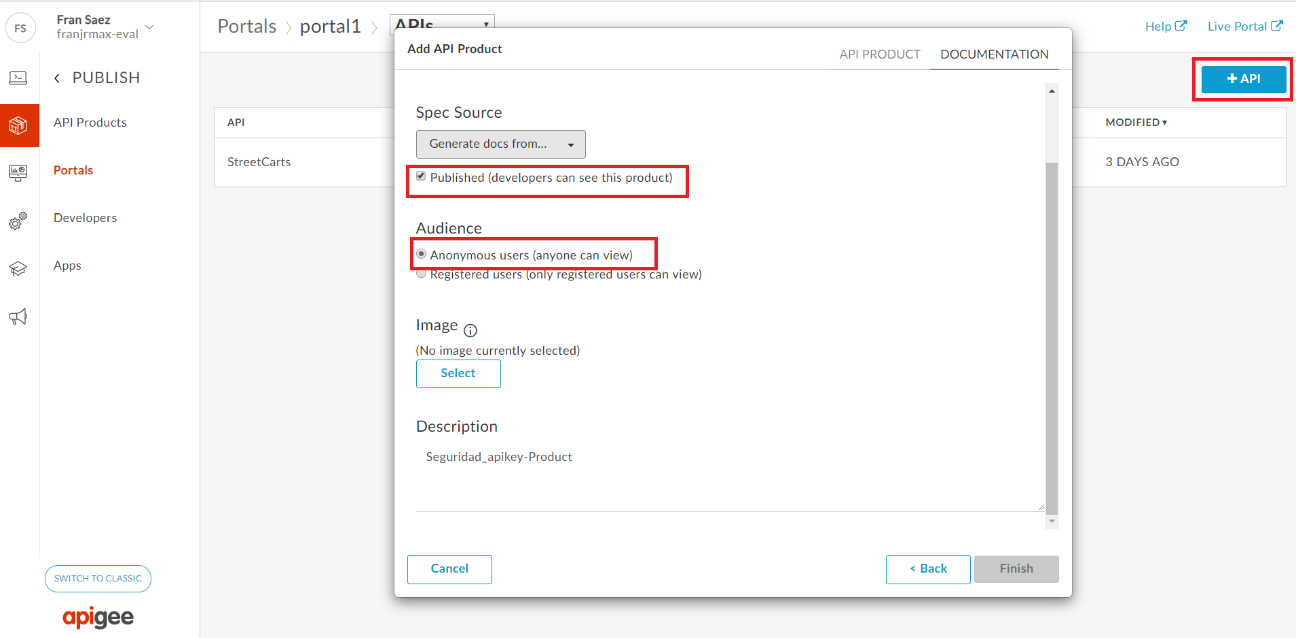
Alcanzado este punto, el objeto desarrollado ya puede ser publicado, pero es necesaria la creación de un entorno que permita dicha publicación, entorno denominado Portal en Apigee Edge. Desde el menú de publicación, dentro del subapartado Portales, se gestiona la creación del mismo, aportando un nombre y una definición y obteniendo una plantilla estándar propuesta por Apigee.



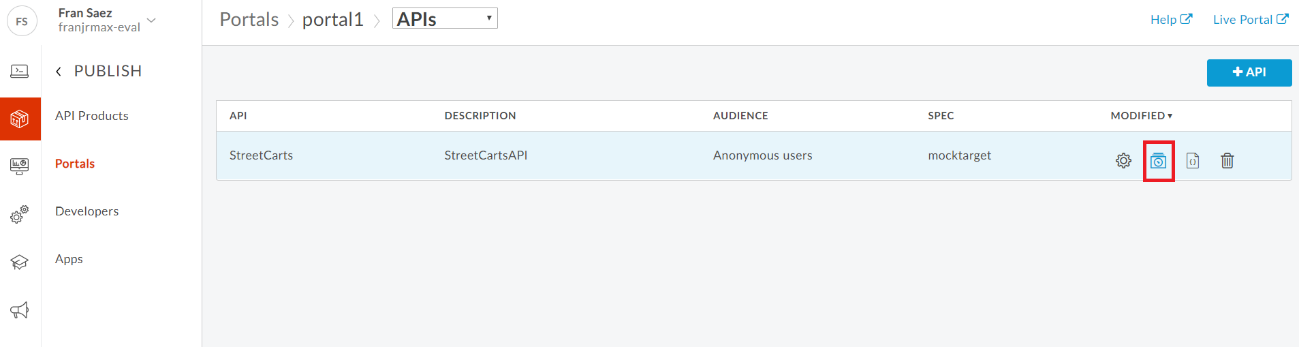


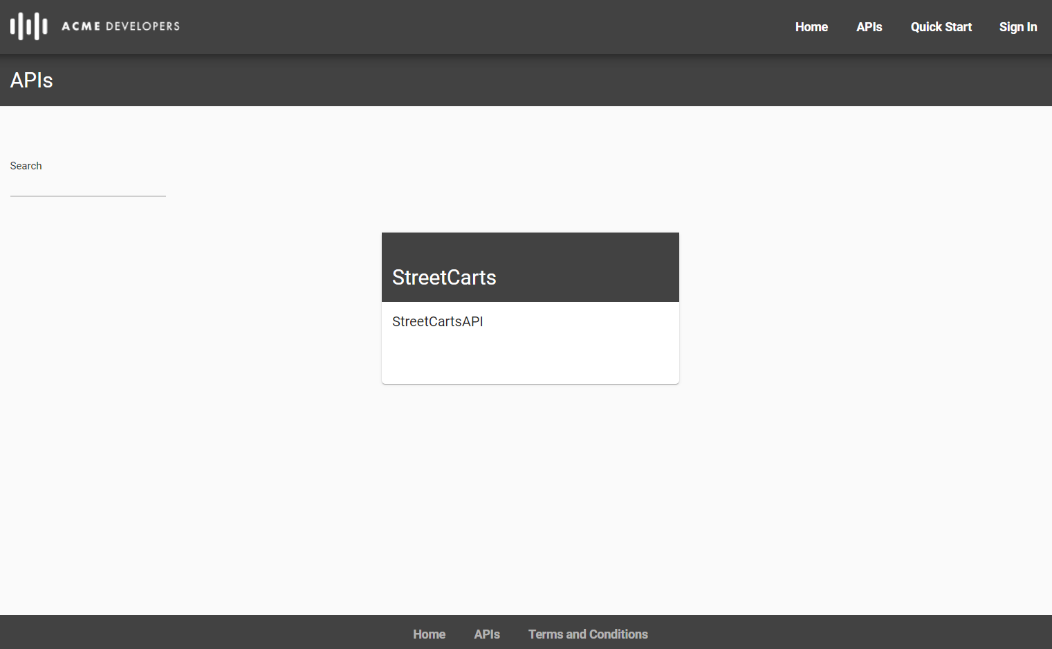


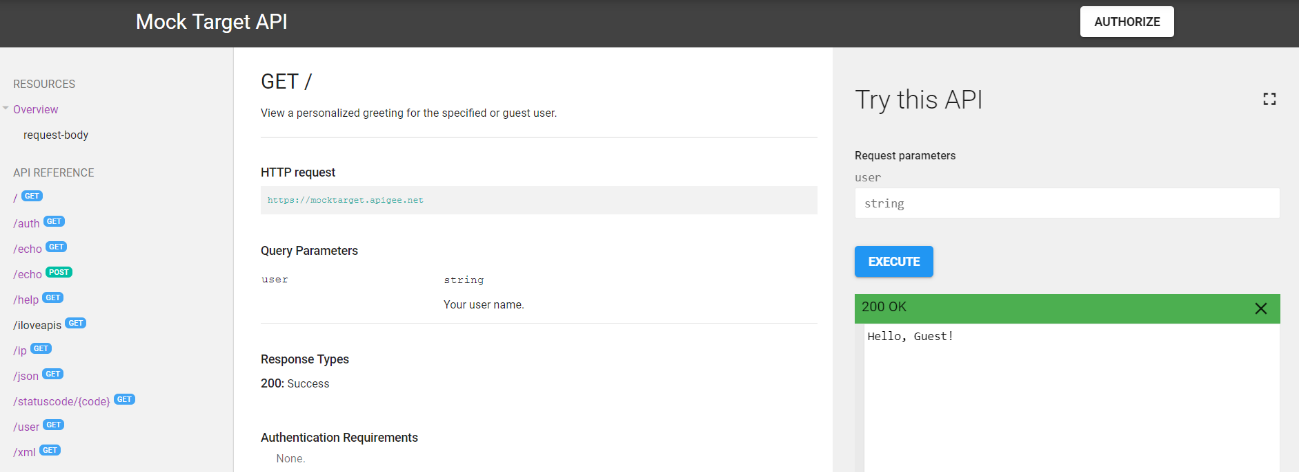
Tal y como podemos observar en la ultima captura, existe un apartado denominado APIs, el cual inicialmente aparece vacío puesto que la API previamente importada aún no ha sido publicada. Con este objetivo en mente, volviendo a acceder al menú principal del portal y al apartado APIs del mismo, se selecciona dicha API activando su publicación y haciéndola accesible para cualquier usuario.



Finalmente, para que la publicación se complete, es necesario que toda API accesible lleve asociada una documentación de referencia que refleje el estado actual de la especificación, siendo por lo tanto necesaria la creación de un snapshot de la misma desde el apartado APIs del portal recientemente creado. Completando este paso es posible acceder a la API desde el portal de la organización.



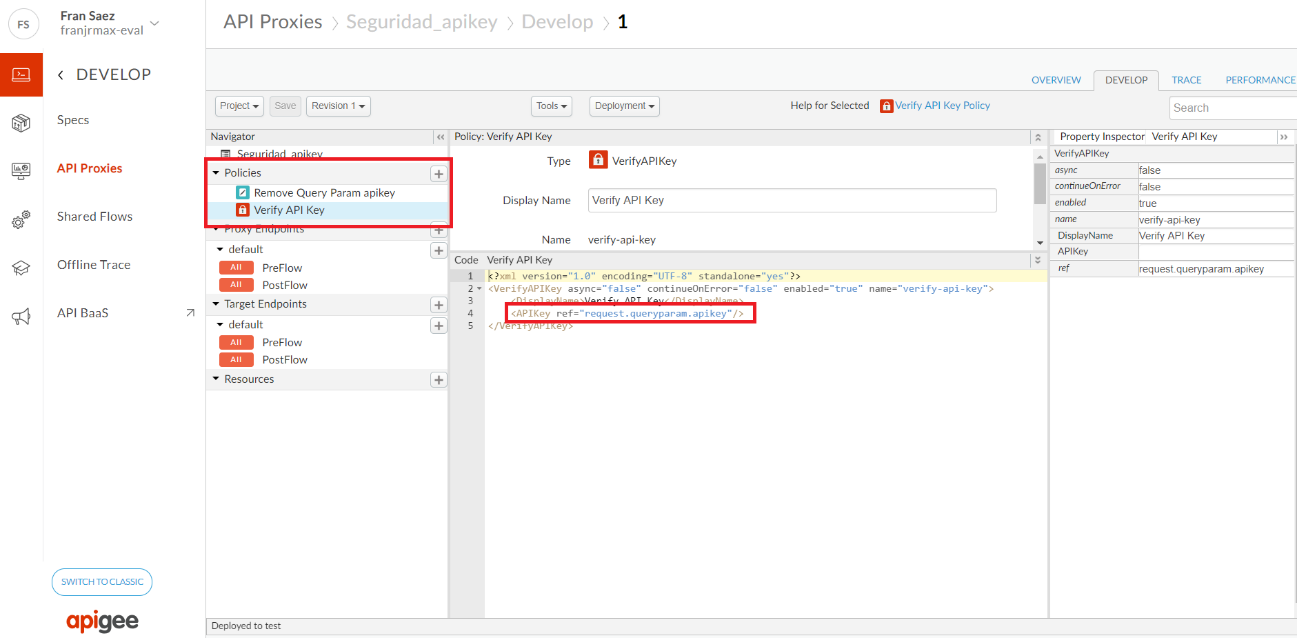




Como último aspecto a tratar referido a los requisitos funcionales definidos al principio de este documento, se mostrará la **implementación de una política de seguridad y autenticación de usuarios**, proceso que lleva directamente asociado la **suscripción** a una determinada API.

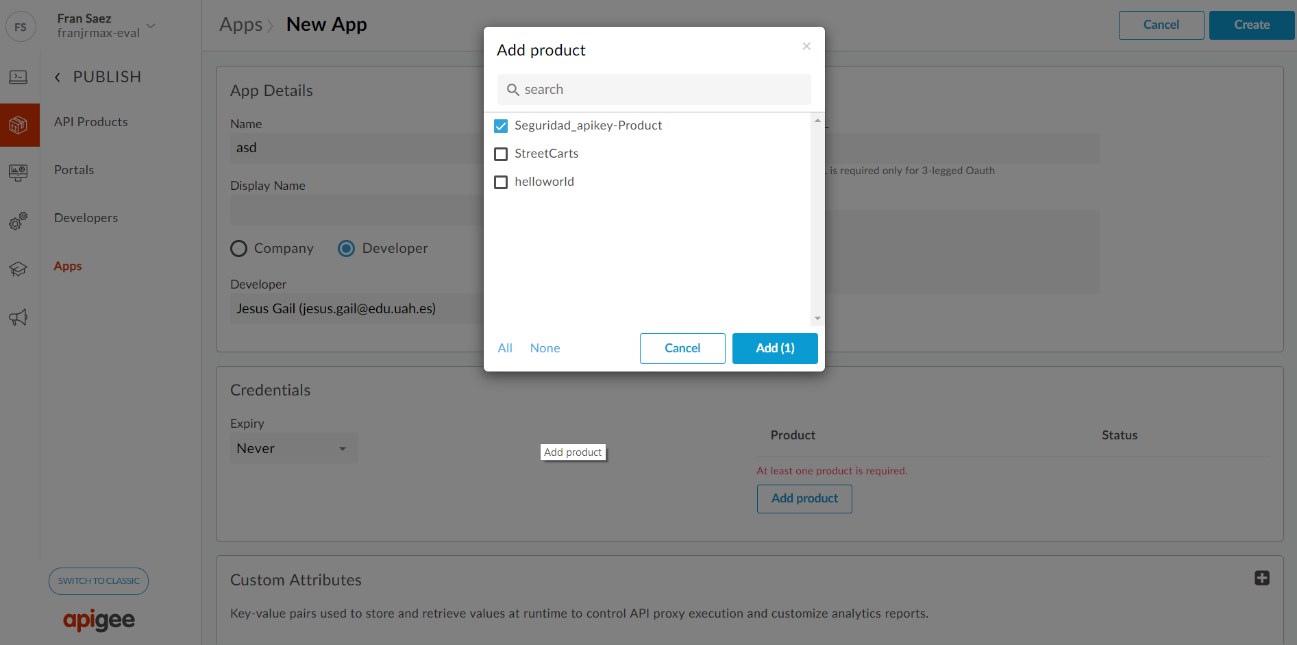
De manera semejante a los procesos descritos anteriormente, la implementación de las políticas de seguridad comienza con la construcción de un servidor proxy sobre la API que va a ser objeto de peticiones. De nuevo el proceso de creación se sustenta sobre las mismas bases, pero, en este caso concreto, es necesario incluir las opciones API Key/Publish API Product en el apartado de seguridad.

Tras ello, seleccionando el nuevo proxy y accediendo a la ventana Develop presente en el mismo, podemos observar que se han implementado dos políticas de seguridad ligadas a la autenticación de los usuarios. Por un lado, Verify API Key comprueba si la clave proporcionada para acceder al API es correcta y, por otro lado, Remove Query Param apikey oculta la clave utilizada tras su uso para evitar la filtración no deseada de la misma.

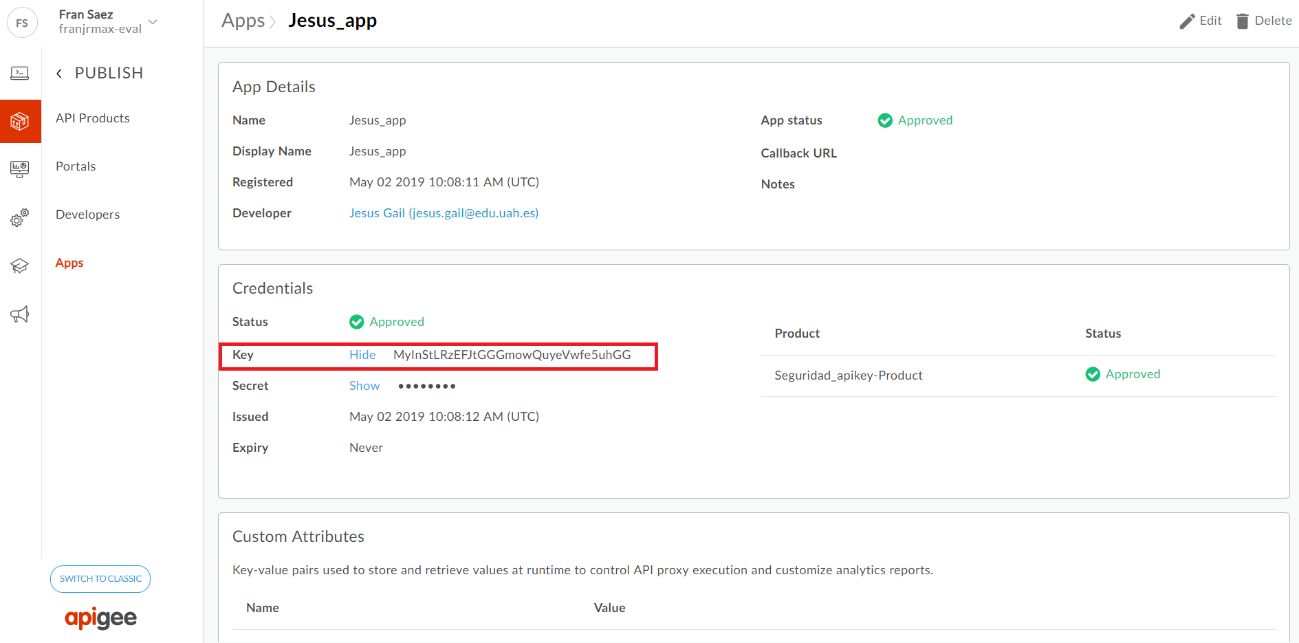


Tal y como se mostrará en el apartado 4.3, si en este momento se desea realizar una petición contra el API Proxy concreto que se acaba de constituir aparecerá un mensaje de error que informará de que no se ha detectado la clave de la API y, por lo tanto, no se puede invocar. Por consiguiente, el siguiente paso consta de la obtención de la mencionada clave secreta, siendo en este punto concreto donde influye el método de suscripción (estructurado por Apigee Edge en base a desarrolladores).

Es necesario, por tanto, comenzar con la creación de la figura del desarrollador desde el subapartado Desarrolladores del menú Publicación, donde, clicando sobre la opción +Developer e introduciendo la información básica del mismo añadiríamos a dicho sujeto a la organización. A continuación, el desarrollador previamente establecido ha de ligarse a una determinada aplicación, permitiendo así el uso y acceso a la misma por parte de este; para ello, de nuevo dentro del menú Publicación y, más concretamente, en el apartado de Apps, se crea una nueva aplicación desde la opción +App donde se selecciona el desarrollador asociado a la misma y el producto concreto al que este podrá acceder.



Para finalizar, tras la creación de la aplicación esta aparecerá en el apartado de Apps, por lo que, si se accede a la misma, aparecerán sus detalles y, además, la clave secreta de acceso ligada al desarrollador autorizado para su uso. Para probar la eficiencia de este sistema de seguridad, se ejecutará una petición hacia la URL <http://franjrmax-eval-test.apigee.net/seguridadapikey?apikey=apikey>, donde el atributo =apikey será sustituido por la clave proporcionada, cuya demostración quedará reflejada de nuevo en el apartado 4.3.

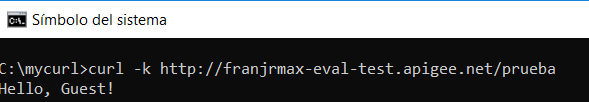


Tras la realización de todos los procesos mencionados en el presente apartado, y siguiendo los pasos descritos, se obtendría finalmente el prototipo definido en un inicio en base a los requisitos funcionales establecidos.

## 4.3 Documentación de pruebas

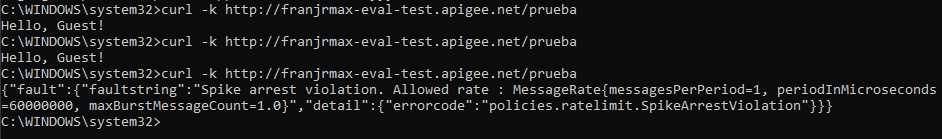
Durante el desarrollo del presente apartado, se mostrarán diferentes pruebas realizadas para comprobar el correcto funcionamiento de las funcionalidades y objetos desarrollados en el punto 4.2 de este mismo documento:

* Envío de petición al servidor proxy inicial: haciendo uso del comando curl desde la consola de comandos, se realiza una petición al servidor proxy creado (copiando la URL del mismo, la cual es accesible a través de la pestaña API Proxies en Apigee Edge), recibiendo la respuesta correspondiente

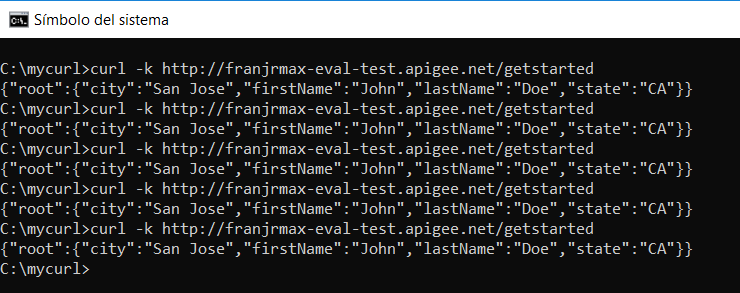


Tal y como se aprecia en la imagen superior, se añade el atributo -k al comando curl. Esto es debido a que, por defecto, el manejo de servidores proxy a través de Curl mantiene un protocolo de autenticación SSL y, si no se dispone el contrato asociado, en ningún momento puede realizarse la conexión; por lo tanto, el comando curl -k permite la conexión al proxy deshabilitando en primer lugar la autenticación mediante SSL.

* Control de acceso: una vez establecida la política de control de acceso dentro del servidor proxy en cuestión, tenemos que comprobar si lo hemos realizado correctamente, para ello abrimos el cmd como administrador y ejecutamos peticiones al servidor proxy, y comprobamos que una vez que realizamos más de dos consultas en un minuto nos muestra un mensaje de error (SpikeArrestViolation, puesto que responde a la política Spike Arrest).

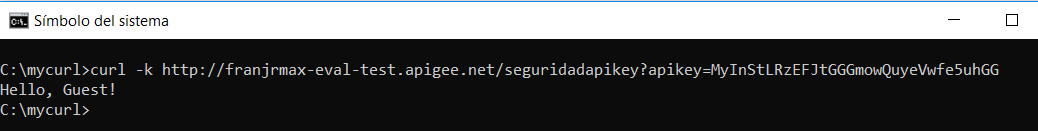


De igual manera podría realizarse la prueba contraria, es decir, enviar peticiones reiteradamente en un corto periodo de tiempo contra un servidor que no posea la política de control de acceso, comprobando que en este caso no se aprecia mensaje de error alguno.

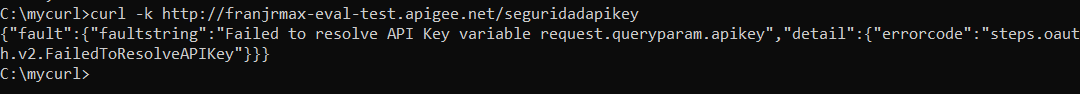


Como añadido, en la presente imagen puede apreciarse el correcto funcionamiento de la política XML to JSON, que modifica el formato en el que los datos se presentan al usuario.

* Autenticación de usuarios: Finalmente, la última prueba destinada a comprobar el correcto funcionamiento del prototipo se centra en realizar una petición al servidor proxy al que se le ha implementado la política de autenticación de usuarios. En base a ello, y tal y como se ha mencionado en el apartado 4.2, realizaremos la petición utilizando el mismo formato de comando empleado hasta el momento (curl -k), pero, como añadido, se incluirá la clave asociada al desarrollado autorizado que aparece en la figura de dicho sujeto en el apartado Apps de la plataforma Edge.



De igual manera que en la prueba anterior, se comprobará el caso contrario, es decir, el envío de la anterior petición sin incluir la clave asociada al desarrollador, obteniendo el siguiente resultado:



Como conclusión a este apartado, cabe destacar que el principal problema surgió con el protocolo de seguridad SSL al realizar las peticiones a los diferentes proxies, puesto que se desconocía tanto el funcionamiento del mencionado protocolo, así como la inhabilitación del mismo. Una vez encontrada la solución a dicho inconveniente, el resto de la implementación del prototipo transcurrió sin fallos destacables, debido principalmente al estudio y comprensión de las funcionalidades y objetos desarrollados desde la plataforma Edge por parte de los miembros del grupo y a la claridad y cantidad de información proporcionada por Apigee a este respecto, lo que facilitó con creces la construcción.

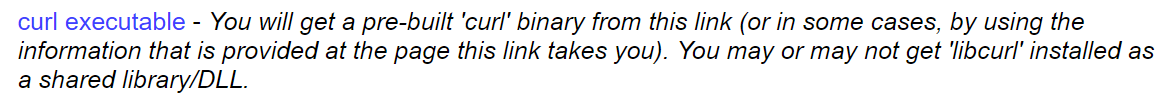
## 4.4 Documentación de instalación

Tal y como se indicó al inicio del apartado 4.2, el grado de detalle en la descripción de los procesos de construcción del prototipo, permite que el mismo pueda ser instalado fácilmente por un usuario ajeno al proyecto siguiendo los pasos que en dicho punto se marcan, por lo que se podría considerar que ambos apartados 4.2 y 4.4 se encuentran desarrollados conjuntamente en el primero de los mencionados apartados.

Es por ello por lo que, en este apartado, a modo de expansión y con el fin de tratar todos los detalles ligados a la construcción, se describirá el proceso de instalación de la herramienta básica de testeo de los servidores: Curl.

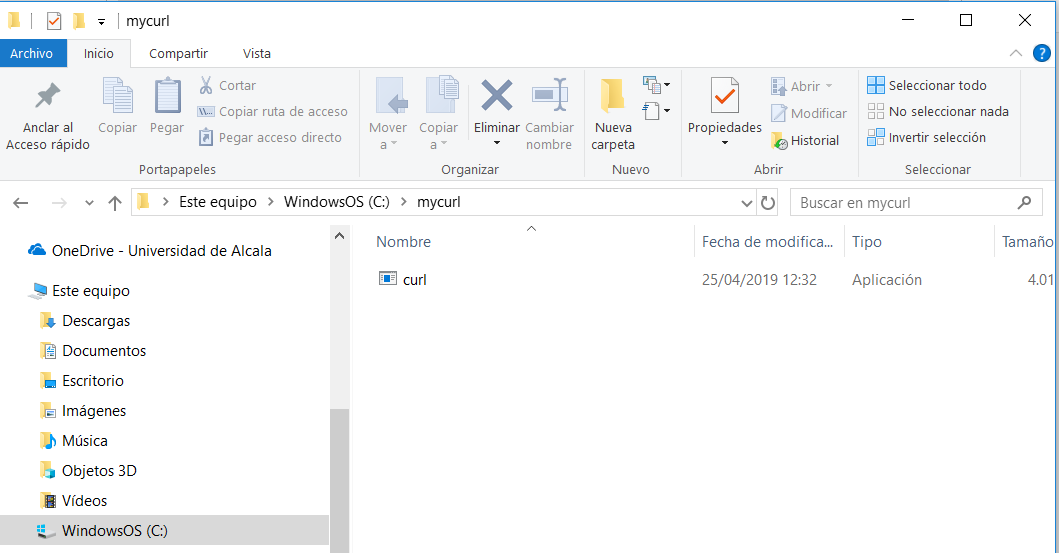
En primer lugar, es necesario acceder al enlace <https://curl.haxx.se/download.html> y, dentro del mismo, seleccionar el apartado download wizard, que nos permitirá realizar una descarga personalizada del ejecutable en base a las condiciones de nuestro sistema operativo.



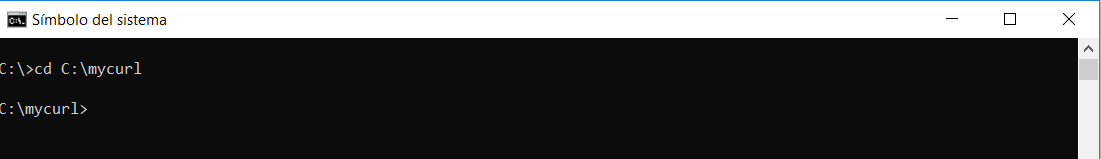


Una vez seleccionado el producto que mejor se adapte al sistema sobre el que se va a trabajar, se descarga un archivo comprimido, dentro del cual aparece un archivo ejecutable denominado curl.

En este punto, dentro del disco C: se crea una nueva carpeta llamada mycurl, donde se copia el ejecutable mencionado con anterioridad.



Una vez realizado este paso, la herramienta y los comandos asociados a Curl ya pueden ser utilizados para la transmisión de datos a través de URLs, siempre y cuando el directorio sobre el que se ejecutan los comandos sea el correspondiente a la ruta en la que se encuentra el ejecutable copiado previamente.



## 4.5 Manual de usuario

Para la **creación y publicación de una API** tenemos que ir al espacio personal de nuestra cuenta, ir al apartado Develop y en el subapartado Specs clicar en crear una nueva especificación, en este apartado crearemos el código fuente de nuestras API. Posteriormente tenemos que ir al apartado Publish, subapartado API Products y en él encontraremos las API creadas anteriormente, donde podremos añadir a nuestra API un servidor que hayamos configurado.

La **suscripción a una API** se realiza desde el menú Publish, subapartado Developers, en este apartado se nos mostrará datos de la API y del creado de esta como el nombre, email, nombre de usuario, apps y desde cuándo es miembro de la plataforma el creador de la API.

Para introducir **políticas de seguridad** o el **control de acceso** es necesario ir al menú Develop, subapartado API Proxies y seleccionamos un proxy creado anteriormente. Una vez seleccionado el proxy tenemos que ir al apartado Develop situado en la parte superior derecha y se nos mostrará un menú donde podremos introducir políticas de seguridad o cualquier variante a este aspecto.

Para ir a la **API Store** tenemos que ir al menú Publish e ir al submenú Portals, aquí podemos modificar la API Store a nuestro gusto como el diseño, las APIs que se nos van a mostrar pero para ir directamente a la API Store tenemos que clicar en el enlace que aparece en la parte superior derecha llamado Live Portal. En Live Portal encontraremos todas nuestras APIs.

Para ir al apartado donde están nuestros **servidores Proxy** es necesario ir al submenú API Proxies, que pertenece al menú Develop. En este apartado encontraremos todos los Proxies creados y también una sección donde podemos crear un nuevo servidor Proxy en el que elegiremos que tipo de servidor queremos y el nombre que le vamos a dar.

Para conocer donde están situadas las **Herramientas de Análisis** debemos dirigirnos la parte de Analyze, dentro de Analyze encontramos diversas secciones donde podremos analizar el rendimiento de la memoria, un análisis del código donde se nos muestra los errores que tenga, un apartado llamado Devices (donde se nos mostrará las plataformas más utilizadas, los tipos de Sistemas operativos más comunes…), etc.

Para encontrar el **Portal de desarrolladores** es necesario ir al apartado Community, situado en la página principal del espacio personal del administrador, en dicho apartado encontraremos todas las preguntas y respuestas de diferentes usuarios que puedan solucionar nuestros problemas.

Para llegar a utilizar el **Soporte de la compañía** deberemos ir al menú Learn y clicar el submenú Support. En este apartado debemos iniciar sesión y elegir el tipo de ayuda que queremos, normalmente elegimos vía chat pero también nos permiten un soporte vía correo electrónico.

La **documentación** es una parte importante para familiarizarnos con la plataforma y se puede acceder a través de la página principal del Administrador, ahí debemos clicar en el apartado Learn More y se nos abrirá una pestaña donde se nos mostrará toda la documentación necesaria para entender la tecnología.

La **personalización** de nuestro espacio dentro de la plataforma APIGEE se realiza desde nuestro portal, una vez que accedemos a nuestro portal podemos personalizarlo tanto estéticamente como funcionalmente.

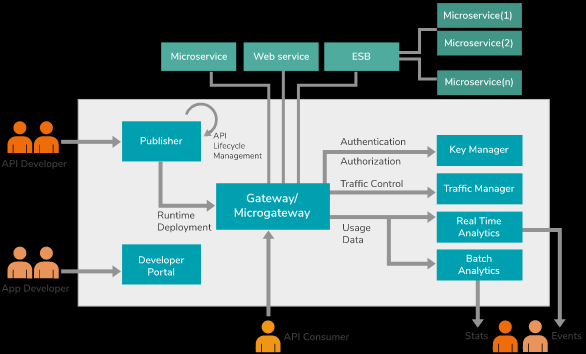
La **testeabilidad** de todo lo que creemos o modifiquemos se encuentra en un apartado dentro de cada sección, por lo que el apartado de test se encuentra en diversos menús.

# 5. Proyecto de implementación de un prototipo del sistema utilizando la tecnología WSO2

Partiendo de la base establecida por los requisitos funcionales y no funcionales definidos en el apartado anterior, y en base a las funcionalidades disponibles en las versiones de prueba que se nos han proporcionado, se plantea elaborar una simulación que recoja las características básicas de una plataforma de API Management, haciendo uso en este apartado concreto de la herramienta WSO2.

## 5.1 Documentación de diseño

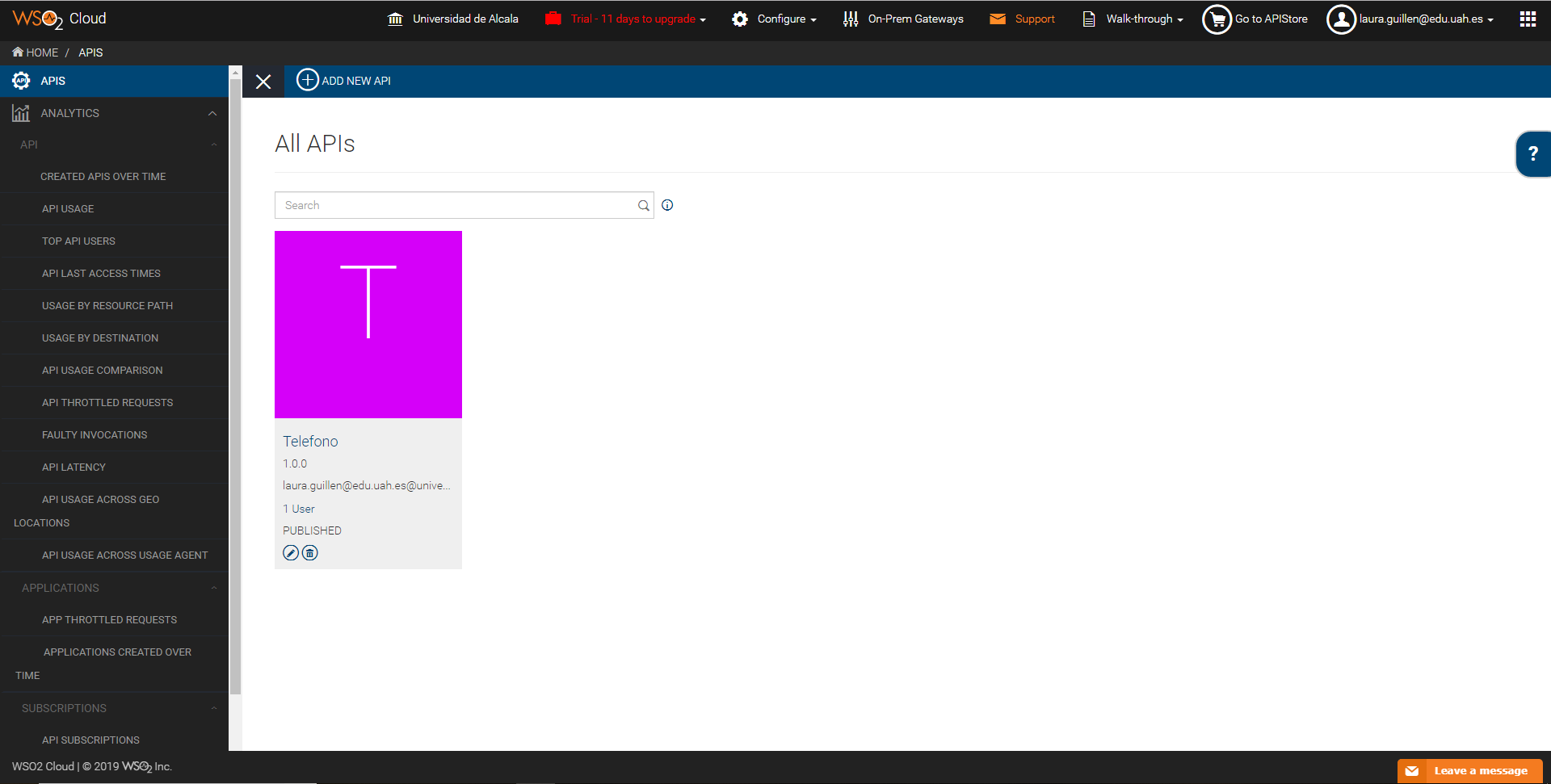
El prototipo que se va a realizar con la herramienta de API management de WSO2 consistirá en que, a partir de un fragmento de código, llevaremos a cabo la creación de una API, su gestión y su supervisión. Todo ello se realizará en la nube que nos brinda la plataforma de WSO2.



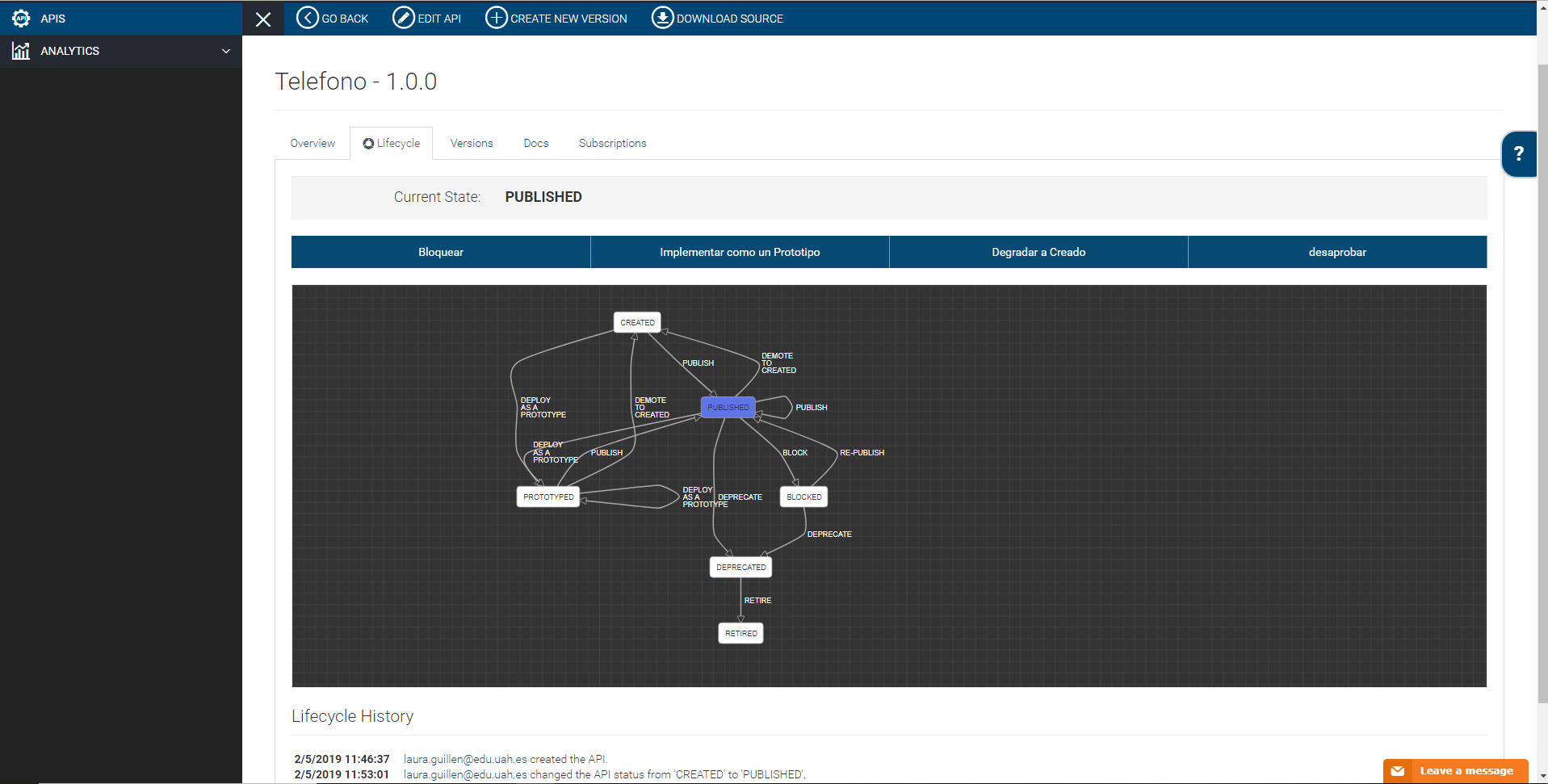
Puesto que se trata de la versión de prueba gratuita a la hora de realizar el prototipo no es posible acceder a todas las funcionalidades que nos ofrece la plataforma, como ejemplo destacable al usar la versión de prueba no es posible monetizar la API creada.

En cuanto a la interfaz de usuario que nos ofrece WSO2 se trata de una interfaz amigable para el usuario, intuitiva, pero con algunas dificultades a la hora de ser utilizada.

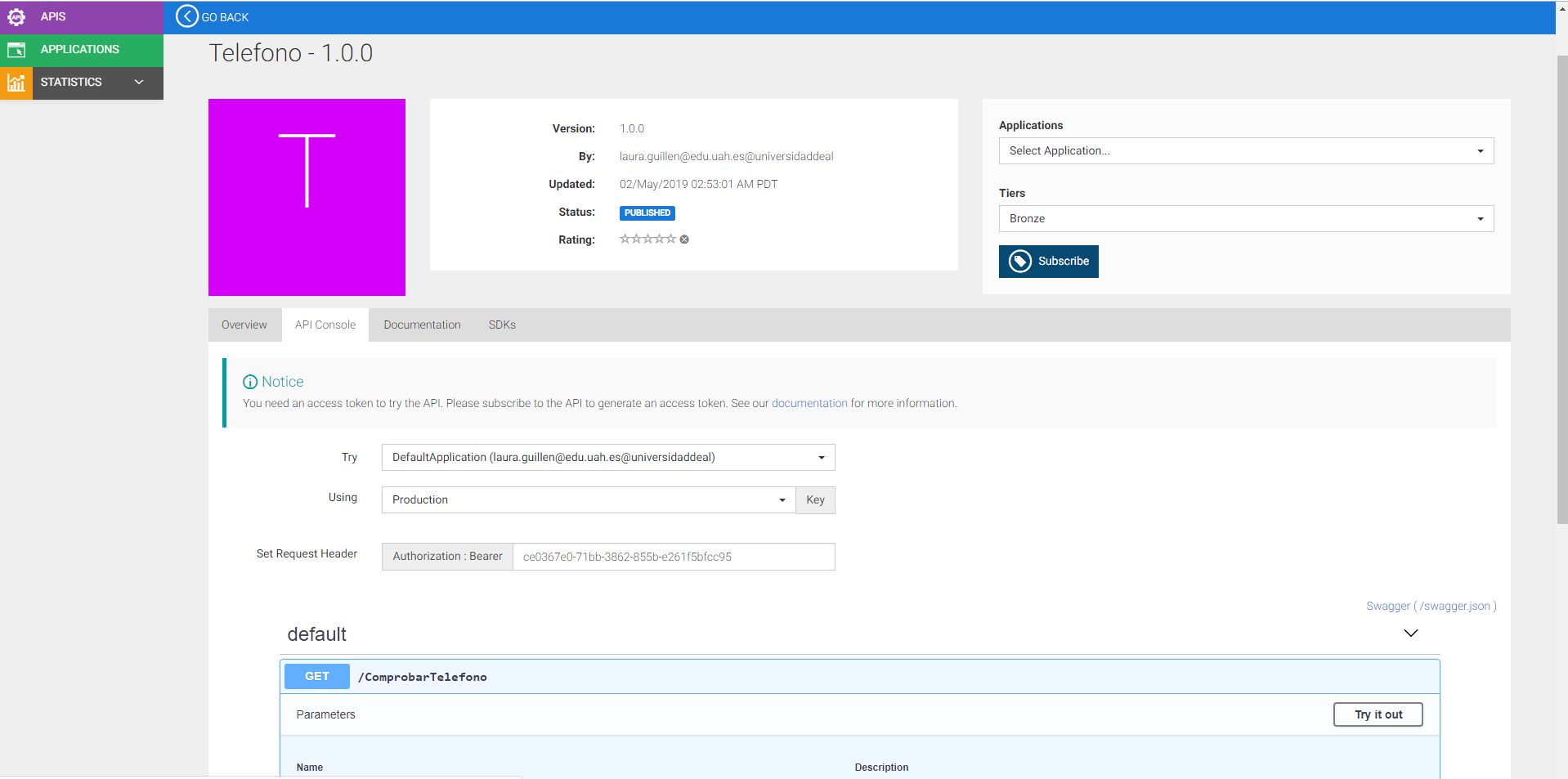
En la primera imagen podemos el diseño de la interfaz de usuario para la nube de WSO2, donde encontramos nuestras APIs creadas y las herramientas de análisis que se encuentran a disposición del creador.



Al clicar en nuestra API creada podemos encontrar la información de nuestra API, su ciclo de vida, sus versiones, la documentación que aportemos como creadores para los futuros usuarios de la API y el número de suscripciones que ha conseguido nuestra API. En la imagen se aprecia esta interfaz en la pestaña del ciclo de vida.

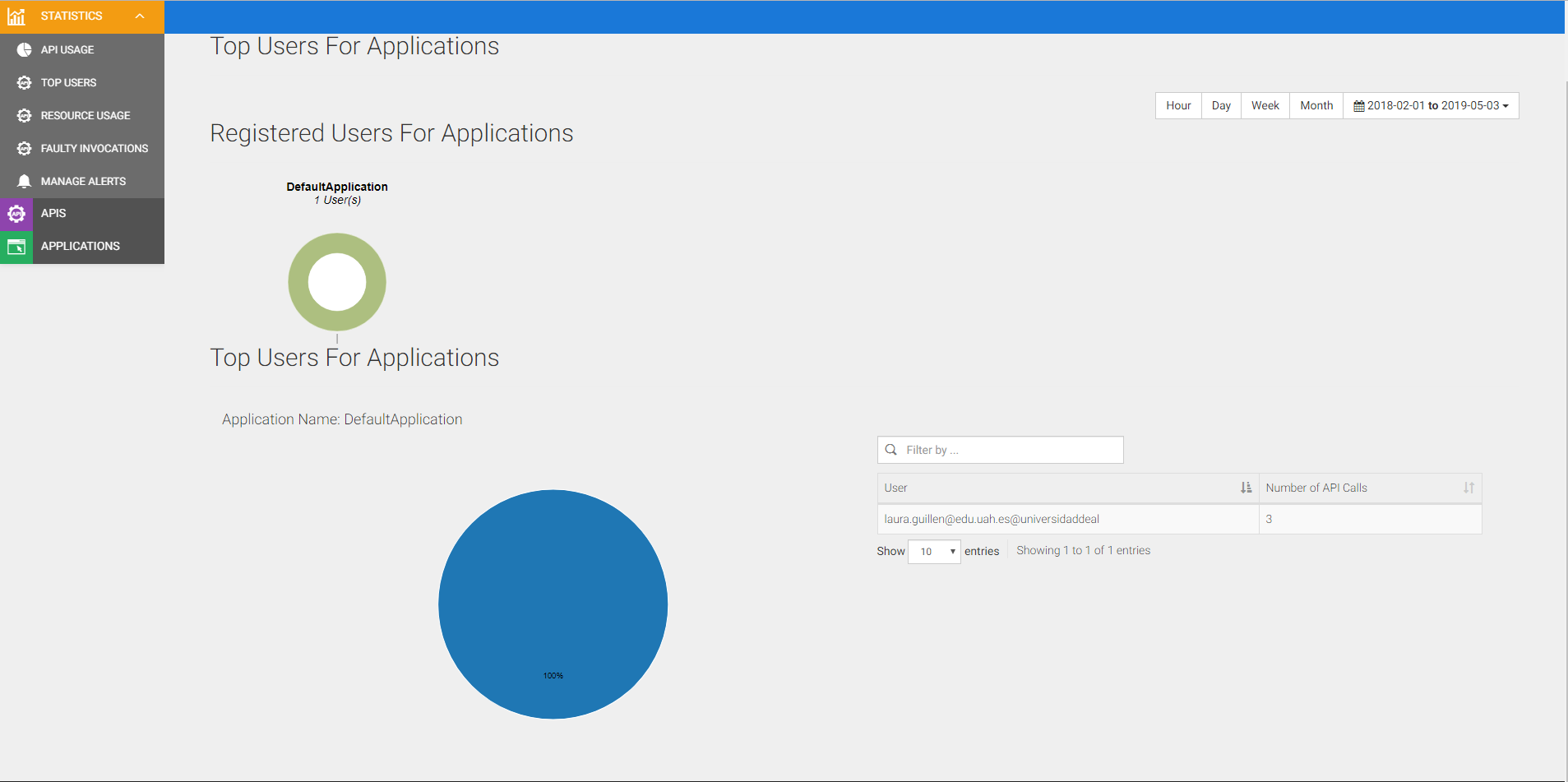


Por último, se mostrará la interfaz de usuario relativa a la API Store, lugar donde se puede llevar a cabo la suscripción a las distintas APIs así como la gestión de la API que hayas creado. Se puede editar, testear, añadir documentación y analizar dicha API a través de las herramientas de análisis que nos ofrece esta sección de la plataforma.



En la imagen nos encontramos en la sección de testeo de la API.

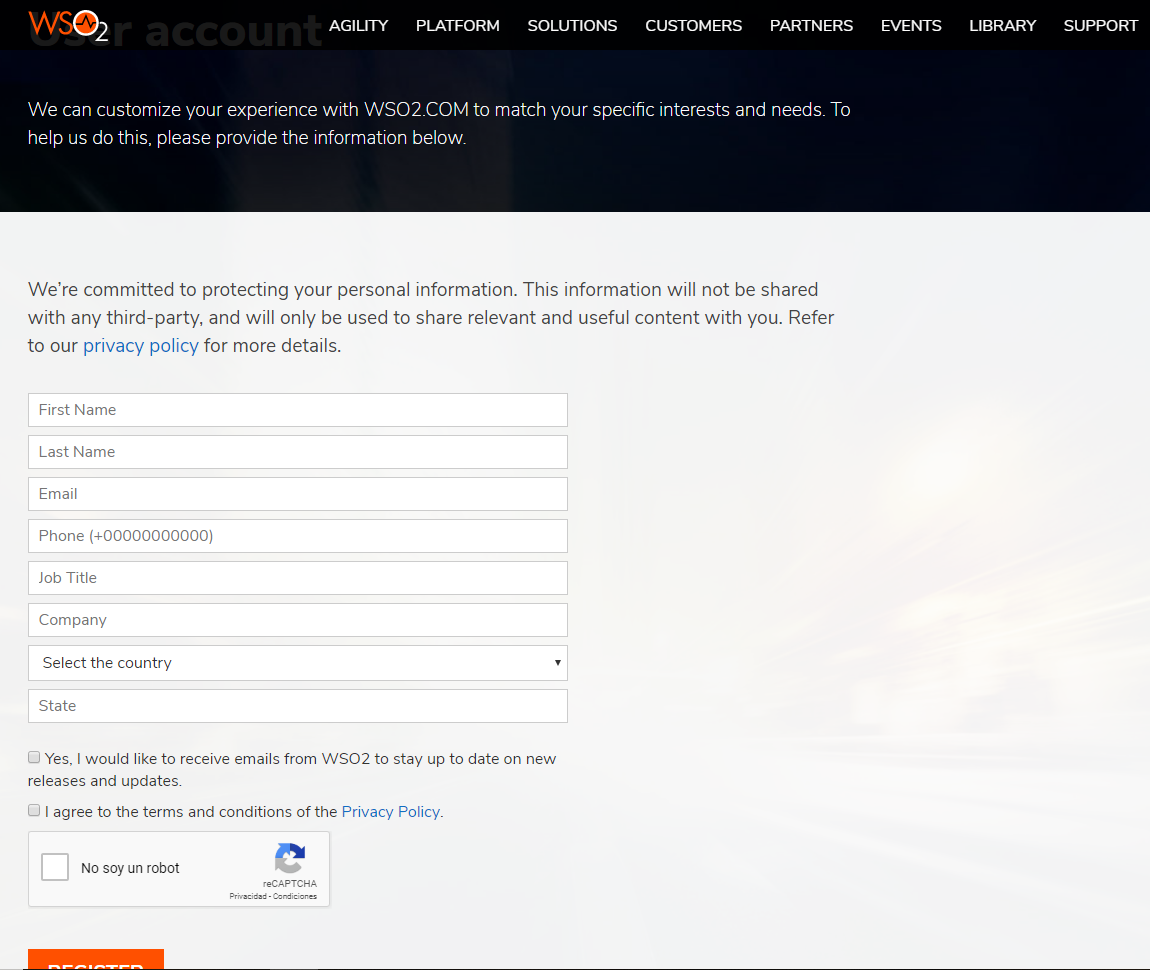
Por otro lado, la interfaz de estadísticas sería:



## 5.2 Documentación de construcción

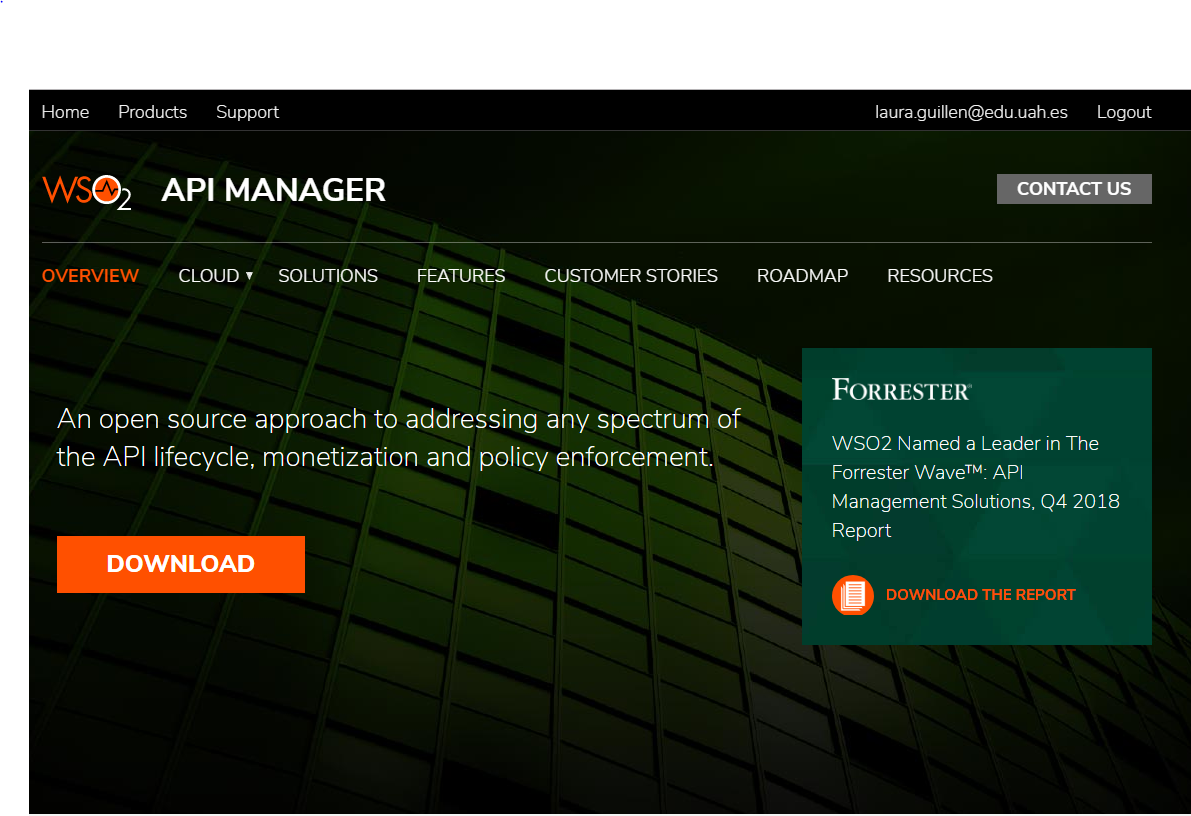
Para la construcción del prototipo con WSO2 debemos destacar que se realiza a través de la versión gratuita que nos permite la utilización de la nube que ofrece la plataforma de WSO2, esto se hace para equiparase con el prototipo que se realizará en la plataforma de APIGEE.

Para comenzar la construcción de nuestro prototipo lo primero que debemos hacer es registrarnos en WSO2 y solicitar la versión de prueba de 15 días.

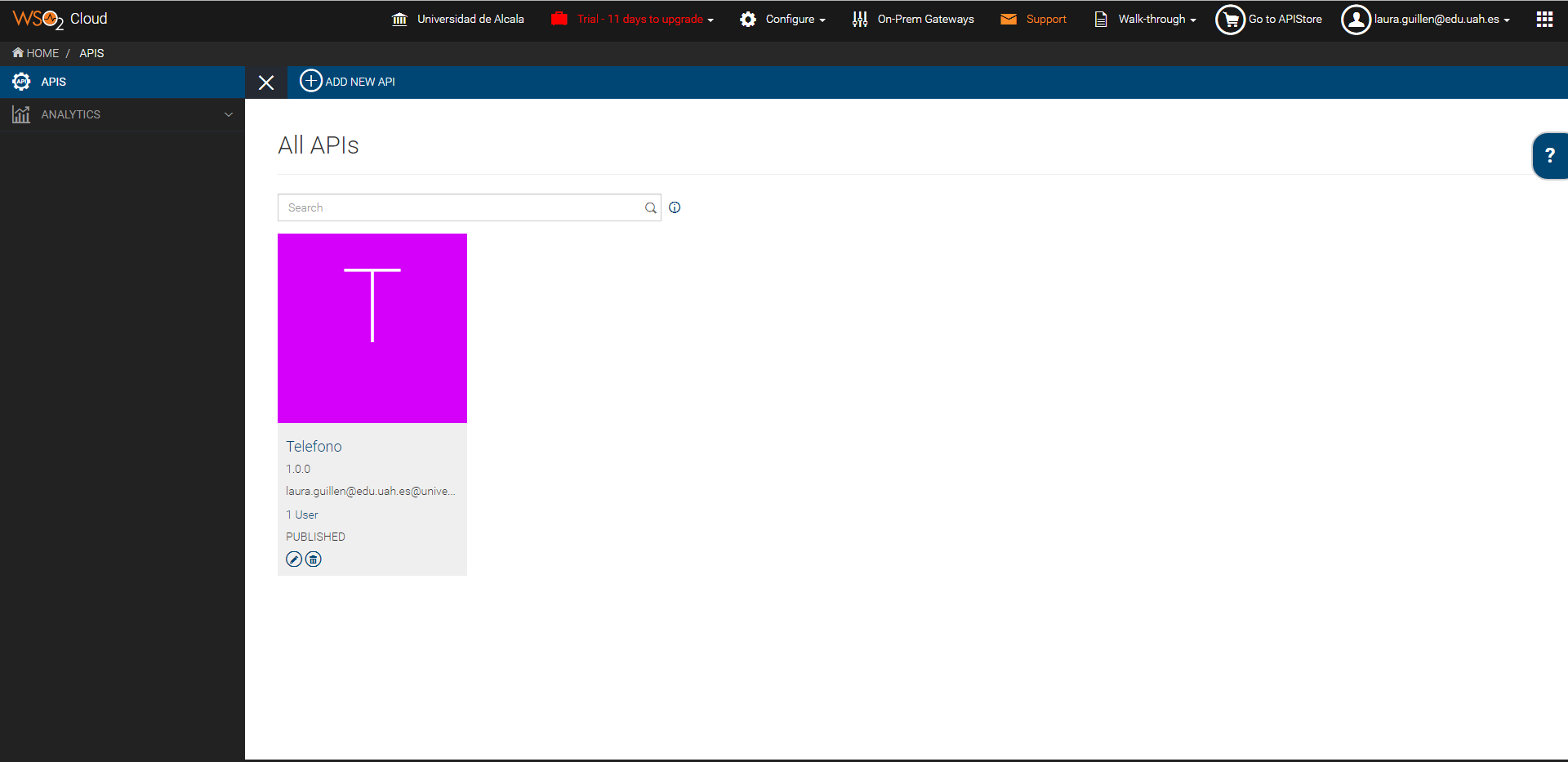


Una vez realizado este registro, WSO2 nos permitirá usar la versión de prueba por lo que deberemos iniciar sesión en la plataforma.

Tras haber iniciado sesión deberemos seleccionar el apartado de API Manager en el apartado llamado PLATAFORM.



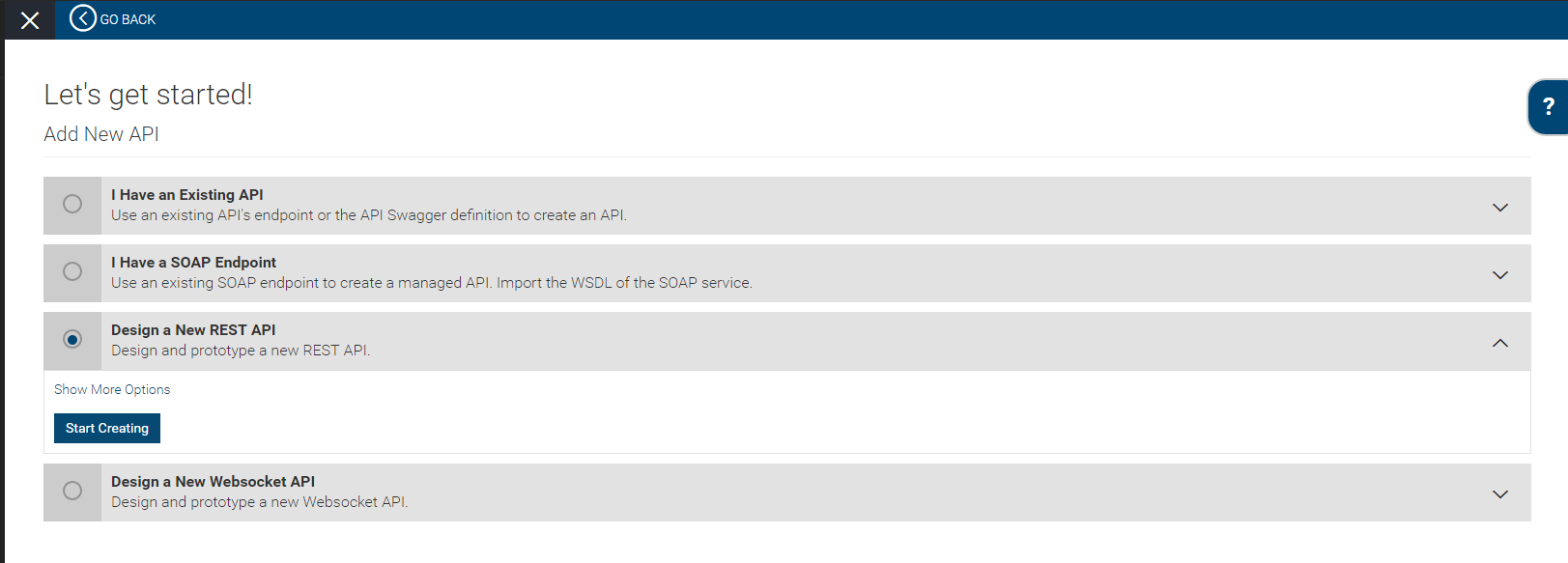
Debido a ello ya nos encontramos en la sección de API Manager que de todas las que hay en la plataforma es la que nos interesa para la construcción de nuestro prototipo. A continuación, seleccionaremos la pestaña de cloud y deberemos iniciar sesión de nuevo con la misma cuenta que se había hecho anteriormente, por lo que nos encontraremos en la siguiente pantalla:



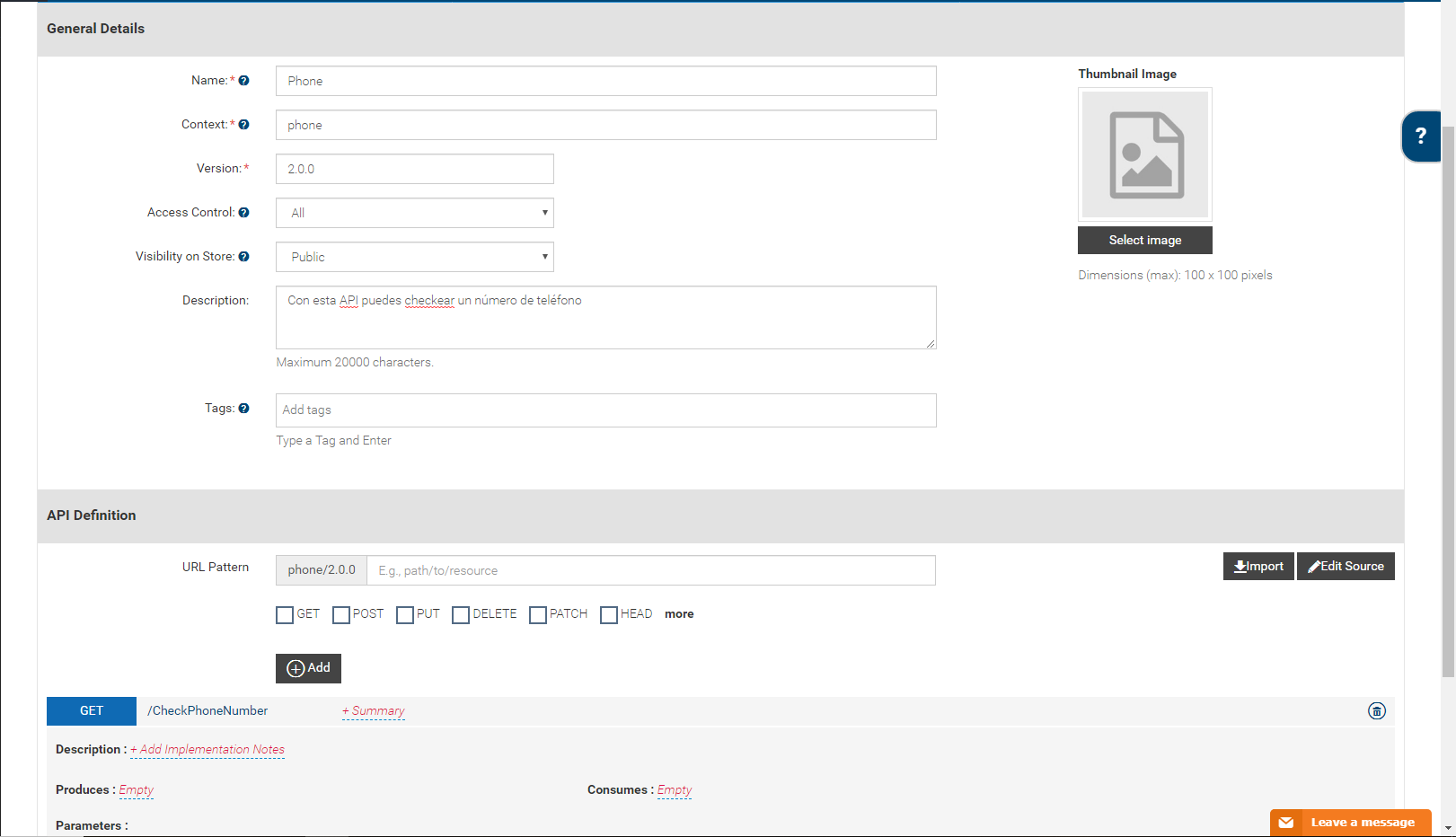
Una vez aquí, comenzaremos con la creación de una nueva API.

Empezaremos seleccionando la pestaña llamada ADD NEW API que se puede apreciaren la parte superior de la imagen.

Tras esto nos aparecerá una nueva pantalla en la que seleccionaremos Desing a NEW REST API.

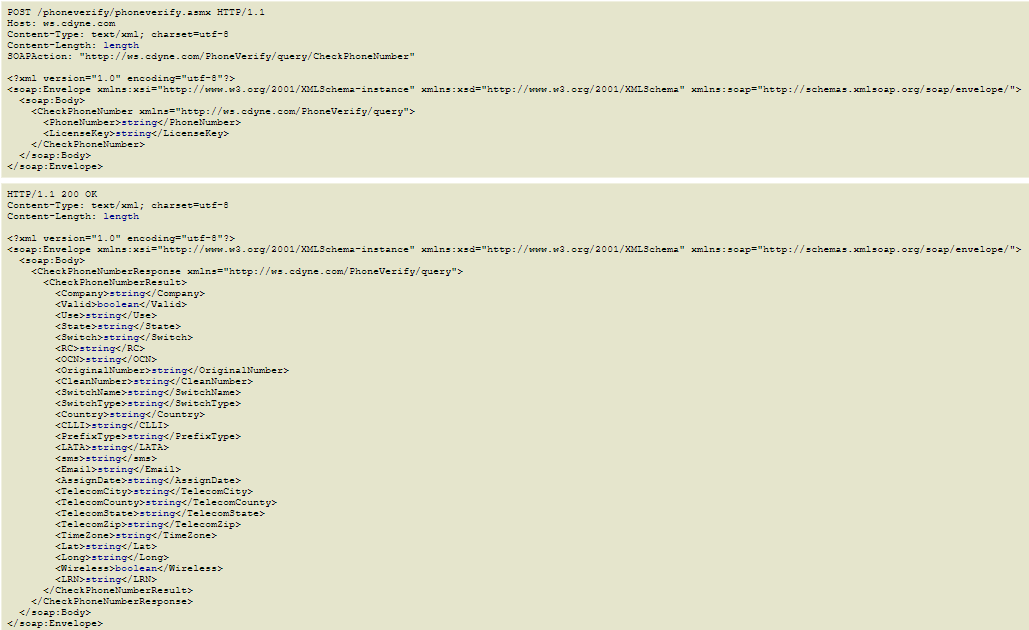


Ahora nos encontramos con una pantalla donde debemos rellenar una serie de campos, los detalles de nuestra API y su definición.



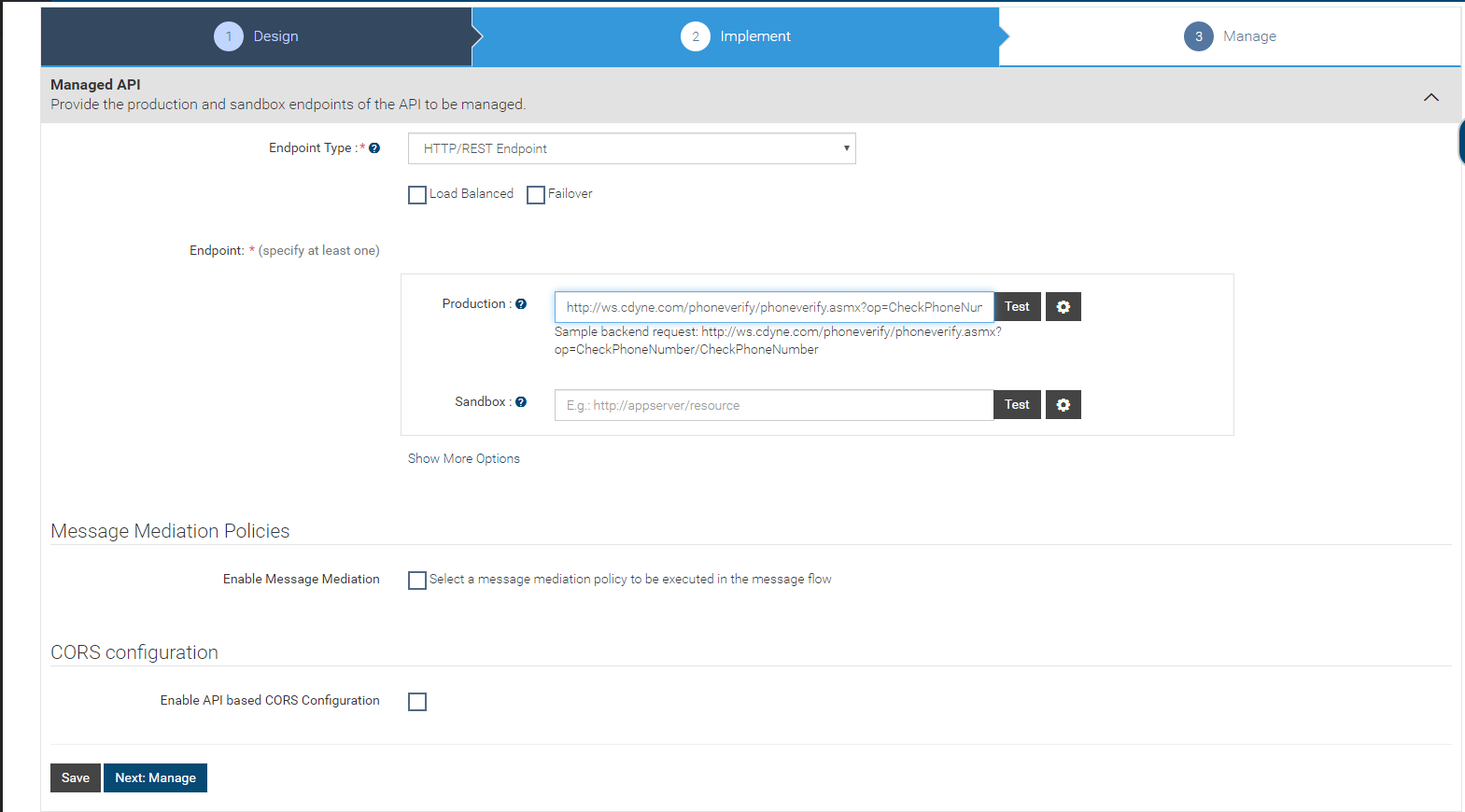
Una vez rellenados estos datos deberemos seleccionar la opción de prueba para una API, que nos permite el uso de todas las funcionalidades del API management dentro del rango de características que nos brinda la versión de prueba.

En este caso utilizaremos como ejemplo el código de una API ya creada en que si tú le introduces como parámetros un número de teléfono y un código de licencia nos verificara si ese número es real o no. Su código sería:



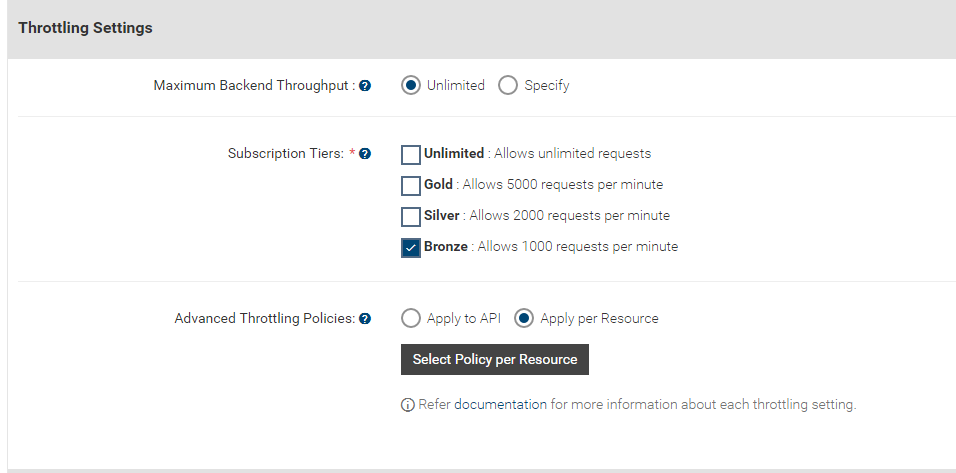
Este código ha sido obtenido de:

(<http://ws.cdyne.com/phoneverify/phoneverify.asmx?op=CheckPhoneNumber>)

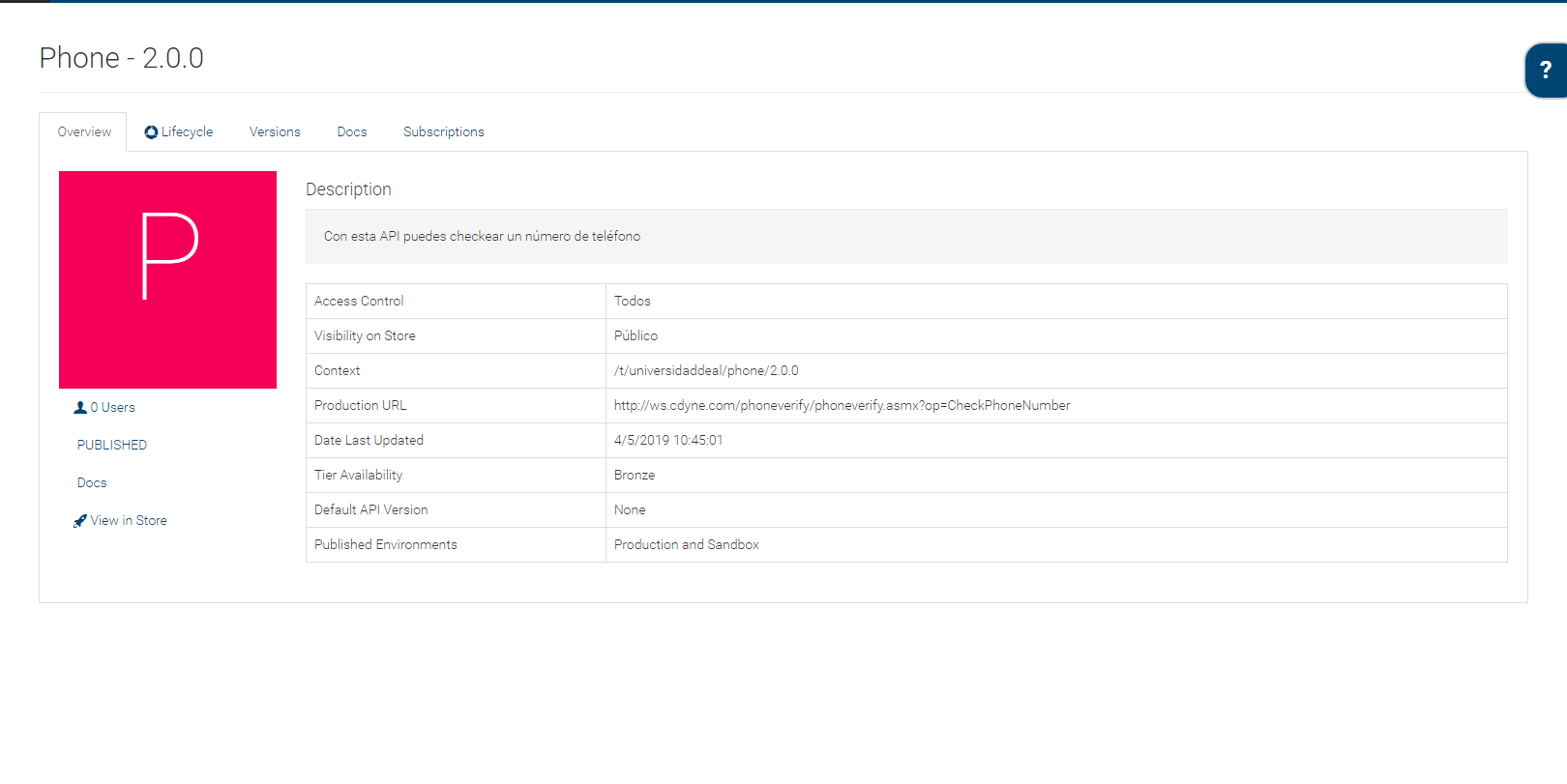


Introducimos dicha URL y continuamos.

Como último paso para la creación de nuestra API seleccionaremos el número de llamadas por minuto que se permitirá, en nuestro caso seleccionaremos la opción de bronce.



Con esto ya abríamos finalizado nuestro proceso de creación de una API, a partir de este momento nuestra API se encuentra en el API Store de WSO2, donde se puede trabajar con las distintas herramientas que ofrece el API Managment para la gestión y explotación de nuestra API. Además, otros usuarios podrán suscribirse a nuestra API para su uso a cambio de una compensación económica para el creador de la API.



## 5.3 Documentación de pruebas

Al comenzar a trabajar en el prototipo planteado de creación de una API para estudiar el funcionamiento de API Management, decidimos utilizar la nube que ofrece la plataforma de WSO2 para así equipararnos al otro prototipo que también se ha llevado a cabo en este proyecto.

El primer intento que se hizo una vez ya inscritos en la plataforma de WSO2 fue utilizar el paquete opensource que ofrece esta compañía para el API Management, cabe destacar que su instalación y puesta en funcionamiento es complicada y además no nos era de gran utilidad para este proyecto pues la comparación con el otro prototipo no sería la adecuada dado que los dos se deberían utilizar desde la nube.

Tras este primer intento conseguimos iniciar una versión de prueba de 15 días en la nube de WSO2. Una vez ya registrados comenzamos a investigar todo los que nos ofrece esta sección de API Management.

El siguiente intento consistió en crear una nueva API a partir de un código realizado por nosotros; este intento fue fallido, por nuestra nula experiencia con este tipo de herramientas y a la hora de crear APIs. Por este motivo después de un largo tiempo de intentos y de documentarnos, decidimos recurrir al soporte que nos ofrece WSO2. Tras recibir su respuesta con una serie de tutoriales y nueva información decidimos crear nuestra API a partir de un código ya existente.

Este último intento y definitivo es el que hemos utilizado como prototipo, puesto que conseguimos crear nuestra API y comenzar a trabajar con ella, en la suscripción, control de acceso, edición, etc.

## 5.4 Documentación de instalación

Nuestro prototipo como tal no necesita ningún tipo de instalación, pues se traba a través de la nube de WSO2 a la que se accede desde la página web de la misma organización (<https://wso2.com/api-management/>). Por tanto, nuestro proceso de instalación para que nuestro prototipo se encuentre operativo sería similar al proceso de creación que ha sido explicado en detalle en el apartado 5.2.

La única instalación que se ha llevado a cabo ha sido la de la herramienta y librería curl para realizar una serie de pruebas en las que se comprueba si existe conexión con el servidor. Curl actúa sobre la consola de comandos del sistema y nos proporciona una respuesta, en caso de que la conexión con el servidor se complete nos devolverá los parámetros introducidos en la prueba y en caso negativo nos devolverá un error. Un ejemplo de un comando para el uso de curl puede ser el siguiente: *curl -k -X GET "https://gateway.api.cloud.wso2.com:443/t/universidaddeal/phone/2.0.0/CheckPhoneNumber?PhoneNumber=918814042&Licencia=0" -H "accept: application/json" -H "Authorization: Bearer ".*

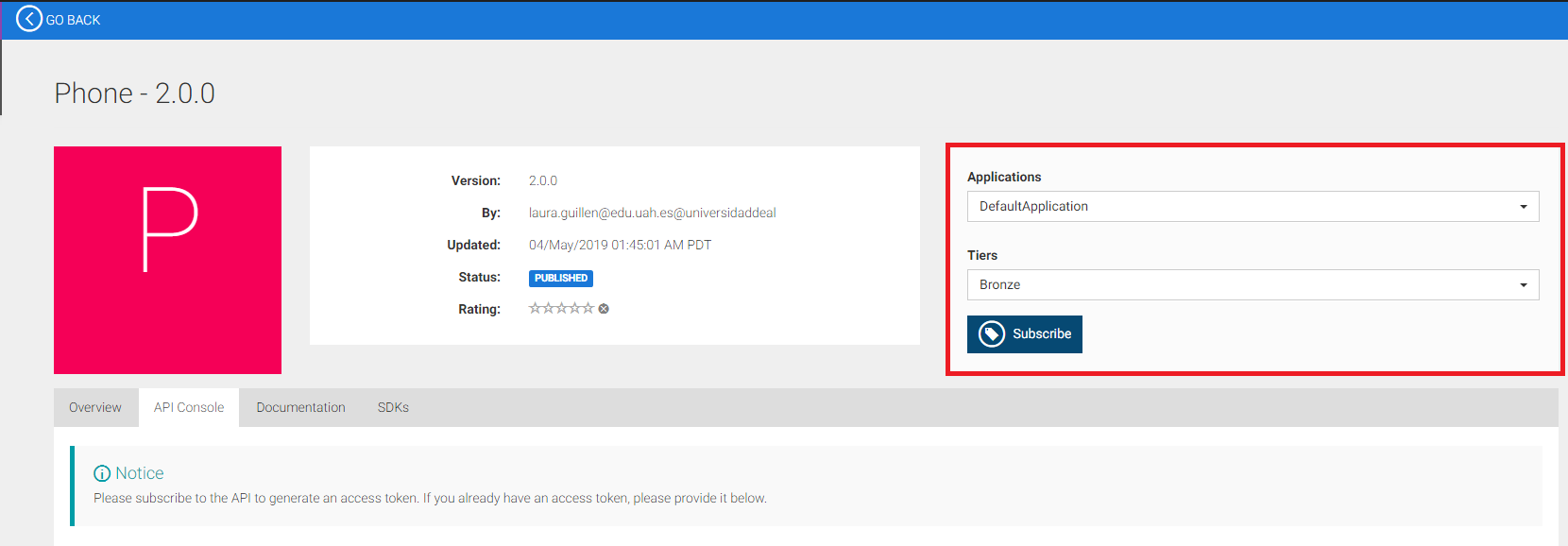
Para la instalación de esta herramienta y librería únicamente es necesario descargarse el archivo .rar que se encuentra en su página web (<https://curl.haxx.se/>). Una vez lo tengamos descargado lo recomendable es crear una carpeta en el disco C: llamada mycurl donde introduciremos el archivo curl de tipo aplicación, esto se hace para que su llamada desde el símbolo del sistema nos sea más sencilla.

## 5.5 Manual de usuario

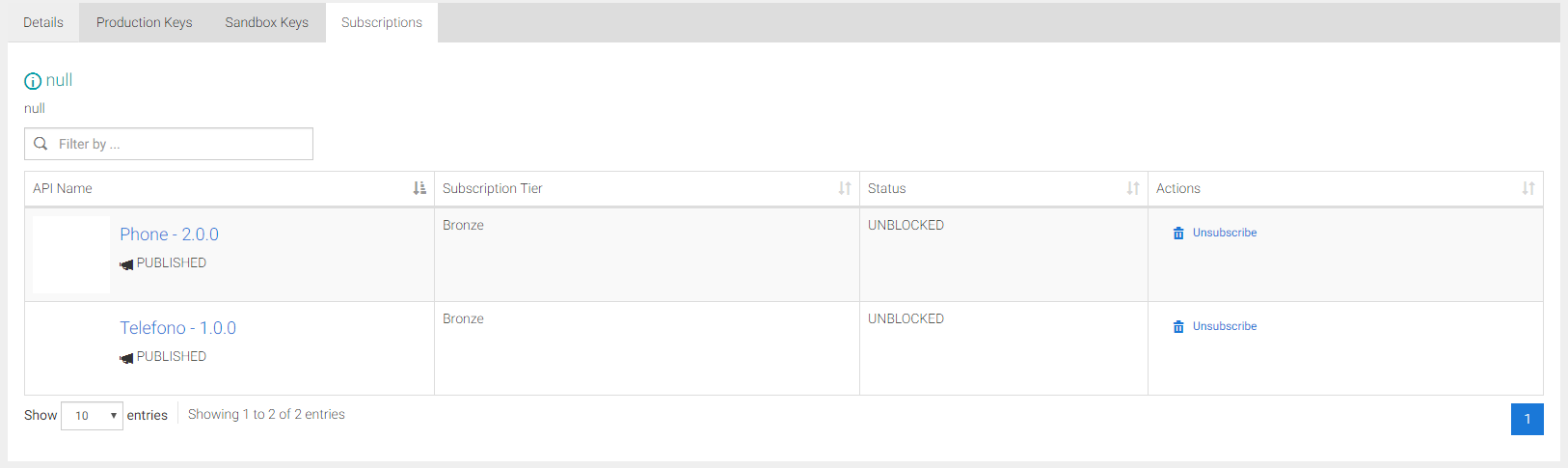
En este caso para el manual de usuario partiremos de la base de que ya tenemos creada nuestra API en la nube de WSO2 como ya ha sido comentado en el apartado 5.2. A partir de este momento se detallarán las distintas funcionalidades que nos ofrece esta plataforma para realizar un proceso de API Management con la API que hemos creado.

En primer momento comenzaremos con la suscripción a una API y su posterior invocación. En este caso realizaremos este proceso con nuestra API para que se veo mejor el proceso y el resultado de la suscripción.

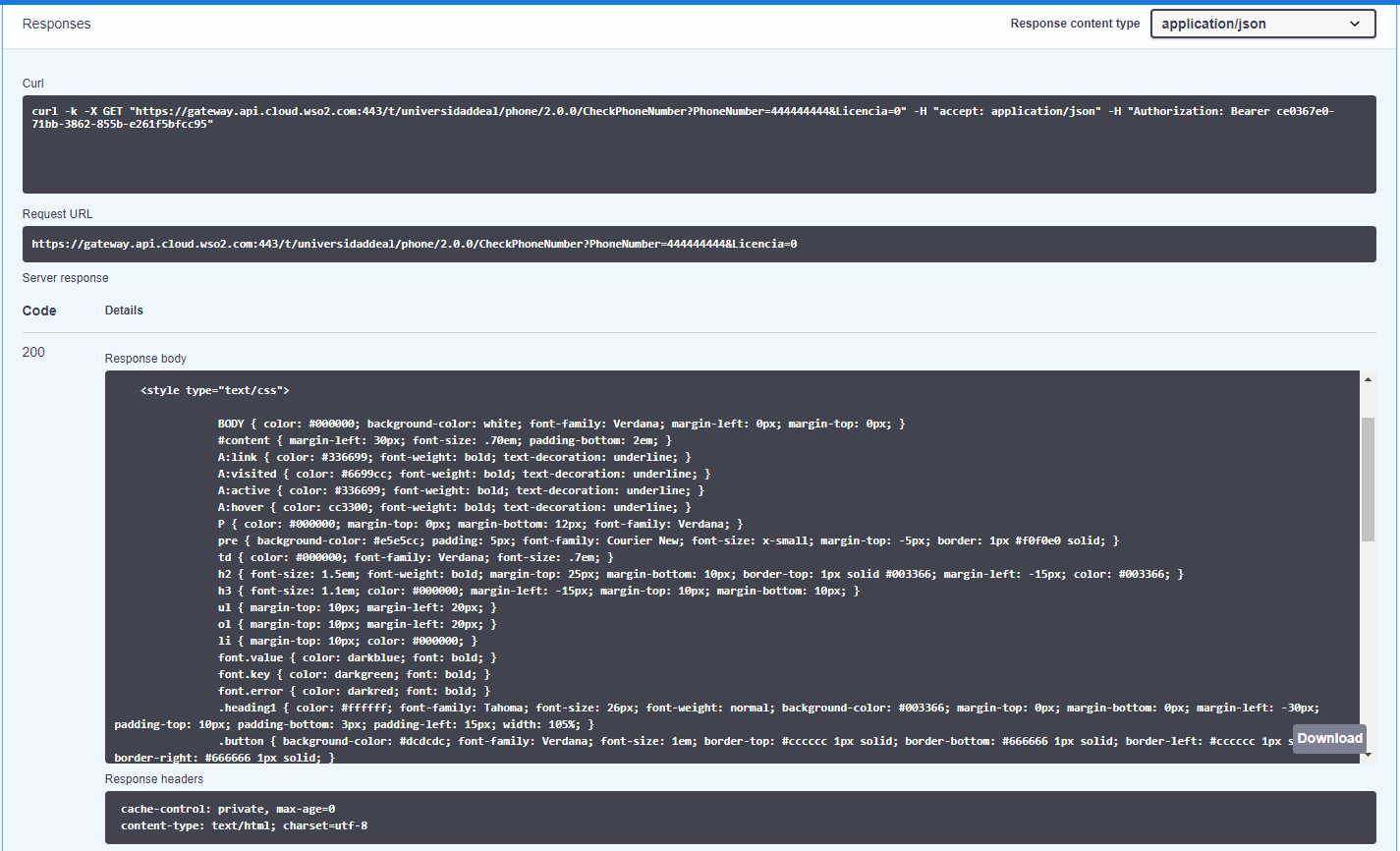
Para ello comenzaremos clicando en el API Store, una vez nos encontremos allí deberemos seleccionar la API a la que queremos suscribirnos, ir al recuadro de suscripción, seleccionar mi aplicación por defecto y el número llamadas por minuto que se realizarán. En la siguiente imagen se aprecia de una mejor forma.



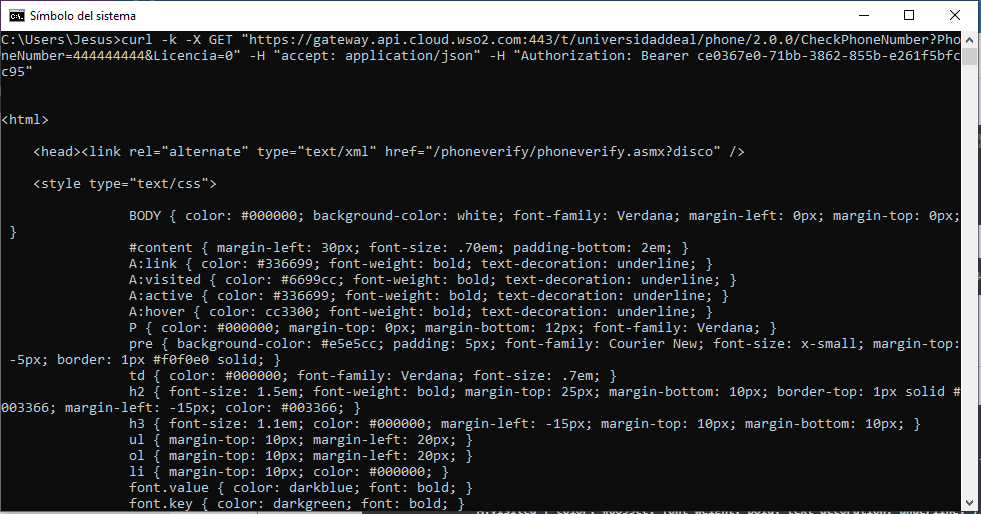
Después de aceptar la suscripción nos aparecerá una pantalla con de nuestra cuenta indicándonos a que APIs estamos suscritos.



Ahora deberemos realizar la invocación de la API, únicamente debemos encontrarnos en la pestaña de API Console donde introduciremos los parámetros que se nos pidan y el resultado sería el siguiente.



Para asegurarnos que la conexión con el servidor se realiza de una forma adecuada podemos usar la línea de curl que se nos proporciona desde el símbolo del sistema (cmd).

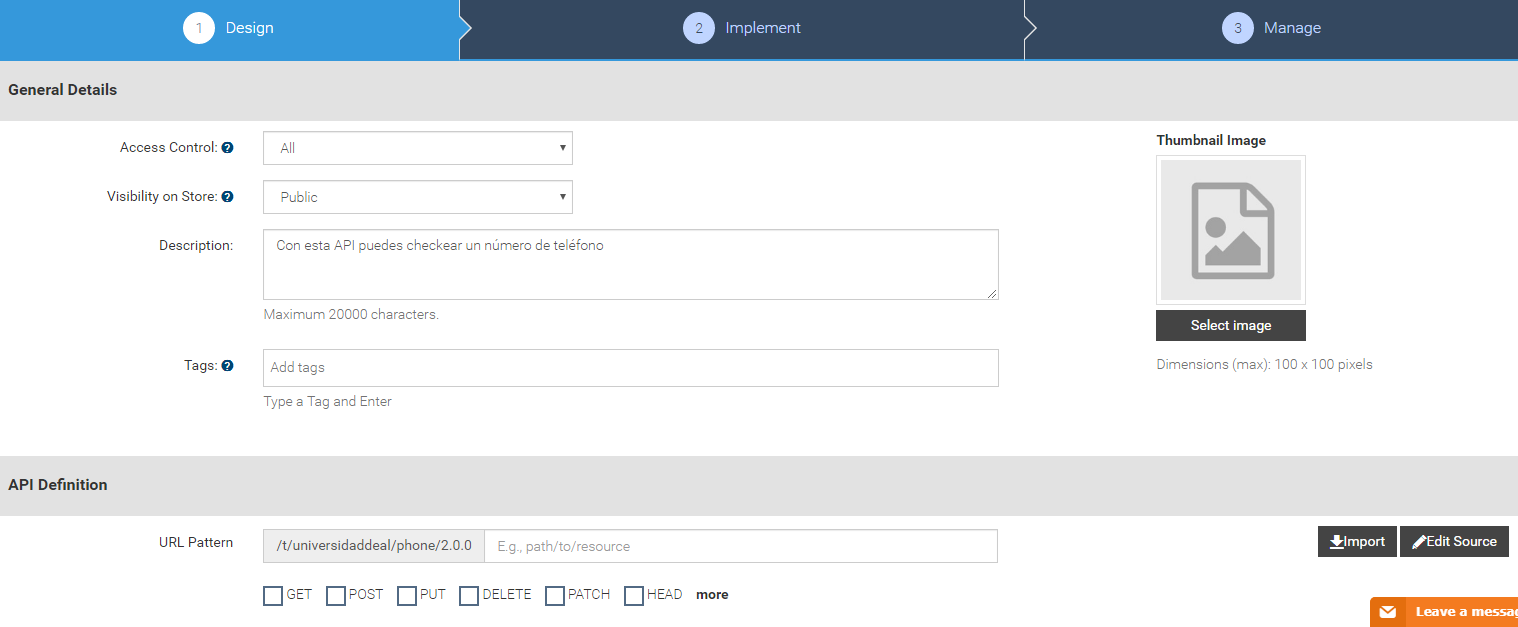


Si el resultado es óptimo como es el caso, la conexión será la adecuada y nos aparecerá el código de la API, de no ser así nos dará como resultado un error. Con todo lo anterior hemos conseguido suscribirnos a una API e invocarla.

A partir de este momento podemos realizar las distintas actividades de API Management como puede ser el control de acceso. Este control se puede seleccionar en el periodo de construcción de la API, donde tu como creador tienes la opción de elegir el acceso concurrente que deseas para tu API, hay 4 niveles a elegir, que serían Unlimited, Gold, Silver y Bronze, en nuestro caso decidimos elegir el más bajo (nivel bronce) que permite 1000 solicitudes por minuto.

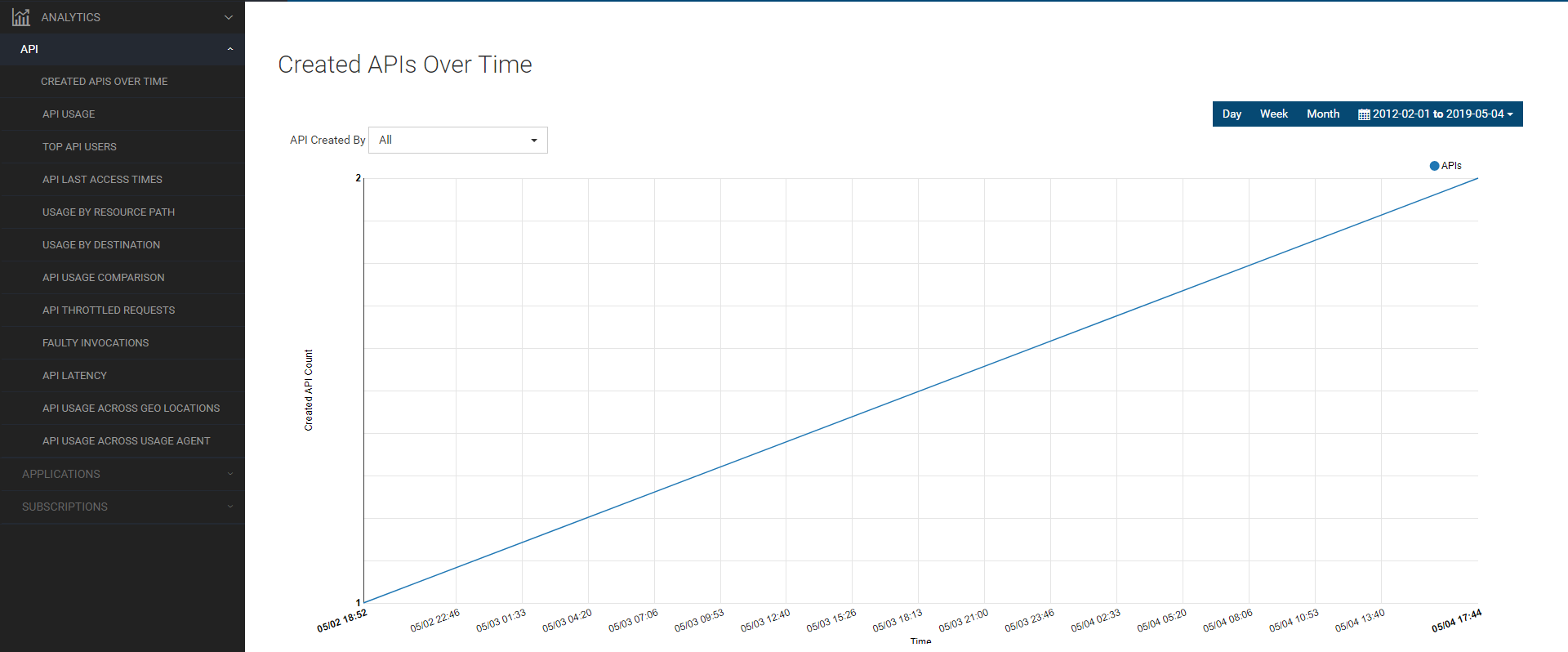
Las políticas de seguridad también se eligen en el proceso de creación, donde debes elegir entre un libre acceso para todos los usuarios o un acceso restringido para una serie de usuarios ordenador por roles. Al ser la versión de prueba en nuestro caso seleccionamos la opción de que todos los usuarios pudieran acceder al contenido de nuestra API.

Para la edición de tu API una vez creada deberás encontrarte en la nube de WSO2 y clicar el botón de EDIT API, donde podrás editar todo lo relacionado con tu API y los puntos mencionados anteriormente como si se tratase de un nuevo proceso de creación.



En el caso de WSO2 no es necesario instalar un servidor proxy si se va a trabajar con la nube como se ha mencionado anteriormente pues WSO2 en este caso lo tiene por defecto.

Por último, para poder acceder a las herramientas de análisis de tu API, es necesario que cliques en la pestaña de ANALYTICS, donde encontrarás un gran número de parámetros a analizar sobre el uso que se está haciendo de tu APIs, como su uso, número de llamadas, comparaciones entre tus APIs, etc.



Por otro lado, si queremos un análisis más individual para cada API y no tan general nos vemos en la obligación de clicar en el API Store donde en el apartado de STATISTICS, podremos observar nuevas estadísticas sobre nuestras APIs.

# 6. Comparación de las dos implementaciones

## 6.1 Evaluación de los criterios en la implementación usando la tecnología Apigee

| **CRITERIO** | **EVALUACIÓN** |
| --- | --- |
| Dificultad del proceso de creación y publicación | La dificultad en el proceso de creación y publicación de nuestras APIs ha sido relativamente sencilla, exceptuando la creación del proxy, que dentro de este proceso es la parte que implica mayor impedimento.  Valoración:3/5 |
| Horas empleadas en la construcción del prototipo | La construcción de este prototipo nos llevó aproximadamente 5 horas, principalmente, como se ha explicado anteriormente, debido a la dificultad de la creación el proxy, una vez sobrepasado este paso se requiere menos tiempo.  Valoración: 5 horas |
| Recursos necesarios | Para la creación del prototipo basado en la tecnología APIGEE es necesario el uso de un servidor Proxy para ejecutar las peticiones del usuario y cURL, una herramienta y librería la cual se aplica sobre la consola del sistema, la cual nos permite transferir datos a través de URL’s  Valoración: servidor Proxy y cURL |
| Configuración del servidor Proxy | Una vez iniciemos sesión en la plataforma podemos crear y configurar nuestro propio proxy en el que podremos indicarle el nombre, la ruta, descripción, seguridad, etc. Y una vez creado podremos testearlo y hacer modificaciones en el futuro.  Valoración: 4/5 |
| Grado de personalización | La plataforma de Google nos permite personalizar nuestras APIs de tal forma que se realiza a través del código fuente de la misma API para que cumpla la funcionalidad que nosotros queramos.  Valoración: 5/5 |
| Calidad del soporte | Por parte de Apigee no hemos recibido respuesta de la empresa a los correos enviados, por lo que su nivel de soporte en este apartado de atención es bastante escaso, por lo menos desde nuestra experiencia. A pesar de ello, el nivel de documentación de la herramienta y su facilidad mejorar la puntuación de este apartado  Valoración: 2/5 |
| Calidad de la documentación | La calidad de la documentación prestada por APIGEE es muy buena debido a que podemos encontrar una gran cantidad de información sobre la mayoría de apartados de la herramienta y, además, dicha información es muy detallada y precisa, cuenta paso a paso los pasos a seguir para los diferentes procesos de la aplicación.  Valoración: 5/5 |
| Facilidad de uso | APIGEE es un entorno muy intuitivo y enfocado a la comodidad de uso por parte del usuario, además, debido a la gran cantidad de documentación y tutoriales podemos encontrar explicaciones de cada apartado de una forma visual, sabiendo qué se puede realizar dentro de cada uno.  Valoración: 5/5 |
| Implementación de seguridad y control de concurrencia | A través del uso de Keys, Apigee permite implementar un método de seguridad la cual da acceso a diferentes métodos/apartados de las API’s según la key que tenga el usuario.  Además, se pueden controlar el número de usuarios que pueden acceder simultáneamente a las API’s.  Valoración: 4/5 |
| Profundidad del análisis | APIGEE permite hacer diferentes pruebas de nuestras APIs, según los parámetros que nosotros escojamos, como puede ser el número de visitas o él número de errores encontrados. El problema lo encontramos debido a que dichos análisis sólo permite realizarlos si contamos con la versión de pago de Apigee.  Valoración: 4/5 |
| Capacidad de prueba | Todas las APIs generadas en esta plataforma pueden ser testeadas para conocer su funcionalidad y claridad.  Valoración: Sí |

## 6.2 Evaluación de los criterios en la implementación usando la tecnología WSO2

| **CRITERIO** | **EVALUACIÓN** |
| --- | --- |
| Dificultad del proceso de creación y publicación | Para poder publicar y crear una API en WSO2 es conveniente leer la documentación sobre ello con anterioridad pues sin saber puede resultar una tarea complicada.  Valoración:4/5 |
| Horas empleadas en la construcción del prototipo | Dado que es necesario documentarse e informarse con anterioridad sobre el proceso de creación, incluiremos dicho tiempo en la creación del prototipo por lo que es una duración aproximada de 6h.  Valoración: 6 horas |
| Recursos necesarios | Puesto que estamos utilizando la versión de prueba en la nube de WSO2, únicamente es necesario el acceso a un navegador web.  Valoración: Conexión a internet |
| Configuración del servidor Proxy | La nube de WSO2 cuenta con un proxy propio ya configurado.  Valoración: 1/5 |
| Grado de personalización | WSO2 cuenta con la función de personalización de una API a gusto del creador, modificación de imagen, llamadas por segundo, seguridad y acceso a esta.  Valoración: 3,5/5 |
| Calidad del soporte | Resulta sencillo ponerse en contacto con el área de soporte de WSO2, además brinda una rápida atención y respuesta.  Valoración: 4/5 |
| Calidad de la documentación | La documentación que ofrece WSO2 es completa, pero en algunos momentos es complicada de entender, además de encontrarse por escrito también existen videotutoriales.  Valoración: 3,5/5 |
| Facilidad de uso | El trabajo con la nube de WSO2 no es especialmente fácil, pues es complicada de entender y de encontrar ciertas funciones, aun que por otro lado su interfaz de usuario si es amigable.  Valoración:3,5/5 |
| Implementación de seguridad y control de concurrencia | WSO2 permite integrar cierta seguridad en tu API a través de la creación de roles y controlar el acceso simultaneo a la API por parte del creador.  Valoración: 4/5 |
| Profundidad del análisis | La plataforma de WSO2 cuenta con una amplia disposición de herramientas de análisis para una API trabajando a través de una serie de parámetros. Sus resultados son de gran ayuda  Valoración: 5/5 |
| Capacidad de prueba | WSO2 permite realizar una prueba del código de una API ejecutando los métodos de esta.  Valoración: Si |

* Configuración del servidor proxy (WSO2): se valora como 1 dado que no existe dificultad alguna pues ya viene integrado en la plataforma.
* Dificultad del proceso de creación (WSO2): Entendemos 5 como una muy alta dificultad y 1 como muy baja dificultad.
* En los apartados que se mida la dificultad la entenderemos como se ha comentado en el punto anterior.

# 7. Comparación de la implementación de las tecnologías

Debe incluir al menos una tabla resumen, en sección de página horizontal, cruzando los criterios y los valores de cada tecnología. Con una columna de comentarios sobre la comparación

| **CRITERIOS** | **APIGEE** | **WSO2** | **COMENTARIOS** |
| --- | --- | --- | --- |
| Dificultad del proceso de creación y publicación | 3/5 | 4/5 | La principal dificultad de Apigee respecto a este criterio reside en la creación del proxy, criterio que evaluaremos independientemente y sin embargo, en WSO2 aun sin tener que crear el proxy, la creación y publicación de APIs requiere un arduo estudio previo de la documentación. |
| Horas empleadas en la construcción del prototipo | 5 horas | 6 horas | Aunque el tiempo empleado en ambas plataformas es prácticamente el mismo, de nuevo la diferencia recae en que en el caso de Apigee ese tiempo incluye la creación del proxy, mientras que en WSO2 únicamente se refiere al tiempo de documentación y creación, pues el proxy lo proporciona la plataforma por defecto. |
| Recursos necesarios | Servidor Proxy y cURL | Conexión a internet | Este criterio está condicionado a la modalidad escogida. En este caso para poder utilizar correctamente Apigee Edge se requiere un servidor proxy y cURL, mientras que para WSO2 solamente conexión a internet pues estamos utilizando la versión de prueba en la nube. |
| Configuración del servidor Proxy | 4/5 | 1 | Apigee nos brinda la posibilidad de crear nuestro propio servidor proxy y personalizarlo, añadiéndole dificultad a la creación del prototipo, mientras que por el contrario la nube de WSO2 cuenta con un proxy propio ya configurado |
| Grado de personalización | 5/5 | 3,5/5 | Ambas plataformas presentan la posibilidad de personalización, sin embargo la plataforma Apigee nos ofrece un abanico de posibilidades más amplio |
| Calidad del soporte | 2/5 | 4/5 | En relación al soporte recibido, la plataforma de WSO2 supera con creces a la de Apigee, por lo menos desde nuestra experiencia. |
| Calidad de la documentación | 5/5 | 3.5/5 | Ambas plataformas ofrecen documentación muy completa, siendo más detallada, precisa y fácil de comprender la de Apigee. |
| Facilidad de uso | 5/5 | 3.5/5 | En relación a la facilidad de uso de ambas plataformas, Apigee es bastante más sencilla e intuitiva de utilizar, siendo la modalidad en la nube de WSO2 complicada de entender y en ocasiones de localizar ciertas funcionalidades. |
| Implementación de seguridad y control de concurrencia | 4/5 | 4/5 | Ambas plataformas nos brindan la posibilidad de implementar seguridad, pudiendo en los dos casos ser más robusta, pero bastante completa. |
| Profundidad del análisis | 4/5 | 5/5 | Tanto Apigee como WSO2 nos permiten realizar análisis en función de diferentes parámetros que son de gran utilidad. El problema de Apigee es que solo se pueden hacer con la versión de pago y no con la prueba gratuita con la que nosotros contamos |
| Capacidad de prueba | Sí | SÍ | En Apigee todas las APIs pueden ser testeadas para conocer su funcionalidad y claridad y WSO2 permite realizar una prueba del código de una API ejecutando los métodos de esta. |

# 8. Conclusiones

Tras realizar la implementación de los prototipos de ambas tecnologías, podríamos concluir diciendo que tanto Apigee como WSO2 presentan funcionalidades muy similares, tal y como se pone de manifiesto en la tabla anterior. No obstante, tras haber trabajado con cada una de ellas hemos comprobado que Apigee tiene aspectos tales como una documentación más precisa y detallada, un entorno de trabajo más intuitivo y un mayor grado de personalización que le hacen destacar por encima de WSO2. Por tanto, aunque funcionalmente son muy parejas, estos criterios adicionales hacen que en nuestro caso consideremos que la plataforma de api management proporcionada por Apigee es una mejor herramienta en rasgos generales, al menos en su versión de prueba principalmente para usuarios sin ninguna experiencia en el sector y en la tecnología.