

Práctica 1: Mule-ESB, *Hello World* y demo

Diseño y Mantenimiento de Servicios

Índice

1. Introducción, expectativas y temporalización	3
1.1. Presentación	3
1.2. <i>Abstract</i>	3
1.3. Temporalización y organización	3
1.4. Expectativas	3
2. Desarrollo de la práctica	4
2.1. Instalación de Mule-ESB	4
2.1.1. Descarga del instalador	4
2.1.2. Comprobación <i>hash</i>	4
2.1.3. Extracción del programa y <i>workspace</i>	5
2.1.4. Añadir regla de <i>Firewall</i> y exclusión a antivirus	6
2.1.5. Primer inicio	6
2.2. Ejemplo <i>Hola Mundo</i>	7
2.2.1. Creación de un nuevo proyecto	7
2.2.2. Añadir y configurar <i>HTTP Listener</i>	8
2.2.3. <i>Set Payload</i>	9
2.2.4. Ejecutar proyecto	10
2.3. Ejemplo personalizado: <i>Hello World</i> multilingüe con HTTPS y <i>Logger</i>	12
2.3.1. Generar KeyStore y certificados HTTPS	13
2.3.2. Configuración del <i>Listener</i>	13
2.3.3. Módulo <i>Set Variable</i>	16
2.3.4. <i>Choice</i> y opciones de este	16
2.3.5. Configuración del <i>Logger</i>	17
2.3.6. Pruebas realizadas	18
2.3.7. Esquema XML <i>MuleSoft</i>	21
3. Conclusiones	22
4. Acceso al vídeo en Youtube. Recursos disponibles	22
5. Bibliografía	23

Índice de figuras

1. Formulario para descargar Mule-ESB	4
2. Interfaz de generación de huellas digitales	5
3. Comprobación de que las huellas digitales coinciden	5
4. Proceso de extracción del ZIP de AnyPoint	5
5. Cuadro de Diálogo para seleccionar la ruta del <i>workspace</i>	6
6. Adición de regla de <i>Firewall</i>	6
7. Adición de excepción en <i>Windows Defender</i>	6
8. Interfaz de AnyPointStudio	7
9. Ruta para crear un nuevo proyecto	7
10. Cuadro de diálogo para crear un nuevo proyecto	8
11. Módulo <i>Listener</i> en el <i>Mule Palette</i>	8

12.	Configuración del <i>HTTP Listener</i>	9
13.	Propiedades y ruta de escucha para el <i>HTTP Listener</i>	9
14.	Búsqueda del módulo <i>Set Payload</i>	10
15.	Configuración de <i>Payload</i>	10
17.	Resultado en el navegador	11
16.	Ejecutar proyecto	11
18.	Flujo del ejemplo personalizado	12
19.	Generación de <i>KeyStore</i>	13
20.	Contenido del <i>KeyStore</i>	13
21.	Módulo <i>Listener</i> en el <i>Mule Palette</i>	14
22.	Configuración del <i>Listener</i> , general	14
23.	Configuración del <i>Listener</i> , detalle TLS	15
24.	Configuración del <i>Listener</i> , MIME y codificación	16
25.	Configuración <i>Set Variable</i> , detalle	16
26.	Detalle de la configuración para el <i>Set Variable</i> cuando el idioma seleccionado es <i>es</i>	17
27.	Configuración del módulo <i>Logger</i>	17
28.	Captura de pantalla del resultado sin indicar parámetro	18
29.	Detalle del certificado usado por el servidor	18
30.	Captura de pantalla del resultado indicando <i>es</i> como parámetro	19
31.	Captura de pantalla del resultado indicando <i>en</i> como parámetro	19
32.	Captura de pantalla del resultado indicando <i>fr</i> como parámetro	19
33.	Captura de pantalla del resultado indicando <i>de</i> como parámetro	20
34.	Registros generados por las peticiones	20

Índice de tablas

1.	Configuraciones de los módulos <i>Set Variable</i> de cada opción del <i>Choice</i>	16
----	---	----

1. Introducción, expectativas y temporalización

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.

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 ejercicio.

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.

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 exercise.

1.3. Temporalización y organización

Tal y como se recoge en la Guía de Estudio de la asignatura, la práctica se engloba dentro del tema 1, *Principios básicos del diseño*. Este tema, según la cronología de la citada Guía, se desarrollará entre los días 13/09/2023 y 03/10/2023. (URJC, 2023)

El trabajo se desarrollará de forma individual.

1.4. Expectativas

Las expectativas de aprendizaje de la presente práctica se basan en la aplicación de los conocimientos básicos sobre SOA adquiridos hasta el momento. A su vez, se espera que este trabajo suponga un primer acercamiento a la herramienta *Mule-ESB*, que se usará en sucesivas prácticas.

2. Desarrollo de la práctica

2.1. Instalación de Mule-ESB

2.1.1. Descarga del instalador

El programa puede descargarse desde la web del desarrollador (MuleSoft): mulesoft.com/lp/dl/anypoint-mule-studio. Aquí será necesario introducir una serie de datos para descargar el instalador. El enlace de descarga será enviado a la dirección de correo electrónico indicada al terminar de cumplimentar el formulario.

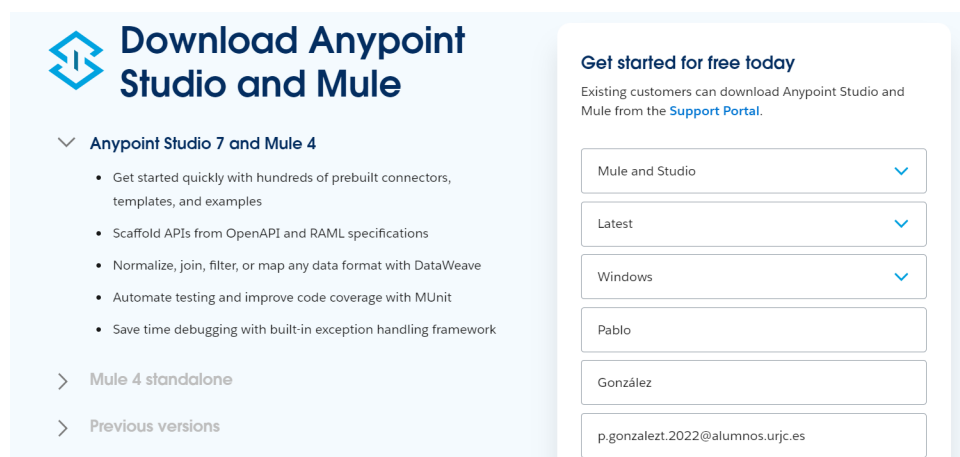


Figura 1: Formulario para descargar Mule-ESB

En el mensaje de correo electrónico recibido, además del propio enlace de descarga encontraremos dos enlaces, en mi opinión, muy interesantes:

- Instrucciones de instalación.¹
- Tutorial para empezar con MuleSoft.² (MuleSoft, 2023a)

2.1.2. Comprobación *hash*

Una vez descargado el ZIP, es altamente recomendable, aunque no estrictamente necesario, comprobar que el *hash*, también llamado «huella digital» del archivo descargado es igual que el mostrado en la web de MuleSoft.

Esto servirá, no solo para verificar que no se ha producido, por ejemplo, un ataque *Man In The Middle*, MITM. Sino para comprobar que no se han producido problemas en la transmisión de los datos que hayan podido pasar inadvertidos.

Si bien la aplicación con la que generemos las huellas digitales sobre los archivos locales es indiferente, en este caso se ha utilizado la aplicación *Autofirma*³ para la generación. Se ha optado por esta ya que hoy en día está instalada en multitud de equipos gracias a ser necesaria para muchas de las relaciones/actividades electrónicas que como ciudadanía mantenemos con la Administración.

¹<https://docs.mulesoft.com/studio/7.12/to-download-and-install-studio>

²<https://developer.mulesoft.com/tutorials-and-howtos/getting-started/hello-mule/>

³<https://firmaelectronica.gob.es/Home/Descargas.html>

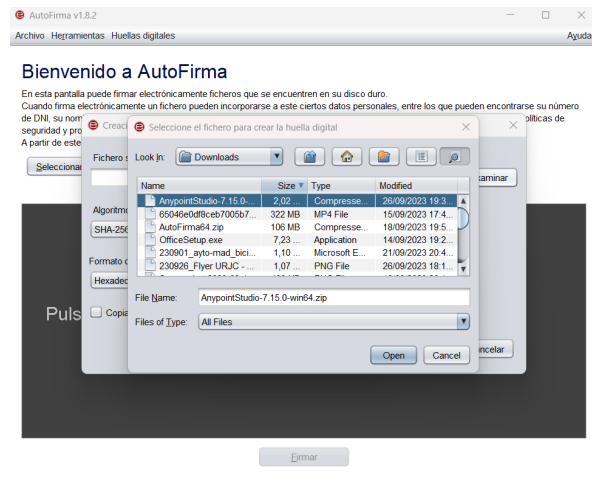


Figura 2: Interfaz de generación de huellas digitales

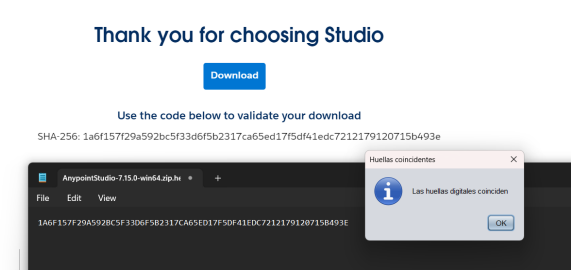


Figura 3: Comprobación de que las huellas digitales coinciden

2.1.3. Extracción del programa y *workspace*

Tal y como se indica en la documentación oficial, para la instalación del programa en Windows, es necesario extraer el archivo ZIP descargado en C:\ (MuleSoft, 2023b). Una vez extraído el archivo

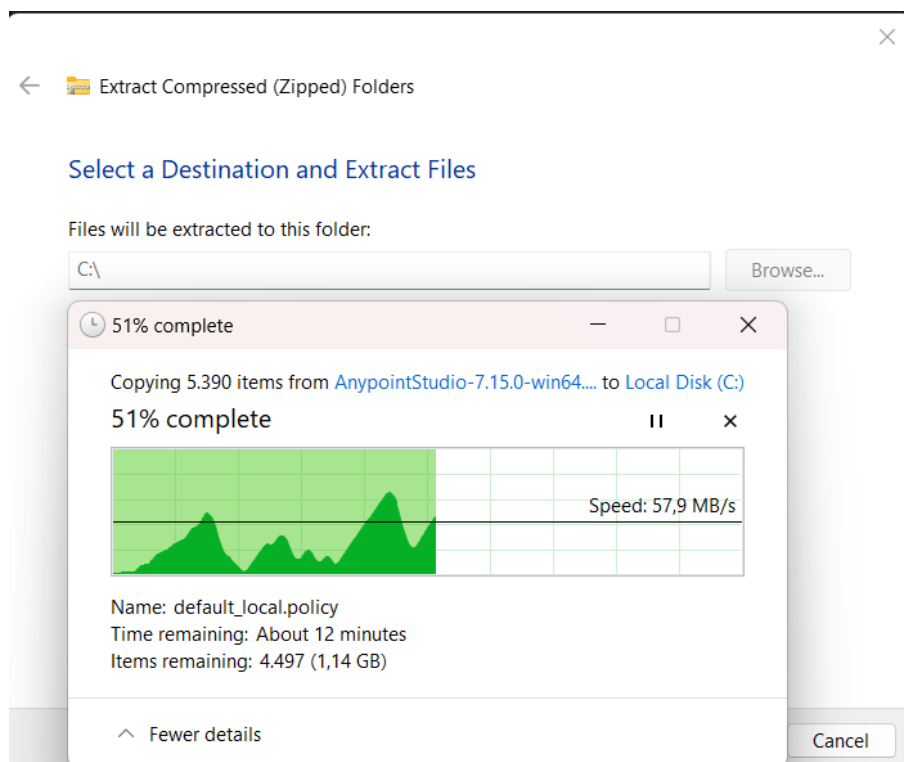


Figura 4: Proceso de extracción del ZIP de AnyPoint

ZIP, tendremos el ejecutable en la ruta C:\AnypointStudio\AnypointStudio.exe, junto con el resto de archivos y recursos.

Al ejecutar el archivo **AnypointStudio.exe**, aparecerá un cuadro de diálogo en el que debemos introducir

el «espacio de trabajo», o *workspace*. El valor que aparece por defecto se puede mantener:

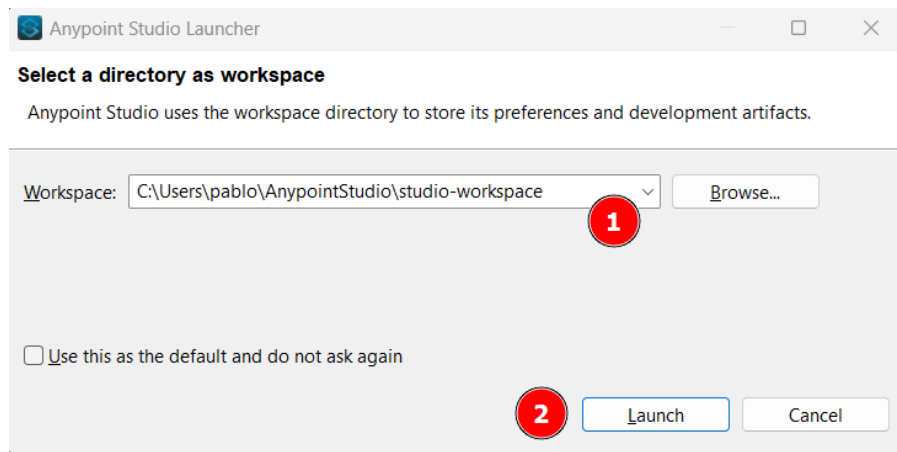


Figura 5: Cuadro de Diálogo para seleccionar la ruta del *workspace*

Haremos clic en *Launch* para iniciar *AnyPointStudio*.

2.1.4. Añadir regla de *Firewall* y exclusión a antivirus

Se añadirá la regla de *Firewall* correspondiente (con especial atención a si trabajaremos con una red pública y/o una red privada). Y, para mejorar el rendimiento, también añadiremos la excepción al antivirus (en mi caso, *Windows Defender*):

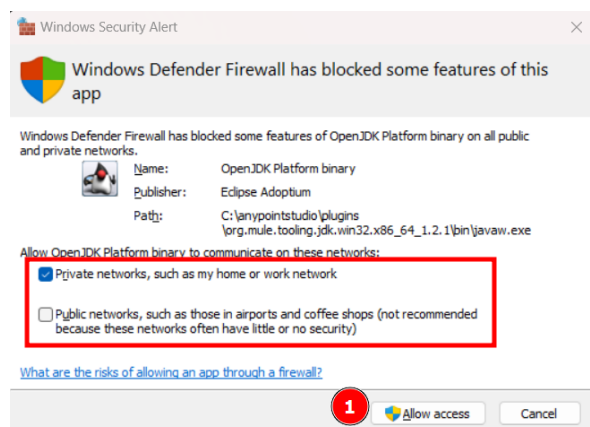


Figura 6: Adición de regla de *Firewall*

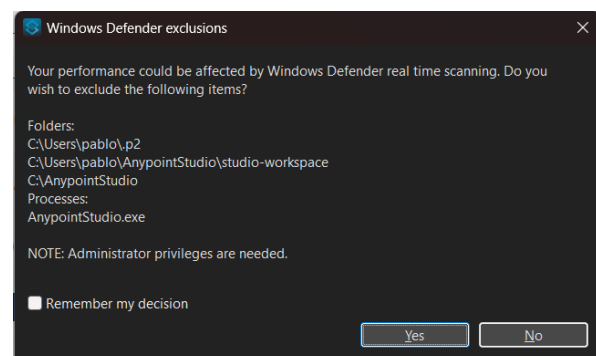


Figura 7: Adición de excepción en *Windows Defender*

2.1.5. Primer inicio

Al completar todos los pasos indicados previamente, cargará la interfaz de *AnyPointStudio*:

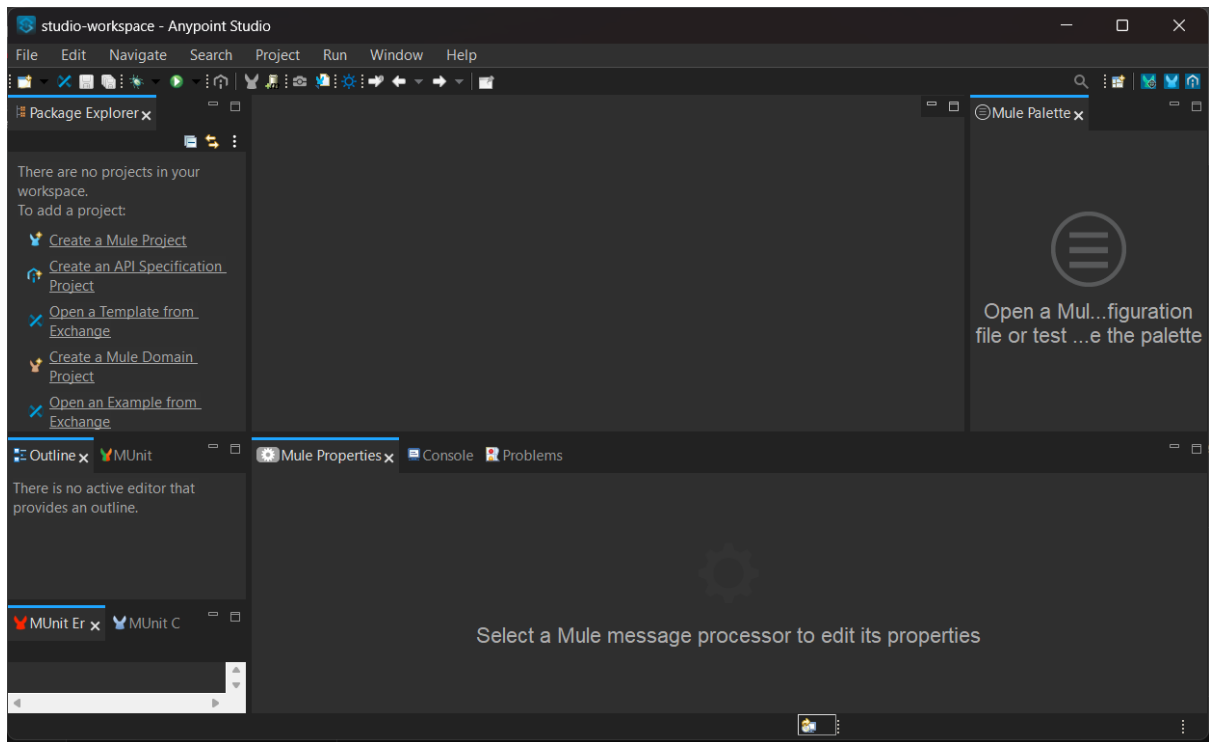


Figura 8: Interfaz de AnyPointStudio

2.2. Ejemplo *Hola Mundo*

2.2.1. Creación de un nuevo proyecto

Una vez definida la ubicación del espacio de trabajo, que hemos realizado en un paso anterior, Extracción del programa y *workspace*, se ha de crear un nuevo proyecto. Este nuevo proyecto agrupará los archivos, configuraciones y definiciones de nuestro desarrollo.

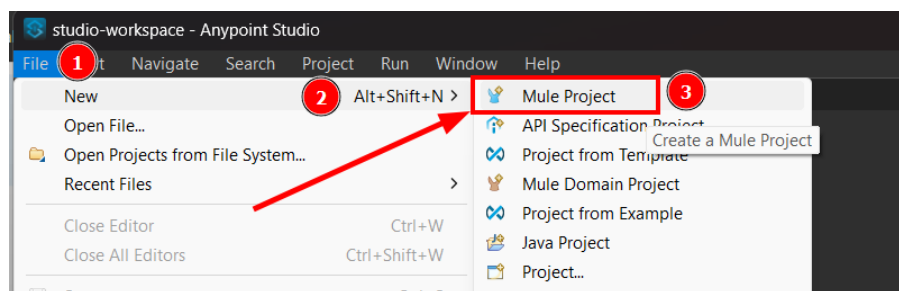


Figura 9: Ruta para crear un nuevo proyecto

En la siguiente ventana deberemos indicar el nombre del nuevo proyecto y el *runtime* deseado. En este caso, tal y como se puede ver en la imagen a continuación, se ha mantenido el *runtime* por defecto y elegido *HelloWorldPablo* como nombre del proyecto:

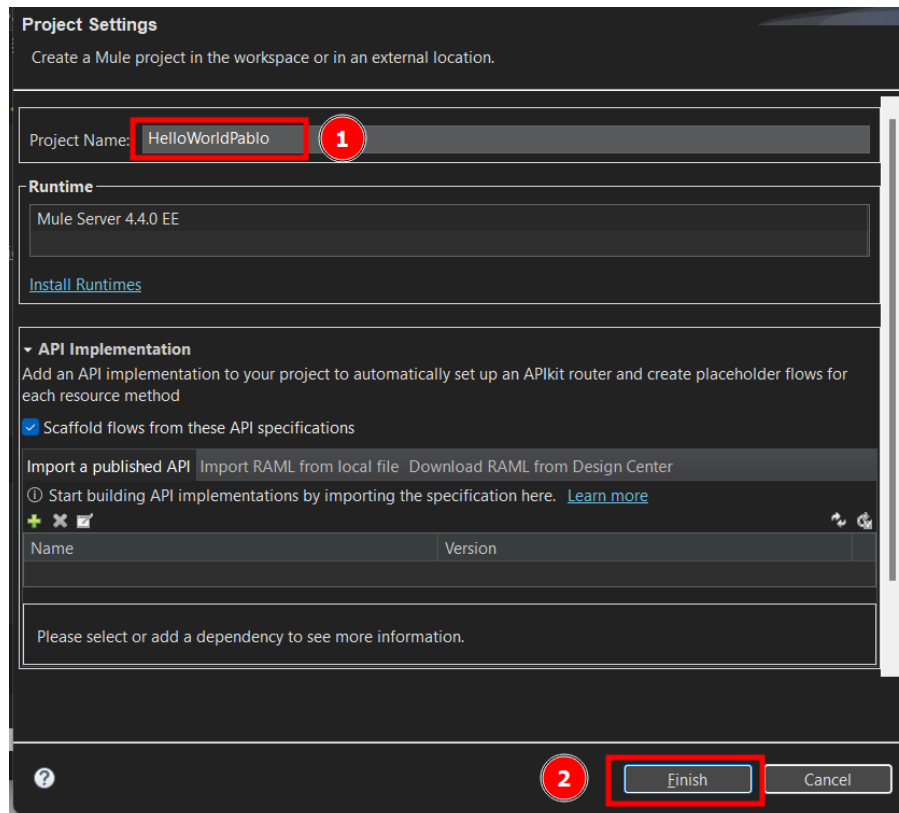
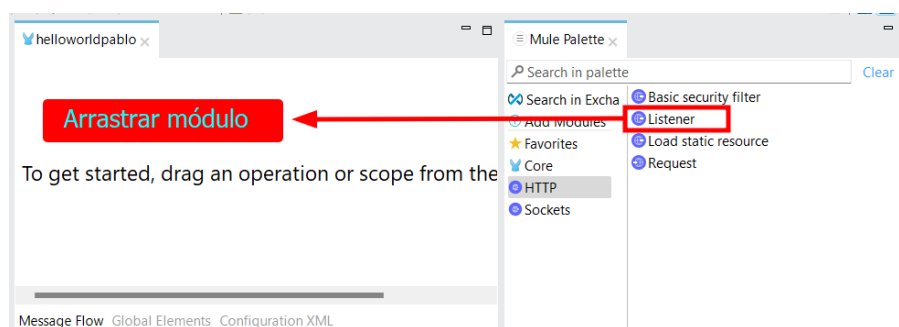


Figura 10: Cuadro de diálogo para crear un nuevo proyecto

Tema oscuro Se ha desactivado el modo oscuro, por preferencia personal. Las sucesivas capturas de pantalla serán con el modo claro.

2.2.2. Añadir y configurar *HTTP Listener*

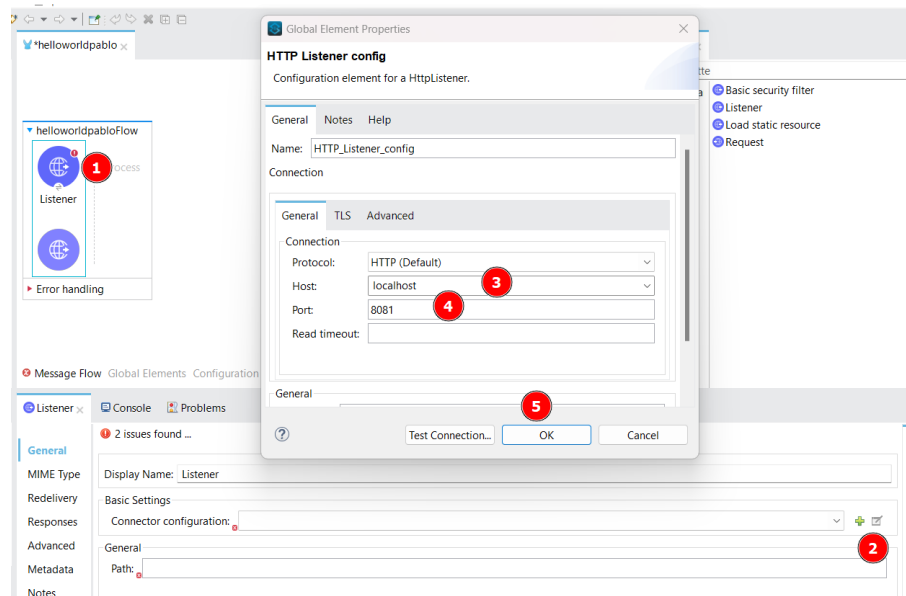
Desde el *Mule Palette*, localizado en la parte superior derecha de la interfaz de Mule Studio, arrastramos el módulo *Listener* al espacio de trabajo (la parte central de la interfaz):

Figura 11: Módulo *Listener* en el *Mule Palette*

Una vez añadido el *Listener*, «escuchador» al espacio de trabajo, debemos configurarlo. Para ello:

1. Seleccionar el *Listener* en el espacio de trabajo.
2. En la vista de propiedades, en la parte inferior de la interfaz, hacer clic en el icono **+** para añadir una nueva configuración. Aquí también podemos cambiar, si lo deseamos, el nombre visible para el elemento, por ejemplo.

3. En el nuevo cuadro de diálogo, se indicarán también los valores para la interfaz en la que escuchará el *Listener* y el puerto por el que lo hará. Si bien para el puerto se ha mantenido el valor que venía por defecto, 8081, para la interfaz de escucha he seleccionado *localhost* en lugar de todas las interfaces.
4. (Opcional) Puede ser útil hacer clic en el botón *Test connection* para comprobar si la configuración es válida y si puede establecerse comunicación a través de la interfaz y puerto indicados.

Figura 12: Configuración del *HTTP Listener*

También se debe indicar cuál será la ruta que activará la ejecución del flujo que se está configurando el AnyPointStudio. En este caso se elige */holamundo*. Es muy importante indicar el *slash* del principio.

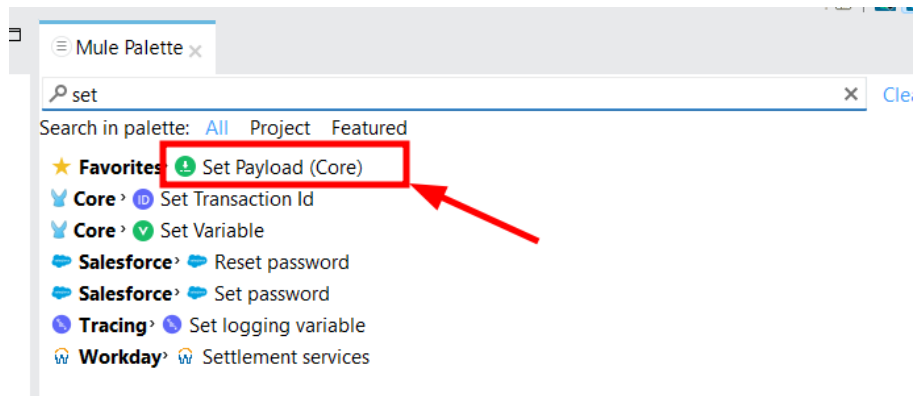
En la parte superior de la sección de propiedades aparecerá el mensaje *There are no errors* si hemos configurado con valores válidos el *Listener*:

Figura 13: Propiedades y ruta de escucha para el *HTTP Listener*

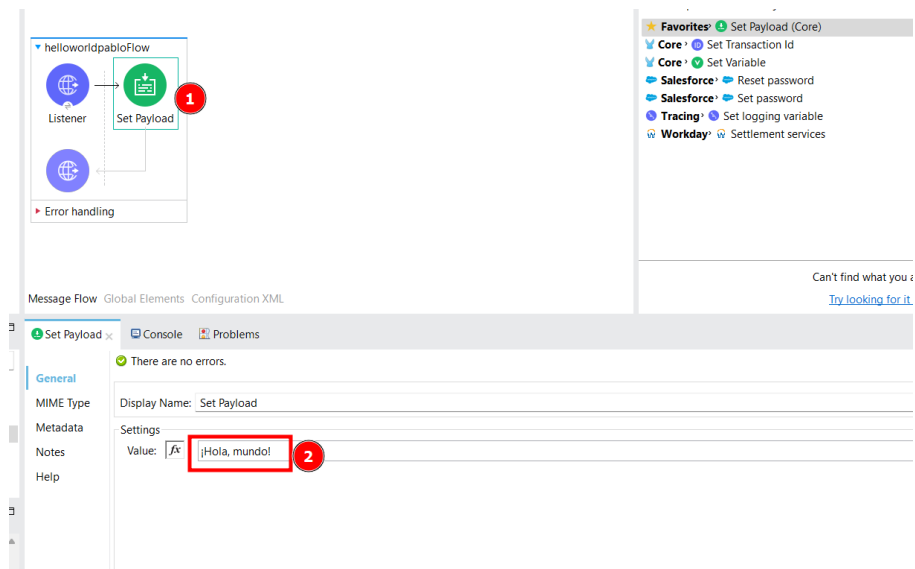
2.2.3. Set Payload

Una vez configurado el *Listener* siguiendo los pasos descritos en el apartado anterior, hemos de configurar el *Payload* («carga útil», en castellano) con el que responderá a las solicitudes HTTP.

Podemos localizar el módulo *Set Payload* en el *Mule Palette* de la parte superior derecha de la pantalla. Lo arrastraremos a la sección *Process* del flujo en el que teníamos el *Listener*.

Figura 14: Búsqueda del módulo *Set Payload*

En el apartado de propiedades del módulo *Set Payload* que acabamos de añadir se puede configurar el mensaje con el que responderá la aplicación a las solicitudes. En este caso, se ha indicado ¡Hola, mundo!:

Figura 15: Configuración de *Payload*

2.2.4. Ejecutar proyecto

Haremos clic derecho sobre el *flow*, «flujo» que hemos creado y, después, seleccionaremos la opción *Run project*. Es posible que nos pida confirmación para guardar el proyecto, haremos clic en guardar.

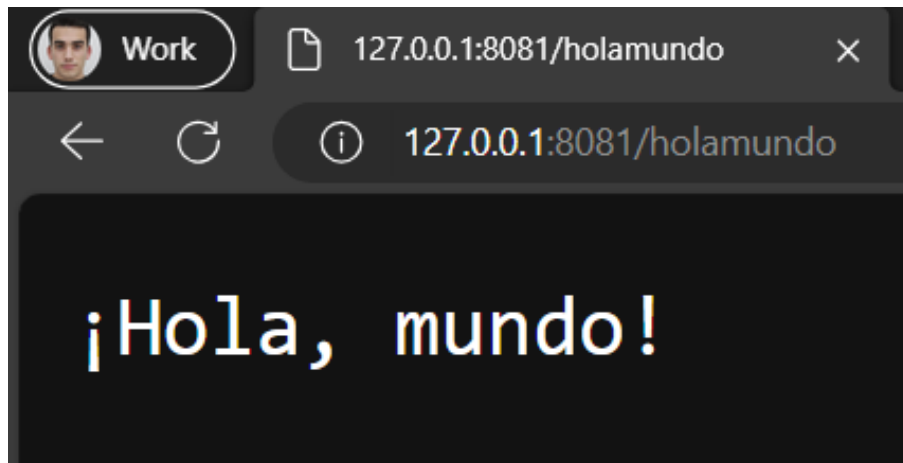


Figura 17: Resultado en el navegador

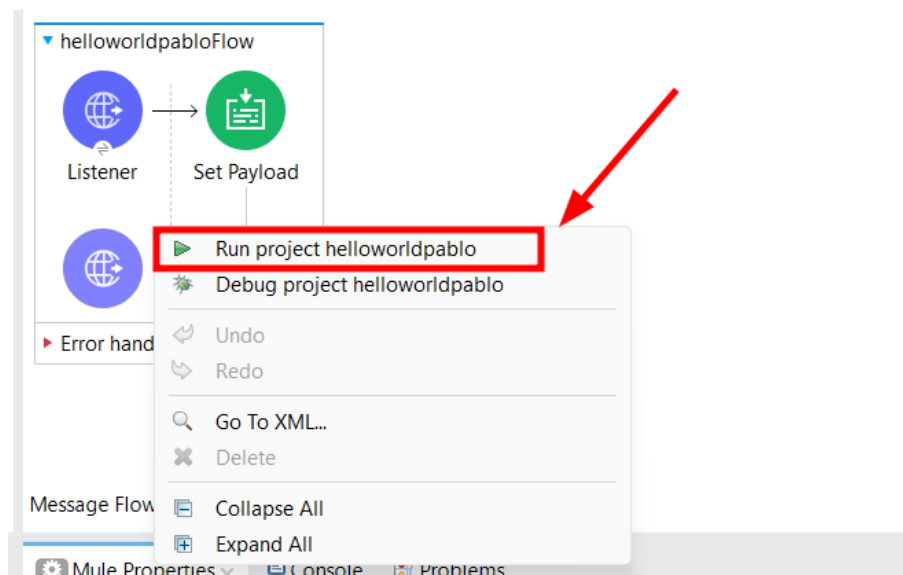


Figura 16: Ejecutar proyecto

En el proceso de ejecución se generaran muchos *logs*, «registros», por consola. En caso de que falle la aplicación puede ser muy útil revisar estos registros.

Al iniciar un navegador e introducir `http://127.0.0.1:8081/holamundo` veremos el mensaje configurado:

CODIFICACIÓN En mi caso, había un carácter especial, «j», que inicialmente el navegador mostraba de forma incorrecta. Para solucionarlo, basta con ajustar en el módulo *Set Payload* la codificación a UTF-8 y la codificación a *plain/text*.

2.3. Ejemplo personalizado: *Hello World* multilingüe con HTTPS y *Logger*

Este ejemplo propuesto se basa en el ejemplo anterior, pero modificándolo con los siguientes factores:

- Se agrega un *Choice* para, según el valor de idioma (almacenado en la variable `lang`, y pasado mediante parámetro en la URL con la misma clave) responda en un idioma u otro con la traducción correspondiente de la frase *Hola, Mundo!*.
- El *Choice* funciona usando la variable que se registra en `vars.lang`, que es fijada con el módulo *Set Variable*, que lee el parámetro HTTP pasado en la URL.
- Se añade el módulo *logger* para registrar cada petición, y así poder detectar errores.
- Se genera un par de certificado-clave para poder cifrar el tráfico HTTP, convirtiéndolo en HTTPS, usando la utilidad `keytool` de Java.

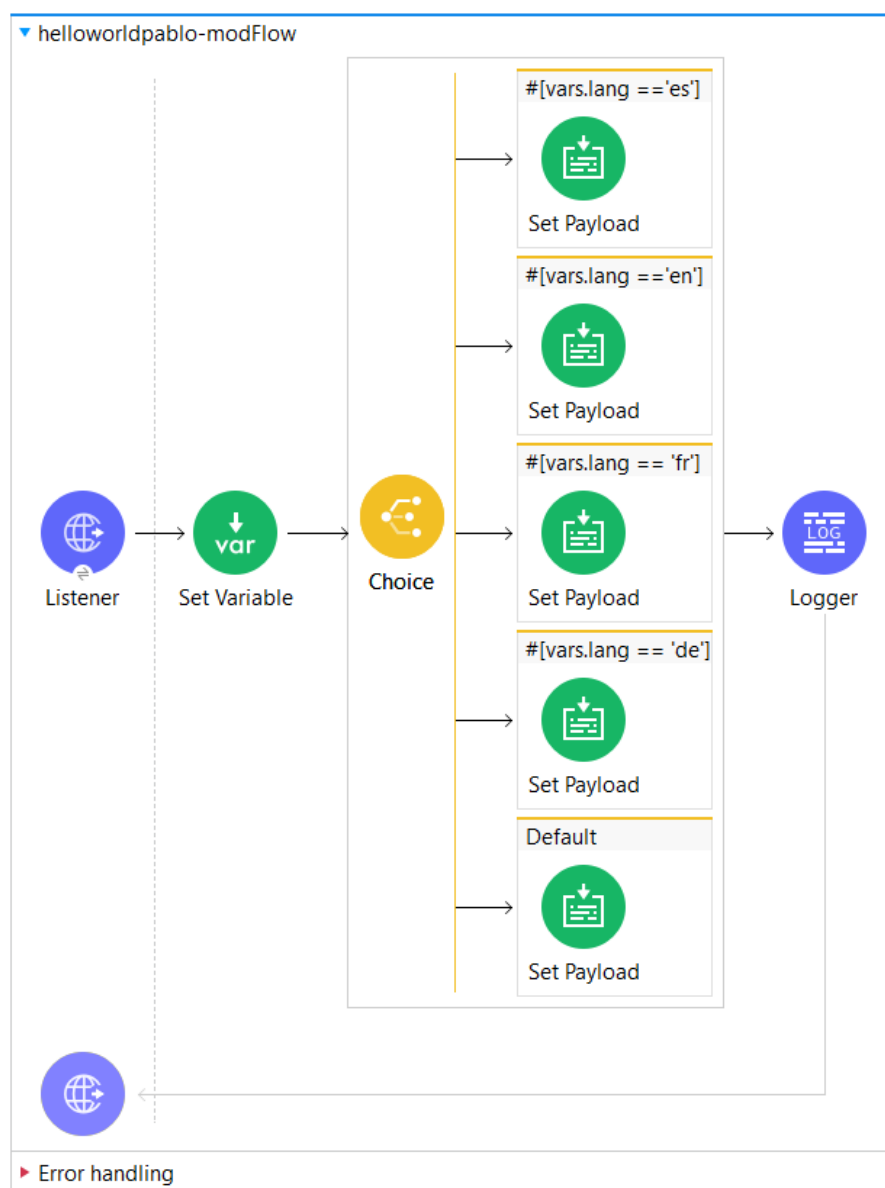


Figura 18: Flujo del ejemplo personalizado

2.3.1. 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 ⁴:

`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]: Diseno 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=Diseno y Mantenimiento de Servicios, L=Madrid, ST=Spain, C=ES correct?
[no]: |
```

Figura 19: 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

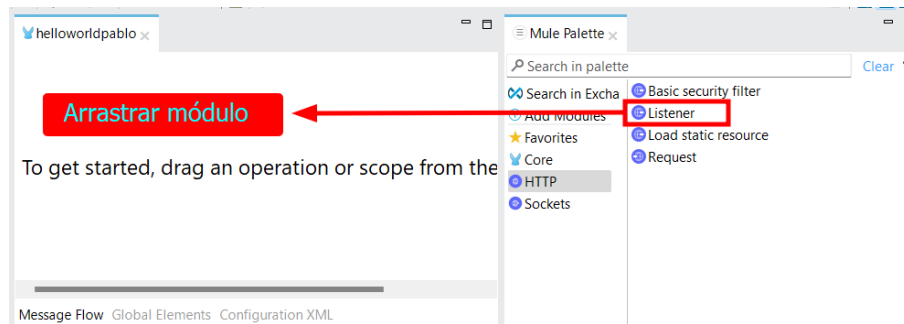
key-alias, 01-oct-2023, PrivateKeyEntry,
Certificate fingerprint (SHA-256): 69:6F:DA:30:B6:76:75:3A:3A:69:10:AF:69:F8:15:BC:05:E9:D3:C5:A8:96:15:AB:D2:8E:B3:B3:E
C:D8:27:5E
```

Figura 20: Contenido del *KeyStore*

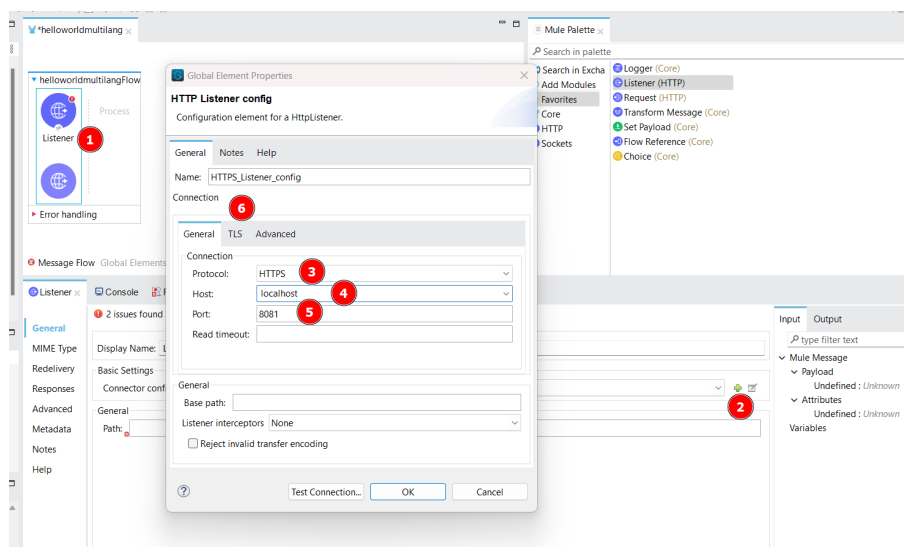
2.3.2. Configuración del *Listener*

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):

⁴Se 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.

Figura 21: Módulo *Listener* en el *Mule Palette*

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

Figura 22: 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:

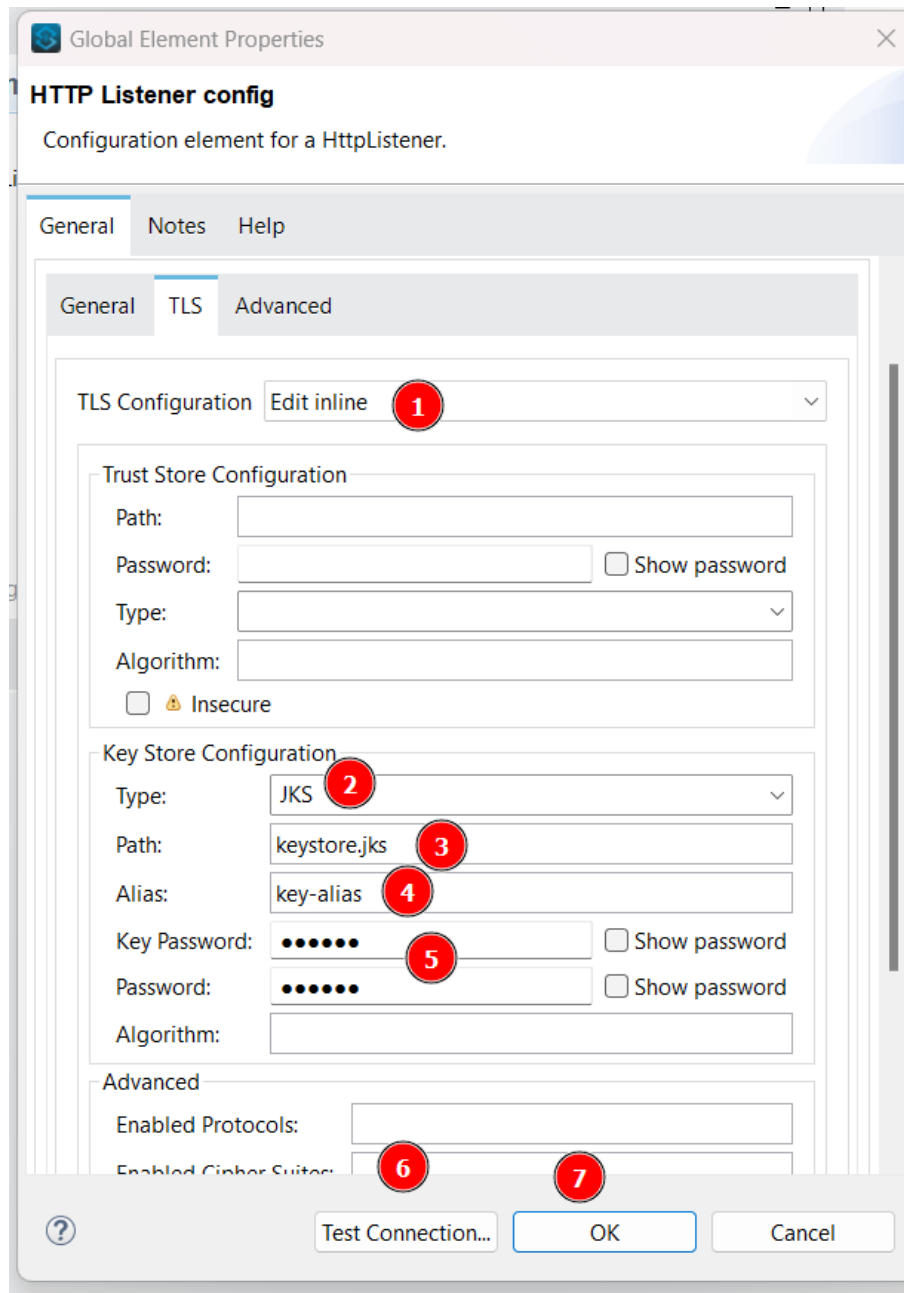


Figura 23: Configuración del *Listener*, detalle TLS

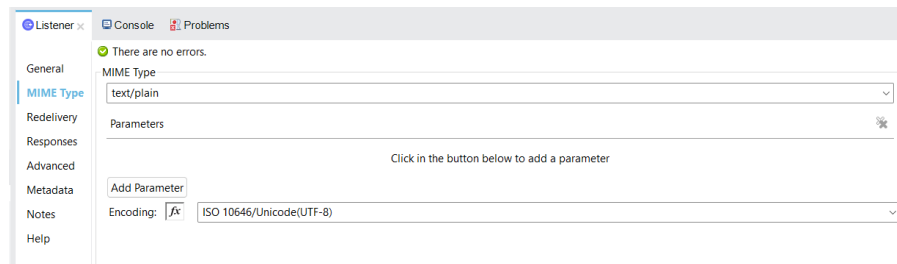
Para la configuración TLS se selecciona *Edit inline* para editar la configuración en la ventana [1], en la configuración *Key Store Configuration* se indica *JKS* como tipo [2], *keystore.jks* [3] como ruta ⁵, *key-alias* como alias de la clave [4]⁶, así como la contraseña en los campos correspondientes [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:

⁵Teniendo en cuenta que está en la carpeta *src/resources*

⁶Se puede consultar el alias de la clave con el comando *keystore -list* indicado previamente

Figura 24: Configuración del *Listener*, MIME y codificación

2.3.3. Módulo *Set Variable*

Se añade un módulo del tipo *Set Variable* al flujo con la siguiente configuración:

- Name: `lang`
- Value: `#[attributes.queryParams.lang]`

Figura 25: Configuración *Set Variable*, detalle

2.3.4. *Choice* y opciones de este

Se añade un módulo de tipo *Choice*, con 4 opciones y un *Default*. Las opciones adicionales se pueden añadir arrastrando módulos del tipo *Set Variable* al árbol del flujo. Las distintas configuraciones son:

Opción del <i>Choice</i>	Valor del <i>Set Variable</i>
<code>#[vars.lang == 'es']</code>	Hola, Mundo!
<code>#[vars.lang == 'en']</code>	Hello, World!
<code>#[vars.lang == 'fr']</code>	Salut le monde!
<code>#[vars.lang == 'de']</code>	Hallo, Welt!
<i>Default</i>	<p><code><h1>ERROR</h1><p>Use “es”, “en”, “fr” o “de” como parámetro en la URL.
</code></p> <p><i>Ejemplo:</i> <code><pre>https://127.0.0.1:8081/holamundo?lang=en</pre></p></code></p>

Cuadro 1: Configuraciones de los módulos *Set Variable* de cada opción del *Choice*

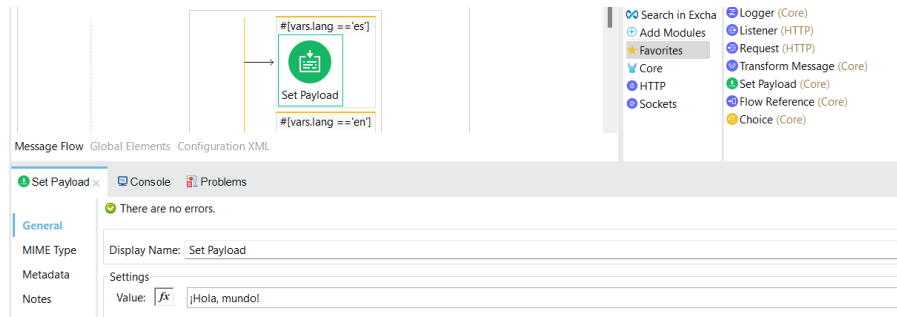


Figura 26: Detalle de la configuración para el *Set Variable* cuando el idioma seleccionado es *es*

2.3.5. Configuración del *Logger*

Por último, se añade un módulo de tipo *Logger* para poder registrar cada petición en el registro. Esto es realmente útil para detectar errores y solucionarlos posteriormente. La configuración de este módulo es sencilla, registrando únicamente el idioma seleccionado por el usuario:

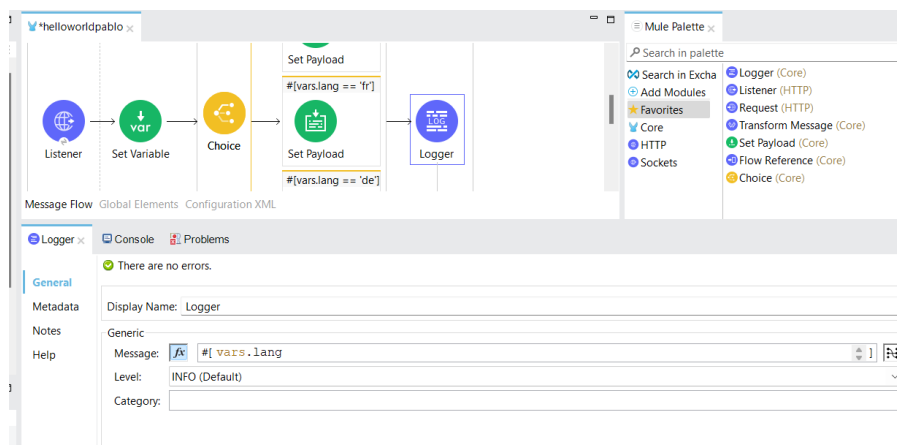


Figura 27: Configuración del módulo *Logger*

2.3.6. Pruebas realizadas

Sin indicar parámetro, o indicando uno no válido Resultado:

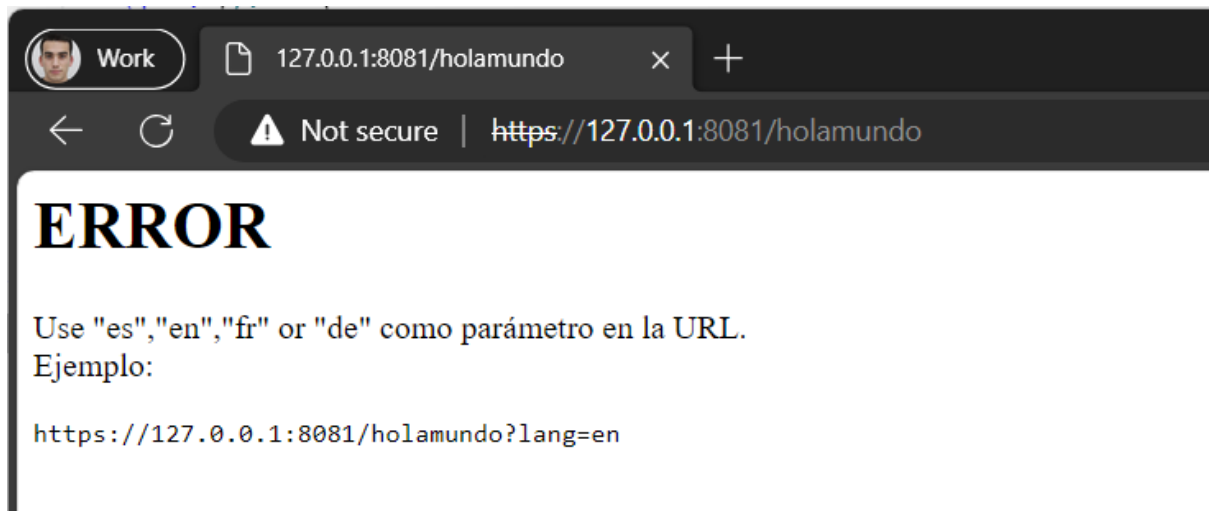


Figura 28: Captura de pantalla del resultado sin indicar parámetro

En este punto cabe detenerse para comentar el **error de certificado** HTTPS mostrado por el navegador. Puesto que el certificado que está usando el servidor no está firmado por ninguna *Autoridad de Certificación*, CA, reconocida, arroja un error para avisar. Sin embargo, la información sí viaja cifrada. También vemos como los detalles introducidos en la generación del certificado se muestran al usuario.

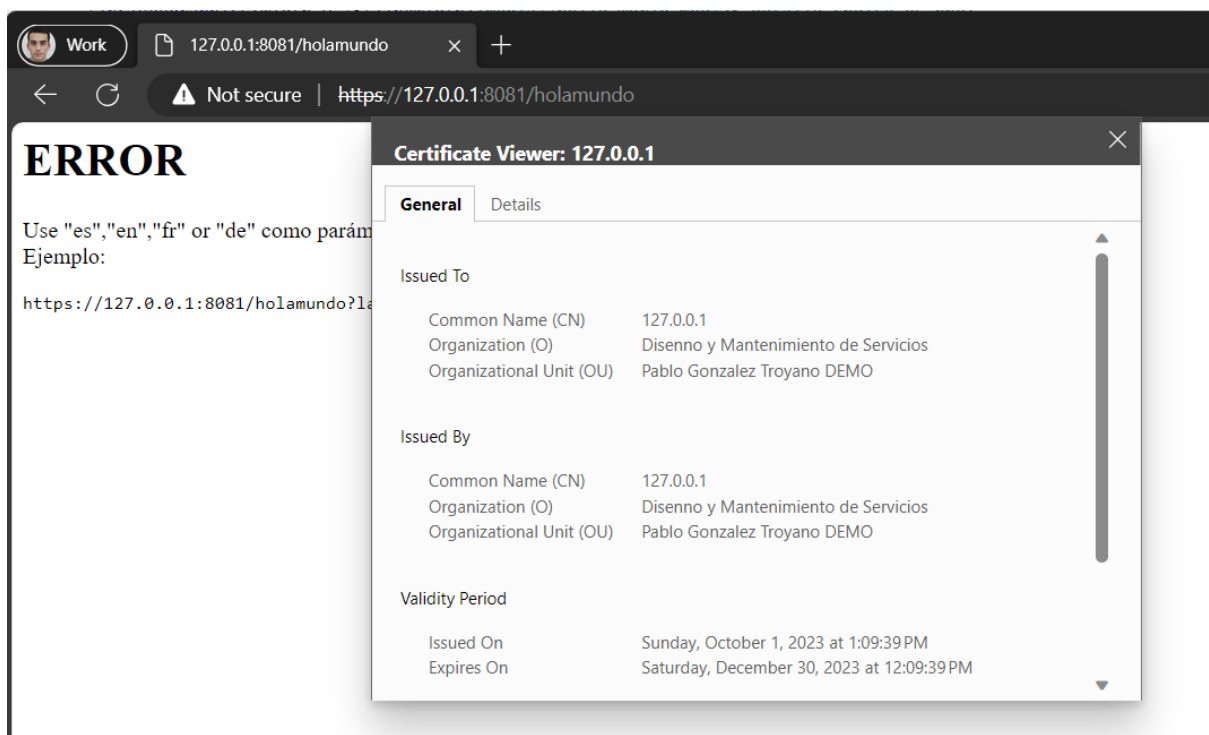


Figura 29: Detalle del certificado usado por el servidor

Indicando *lang=es* Resultado:

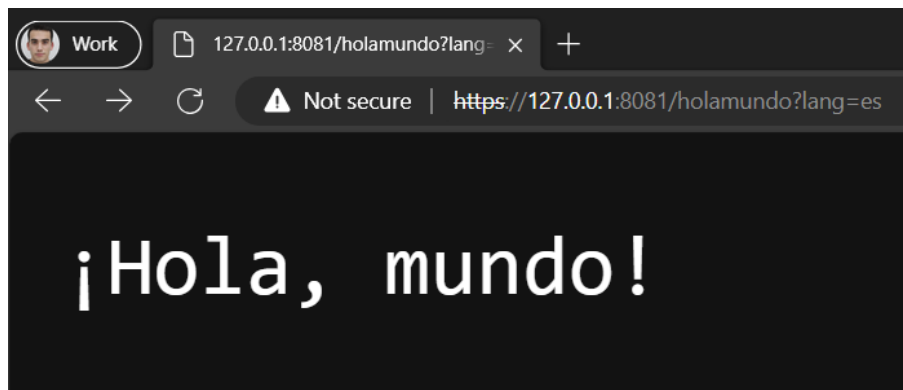


Figura 30: Captura de pantalla del resultado indicando *es* como parámetro

Indicando *lang=en* Resultado:

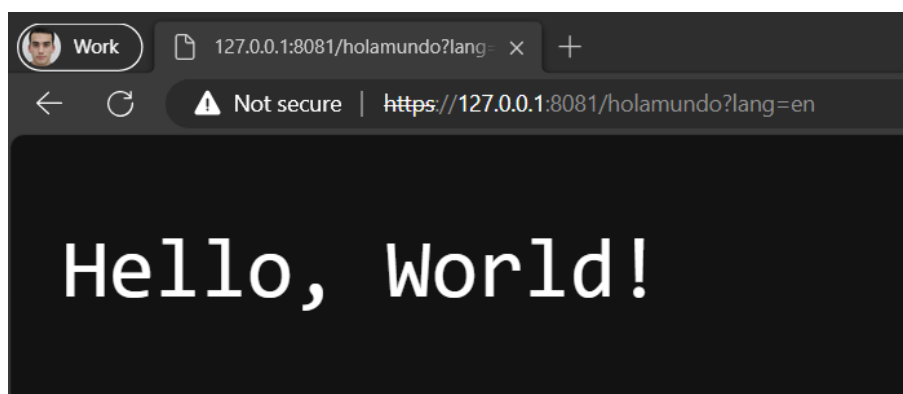


Figura 31: Captura de pantalla del resultado indicando *en* como parámetro

Indicando *lang=fr* Resultado:



Figura 32: Captura de pantalla del resultado indicando *fr* como parámetro

Indicando *lang=de* Resultado:



Figura 33: Captura de pantalla del resultado indicando *de* como parámetro

Registro de peticiones Podemos comprobar si el módulo *Logger* está funcionando correctamente simplemente revisando las entradas en el registro. Habrá una para cada solicitud realizada, con el valor de la variable *lang* correspondiente:

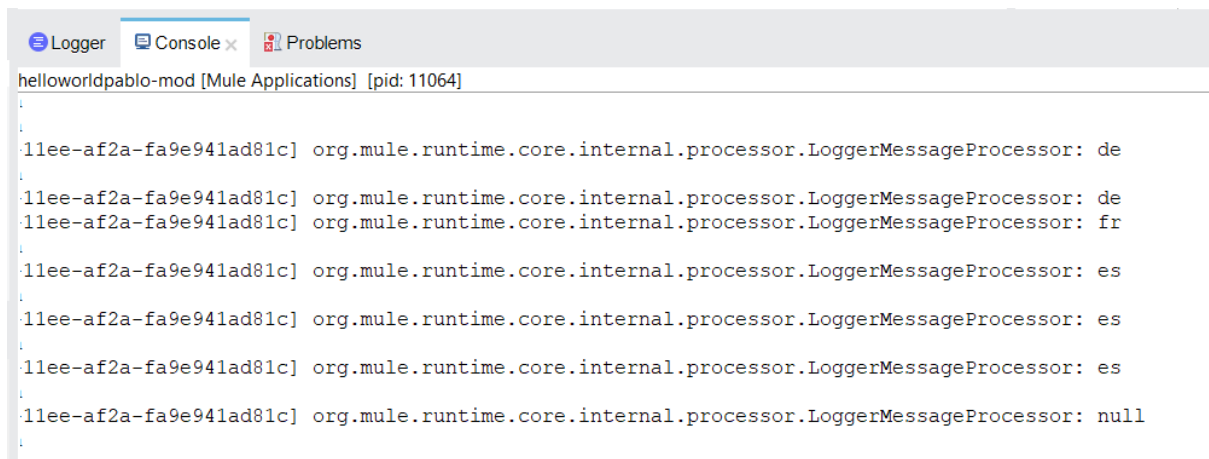


Figura 34: Registros generados por las peticiones

2.3.7. Esquema XML *MuleSoft*

```

1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <mule xmlns:tls="http://www.mulesoft.org/schema/mule/tls" xmlns:http="http://www.mulesoft.org/
  schema/mule/http"
4   xmlns="http://www.mulesoft.org/schema/mule/core"
5   xmlns:doc="http://www.mulesoft.org/schema/mule/documentation" xmlns:xsi="http://www.w3.org
  /2001/XMLSchema-instance" xsi:schemaLocation="http://www.mulesoft.org/schema/mule/core
  http://www.mulesoft.org/schema/mule/core/current/mule.xsd
6 http://www.mulesoft.org/schema/mule/http http://www.mulesoft.org/schema/mule/http/current/mule-
  http.xsd
7 http://www.mulesoft.org/schema/mule/tls http://www.mulesoft.org/schema/mule/tls/current/mule-
  tls.xsd">
8   <http:listener-config name="HTTP_Listener_config" doc:name="HTTP Listener config" doc:id="
  8044be58-0bd7-4b79-b7dd-742cb50fc294" >
9     <http:listener-connection host="localhost" port="8081" protocol="HTTPS">
10       <tls:context >
11         <tls:key-store type="jks" path="keystore.jks" keyPassword="123456" password="123456"
  alias="key-alias"/>
12       </tls:context>
13     </http:listener-connection>
14   </http:listener-config>
15   <flow name="helloworldpablo-modFlow" doc:id="4806d7ad-8e6f-4fda-9005-1e96d286a675" >
16     <http:listener doc:name="Listener" doc:id="cfb18e4d-1bea-4775-833f-c4f349878f50" config-
  ref="HTTP_Listener_config" path="/holamundo"/>
17     <set-variable value="#[attributes.queryParams.lang]" doc:name="Set Variable" doc:id="457
  ef8b4-9f32-4e6a-ade1-d021dc79ce50" variableName="lang" mimeType="text/plain" encoding="UTF
  -8"/>
18     <choice doc:name="Choice" doc:id="228e7216-fca4-4885-a366-7aa5b233eeb6">
19       <when expression="#[vars.lang == 'es']">
20         <set-payload value="¡Hola, mundo!" doc:name="Set Payload" doc:id="0d37a915-3db6-453d-
  a102-225202267020" mimeType="text/plain" encoding="UTF-8" />
21       </when>
22       <when expression="#[vars.lang == 'en']">
23         <set-payload doc:name="Set Payload" doc:id="d9ae812c-33c3-4db0-8cb5-e0e05e46cc9d"
  value="Hello, World!"/>
24       </when>
25       <when expression="#[vars.lang == 'fr']">
26         <set-payload doc:name="Set Payload" doc:id="dab65e6f-dc36-410b-a750-c91a89d6eb75"
  value="Salut le monde!"/>
27       </when>
28       <when expression="#[vars.lang == 'de']">
29         <set-payload value="Hallo, Welt!" doc:name="Set Payload" doc:id="2fcb58b1-5917-4f1c-
  b454-fb6ffd157d98" />
30       </when>
31       <otherwise >
32         <set-payload value='&lt;h1&gt;ERROR&lt;/h1&gt;&lt;p&gt;Use "es","en","fr" or "de" como
  parámetro en la URL. &lt;br&gt; Ejemplo: &lt;pre&gt;https://127.0.0.1:8081/holamundo?
  lang=en&lt;/pre&gt; &lt;/p&gt; ' doc:name="Set Payload" doc:id="96da8b59-c8a7-455a-9ca4-009
  c5650fd88" mimeType="text/html" encoding="UTF-8"/>
33       </otherwise>
34     </choice>
35     <logger level="INFO" doc:name="Logger" doc:id="ea0a7774-44f9-47df-8c98-9383f063a235"
  message="#[vars.lang]"/>
36   </flow>
37 </mule>

```

3. Conclusiones

Para concluir podría destacar la facilidad con la que, usando Mule-ESB/AnyPointStudio podemos desarrollar Servicios. A su vez, se ha comprobado como las definiciones en XML estandarizadas ayudan a replicar y compartir los desarrollos de forma sencilla; además de facilitar el control de versiones.

4. Acceso al vídeo en Youtube. Recursos disponibles

El vídeo está disponible en el siguiente enlace: <https://www.youtube.com/watch?v=mvONyOG7rQ4>

Ejemplo Práctico MuleSoft Práctica 1



Diseño y Mantenimiento de Servicios

Pablo González Troyano

A su vez, está disponible:

- El archivo XML de la definición del flujo en MuleSoft: en la sección 2.3.7, y en este enlace a GitHub⁷.
- El archivo `.jar` del proyecto Mule, en este enlace a GitHub⁸.

⁷<https://github.com/gonzaleztroiano/CGIS.DMS/blob/main/helloworldpablo-mod/helloworldpablo-mod/src/main/mule/helloworldpablo.xml>

⁸<https://github.com/gonzaleztroiano/CGIS.DMS/blob/main/helloworldpablo-mod/helloworldpablo-mod.jar>

5. Bibliografía

MuleSoft. (2023a). *Build your first Hello Mule application*. Consultado el 28 de septiembre de 2023, desde <https://developer.mulesoft.com/tutorials-and-howtos/getting-started/hello-mule/>

MuleSoft. (2023b). *Download and Install Anypoint Studio | MuleSoft Documentation*. Consultado el 26 de septiembre de 2023, desde <https://docs.mulesoft.com/studio/7.12/to-download-and-install-studio>

URJC. (2023). Guía de Estudio de la asignatura Diseño y Mantenimiento de Servicios.