

# Práctica 3

## Implementación de servicios SOAP y REST en *Enterprise Service Bus*

Diseño y Mantenimiento de Servicios

# Índice

|                                                                                                                                                                   |           |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
| <b>1. Introducción, expectativas y temporización</b>                                                                                                              | <b>4</b>  |
| 1.1. Presentación . . . . .                                                                                                                                       | 4         |
| 1.2. <i>Abstract</i> . . . . .                                                                                                                                    | 4         |
| 1.3. Temporización y organización . . . . .                                                                                                                       | 4         |
| 1.4. Expectativas . . . . .                                                                                                                                       | 4         |
| <b>2. Apartado 1: Desarrollo e implementación de servicios propios y externos a través de la plataforma ESB Mule</b>                                              | <b>5</b>  |
| 2.1. Parte común: Listener con HTTPS . . . . .                                                                                                                    | 5         |
| 2.1.1. Generar KeyStore y certificados HTTPS . . . . .                                                                                                            | 5         |
| 2.1.2. Configuración del <i>Listener</i> . . . . .                                                                                                                | 6         |
| 2.2. Servicios SOAP . . . . .                                                                                                                                     | 8         |
| 2.2.1. SOAP Interno . . . . .                                                                                                                                     | 8         |
| Configuración en AnyPoint Studio . . . . .                                                                                                                        | 8         |
| Configuración en <i>backend</i> Flask . . . . .                                                                                                                   | 10        |
| Acceso al vídeo explicativo . . . . .                                                                                                                             | 10        |
| 2.2.2. SOAP Externo . . . . .                                                                                                                                     | 11        |
| Configuración en AnyPoint Studio . . . . .                                                                                                                        | 11        |
| Acceso al vídeo explicativo . . . . .                                                                                                                             | 15        |
| 2.3. Servicios REST . . . . .                                                                                                                                     | 16        |
| 2.3.1. REST Interno . . . . .                                                                                                                                     | 16        |
| Configuración en AnyPoint Studio . . . . .                                                                                                                        | 16        |
| Configuración en <i>backend</i> Flask . . . . .                                                                                                                   | 18        |
| Acceso al vídeo explicativo . . . . .                                                                                                                             | 18        |
| 2.3.2. REST Externo . . . . .                                                                                                                                     | 18        |
| Configuración en AnyPoint Studio . . . . .                                                                                                                        | 18        |
| Configuración en AnyPoint Studio . . . . .                                                                                                                        | 18        |
| Acceso al vídeo explicativo . . . . .                                                                                                                             | 20        |
| 2.4. Código . . . . .                                                                                                                                             | 21        |
| 2.4.1. De proyecto en AnyPoint Studio . . . . .                                                                                                                   | 21        |
| 2.4.2. De aplicativo desarrollado en Flask . . . . .                                                                                                              | 24        |
| <b>3. Apartado 2: Estudiar las características del Mule ESB como herramienta para implementar servicios, haciendo especial hincapié en el “Management Center”</b> | <b>27</b> |
| 3.1. API Manager . . . . .                                                                                                                                        | 28        |
| 3.2. Analytics . . . . .                                                                                                                                          | 29        |
| 3.3. RunTime Manager . . . . .                                                                                                                                    | 32        |
| 3.4. Access Management . . . . .                                                                                                                                  | 33        |
| <b>4. Apartado 3: Implementación de alguno de los elementos anteriores</b>                                                                                        | <b>34</b> |
| <b>5. Conclusiones</b>                                                                                                                                            | <b>39</b> |
| <b>6. Acceso a los vídeos en YouTube</b>                                                                                                                          | <b>40</b> |
| <b>7. Bibliografía</b>                                                                                                                                            | <b>41</b> |

## Índice de figuras

|     |                                                                                                       |    |
|-----|-------------------------------------------------------------------------------------------------------|----|
| 1.  | Generación de <i>KeyStore</i> . . . . .                                                               | 5  |
| 2.  | Contenido del <i>KeyStore</i> . . . . .                                                               | 5  |
| 3.  | Módulo <i>Listener</i> en el <i>Mule Palette</i> . . . . .                                            | 6  |
| 4.  | Configuración del <i>Listener</i> , general . . . . .                                                 | 6  |
| 5.  | Configuración del <i>Listener</i> , detalle TLS . . . . .                                             | 7  |
| 6.  | Configuración del <i>Listener</i> , MIME y codificación . . . . .                                     | 8  |
| 7.  | Pipeline del Servicio SOAP Interno . . . . .                                                          | 8  |
| 8.  | Adición de <i>Listener</i> al área de trabajo . . . . .                                               | 9  |
| 9.  | Configuración de <i>Set Variable</i> para SOAP Interno . . . . .                                      | 9  |
| 10. | Configuración de <i>Logger</i> para SOAP Interno . . . . .                                            | 9  |
| 11. | Configuración de <i>HTTP_Request_configuration-INTERNO</i> para SOAP Interno . . . . .                | 10 |
| 12. | Detalle de configuración <i>URI Params</i> en el <i>Request</i> de SOAP Interno . . . . .             | 10 |
| 13. | Pipeline completo para <i>SOAP Externo</i> . . . . .                                                  | 11 |
| 14. | Adición de <i>Listener</i> al área de trabajo . . . . .                                               | 11 |
| 15. | Parejas de <i>Variable</i> + <i>Logger</i> en el pipeline de <i>SOAP Externo</i> . . . . .            | 12 |
| 16. | Contenido de cada uno de los <i>ifs</i> del <i>Choice</i> en el <i>SOAP Externo</i> . . . . .         | 13 |
| 17. | Configuración del módulo <i>Consume</i> para el SOAP Externo . . . . .                                | 14 |
| 19. | <i>Default response</i> en SOAP Externo . . . . .                                                     | 15 |
| 18. | Selector de operación en el módulo <i>Consume</i> del SOAP Externo . . . . .                          | 15 |
| 20. | Pipeline del Servicio REST Interno . . . . .                                                          | 16 |
| 21. | Adición de <i>Listener</i> al área de trabajo . . . . .                                               | 16 |
| 22. | Configuración de <i>Set Variable</i> para REST Interno . . . . .                                      | 17 |
| 23. | Configuración de <i>Logger</i> para REST Interno . . . . .                                            | 17 |
| 24. | Configuración de <i>HTTP_Request_configuration-INTERNO</i> para REST Interno . . . . .                | 17 |
| 25. | Detalle de configuración <i>URI Params</i> en el <i>Request</i> de REST Interno . . . . .             | 18 |
| 26. | Pipeline del Servicio REST Externo . . . . .                                                          | 19 |
| 27. | Adición de <i>Listener</i> al área de trabajo en REST Externo . . . . .                               | 19 |
| 28. | Configuración de <i>Set Variable</i> para REST Externo . . . . .                                      | 19 |
| 29. | Configuración de <i>Logger</i> para REST EXTERNO . . . . .                                            | 20 |
| 30. | Configuración de <i>HTTP_Request_configuration-INTERNO</i> para SOAP Interno . . . . .                | 20 |
| 31. | Detalle de configuración <i>URI Params</i> en el <i>Request</i> de REST Externo . . . . .             | 20 |
| 32. | Diferencia de latencias a regiones Azure. Obtenido de AzureSpeed.com . . . . .                        | 27 |
| 33. | Captura de API Manager. (Extraído de MuleSoft, 2023c) . . . . .                                       | 28 |
| 34. | Captura de API Manager, <i>Service Mesh</i> . (Extraído de MuleSoft, 2023c) . . . . .                 | 29 |
| 35. | Captura de API Manager, <i>API Analysis</i> . (Extraído de MuleSoft, 2023c) . . . . .                 | 29 |
| 36. | Captura de API Manager, <i>API Analytics</i> . (Extraído de MuleSoft, 2023a) . . . . .                | 30 |
| 37. | Captura de API Manager, <i>User Behavior</i> . (Extraído de MuleSoft, 2023a) . . . . .                | 30 |
| 38. | Captura de API Manager, <i>Compliance</i> . (Extraído de MuleSoft, 2023a) . . . . .                   | 31 |
| 39. | Captura de API Manager, <i>Key API trends</i> . (Extraído de MuleSoft, 2023a) . . . . .               | 31 |
| 40. | Captura de API Manager, <i>Gráficos personalizados</i> . (Extraído de MuleSoft-Docs, 2023e) . . . . . | 32 |
| 41. | Escenarios de despliegue en <i>Runtime Manager</i> . (Extraído de MuleSoft-Docs, 2023a) . . . . .     | 33 |
| 42. | Ejemplo de gestión de identidad. (Extraído de MuleSoft-Blog, 2016) . . . . .                          | 34 |
| 43. | Seleccionar <i>API Manager</i> en el menú. . . . .                                                    | 35 |
| 44. | Botón <i>Automated Policies</i> . . . . .                                                             | 35 |
| 45. | Botón <i>Add Automated Policy</i> . . . . .                                                           | 36 |

|     |                                                                            |    |
|-----|----------------------------------------------------------------------------|----|
| 46. | Seleccionar opción <i>IP Blocklist</i> . . . . .                           | 36 |
| 47. | RIPEStat muestra, entre otra información, la ruta de las IP . . . . .      | 37 |
| 48. | Añadir rango IP a bloquear . . . . .                                       | 37 |
| 49. | Regla <i>IP Blocklist</i> creada desde <i>Automated Policies</i> . . . . . | 38 |

## Índice de tablas

*Este documento no contiene tablas.*

## Listings

|    |                                                                             |    |
|----|-----------------------------------------------------------------------------|----|
| 1. | SOAP-Externo - Transform Dividir . . . . .                                  | 13 |
| 2. | SOAP-Externo - Transform Multiplicar . . . . .                              | 13 |
| 3. | SOAP-Externo - Transform Sumar . . . . .                                    | 13 |
| 4. | SOAP-Externo - Transform Restar . . . . .                                   | 14 |
| 5. | practica3.xml - Código de flujos en AnyPoint Studio . . . . .               | 21 |
| 6. | palabras.py - Código que genera el backend para servicios propios . . . . . | 24 |

## 1. Introducción, expectativas y temporización

### 1.1. Presentación

En el presente trabajo se aplicarán los conocimientos adquiridos hasta el momento en la asignatura de Diseño y Mantenimiento de Servicios, perteneciente al Grado en Ciencia, Gestión e Ingeniería de Servicios por la Universidad Rey Juan Carlos. En concreto, el objetivo es incrementar los conocimientos adquiridos sobre el diseño y mantenimiento de servicios a través de un ESB (Enterprise Service Bus).

Durante todo el desarrollo de la práctica se incluirán capturas y recursos gráficos para demostrar la realización de las pruebas, así como para ejemplificar el proceso y facilitar la comprensión del desarrollo del trabajo.

### 1.2. *Abstract*

This work will apply the knowledge acquired so far in the subject of *Diseño y Mantenimiento de Servicios*, belonging to the Degree in Science, Management and Engineering of Services (Ciencia, Gestión e Ingeniería de Servicios) by the Universidad Rey Juan Carlos. Specifically, the objective is to increase the knowledge acquired about the design and maintenance of services through an ESB (Enterprise Service Bus).

Throughout the development of the practice, screenshots and graphic resources will be included to demonstrate the realization of the tests, as well as to illustrate the process and facilitate the understanding of the development of the work.

### 1.3. Temporización y organización

Tal y como se recoge en la Guía de Estudio de la asignatura, la práctica se engloba dentro del tema 3, *Patrones Avanzados*. Esta entrega, según la cronología de la citada Guía, se desarrollará entre los días 25/10/2023 y 20/11/2023. (URJC, 2023) Sí cabe mencionar que la fecha límite de entrega de la presente práctica se ha fijado para el día 23/12/2023 en el Aula Virtual.

El trabajo se desarrollará de forma individual.

### 1.4. Expectativas

Las expectativas de aprendizaje de la presente práctica se basan en la completa comprensión del funcionamiento de REST y SOA, así como implementación de servicios que los usan en un ESB (MuleSoft AnyPoint, en este caso).

## 2. Apartado 1: Desarrollo e implementación de servicios propios y externos a través de la plataforma ESB Mule

*Se proveerá al alumno de un conjunto de vídeos de ejemplo para la implementación de servicios REST y SOAP a través del ESB. A partir de los cuales, el alumno realizará servicios similares de igual complejidad. En total se deberán implementar 4 servicios.*

### 2.1. Parte común: Listener con HTTPS

#### 2.1.1. Generar KeyStore y certificados HTTPS

El *KeyStore* es el formato en el que Java guarda certificados y claves (privadas). Para generarlo, podemos usar la utilidad `keytool.exe` del *Java Development Kit* (en adelante, *JDK*).

El *JDK* viene incluido en la instalación de AnyPoint Studio. Para encontrarlo podemos revisar los *logs*, registros de alguna ejecución previa. En mi caso, la ruta de la utilidad es:

`C:\AnypointStudio\plugins\org.mule.tooling.jdk.v8.win32.x86_64_1.2.1\bin\keytool.exe`

Nos situaremos en la carpeta de recursos del proyecto actual, en mi caso:

PS `C:\Users\pablo\AnypointStudio\studio-workspace\helloworldpablo-mod\src\main\resources`

Será en esta ruta en la que ejecutaremos el comando a continuación para generar las claves <sup>1</sup>:

`keytool.exe -genkey -alias key-alias -keystore keystore-name.jks -keyalg RSA -storetype JKS`

Al ejecutar el comando se nos irá solicitando por terminal una serie de valores. El más importante es el primero, pues será el CN del certificado. Deberá coincidir con la dirección por la que accedamos al servidor: 127.0.0.1 en nuestro caso.

```
PS C:\Users\pablo\AnypointStudio\studio-workspace\helloworldpablo-mod\src\main\resources> C:\AnypointStudio\plugins\org.mule.tooling.jdk.v8.win32.x86_64_1.2.1\bin\keytool.exe -genkey -alias key-alias -keystore keystore-name.jks -keyalg RSA -storetype JKS
Enter keystore password:
Keystore password is too short - must be at least 6 characters
Enter keystore password:
Re-enter new password:
What is your first and last name?
[Unknown]: 127.0.0.1
What is the name of your organizational unit?
[Unknown]: Pablo Gonzalez Troyano DEMO
What is the name of your organization?
[Unknown]: Disenno y Mantenimiento de Servicios
What is the name of your City or Locality?
[Unknown]: Madrid
What is the name of your State or Province?
[Unknown]: Spain
What is the two-letter country code for this unit?
[Unknown]: ES
Is CN=127.0.0.1, OU=Pablo Gonzalez Troyano DEMO, O=Disenno y Mantenimiento de Servicios, L=Madrid, ST=Spain, C=ES correct?
[no]: |
```

Figura 1: Generación de *KeyStore*

Con `keytool.exe -list -keystore .\keystore.jks` podremos ver el certificado que contiene:

```
PS C:\Users\pablo\AnypointStudio\studio-workspace\helloworldpablo-mod\src\main\resources> C:\AnypointStudio\plugins\org.mule.tooling.jdk.v8.win32.x86_64_1.2.1\bin\keytool.exe -list -keystore .\keystore.jks
Enter keystore password:
Keystore type: jks
Keystore provider: SUN

Your keystore contains 1 entry

key-alias, 01-oct-2023, PrivateKeyEntry,
Certificate fingerprint (SHA-256): 69:6F:DA:30:B6:76:75:3A:3A:69:10:AF:69:F8:15:BC:05:E9:D3:C5:A8:96:15:AB:D2:8E:B3:E
C:D8:27:5E
```

Figura 2: Contenido del *KeyStore*

<sup>1</sup>Se omite la ruta completa al ejecutable. Se debería incluir o añadir la carpeta en la que se ubica a la variable `%PATH%` del sistema operativo.

### 2.1.2. Configuración del *Listener*

Desde el *Mule Palette*, localizado en la parte superior derecha de la interfaz de Mule Studio, arrastramos el módulo *Listener* al espacio de trabajo (la parte central de la interfaz):

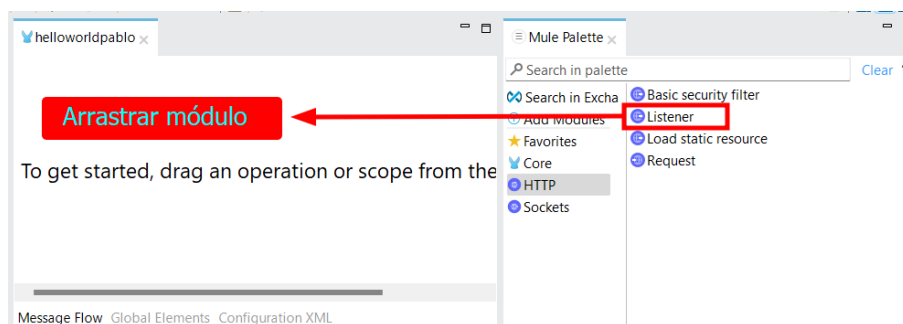


Figura 3: Módulo *Listener* en el *Mule Palette*

La configuración del *Listener* difiere parcialmente, tal y como se puede ver a continuación:

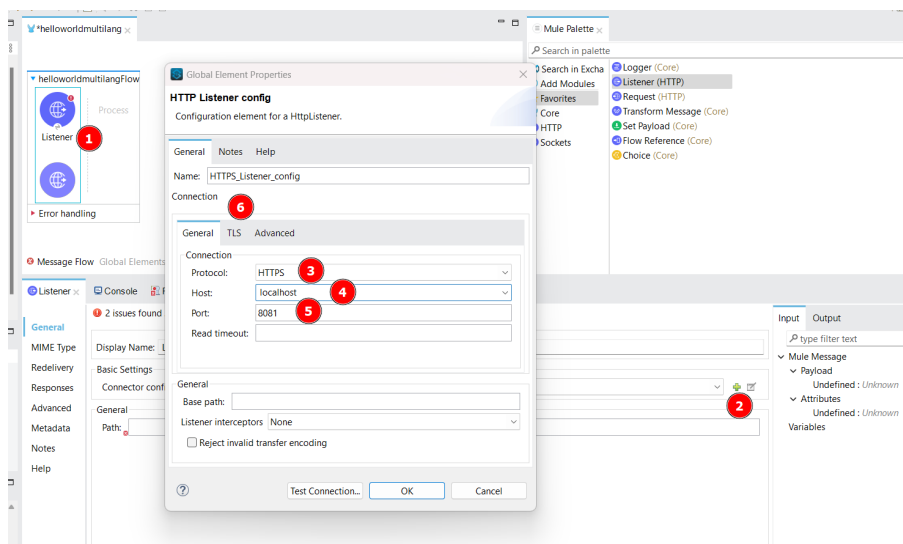


Figura 4: Configuración del *Listener*, general

Primero, se selecciona el módulo en el flujo [1], en la configuración del módulo se hace clic en «+» para añadir una nueva *HTTP Listener config* [2]. En la configuración se indicará que se usa el protocolo HTTPS [3], que el host será *localhost* [4], y el puerto lo mantenemos en el 8081 [5].

En la pestaña *TLS* [6], debemos indicar los detalles del *KeyStore* generado en el paso anterior:

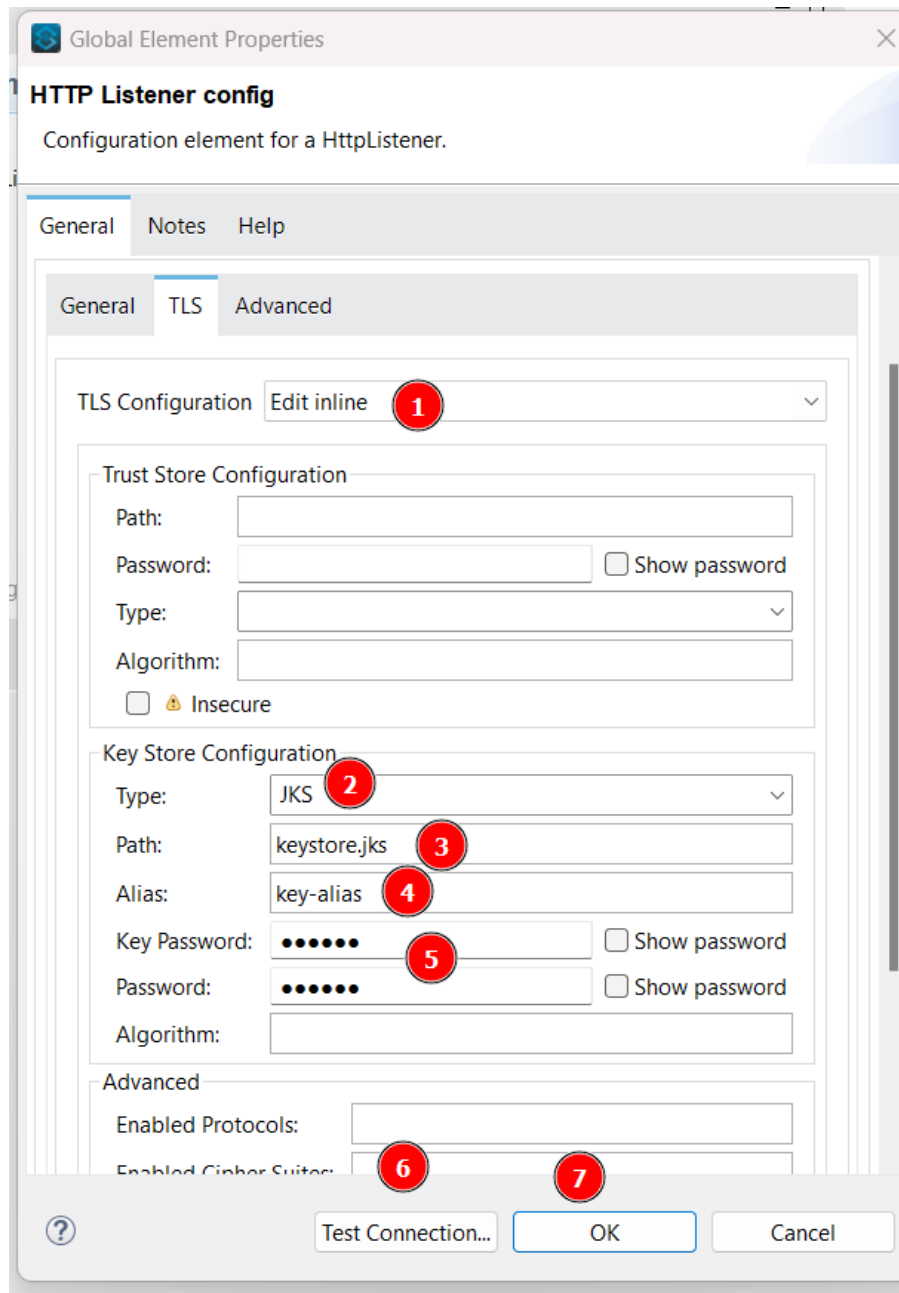


Figura 5: Configuración del *Listener*, detalle TLS

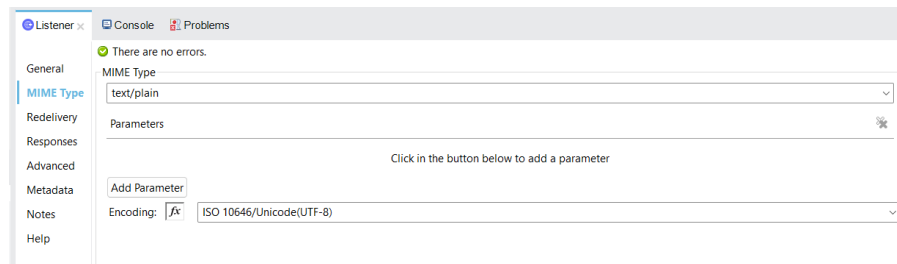
Para la configuración TLS se selecciona *Edit inline* para editar la configuración en la ventana [1], en la configuración *Key Store Configuration* se indica *JKS* como tipo [2], *keystore.jks* [3] como ruta <sup>2</sup>, *key-alias* como alias de la clave [4]<sup>3</sup>, así como la contraseña en los campos correspondientes [5].

Es altamente recomendable probar la conexión [6] antes de guardarlo [7].

Indicamos */holamundo* como *path* en la configuración y editamos el *MIME Type* a *text/plain* y encoding a UTF-8:

<sup>2</sup>Teniendo en cuenta que está en la carpeta *src/resources*

<sup>3</sup>Se puede consultar el alias de la clave con el comando *keystore -list* indicado previamente

Figura 6: Configuración del *Listener*, MIME y codificación

## 2.2. Servicios SOAP

### 2.2.1. SOAP Interno

Para el servicio SOAP Interno se ha usado *Flask* (Python) para simular una serie de entradas en el diccionario.

#### Configuración en AnyPoint Studio

En AnyPoint Studio el *pipeline* para esta demostración es el siguiente:

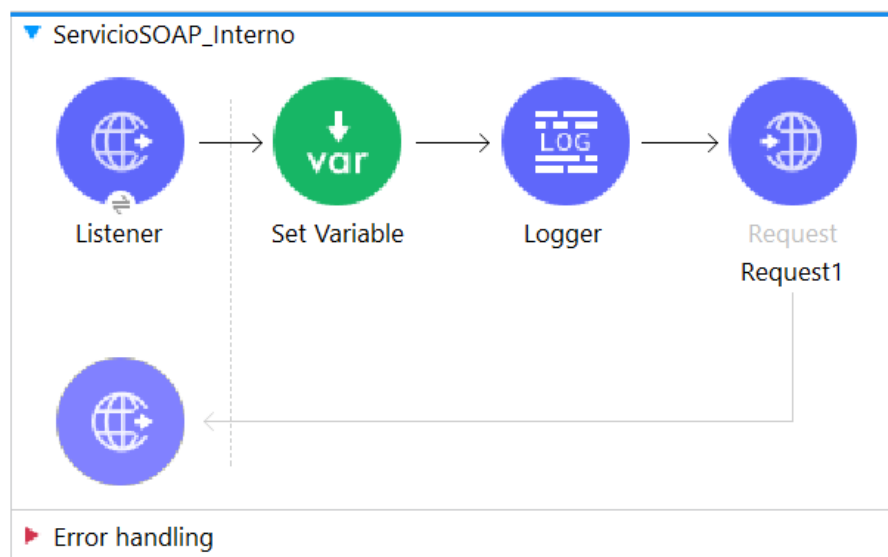
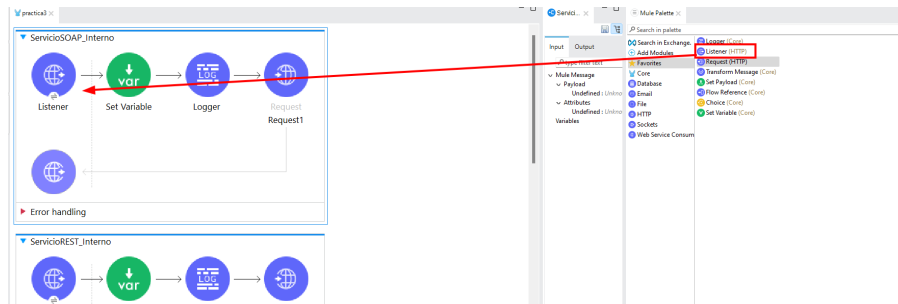


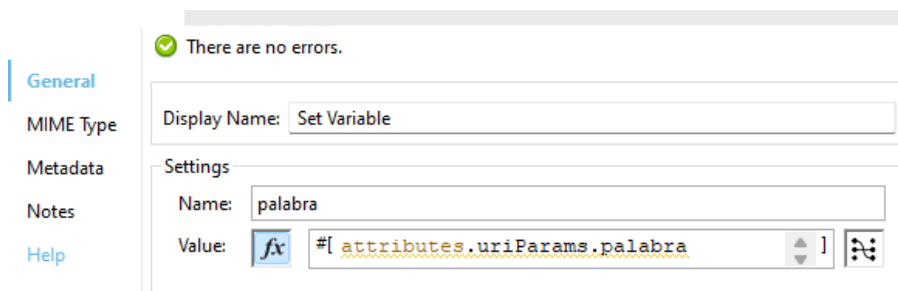
Figura 7: Pipeline del Servicio SOAP Interno

Para la creación de cada uno de los elementos que lo componen se han seguido los siguientes pasos:

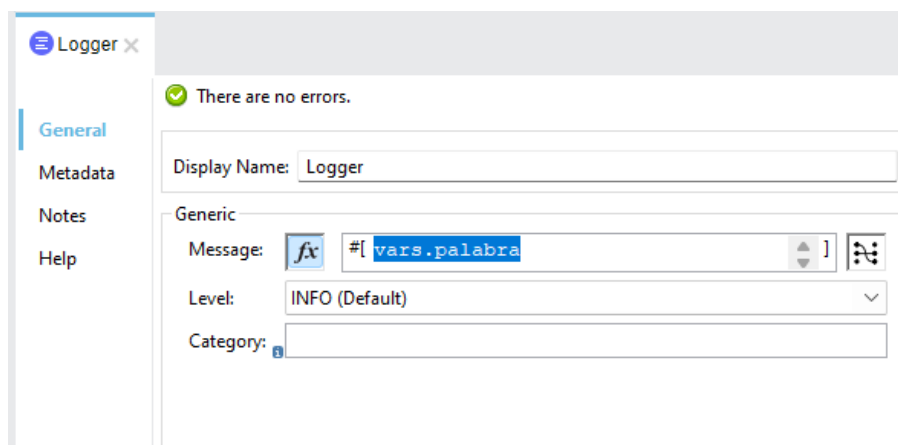
1. Añadir el *Listener* al área de trabajo, arrastrándolo desde el *Mule Palette*:

Figura 8: Adición de *Listener* al área de trabajo

- La configuración del parámetro (además de lo ya explicado anteriormente en lo relativo al HTTPS) es sencilla. La única variable que se ha de editar es el *Path*:  
`/soap_interno/palabra/{palabra}`
- Se usará el parámetro de URI *palabra* para identificar qué es lo que se quiere solicitar. Para hacerlo, hemos de fijar la variable en el código durante la ejecución usando el módulo *Set Variable*:

Figura 9: Configuración de *Set Variable* para SOAP Interno

- Para comprobar el correcto funcionamiento del *pipeline*, así como para facilitar la solución de posibles errores que puedan surgir, se añade un módulo *Logger*, que añadirá al log de la ejecución el resultado de `#[vars.palabra]`:

Figura 10: Configuración de *Logger* para SOAP Interno

- El siguiente módulo, el de *Request* es el más importante de toda la *pipeline*. Para la configuración de este:

- Se ha añadido un nuevo elemento del tipo *HTTP\_Request\_configuration* con los detalles del servidor Flask que está *escuchando*:

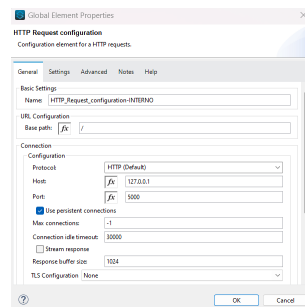


Figura 11: Configuración de *HTTP\_Request\_configuration-INTERNO* para SOAP Interno

- El *path* se ha definido otro *URI* param, */soap/palabras/{palabra}*. En este caso coincide con el nombre de la variable que se ha utilizado en *Set Variable*, pero puede no hacerlo.<sup>4</sup> La configuración de este parámetro *URI* es necesario hacerla en la pestaña al efecto:

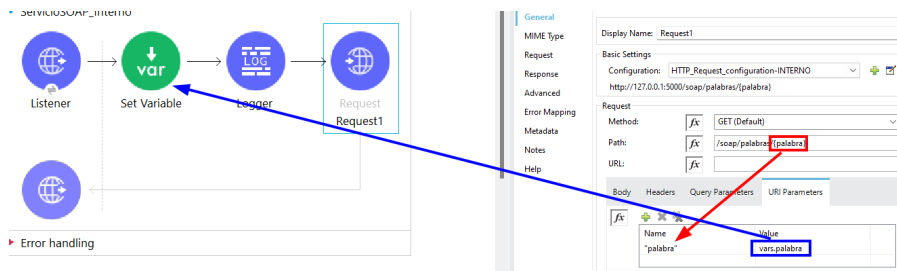


Figura 12: Detalle de configuración *URI Params* en el *Request* de SOAP Interno

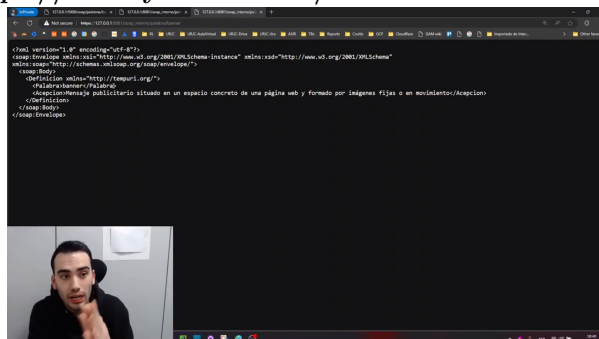
### Configuración en *backend* Flask

El código es consultable en la sección 2.4.2 (*De aplicativo desarrollado en Flask*).

### Acceso al vídeo explicativo

Se puede acceder al vídeo explicativo desde el siguiente enlace:

<https://www.youtube.com/watch?v=saPd7OvC3Ag>



<sup>4</sup>De hecho, en la implementación de otros casos se ha cambiado para comprobar que sigue funcionando

### 2.2.2. SOAP Externo

Para esta sección se usará el servicio de calculadora, indicado por el profesor.

#### Configuración en AnyPoint Studio

Para el servicio SOAP externo se ha generado el siguiente *pipeline*:

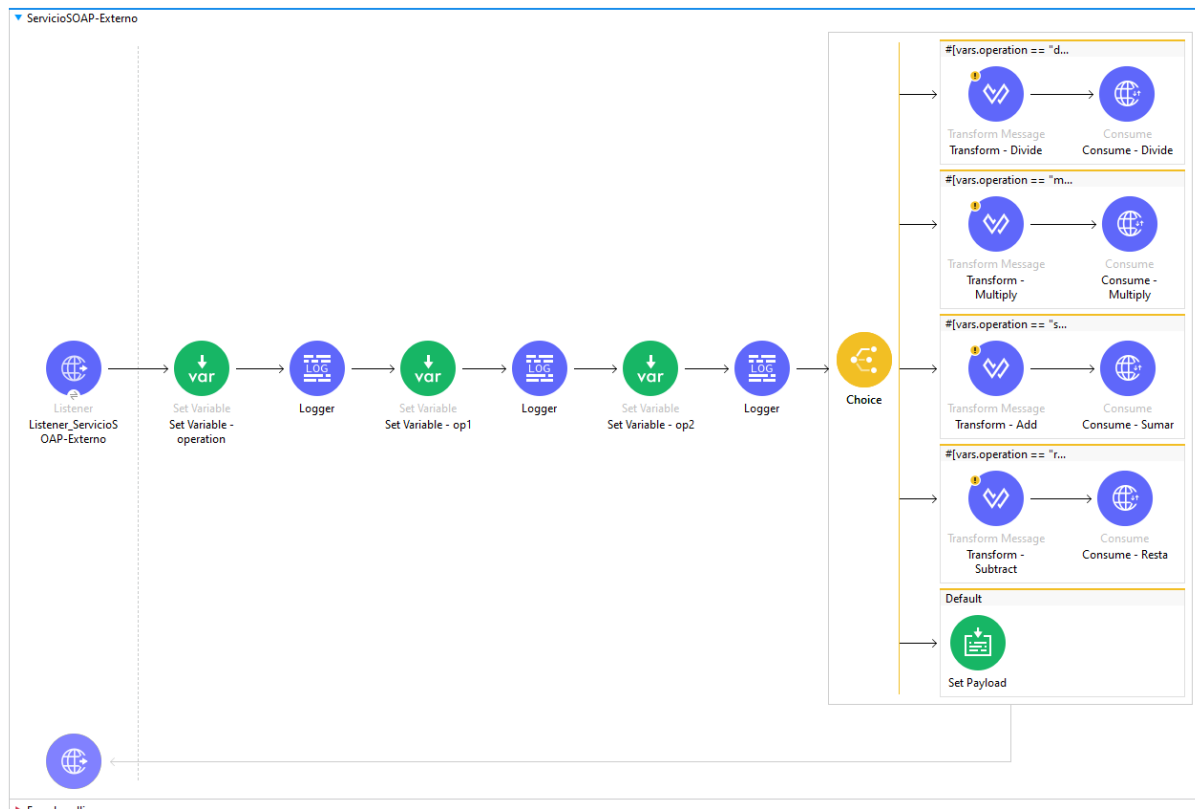


Figura 13: Pipeline completo para *SOAP Externo*

Para la creación de cada uno de los elementos que lo componen se han seguido los siguientes pasos:

1. Añadir el *Listener* al área de trabajo, arrastrándolo desde el *Mule Palette*:

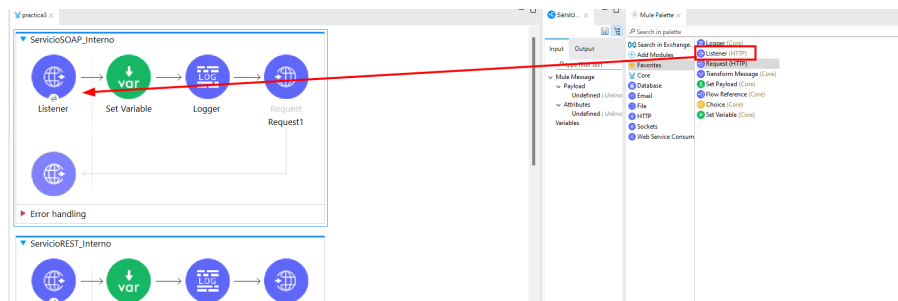


Figura 14: Adición de *Listener* al área de trabajo

2. La configuración del parámetro (además de lo ya explicado anteriormente en lo relativo al HTTPS) es sencilla. La única variable que se ha de editar es el *Path*:  
`/soap_externo`

3. Se han definido 3 variables, cada una con su relativo *Logger* a continuación del mismo, para permitir registrar los valores que toman las variables (facilitando el posterior *troubleshooting*):

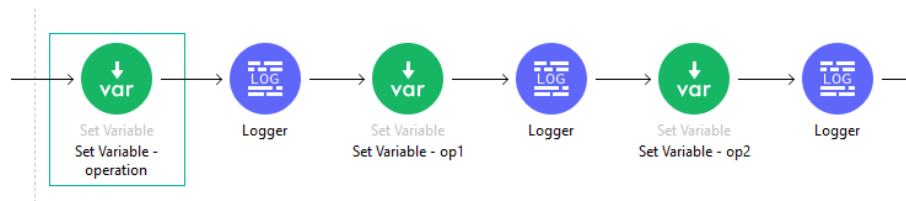


Figura 15: Parejas de *Variable* + *Logger* en el pipeline de *SOAP Externo*

Todos los módulos *Logger* están configurados para añadir registrar la información de módulo *Set variable* que le precede. De esta forma, la configuración del *Message* es (en orden de aparición en el *pipeline*):

- a) `Operation: #[vars.operation]`
- b) `operando1: #[vars.op1]`
- c) `operando2: #[vars.op2]`

Los módulos *Set Variable* están configurados de la siguiente manera (clave-valor):

- a) `operation = #[attributes.queryParams.operation]`
- b) `op1 = #[attributes.queryParams.op1]`
- c) `op2 = #[attributes.queryParams.op2]`

4. A continuación, tenemos un *CHOICE* que, en función del valor de la variable *operation*, se ejecutará una u otra petición. Para cada opción hay una condición. Las condiciones son:

- a) `vars.operation == "dividir"`
- b) `vars.operation == "multiplicar"`
- c) `vars.operation == "sumar"`
- d) `vars.operation == "restar"`
- e) Opción por defecto (`else`)

Como se puede extraer de los posibles valores de variable anteriores, no es necesario que coincidan literalmente con la operación que luego se enviará al servicio SOAP externo.

5. Cada uno de los *choice* ejecuta 2 módulos: un *Transform Message* y un *Consume*.

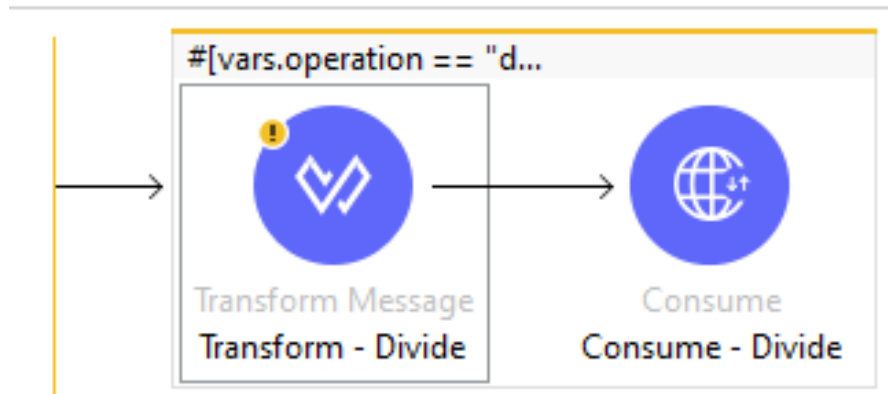


Figura 16: Contenido de cada uno de los *ifs* del *Choice* en el *SOAP Externo*

Este módulo, el *Transform Message*, tiene una configuración distinta para cada opción:

a) Dividir

```

1 %dw 2.0
2 output application/xml
3 ns ns0 http://tempuri.org/
4 ---
5 {
6   ns0#Divide: {
7     ns0#intA: vars.op2 as Number,
8     ns0#intB: vars.op1 as Number
9   }
10 }
```

Listing 1: SOAP-Externo - Transform Dividir

b) Multiplicar

```

1 %dw 2.0
2 output application/xml
3 ns ns0 http://tempuri.org/
4 ---
5 {
6   ns0#Multiply: {
7     ns0#intA: vars.op2 as Number,
8     ns0#intB: vars.op1 as Number
9   }
10 }
```

Listing 2: SOAP-Externo - Transform Multiplicar

c) Sumar

```

1 %dw 2.0
2 output application/xml
3 ns ns0 http://tempuri.org/
4 ---
5 {
6   ns0#Add: {
7     ns0#intA: vars.op2 as Number,
8     ns0#intB: vars.op1 as Number
9   }
10 }
```

```
10 }
```

Listing 3: SOAP-Externo - Transform Sumar

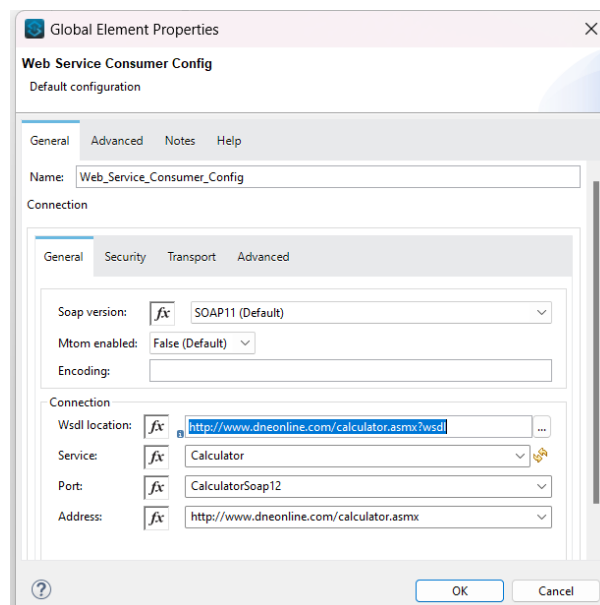
d) restar

```
1 %dw 2.0
2 output application/xml
3 ns ns0 http://tempuri.org/
4 ---
5 {
6   ns0#Subtract: {
7     ns0#intA: vars.op2 as Number,
8     ns0#intB: vars.op1 as Number
9   }
10 }
```

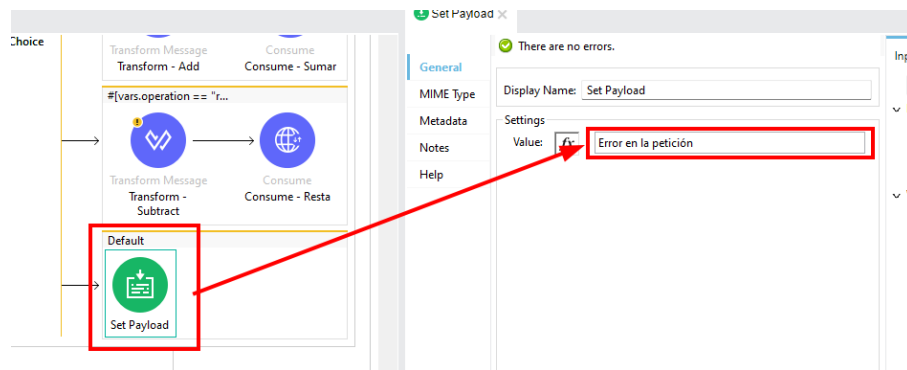
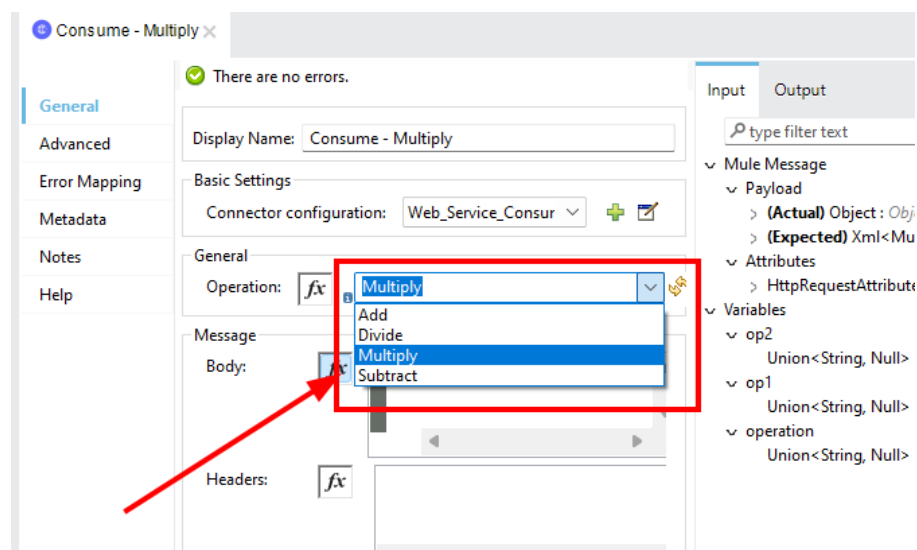
Listing 4: SOAP-Externo - Transform Restar

En el módulo *Consumo* se ha añadido en la opción *Connector configuration* el WSDL proporcionado por el servicio:

<http://www.dneonline.com/calculator.asmx?wsdl>

Figura 17: Configuración del módulo *Consume* para el SOAP Externo

Una vez añadido, basta con seleccionar la operación debida en el selector:

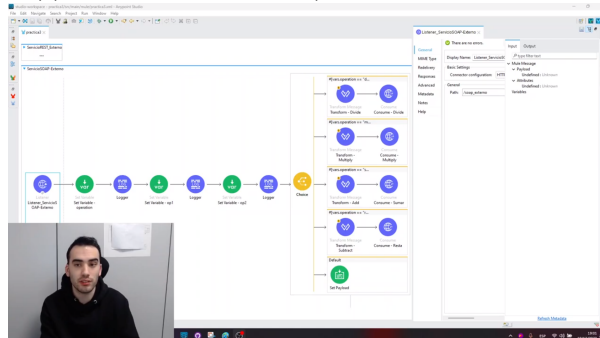
Figura 19: *Default response* en SOAP ExternoFigura 18: Selector de operación en el módulo *Consume* del SOAP Externo

6. Por último, si no ha coincidido con ninguna de las opciones predefinidas (dividir, multiplicar, suma o restar), en lugar de hacer ninguna petición, se devuelve como respuesta un mensaje de error:

### Acceso al vídeo explicativo

Se puede acceder al vídeo explicativo desde el siguiente enlace:

<https://www.youtube.com/watch?v=EEmueHup-b8>



## 2.3. Servicios REST

### 2.3.1. REST Interno

Para el servicio REST Interno se ha usado *Flask* (Python) para simular una serie de entradas en el diccionario.

#### Configuración en AnyPoint Studio

En AnyPoint Studio el *pipeline* para esta demostración es el siguiente:

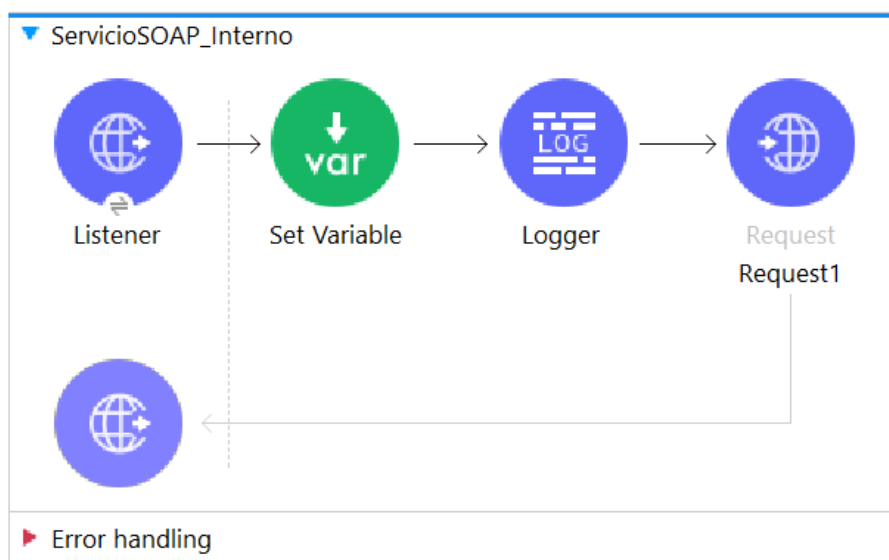


Figura 20: Pipeline del Servicio REST Interno

Para la creación de cada uno de los elementos que lo componen se han seguido los siguientes pasos:

1. Añadir el *Listener* al área de trabajo, arrastrándolo desde el *Mule Palette*:

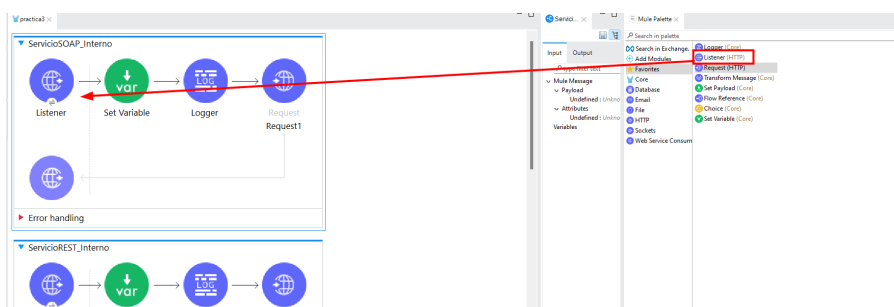
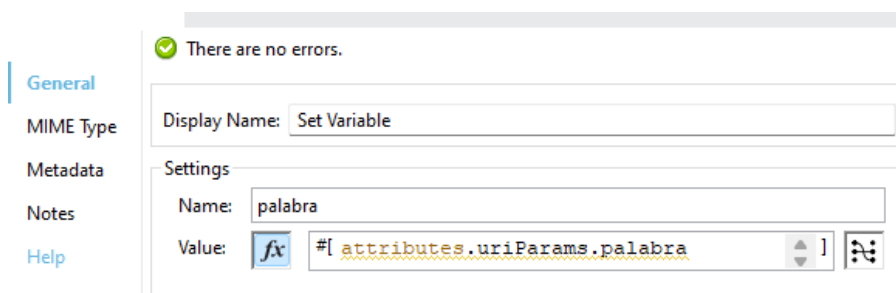
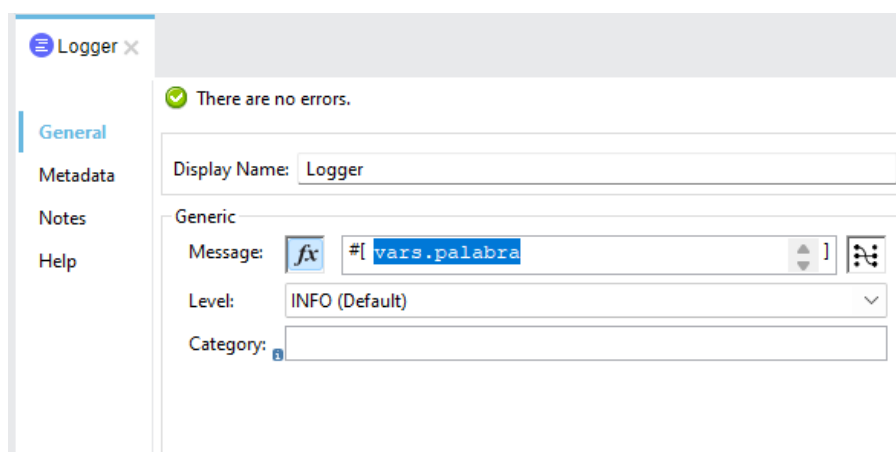


Figura 21: Adición de *Listener* al área de trabajo

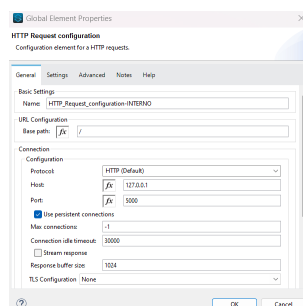
2. La configuración del parámetro (además de lo ya explicado anteriormente en lo relativo al HTTPS) es sencilla. La única variable que se ha de editar es el *Path*:  
`/rest_interno/palabra/{palabra}`
3. Se usará el parámetro de URI *palabra* para identificar qué es lo que se quiere solicitar. Para hacerlo, hemos de fijar la variable en el código durante la ejecución usando el módulo *Set Variable*:

Figura 22: Configuración de *Set Variable* para REST Interno

4. Para comprobar el correcto funcionamiento del *pipeline*, así como para facilitar la solución de posibles errores que puedan surgir, se añade un módulo *Logger*, que añadirá al log de la ejecución el resultado de `#[vars.palabra]`:

Figura 23: Configuración de *Logger* para REST Interno

5. El siguiente módulo, el de *Request* es el más importante de toda la *pipeline*. Para la configuración de este:
- Se ha añadido un nuevo elemento del tipo *HTTP\_Request\_configuration* con los detalles del servidor Flask que está *escuchando*:

Figura 24: Configuración de *HTTP\_Request\_configuration-INTERNO* para REST Interno

- El *path* se ha definido otro *URI* param, `/rest/palabras/{palabra}`. En este caso coincide con el nombre de la variable que se ha utilizado en *Set Variable*, pero puede no hacerlo.<sup>5</sup> La

<sup>5</sup>De hecho, en la implementación de otros casos se ha cambiado para comprobar que sigue funcionando

configuración de este parámetro URI es necesario hacerla en la pestaña al efecto:

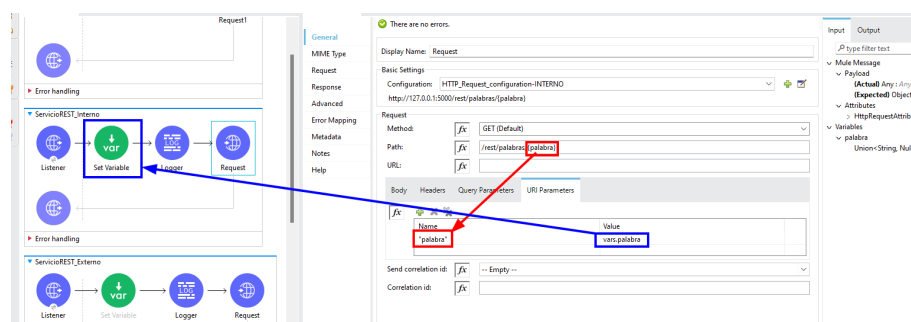


Figura 25: Detalle de configuración *URI Params* en el *Request* de REST Interno

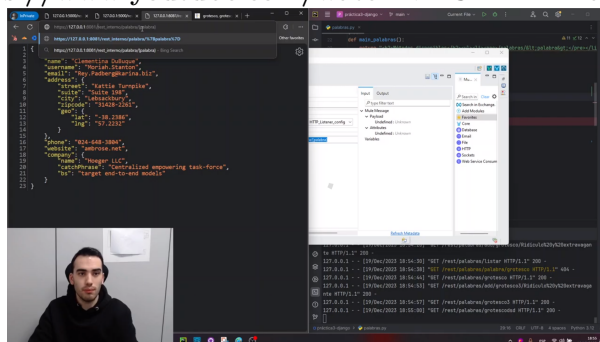
### Configuración en *backend* Flask

El código es consultable en la sección 2.4.2 (*De aplicativo desarrollado en Flask*).

### Acceso al vídeo explicativo

Se puede acceder al vídeo explicativo desde el siguiente enlace:

<https://www.youtube.com/watch?v=G1P2m1NaCWI>



### 2.3.2. REST Externo

Se usará en esta sección el servicio de *jsonplaceholder.typicode.com*, indicado por el profesor de la asignatura.

### Configuración en AnyPoint Studio

#### Configuración en AnyPoint Studio

En AnyPoint Studio el *pipeline* para esta demostración es el siguiente:

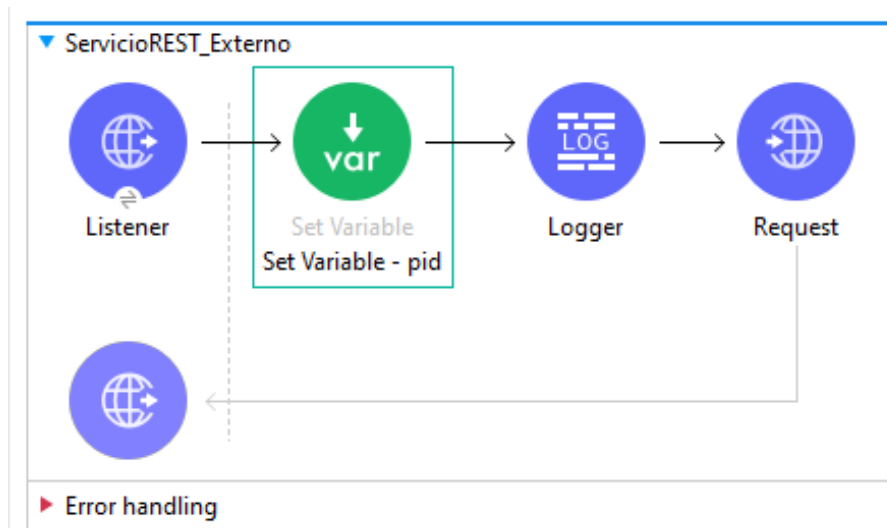
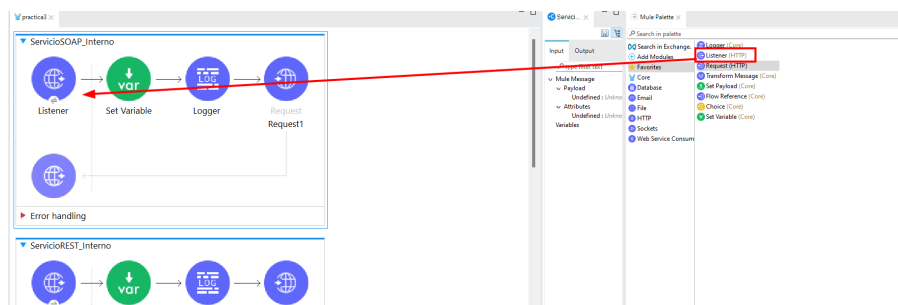


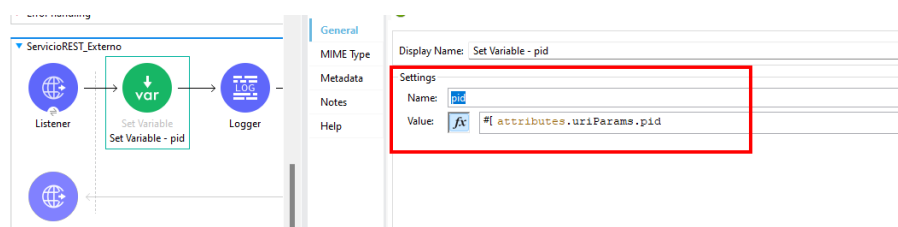
Figura 26: Pipeline del Servicio REST Externo

Para la creación de cada uno de los elementos que lo componen se han seguido los siguientes pasos:

1. Añadir el *Listener* al área de trabajo, arrastrándolo desde el *Mule Palette*:

Figura 27: Adición de *Listener* al área de trabajo en REST Externo

2. La configuración del parámetro (además de lo ya explicado anteriormente en lo relativo al HTTPS) es sencilla. La única variable que se ha de editar es el *Path*:  
`/rest_externo/person/{pid}`
3. Se usará el parámetro de URI *pid* para identificar qué es lo que se quiere solicitar. Para hacerlo, hemos de fijar la variable en el código durante la ejecución usando el módulo *Set Variable*:

Figura 28: Configuración de *Set Variable* para REST Externo

4. Para comprobar el correcto funcionamiento del *pipeline*, así como para facilitar la solución de posibles errores que puedan surgir, se añade un módulo *Logger*, que añadirá al log de la ejecución

el resultado de `#[pid: #[vars.pid]]`:

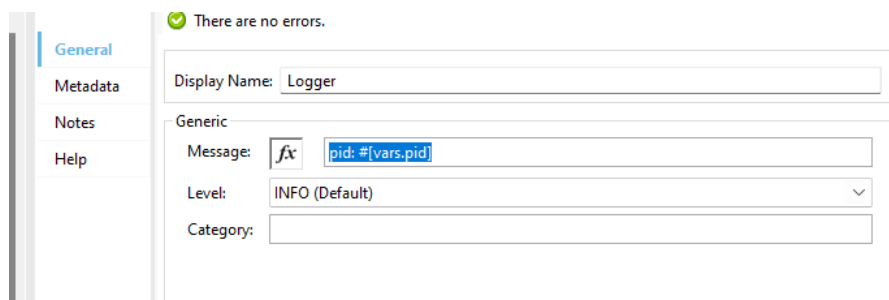


Figura 29: Configuración de *Logger* para REST EXTERNO

5. El siguiente módulo, el de *Request* es el más importante de toda la *pipeline*. Para la configuración de este:

- Se ha añadido un nuevo elemento del tipo *HTTP\_Request\_configuration* con los detalles del servidor externo que responderá a la petición:

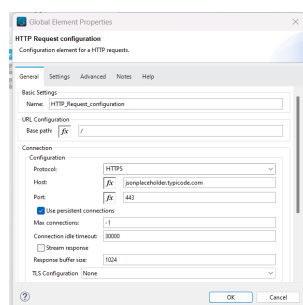


Figura 30: Configuración de *HTTP\_Request\_configuration-INTERNO* para SOAP Interno

- El *path* se ha definido otro *URI param*, `/users/{user}`. En este caso NO coincide con el nombre de la variable que se ha utilizado en *Set Variable*, pero puede no hacerlo (como se ha visto en otros ejemplos). La configuración de este parámetro URI es necesario hacerla en la pestaña al efecto:

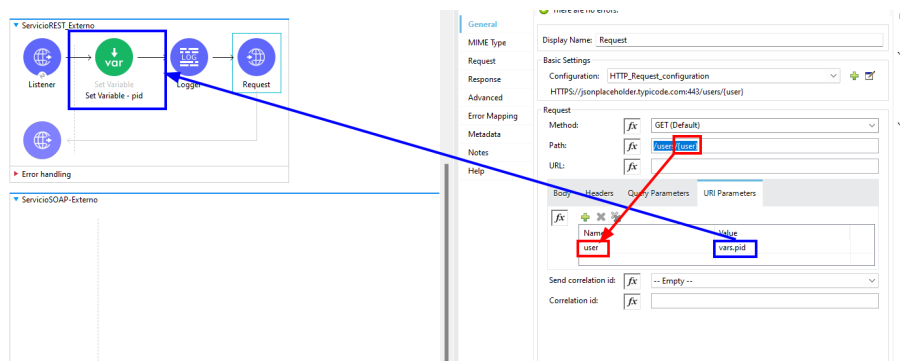
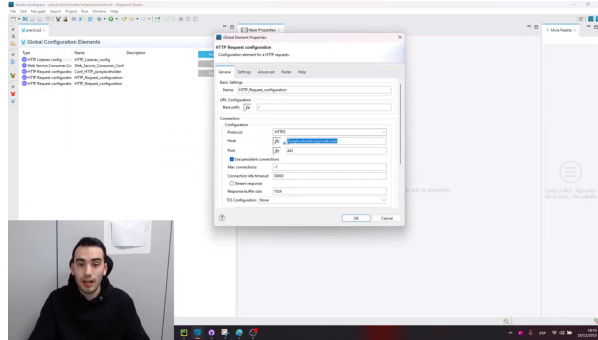


Figura 31: Detalle de configuración *URI Params* en el *Request* de REST Externo

### Acceso al vídeo explicativo

Se puede acceder al vídeo explicativo desde el siguiente enlace:

<https://www.youtube.com/watch?v=YqBN63LUNQs>



## 2.4. Código

### 2.4.1. De proyecto en AnyPoint Studio

```

1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <mule xmlns:tracking="http://www.mulesoft.org/schema/mule/ee/tracking" xmlns:file="http://www.
4   mulesoft.org/schema/mule/file"
5   xmlns:ee="http://www.mulesoft.org/schema/mule/ee/core"
6   xmlns:chatgpt="http://www.mulesoft.org/schema/mule/chatgpt" xmlns:wsc="http://www.mulesoft.
7   org/schema/mule/wsc" xmlns:tls="http://www.mulesoft.org/schema/mule/tls" xmlns:http="http:
8   //www.mulesoft.org/schema/mule/http" xmlns="http://www.mulesoft.org/schema/mule/core"
9   xmlns:doc="http://www.mulesoft.org/schema/mule/documentation" xmlns:xsi="http://www.w3.org
10  /2001/XMLSchema-instance" xsi:schemaLocation="http://www.mulesoft.org/schema/mule/core
11  http://www.mulesoft.org/schema/mule/core/current/mule.xsd
12  http://www.mulesoft.org/schema/mule/http http://www.mulesoft.org/schema/mule/http/current/mule-
13  http.xsd
14  http://www.mulesoft.org/schema/mule/tls http://www.mulesoft.org/schema/mule/tls/current/mule-
15  tls.xsd
16  http://www.mulesoft.org/schema/mule/wsc http://www.mulesoft.org/schema/mule/wsc/current/mule-
17  wsc.xsd
18  http://www.mulesoft.org/schema/mule/chatgpt http://www.mulesoft.org/schema/mule/chatgpt/
19  current/mule-chatgpt.xsd
20  http://www.mulesoft.org/schema/mule/ee/core http://www.mulesoft.org/schema/mule/ee/core/
21  current/mule-ee.xsd
22  http://www.mulesoft.org/schema/mule/file http://www.mulesoft.org/schema/mule/file/current/mule-
23  file.xsd
24  http://www.mulesoft.org/schema/mule/ee/tracking http://www.mulesoft.org/schema/mule/ee/
25  tracking/current/mule-tracking-ee.xsd">
26   <http:listener-config name="HTTP_Listener_config" doc:name="HTTP Listener config" doc:id="
27     509957d4-0a5e-46e1-a391-e311648616db" >
28     <http:listener-connection protocol="HTTPS" host="localhost" port="8081" >
29       <tls:context >
30         <tls:key-store type="jks" path="keystore.jks" alias="key-alias" keyPassword="12345678"
31         password="12345678" />
32       </tls:context>
33     </http:listener-connection>
34   </http:listener-config>
35   <wsc:config name="Web_Service_Consumer_Config" doc:name="Web Service Consumer Config" doc:id=
36     "5cc28f55-bb07-4568-8319-4d9fdcb76fe" >
37     <wsc:connection wsdlLocation="http://www.dneonline.com/calculator.asmx?wsdl" service="
38     Calculator" port="CalculatorSoap12" address="http://www.dneonline.com/calculator.asmx" />
39   </wsc:config>
40   <http:request-config name="Conf_HTTP_jsonplaceholder" doc:name="HTTP Request configuration"
41   doc:id="f74c3cdb-0830-42ac-8cee-6a0a520ee5ea" >
42     <http:request-connection protocol="HTTPS" host="jsonplaceholder.typicode.com" port="443" /

```

```

    >
25 </http:request-config>
26 <http:request-config name="HTTP_Request_configuration-INTERNO" doc:name="HTTP Request
    configuration" doc:id="22ab49ac-96fe-41bd-a043-e11680928310" >
27   <http:request-connection host="127.0.0.1" port="5000" />
28 </http:request-config>
29 <http:request-config name="HTTP_Request_configuration" doc:name="HTTP Request configuration"
    doc:id="ed7d3d69-3544-4838-a527-eca154ffb1f4" >
30   <http:request-connection protocol="HTTPS" host="jsonplaceholder.typicode.com" port="443" /
    >
31 </http:request-config>
32 <flow name="ServicioSOAP_Interno" doc:id="4e0a43da-c1ab-49f4-b8d7-9b1f9eda834a" >
33   <http:listener doc:name="Listener" doc:id="e162dd14-e91e-4a5e-ac6c-4a88d87e220a" config-
    ref="HTTP_Listener_config" path="/soap_interno/palabra/{palabra}"/>
34   <set-variable value="#[attributes.uriParams.palabra]" doc:name="Set Variable" doc:id="491
    be4d0-6d11-4569-82ba-3d1c0f452e12" variableName="palabra"/>
35   <logger level="INFO" doc:name="Logger" doc:id="f0b9f604-0c9f-4ebe-a7f3-b2d7e916baad"
    message="#[vars.palabra]"/>
36   <http:request method="GET" doc:name="Request1" doc:id="b82ea9df-8ee8-4779-8be9-669
    bd72afa3a" config-ref="HTTP_Request_configuration-INTERNO" path="/soap/palabras/{palabra}"
    outputMimeType="text/plain">
37     <http:uri-params><![CDATA#[output application/java
---
38 {
39   "palabra" : vars.palabra
40 }]]]></http:uri-params>
41   </http:request>
42 </flow>
43 <flow name="ServicioREST_Interno" doc:id="95411434-b973-4ac5-b6a2-ed30702bb84a">
44   <http:listener doc:name="Listener" doc:id="0cfa35b4-ab5e-415d-b173-9d7d519ea493" config-
    ref="HTTP_Listener_config" path="/rest_interno/palabra/{palabra}"/>
45   <set-variable value="#[attributes.uriParams.palabra]" doc:name="Set Variable" doc:id="
    c30ffe15-e8e3-435b-9eb5-86a89dd18458" variableName="palabra"/>
46   <logger level="INFO" doc:name="Logger" doc:id="f5d43108-007e-43c8-8291-125737bd309f"
    message="#[vars.palabra]"/>
47   <http:request method="GET" doc:name="Request" doc:id="8b2bb9b6-b47f-4308-a637-516c1c3f11a9
    " config-ref="HTTP_Request_configuration-INTERNO" path="/rest/palabras/{palabra}">
48     <http:uri-params ><![CDATA#[output application/java
---
49 {
50   "palabra" : vars.palabra
51 }]]]></http:uri-params>
52   </http:request>
53 </flow>
54 <flow name="ServicioREST_Externo" doc:id="372c20eb-ac90-44cc-8fd8-995573d5a69e" >
55   <http:listener doc:name="Listener" doc:id="4df9c025-61ac-41b8-9fba-0a4ed782fa33" config-
    ref="HTTP_Listener_config" path="/rest_externo/person/{pid}"/>
56   <set-variable value="#[attributes.uriParams.pid]" doc:name="Set Variable - pid" doc:id="8
    aa67433-88d5-47a5-abb6-a332c265d233" variableName="pid"/>
57   <logger level="INFO" doc:name="Logger" doc:id="f71a7d2e-24e5-4929-832f-301fc7f599a6"
    message="pid: #[vars.pid]"/>
58   <http:request method="GET" doc:name="Request" doc:id="4d5b9d94-3408-4814-8c1a-ffc158e944ad
    " config-ref="HTTP_Request_configuration" path="/users/{user}">
59     <http:uri-params ><![CDATA#[output application/java
---
60 {
61   user : vars.pid
62 }]]]></http:uri-params>
63   </http:request>
64 </flow>
65 </http:request>
66

```

```

67 </flow>
68 <flow name="ServicioSOAP-Externo" doc:id="2eeac828-68e5-4590-b7a3-94dd7acd47cb"
    tracking:enable-default-events="true">
69     <http:listener doc:name="Listener_ServicioSOAP-Externo" doc:id="6ba899d5-66ae-4947-9a88-
        c4bc3be0a8f4" config-ref="HTTP_Listener_config" path="/soap_externo"/>
70     <set-variable value="#[attributes.queryParams.operation]" doc:name="Set Variable -
        operation" doc:id="9b141743-d371-4e00-83f3-912804a514b3" variableName="operation"/>
71     <logger level="INFO" doc:name="Logger" doc:id="fa61a5f9-58bf-415d-a69f-4dd4a22ee419"
        message="Operation: #[vars.operation]"/>
72     <set-variable value="#[attributes.queryParams.op1]" doc:name="Set Variable - op1" doc:id="
        45dc7757-0835-410e-bfb5-f975d4960b57" variableName="op1"/>
73     <logger level="INFO" doc:name="Logger" doc:id="8208da29-fe7e-4a26-9ade-7ee10e6f7ae9"
        message="operando1: #[vars.op1]"/>
74     <set-variable value="#[attributes.queryParams.op2]" doc:name="Set Variable - op2" doc:id="
        31b43588-08ca-4844-922c-5c39443f5393" variableName="op2"/>
75     <logger level="INFO" doc:name="Logger" doc:id="ada46450-08df-4926-9725-f33984b3398f"
        message="operando2: #[vars.op2]"/>
76     <choice doc:name="Choice" doc:id="ddeea3c5-8137-4484-88a5-84da87e296ce" tracking:enable-
        default-events="true">
77         <when expression="#[vars.operation == 'dividir']">
78             <ee:transform doc:name="Transform - Divide" doc:id="ea893f5c-3316-424e-b368-
                b3fdd566bd4c">
79                 <ee:message>
80                     <ee:set-payload><![CDATA[%dw 2.0
81 output application/xml
82 ns ns0 http://tempuri.org/
83 ---
84 {
85     ns0#Divide: {
86         ns0#intA: vars.op2 as Number,
87         ns0#intB: vars.op1 as Number
88     }
89 }]]></ee:set-payload>
90         </ee:message>
91         </ee:transform>
92         <wsc:consume operation="Divide" doc:name="Consume - Divide" doc:id="ded94a10-c73e-45f0-
            abbb-87098d2a13b4" config-ref="Web_Service_Consumer_Config" />
93         </when>
94         <when expression="#[vars.operation == 'multiplicar']">
95             <ee:transform doc:name="Transform - Multiply" doc:id="d03c4537-0bf5-48bc-bd06-30120
                b0ecb53">
96                 <ee:message>
97                     <ee:set-payload><![CDATA[%dw 2.0
98 output application/xml
99 ns ns0 http://tempuri.org/
100 ---
101 {
102     ns0#Multiply: {
103         ns0#intA: vars.op2 as Number,
104         ns0#intB: vars.op1 as Number
105     }
106 }]]></ee:set-payload>
107             </ee:message>
108             </ee:transform>
109             <wsc:consume operation="Multiply" doc:name="Consume - Multiply" doc:id="cbf5bcd1-d1fc-
                4843-be0c-6eff64a46953" config-ref="Web_Service_Consumer_Config" />
110             </when>
111             <when expression="#[vars.operation == 'sumar']">
112                 <ee:transform doc:name="Transform - Add" doc:id="8cb48fd1-7963-49cd-912a-e184ed720ad6"

```

```

>
113     <ee:message>
114         <ee:set-payload><![CDATA[%dw 2.0
115 output application/xml
116 ns ns0 http://tempuri.org/
117 ---
118 {
119     ns0#Add: {
120         ns0#intA: vars.op2 as Number,
121         ns0#intB: vars.op1 as Number
122     }
123 }]]></ee:set-payload>
124     </ee:message>
125     </ee:transform>
126     <wsc:consume operation="Add" doc:name="Consume - Sumar" doc:id="bcba9c19-4c6a-4b28-
127 ace6-233a2dfe9eb4" config-ref="Web_Service_Consumer_Config" />
128     </when>
129     <when expression="#[vars.operation == "restar"]">
130         <ee:transform doc:name="Transform - Subtract" doc:id="4d903904-aa15-4f25-be81-1
131 d539f33a3b9">
132             <ee:message>
133                 <ee:set-payload><![CDATA[%dw 2.0
134 output application/xml
135 ns ns0 http://tempuri.org/
136 ---
137 {
138     ns0#Subtract: {
139         ns0#intA: vars.op2 as Number,
140         ns0#intB: vars.op1 as Number
141     }
142 }]]></ee:set-payload>
143             </ee:message>
144             </ee:transform>
145             <wsc:consume doc:name="Consume - Resta" doc:id="9f65bed0-6fee-44d1-8436-9dd5f2fef69e"
146 config-ref="Web_Service_Consumer_Config" operation="Subtract"/>
147             </when>
148             <otherwise >
149                 <set-payload value="Error en la petición" doc:name="Set Payload" doc:id="d68cae3b
150 -1179-43c1-ae69-8f8b9ec66133" />
151             </otherwise>
152         </choice>
153     </flow>
154 </mule>

```

Listing 5: practica3.xml - Código de flujos en AnyPoint Studio

#### 2.4.2. De aplicativo desarrollado en Flask

```

1 from flask import Flask, make_response
2
3 app = Flask(__name__)
4
5 palabras = {"perreo": "Baile que se ejecuta generalmente a ritmo de reguetón, con eróticos
6 movimientos de caderas, y en el que, cuando se baila por parejas, el hombre se coloca
habitualmente detrás de la mujer con los cuerpos muy juntos",
7
8 "parkour": "Actividad deportiva desarrollada al aire libre, principalmente en
entornos urbanos, que consiste en ir salvando los obstáculos que se encuentran en el
camino, como diferencias de altura, escaleras, bancos, etc., mediante saltos, volteretas u
otros movimientos.",
9
10 }

```

```

7         "banner": "Mensaje publicitario situado en un espacio concreto de una página web y
        formado por imágenes fijas o en movimiento",
8         "biometano": "Gas compuesto mayoritariamente de metano, obtenido mediante un
        proceso de depuración del biogás.",
9         "cookie": "Pequeño archivo de texto enviado por un sitio web y almacenado en el
        navegador del usuario, cuyas actividades y preferencias captura. U. menos c. m.",
10        "escriturista": "Persona especializada en el conocimiento de los libros sagrados,
        o que profesa sus enseñanzas."}
11
12
13 @app.route('/')
14 def inicio():
15     return ('<h2>Bienvenido al programa que te mostrará las nuevas palabras de la RAE</h2>'
16            '<h3>Protocolos disponibles</h3>'
17            '<ul><li><pre>/soap</pre></li>'
18            '<li><pre>/rest</pre></li></ul>')
19
20
21 @app.route("/rest/palabras")
22 def main_palabras():
23     return "<h2>Métodos disponibles</h2><ul><li><pre>/palabras/&lt;palabra&gt;</pre></li><li><pre>/palabras/listar</pre></li><li><pre>/palabras/comprobar/&lt;palabra&gt;</pre></li></ul>"
24
25
26 @app.route("/rest/palabras/add/<palabra>/<definicion>")
27 def add_palabra(palabra, definicion):
28     palabras[palabra] = definicion
29     return "OK: " + palabra
30
31
32 @app.route("/rest/palabras/<palabra>")
33 def mostrar_palabra(palabra):
34     if check_palabra(palabra) == "true":
35         return palabras.get(palabra)
36     else:
37         return "Palabra inexistente en base de datos"
38
39
40 @app.route("/rest/palabras/listar")
41 def listar_palabra():
42
43     cadena = ""
44     for palabra in palabras.keys():
45         cadena += palabra
46         cadena += "\n<br/>"
47
48     return str(cadena)
49
50
51 @app.route("/rest/soap/palabras/comprobar/<palabra>")
52 def check_palabra(palabra):
53
54     if palabra in palabras:
55         return "true"
56     else:
57         return "false"
58
59

```

```
60 soap_start = ('<?xml version="1.0" encoding="utf-8"?>\n<soap:Envelope xmlns:xsi="http://www.w3\n
61 .org/2001/XMLSchema-\n
62 'instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"\nxmlns:soap="http://\n
schemas.xmlsoap.org/soap/'\n
63 'envelope/">\n <soap:Body>\n <Definicion xmlns="http://tempuri.org/">\n\n
64 <Palabra>')\n
65 soap_middle = '</Palabra>\n <Acepcion>'\n
66 soap_end = '</Acepcion>\n </Definicion>\n </soap:Body>\n</soap:Envelope>'\n
67\n
68 @app.route("/soap/palabras/<palabra>")\n
69 def generate_soap_response(palabra):\n
70     if check_palabra(palabra) == "true":\n
71         return soap_start + palabra + soap_middle + palabras.get(palabra) + soap_end\n
72     else:\n
73         return "Palabra inexistente en base de datos"\n
74\n
75\n
76 def mostrar_palabra_soap(palabra):\n
77     response = make_response((generate_soap_response(palabra)), 200)\n
78     response.mimetype = "plain/text"\n
79     return response\n
80\n
81\n
82 if __name__ == '__main__':\n
83     app.run()
```

Listing 6: palabras.py - Código que genera el backend para servicios propios

### 3. Apartado 2: Estudiar las características del Mule ESB como herramienta para implementar servicios, haciendo especial hincapié en el “Management Center”

Hasta este momento, el trabajo se ha basado de forma casi exclusiva en programar/configurar los servicios a bajo nivel. Pero que funcione en las pruebas que hacemos en nuestra máquina no es sinónimo de que vaya a funcionar una vez desplaguemos. En un despliegue *real* se han de tener en cuenta muchas más cuestiones, por ejemplo:

- El número de usuarios, desde el punto de vista de la escalabilidad. No es lo mismo desplegar para 10 que 10.000. La capacidad que hemos de asignar al servicio será diferente.
- Si bien la cuestión de la escalabilidad es un mundo, requiere conocer conceptos como escalabilidad horizontal, escalabilidad vertical, coordinación de procesos distribuidos, conocer el modelo de capas para dividir nuestra aplicación... Nunca es *desarrollo y ya*.
- Además del número, hemos de saber dónde están nuestros usuarios. Si nuestra base de usuarios está en Europa, desplegar físicamente los servicios en Australia de poco nos servirá, y el impacto sobre el servicio será importante. En la siguiente captura se puede ver la diferencia de latencia desde Madrid (mi equipo, concretamente) a diferentes centros de datos de Microsoft Azure por el mundo:

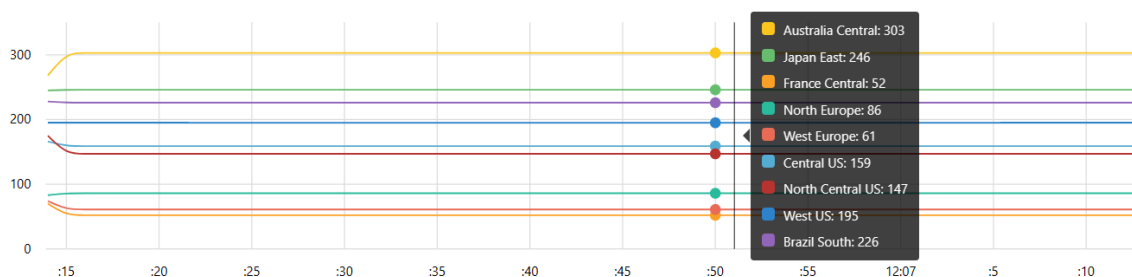


Figura 32: Diferencia de latencias a regiones Azure. Obtenido de AzureSpeed.com

- Una vez conocemos todo lo anterior, es vital definir SLO y alertado para remediar los problemas que puntualmente pueda producirse.
- El alertado se combinará con sistemas para conocer en todo momento cómo está funcionando nuestro sistema.

Todas estas características, y más, están disponibles en las soluciones complementarias que proporciona MuleSoft. En esta sección trataremos las siguientes características, que son las incluidas en el **Management Center**:

- *API Manager*
- *Analytics*
- *RunTime Manager*
- *Access Management*

### 3.1. API Manager

Esta característica está incluida en la suite *AnyPoint Platform*. (MuleSoft, 2023b) En concreto, la funcionalidad *AnyPoint API Manager* permite *operar y escalar un programa de APIs de manera eficiente* (MuleSoft, 2023c)

Permite definir políticas, a nivel de organización y/o grupo, de forma que las características y la seguridad de todos los servicios que nuestra organización expone. En la imagen siguiente, en la que se visualiza la interfaz de *API Manager* se puede ver como existen 2 políticas aplicadas (sección *Applied policies*) y otras 3 que todavía no se aplican al servicio concreto, pero que pueden aplicarse de forma sencilla. Esta reutilización permite mantener la coherencia en todos los servicios desplegados.

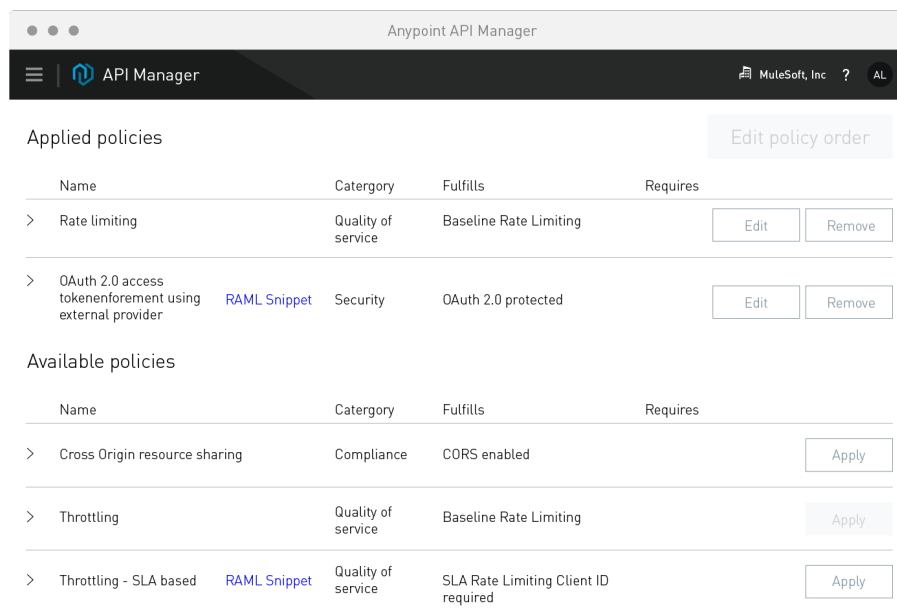
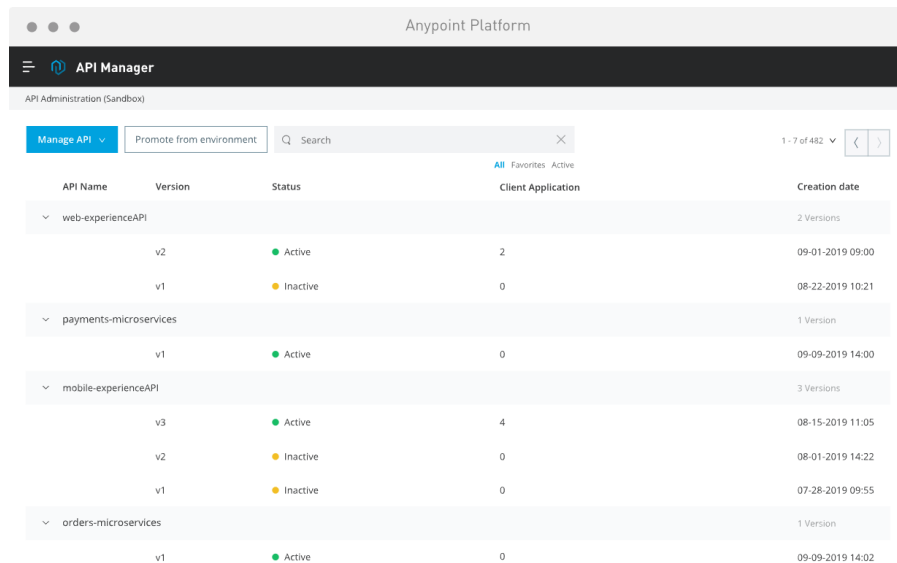


Figura 33: Captura de API Manager. (Extraído de MuleSoft, 2023c)

También permite la federación de identidades con otros proveedores, para simplificar el acceso y la gestión de perfiles en la plataforma. La conexión es posible mediante estándares como SAML y OAuth2, así como con proveedores como Okta, PingFederate y OpenAM, entre otras. (MuleSoft, 2023c)

La siguiente gran ventaja radica en que API Manager permite mantener una *mallá de servicios* (en inglés, *Service Mesh*). Esta característica permite aplicar *una seguridad y una gobernanza uniformes en los microservicios, sin importar dónde se encuentren, desde una única interfaz*. (MuleSoft, 2023c)



The screenshot shows the 'API Manager' interface in the 'Anypoint Platform'. It displays a table of APIs and their versions. The table has columns for API Name, Version, Status, Client Application, and Creation date. The APIs listed are 'web-experienceAPI', 'payments-microservices', 'mobile-experienceAPI', and 'orders-microservices'.

| API Name               | Version | Status   | Client Application | Creation date    |
|------------------------|---------|----------|--------------------|------------------|
| web-experienceAPI      | v2      | Active   | 2                  | 09-01-2019 09:00 |
|                        | v1      | Inactive | 0                  | 08-22-2019 10:21 |
| payments-microservices | v1      | Active   | 0                  | 09-09-2019 14:00 |
| mobile-experienceAPI   | v3      | Active   | 4                  | 08-15-2019 11:05 |
|                        | v2      | Inactive | 0                  | 08-01-2019 14:22 |
|                        | v1      | Inactive | 0                  | 07-28-2019 09:55 |
| orders-microservices   | v1      | Active   | 0                  | 09-09-2019 14:02 |

Figura 34: Captura de API Manager, *Service Mesh*. (Extraído de MuleSoft, 2023c)

Por último, que todas las API se encuentren desplegadas en la misma ubicación permite visualizar de forma agrupada las estadísticas y métricas de todas ellas. Esto permite correlar errores entre API relacionadas, lo que puede ser realmente útil a la hora de detectar y solventar problemas.

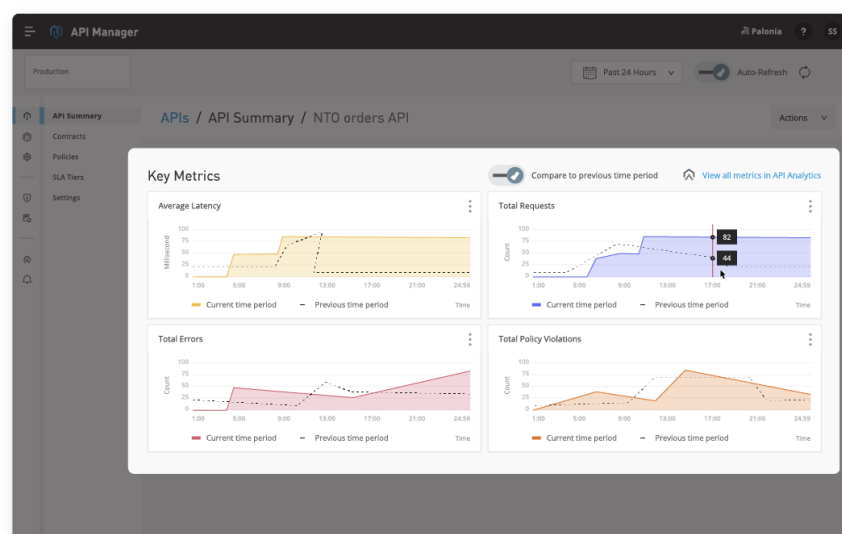


Figura 35: Captura de API Manager, *API Analytics*. (Extraído de MuleSoft, 2023c)

### 3.2. Analytics

Como se ha expuesto anteriormente, es vital poder ver cómo están funcionando los servicios desplegados. Las características de *Anypoint API Analytics* permiten justamente eso: *mejorar la fiabilidad, rendimiento y el valor completo de las API con “insights” críticos*. (MuleSoft, 2023a)

Una de las características vitales es poder analizar en tiempo real el rendimiento de las API desplegadas. Dispone de una serie de gráficos predeterminados que permiten, de forma sencilla, monitorizar errores,

rendimiento, latencias y cumplimiento.

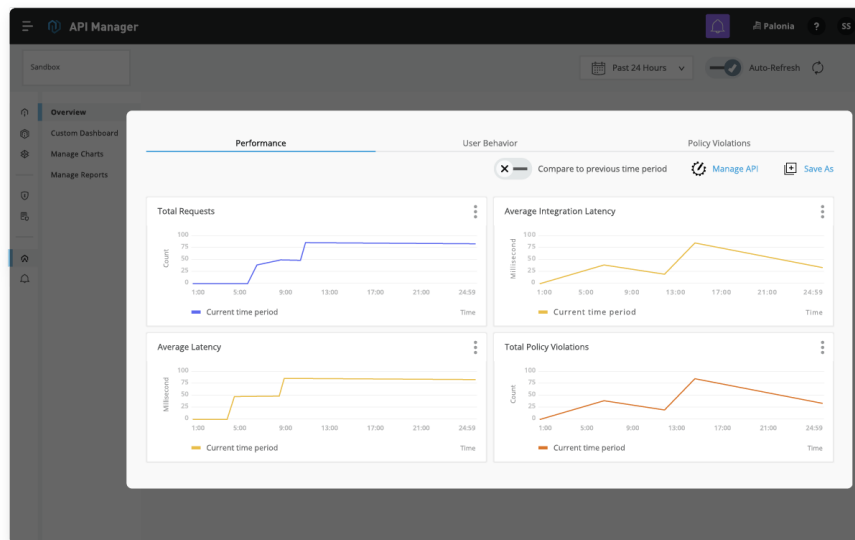


Figura 36: Captura de API Manager, *API Analytics*. (Extraído de MuleSoft, 2023a)

Permite conocer cómo es la interacción de los consumidores y usuarios. Esta información resulta vital tanto a desarrolladores como a gestores, para entender qué necesitan los consumidores a la hora de desarrollar el producto.

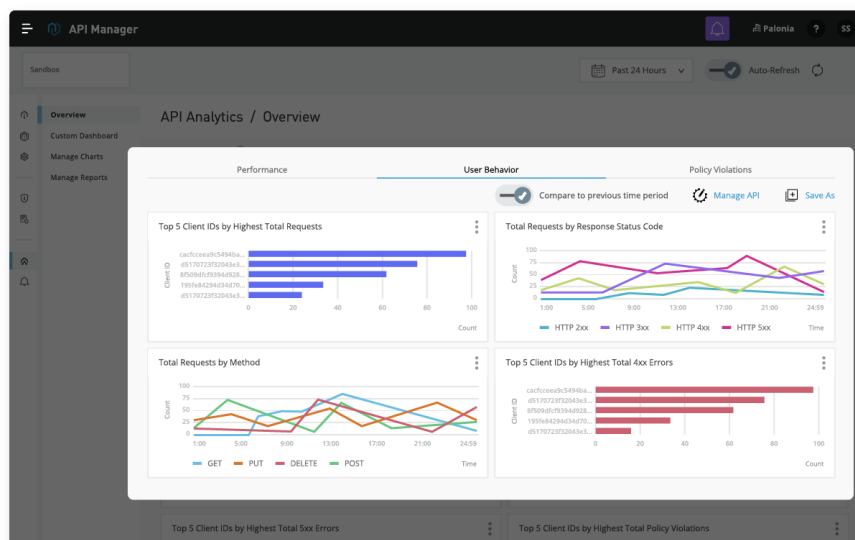


Figura 37: Captura de API Manager, *User Behavior*. (Extraído de MuleSoft, 2023a)

Además de comprender cómo usan las API expuestas los usuarios, permite asegurar el cumplimiento, evitando problemas de calidad y legales. Esta parte es también importante, más aún cuando son varios los equipos los encargados del diseño, implementación y mantenimiento de los servicios.

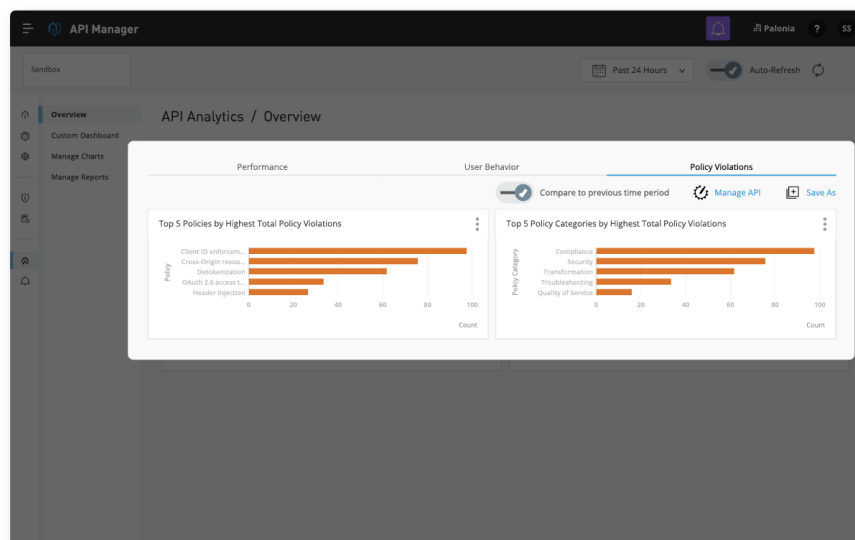


Figura 38: Captura de API Manager, *Compliance*. (Extraído de MuleSoft, 2023a)

Disponer de todas las API en la plataforma de MuleSoft permite identificar tendencias *macro* en todos los ámbitos anteriormente indicados, así como agrupar las mismas en función de datos clave. La toma de decisiones informada es más sencilla con las métricas que AnyPoint Platform ofrece.

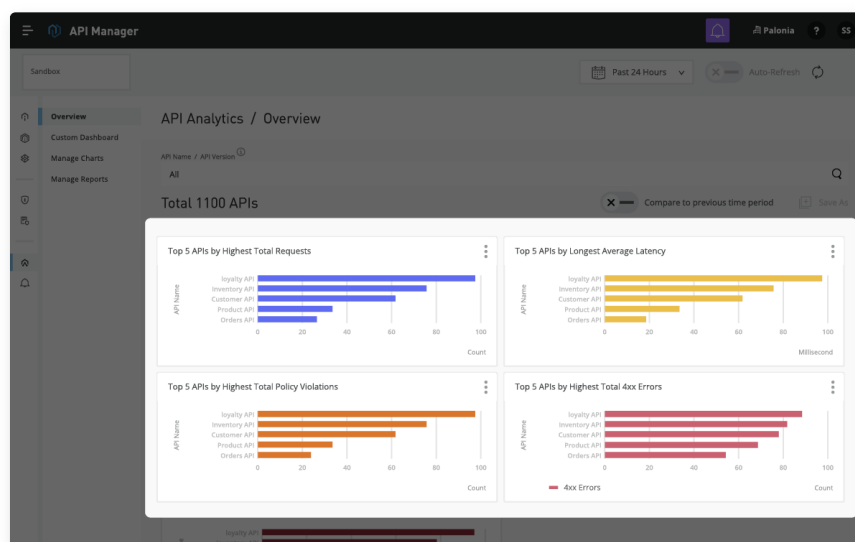


Figura 39: Captura de API Manager, *Key API trends*. (Extraído de MuleSoft, 2023a)

En la base de conocimiento de la empresa se puede obtener mucha más información al respecto. (Docs, 2023) También se muestran capturas reales de la herramienta, que facilitan comprender cómo es el producto final:



Figura 40: Captura de API Manager, *Gráficos personalizados*. (Extraído de MuleSoft-Docs, 2023e)

### 3.3. RunTime Manager

*Anypoint Runtime Manager* es la interfaz de la plataforma que ofrece una vista unificada de todas las aplicaciones, servidores y API desplegadas. (MuleSoft-Docs, 2023a)

Utilizando esta funcionalidad es posible desplegar y gestionar todos los entornos (*sandbox*, preproducción y producción) y las fases de integración y despliegue continuo. Usando la aplicación, es indiferente si el servidor físicamente está en entornos *on-premise* o en la nube.

El funcionamiento es sencillo: se ha de instalar un agente en cada uno de los equipos sobre los que se desean poder desplegar API y servicios. Será este agente el que se comuniquen de forma segura con los servidores centrales de Runtime Manager para actualizar la configuración a lo definido en el panel de control. (MuleSoft-Docs, 2023f)

Runtime Manager permite su uso en diferentes escenarios:

- **CloudHub:** es el entorno de despliegue propio de MuleSoft.
- **Híbrido:** en este caso, el despliegue se hace en servidores propios, pero las aplicaciones y servicios se gestionan con la plataforma de MuleSoft.
- **Anypoint Platform Private Cloud Edition:** Es un caso similar al anterior, los servidores siguen siendo propios y se sigue usando el controlador de MuleSoft. Sin embargo, este controlador está alojado también en un servidor propio.
- **Anypoint Runtime Fabric:** unos servidores especiales, que forman la *Runtime Fabric* son insta-

lados en equipos propios, y se utiliza la plataforma de MuleSoft para gestionarlos.

El siguiente diagrama muestra los diferentes escenarios:

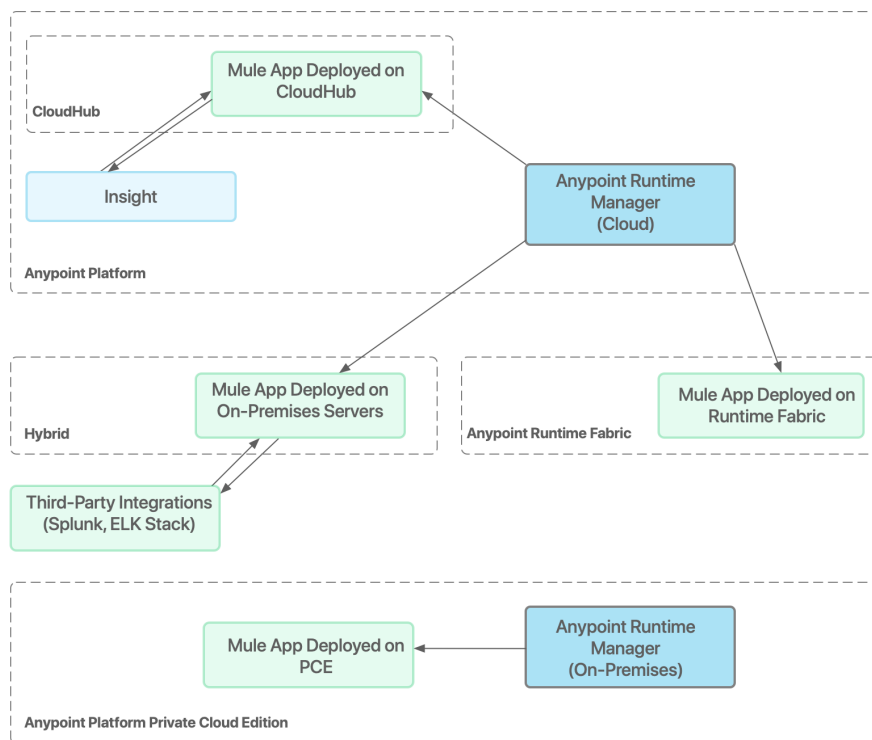


Figura 41: Escenarios de despliegue en *Runtime Manager*. (Extraído de MuleSoft-Docs, 2023a)

### 3.4. Access Management

La gestión de acceso en el entorno de MuleSoft tiene dos vertientes diferenciadas:

- Por un lado, el acceso al entorno de gestión de MuleSoft. Aquí se configurarán qué usuarios tienen qué permisos. La identidad de los usuarios se puede delegar en *Identity Providers* (IdP) externos, de forma que pueda usar las claves que ya utiliza para iniciar sesión en el entorno MuleSoft. (MuleSoft-Docs, 2023d)
- Por otro lado, tenemos la securización y la autenticación de usuarios que acceden a las propias API que hemos publicado. Para la gestión de estos accesos tenemos diferentes vías:
  1. Usuario y contraseña.
  2. Usuario y contraseña + multifactor
  3. Uso de tokens.
  4. Autenticación federada.

Lo normal es que se combinen diferentes métodos de autenticación en un desarrollo grande. Un ejemplo podría ser:

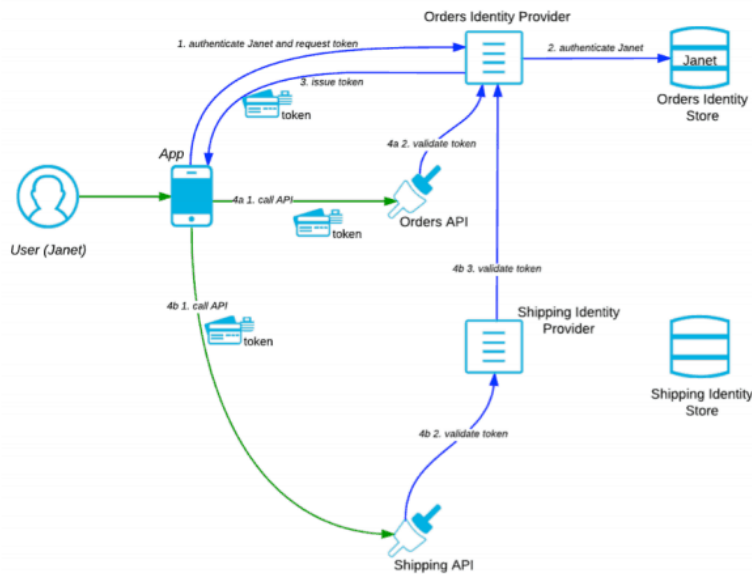


Figure 4: Federated Identity

Figura 42: Ejemplo de gestión de identidad. (Extraído de MuleSoft-Blog, 2016)

#### 4. Apartado 3: Implementación de alguno de los elementos anteriores

En este apartado, a través de la Consola de la plataforma *Anypoint Platform*, forzaremos la aplicación de una *política automatizada*.

Las políticas automatizadas permiten *desplegar API con confianza, porque el entorno objetivo automáticamente fuerza políticas críticas, reduciendo los errores. De esta forma, el administrador ya no necesita recordar configurar todas las políticas siempre*. (MuleSoft-Docs, 2023c)

Las instrucciones para su aplicación están disponibles en la documentación oficial. (MuleSoft-Docs, 2023b)

También se recogerán en el presente documento.

1. Acceder a API manager desde [anypoint.mulesoft.com](https://anypoint.mulesoft.com).
2. Una vez ha cargado el portal de gestión, se ha de seleccionar la opción *API Manager*, de entre las opciones del menú de la izquierda.

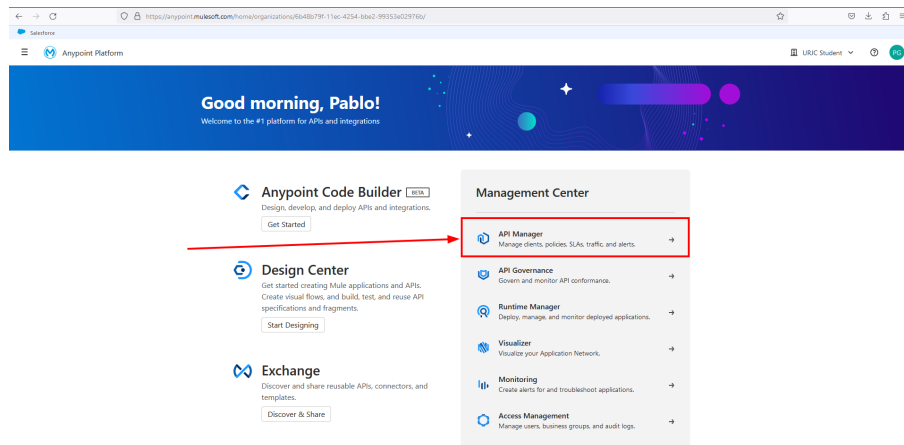


Figura 43: Seleccionar *API Manager* en el menú.

3. En esta nueva pantalla, se ha de hacer clic en *Automated Policies*, dentro del menú lateral derecho.

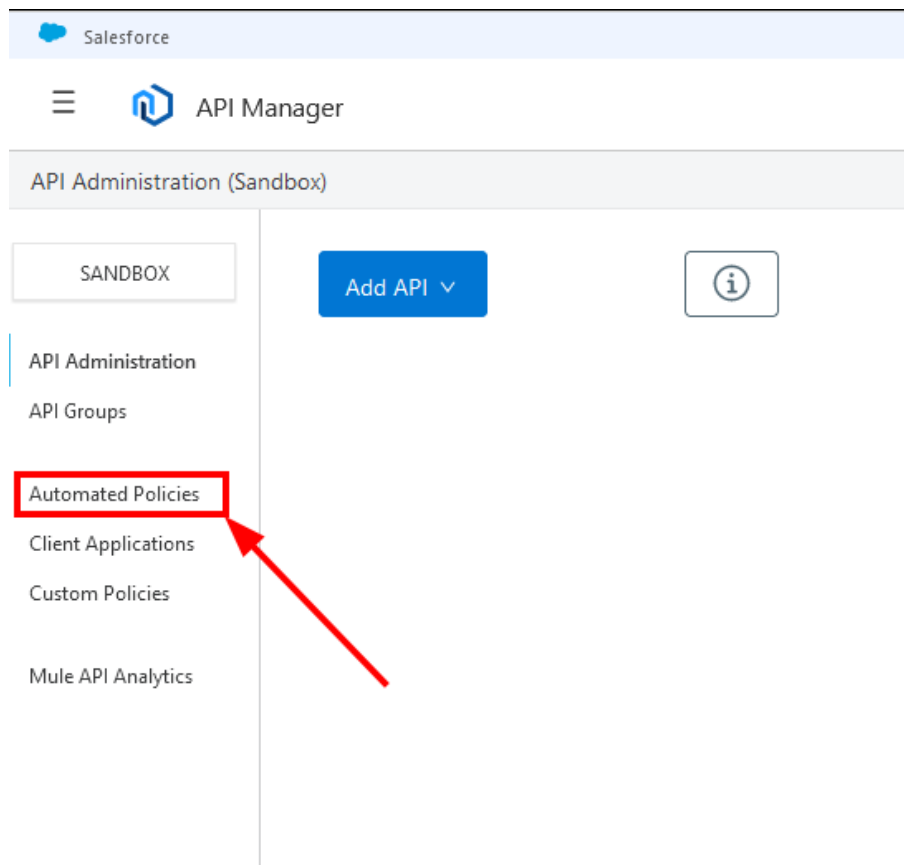
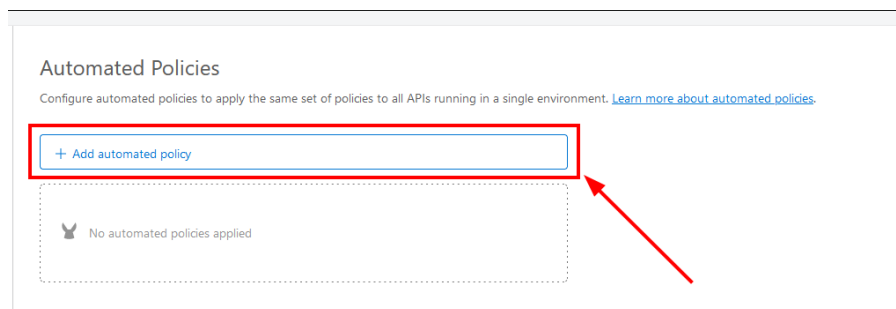


Figura 44: Botón *Automated Policies*

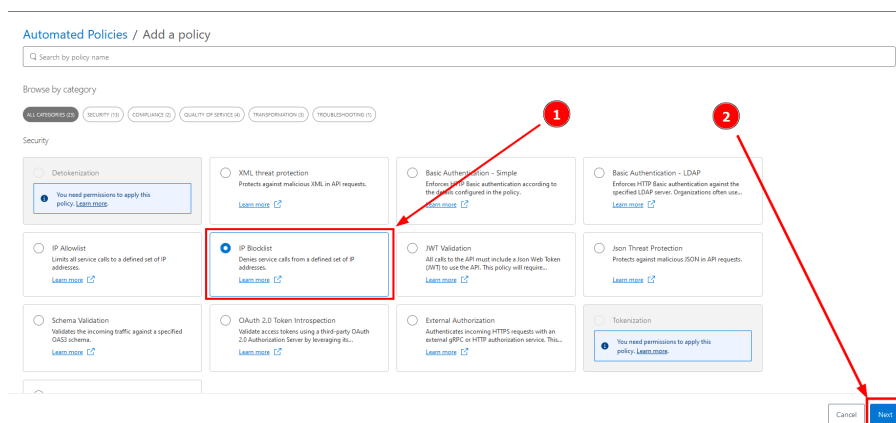
4. En esta ventana, en el caso de tener implementada ya alguna política, se mostraría. No es nuestro caso. Hacemos clic en el botón *Add automated policy*, situado en la parte superior de la ventana:

Figura 45: Botón *Add Automated Policy*

5. Se nos mostrarán multitud de opciones, agrupadas en las siguientes categorías:

- *Security*
- *Transformation*
- *Compliance*
- *Troubleshooting*
- *Quality of service*

Seleccionaremos la deseada. En este caso, *IP Blocklist*. La seleccionamos y hacemos clic en *Next*:

Figura 46: Seleccionar opción *IP Blocklist*

6. Como para nuestro caso de uso no tenemos ningún caso de IP que deseemos bloquear, usaremos un rango dentro del cual está la Universidad Rey Juan Carlos (anunciado por RedIRIS):

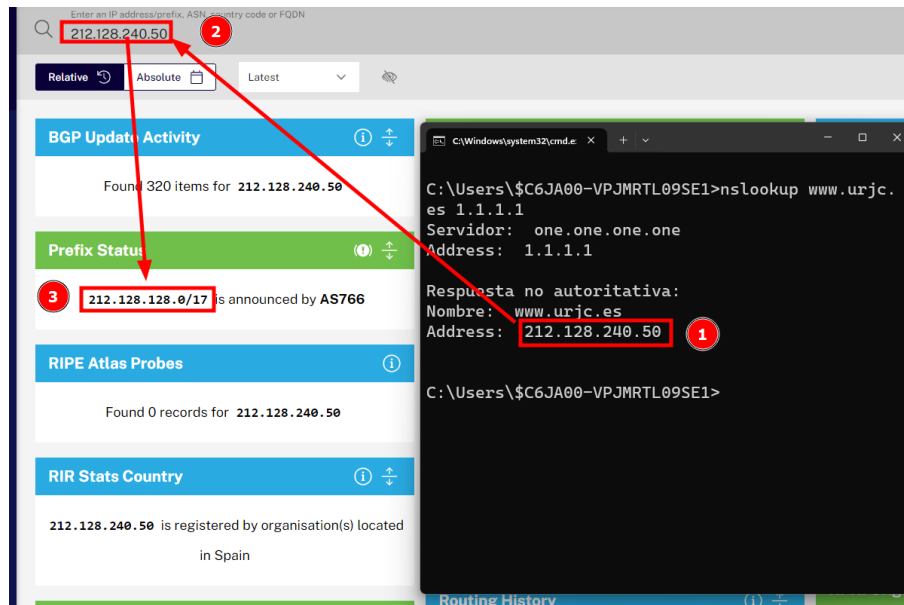


Figura 47: RIPEStat muestra, entre otra información, la ruta de las IP

7. Añadimos el rango, en formato CIDR en el campo (pudiendo añadir varias haciendo clic en *Add* y hacemos para guardar los cambios seleccionamos *Apply*)

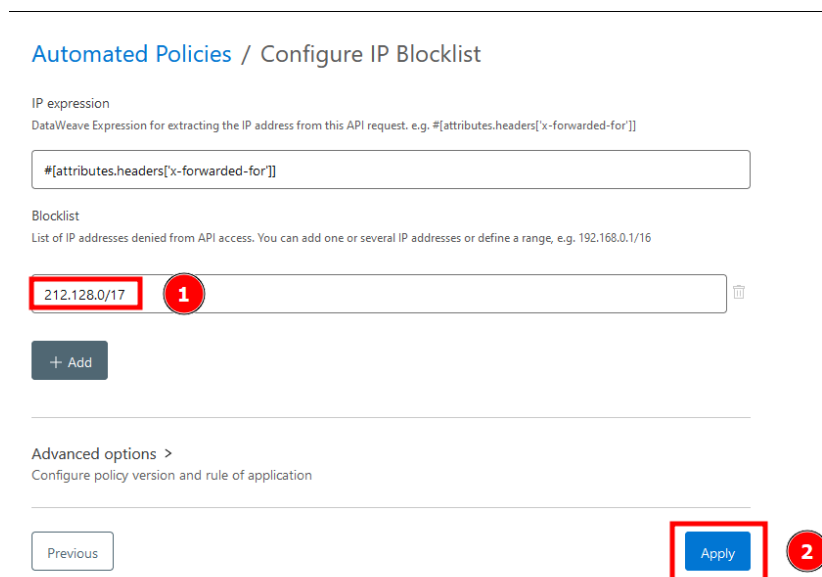


Figura 48: Añadir rango IP a bloquear

8. En este momento, volveremos a la pantalla inicial, donde ya podremos ver la regla creada:

## Automated Policies

Configure automated policies to apply the same set of policies to all APIs running in a single environment. [Learn](#)

+ Add automated policy

### IP Blocklist

Denies service calls from a defined set of IP addresses.

Version: 1.1.1

Rule of application: All runtimes

SECURITY



Figura 49: Regla *IP Blocklist* creada desde *Automated Policies*

## 5. Conclusiones

Para concluir el presente trabajo podemos afirmar que la implementación de sistemas basados en los estándares SOAP y REST es realmente útil para (1) el intercambio de la información respecto a los servicios que una organización ofrece; y (2) el intercambio de las características de la presentación y consumo de la información que se ofrece.

Si bien los casos que hemos visto son muy concretos y adaptados para el entorno educativo en el que se da, considero que hubiera sido muy interesante ver casos de implementación de SOAP y REST, junto con tecnologías relacionadas como WSDL, de organizaciones reales.

En tanto al posible margen de mejora, siempre es posible aumentar la calidad de una entrega. Por ejemplo, se podrían haber buscado ejemplos de casos de uso de empresas y organizaciones que ya estuvieran implementando soluciones similares a las descritas en el presente trabajo.

## 6. Acceso a los vídeos en YouTube

Además de en los enlaces anteriormente referenciados, se reflejan aquí los vídeos grabados:

- SOAP Interno - <https://youtu.be/saPd7OvC3Ag>
- SOAP Externo - <https://youtu.be/EEmueHup-b8>
- REST Interno - <https://youtu.be/G1P2m1NaCWI>
- REST Externo - <https://youtu.be/YqBN63LUNQs>

## 7. Bibliografía

- Docs, M. (2023). Mule API Analytics | MuleSoft Documentation. Consultado el 17 de diciembre de 2023, desde <https://docs.mulesoft.com/api-manager/latest/analytics-landing-page>
- MuleSoft. (2023a). API Analytics | API Reporting. Consultado el 17 de diciembre de 2023, desde <https://www.mulesoft.com/platform/api/anypoint-analytics>
- MuleSoft. (2023b). Plataforma de gestión de APIs | API Manager. Consultado el 17 de diciembre de 2023, desde <https://www.mulesoft.com/es/platform/api/manager>
- MuleSoft. (2023c). Plataforma de integración empresarial híbrida | Anypoint Platform. Consultado el 17 de diciembre de 2023, desde <https://www.mulesoft.com/es/platform/enterprise-integration>
- MuleSoft-Blog. (2016, agosto). API security: Ways to authenticate and authorize [Section: Dev guides]. Consultado el 17 de diciembre de 2023, desde <https://blogs.mulesoft.com/dev-guides/api-security-ways-to-authenticate-and-authorize/>
- MuleSoft-Docs. (2023a). Anypoint Runtime Manager | MuleSoft Documentation. Consultado el 17 de diciembre de 2023, desde <https://docs.mulesoft.com/runtime-manager/>
- MuleSoft-Docs. (2023b). Applying Automated Policies | MuleSoft Documentation. Consultado el 18 de diciembre de 2023, desde <https://docs.mulesoft.com/gateway/latest/policies-automated-applying>
- MuleSoft-Docs. (2023c). Automated Policies | MuleSoft Documentation. Consultado el 18 de diciembre de 2023, desde <https://docs.mulesoft.com/gateway/latest/policies-automated-overview>
- MuleSoft-Docs. (2023d). Manage Identity Providers | MuleSoft Documentation. Consultado el 17 de diciembre de 2023, desde <https://docs.mulesoft.com/access-management/external-identity-index>
- MuleSoft-Docs. (2023e). Reviewing Mule API Analytics Usage | MuleSoft Documentation. Consultado el 17 de diciembre de 2023, desde <https://docs.mulesoft.com/api-manager/latest/viewing-api-analytics>
- MuleSoft-Docs. (2023f). Runtime Manager Agent | MuleSoft Documentation. Consultado el 17 de diciembre de 2023, desde <https://docs.mulesoft.com/runtime-manager/runtime-manager-agent>
- URJC. (2023). Guía de Estudio de la asignatura Diseño y Mantenimiento de Servicios.