Aplicaciones con Enfoque Orientado a Servicios

Tema Nº8:Trabajando con Mule ESB - I

Indicador de logro Nº8:Desarrolla aplicaciones distribuidas utilizando Mule ESB

**TEMA 01 Teoría de los**

Imagen que contiene Icono

Descripción generada automáticamente

**TEMA Nº8:**

Trabajando con Mule ESB - I

**Subtema 8.1:**

Introducción a Mule ESB

Mule ESB es un ESB creado por la compañía Mulesoft basado en el lenguaje de programación Java. Ofrece herramientas para la integración de sistemas a través de tecnologías web.

Desde el punto de vista del desarrollador, este software ofrece una interfaz amigable, basada en “cajitas”, sobre la que es sencillo y rápido programar. Si bien es cierto que son necesarios conocimientos de Java y del funcionamiento de servicios web, se trata de un lenguaje muy intuitivo basado en el meta-lenguaje XML y una interfaz gráfica que nos ayuda a entender de un solo vistazo el funcionamiento de un servicio.

**Subtema 8.2:**

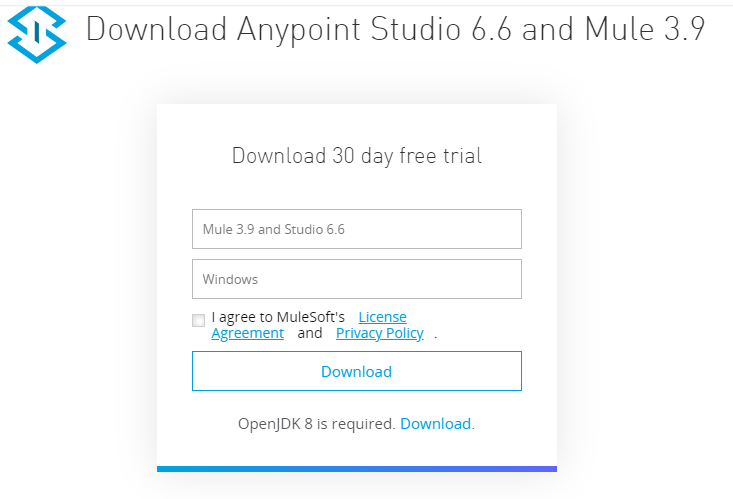
Instalación de herramientas

Las herramientas que usaremos son: Anypoint Studio y Mule StandAlone.

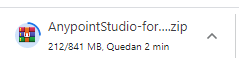
**Anypoint Studio** es el entorno de desarrollo integrado (IDE – *Integrated Development Environment*) en el que programaremos nuestras aplicaciones en Mule ESB. Éste tiene ciertas características que permiten programar flujos de manera visual.

***Nota*:** Como pre requisito, debe tener instalado Java jdk 8 y Maven.

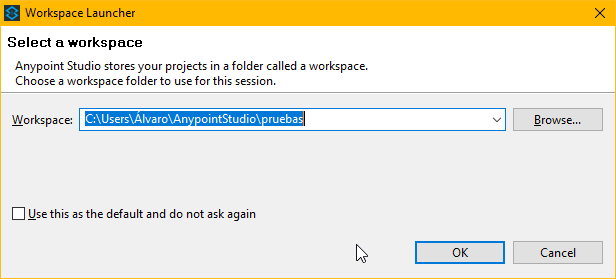
1. Ingresamos a <https://www.mulesoft.com/lp/dl/studio/previous>
2. Seleccionamos la opción Mule 3.9 and Studio 6.6 y el sistema operativo del equipo donde realizaremos la instalación.



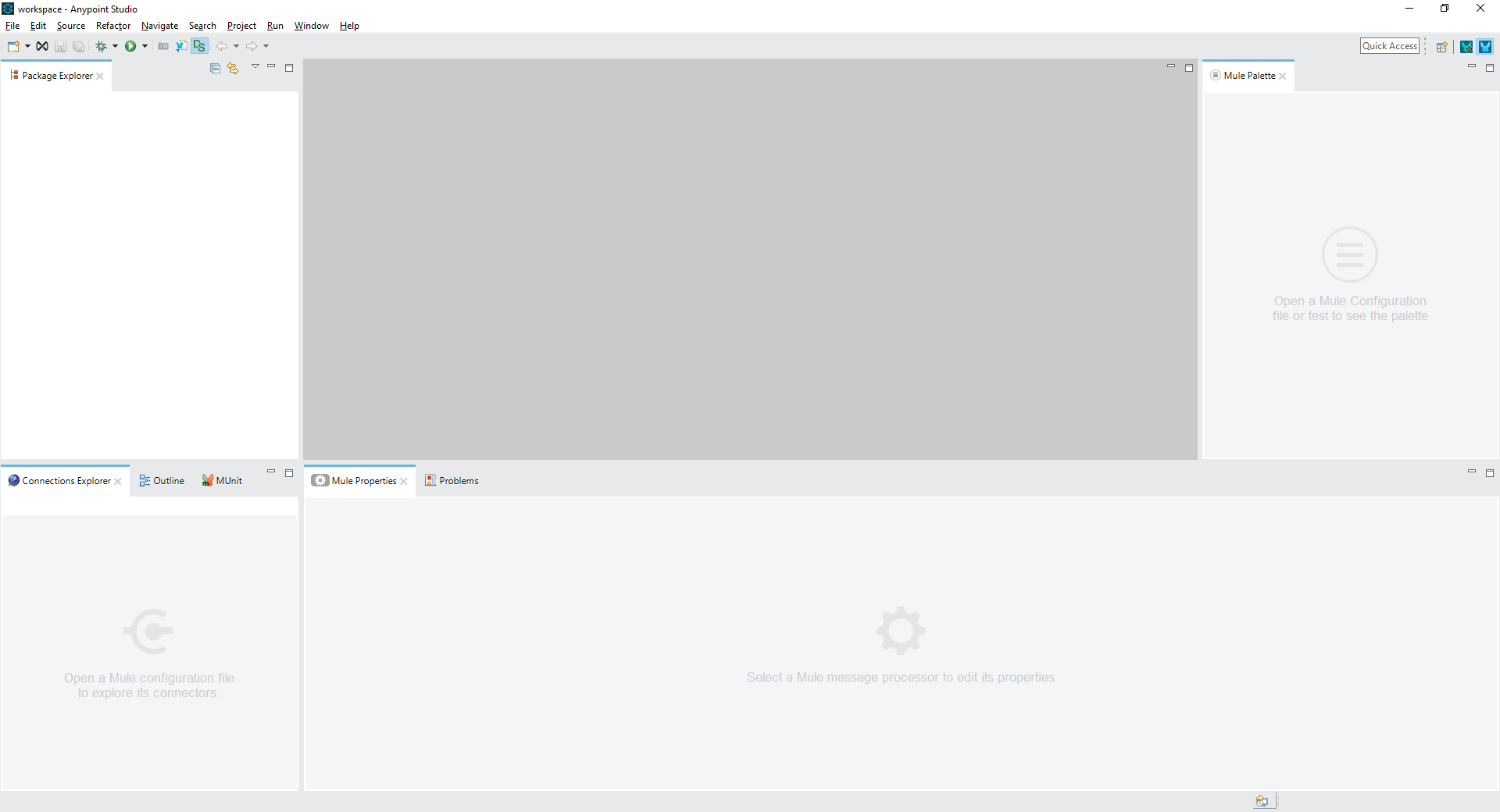
1. Marque la casilla de acuerdos de licencia y haga clic en el botón *Download*.



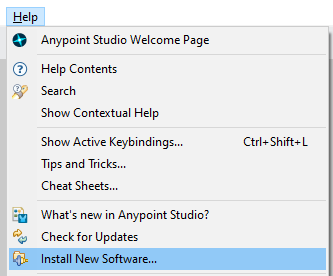
1. Luego de finalizada la descarga, descomprima la carpeta en la ubicación de su preferencia. (Por ejemplo: C:\AnyPointStudio)
2. Ejecute el archivo de nombre **AnypointStudio** ubicado en la carpeta donde extrajo los archivos.
3. Seleccione un espacio de trabajo *workspace* para almacenar sus proyectos. Puede ser un nuevo directorio o uno ya existente.



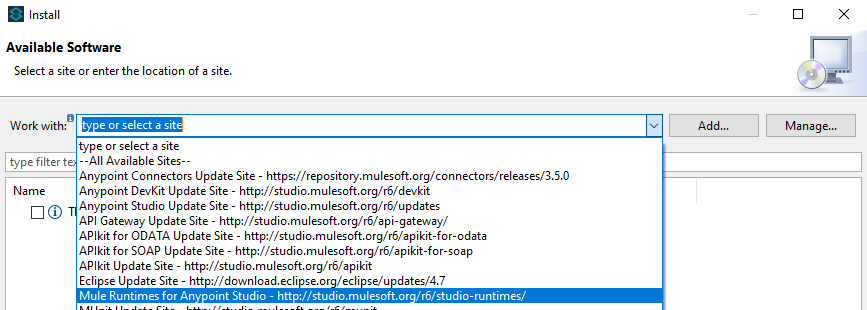
1. Verifique su espacio de trabajo



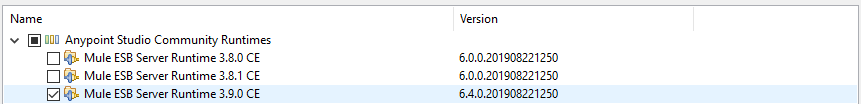
1. Para instalar la versión **Community**, vaya al menú *Help > Install new software*



1. En el campo *Work with*, seleccione la opción *Mule Runtimes for Anypoint Studio…*



1. Luego que aparezcan las versiones el Runtime que puede instalar, seleccione la última de la sección *Anypoint Studio Community Runtimes.* En este caso la última versión es la **3.9.0** CE.

**

1. Haga clic en *Next* y acepte los términos y condiciones para continuar con la instalación.
2. Reinicie la aplicación.

**Subtema 8.3:**

Mule Expression Language (MEL)

MEL es un lenguaje de expresión liviano y específico de Mule que puede usar para acceder y evaluar los datos en la carga útil, las propiedades y las variables de un mensaje de Mule. Accesible y utilizable desde prácticamente todos los procesadores de mensajes de Mule, MEL le permite filtrar, enrutar o actuar de manera rápida sobre las diferentes partes del objeto de mensaje de Mule.

**Variables**

Mule tiene variables importantes para la ejecución de los flujos están contenidas en un objeto llamado *message* el cual agrupa toda esa información.

* payload

Contiene la carga útil del mensaje que se está procesando. Es muy importante porque reflejan el resultado de las invocaciones a servicios web, el resultado de transformaciones, etc. Por ejemplo, si hacemos una petición a un web service externo, el body de la petición realizada se quedará almacenado en el payload.

* flowVars

Las variables de flujo son aquellas cuyo ciclo de vida es lo que dura la ejecución del flujo donde se encuentra. Estas variables son débilmente tipadas, así que podemos asignarle valores sin importar su tipo, ya que, aunque tienen tipo puede cambiar a lo largo de su ciclo de vida.

* sessionVars

Las variables de sesión son variables globales, es decir, se puede acceder a ellas desde cualquier flujo en cualquier momento. También son débilmente tipadas.

* exception

Se trata del objeto que contiene las excepciones que se producen en el flujo y solo están accesibles cuando se haya producido una excepción, el resto del tiempo estarán vacías.

* inbound properties

Se trata de información que viene contenida en la petición (request), específicamente de la cabecera de un mensaje HTTP. Estas propiedades son inmutables, no pueden cambiar a lo largo de la ejecución del flujo y son un mapa (par de clave y valor), así que podremos encontrar el valor de una cabecera conociendo la clave de la misma.

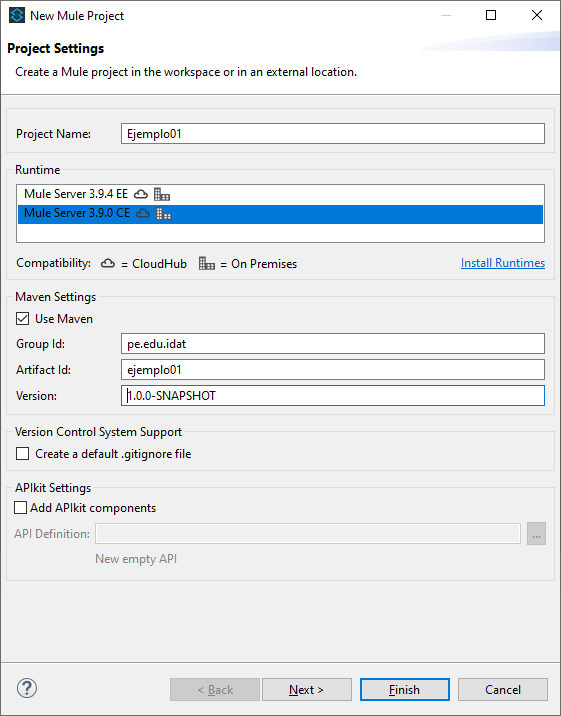
* outbound properties

Son elementos que se añaden cuando hacemos una request a un sistema externo. Por ejemplo, si quisiéramos mandar una cabecera a un servicio web podríamos añadirlo al mapa de *outbound properties*, indicando la clave y el valor.

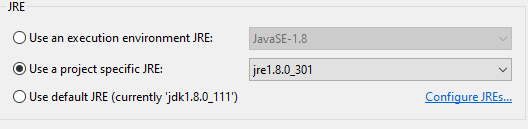
**Ejemplos:**

**Creación de proyecto**

1. Haga clic en el menú *File > New > Mule Project*.
2. Configuramos nuestro proyecto y hacemos clic en *Next*:



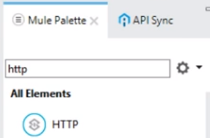
1. Seleccionamos **jdk1.8.x** y hacemos clic en *Next*.



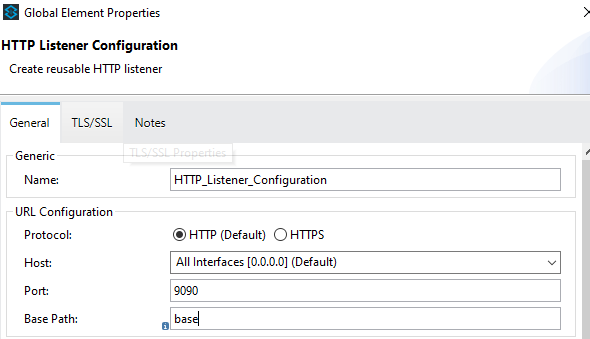
1. Hacemos clic en *Finish*.

**Trabajando con componentes**

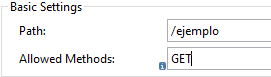
1. De la paleta de componentes, seleccionamos HTTP y arrastramos el lienzo en blanco. Este componente permitirá “escuchar” el request de una invocación a un HTTP.



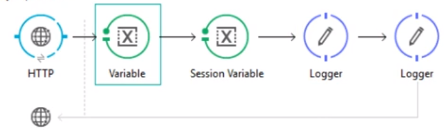
1. En la opción de *General Settings > Connector Configuration*, hacemos clic en el botón **+** y realizamos la siguiente configuración.



1. En *Basic Settings*, configuramos:

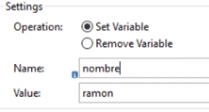


1. De la paleta arrastramos un control *Variable, un control Session Variable y dos controles* Logger.

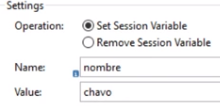


1. Ahora configuramos cada control.

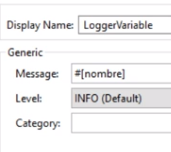
* Variable



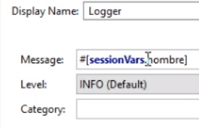
* Session Variable



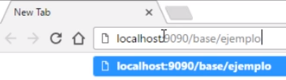
* Logger



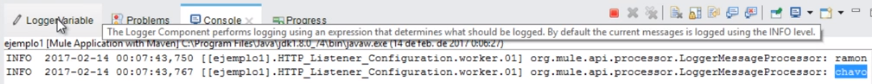
* Logger



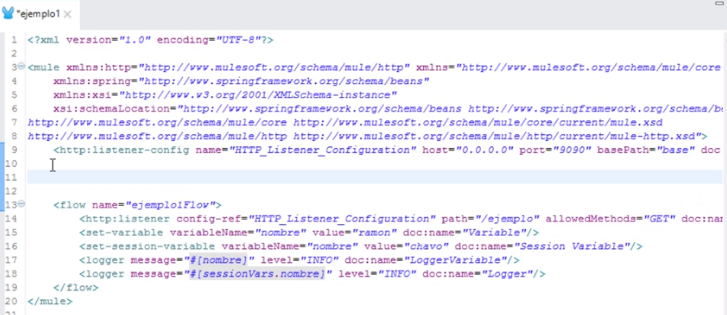
1. Ejecute la aplicación haciendo clic derecho a la aplicación y seleccione *Run As > Mule Application with Maven*.
2. Abra su navegador e ingrese la URL [*http://localhost:9090/base/ejemplo*](http://localhost:9090/base/ejemplo)



1. Verificamos en la consola los mensajes de log.



1. Abra el archivo en la vista XML.



**Subtema 8.4:**

Flujos y colas de mensajería

El código de los programas escritos en Mule también deben considerar una estructura. De forma similar a otros lenguajes de programación donde existen funciones, métodos o procedimientos para agrupar el código.

En Mule existen los siguientes elementos:

**Flujos**

Aquí es donde comienzan las ejecuciones de los servicios/programas, y se pueden dividir en dos categorías:

* flow

Los flujos deben tener un conector de entrada, es decir, el punto donde se inicia la ejecución, y podrán definir capturadores de excepciones.



* subflow

Es una agrupación de código que no puede iniciarse desde el exterior, sólo puede utilizarse si se hace referencia a éste desde otro flujo o subflujo. Además no pueden definirse manejadores de excepciones para los subflujos, así que cuando se produce una excepción, esta es capturada por el manejador del flujo que llame a este subflujo.



**Colas**

Una cola es un área de almacenamiento temporal para un mensaje. Puede crear colas, colas FIFO y colas de mensajes no entregados en Anypoint Platform. También puede enviar y recibir mensajes desde una cola en Anypoint Studio. Tenemos:

* JMS

El JMS *endpoint* permite que las aplicaciones de Mule envíen y reciban mensajes a las colas a través la API de Java Message Service (JMS).



* VM

El *Memory endpoint* (VM) permite que las aplicaciones Mule utilicen comunicaciones intra-JVM.

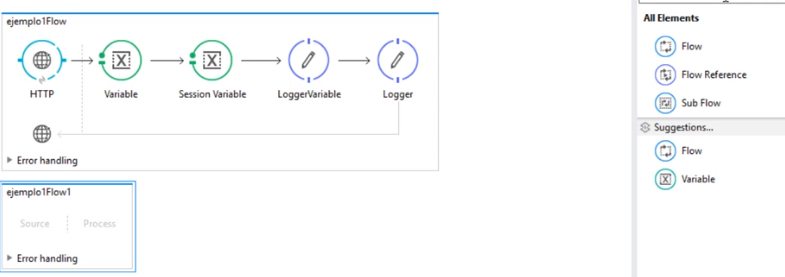


**Ejemplos:**

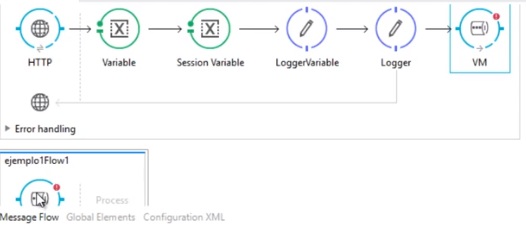
**Flujo**

A partir del ejemplo inicial (MEL), extenderemos el flujo para que invoque a un segundo *endpoint*. Tenga en cuenta que las variables de flujo solo existen en su correspondiente flujo, mientras que las variables de sesión existen en todo el programa.

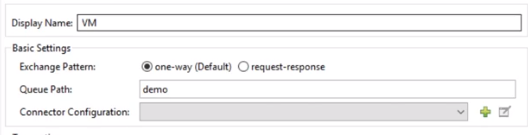
1. De la paleta, seleccione un control *Flow* (flujo) y lo agregamos debajo del primer flujo.



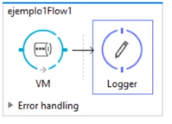
1. Agregue un componente *VM* al final del primer flujo y otro al inicio del segundo flujo.



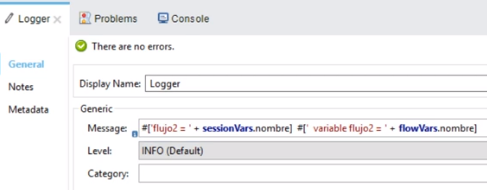
1. A ambos componentes coloque como *Queue Path* (Ruta de cola) el nombre *demo*.



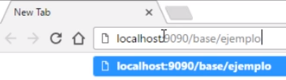
1. Añada un componente Logger al final del segundo flujo



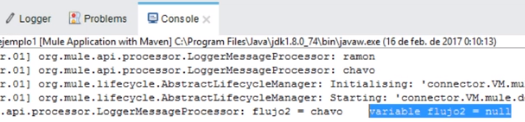
1. Configure el componente para que muestre el valor de la variable de sesión creada en el primer flujo. Además, agregue el código para intentar mostrar la variable de flujo.



1. Ejecute la aplicación haciendo clic derecho a la aplicación y seleccione *Run As > Mule Application with Maven*.
2. Abra su navegador e ingrese la URL [*http://localhost:9090/base/ejemplo*](http://localhost:9090/base/ejemplo)

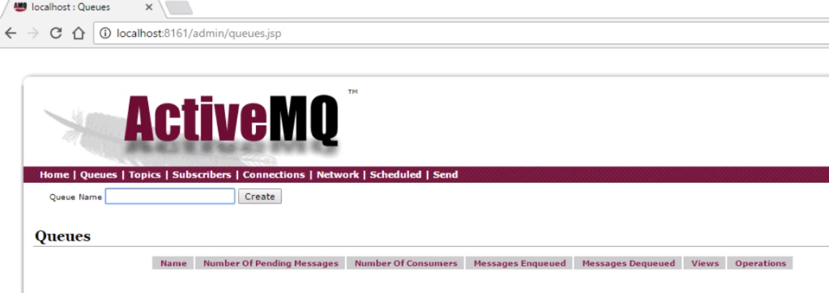


1. Verificamos en la consola los mensajes de log.



**Colas – Enviando mensaje**

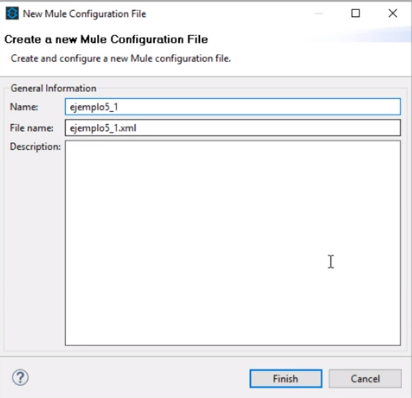
1. Inicie el servicio Apache ActiveMQ



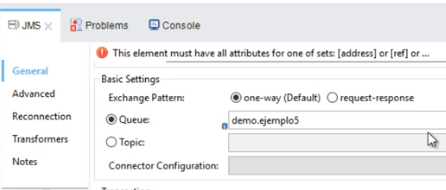
1. Cree una cola.



1. En *Anypoint Studio*, cree un nuevo archivo de configuración de Mule *(New Mule Configuration File)*



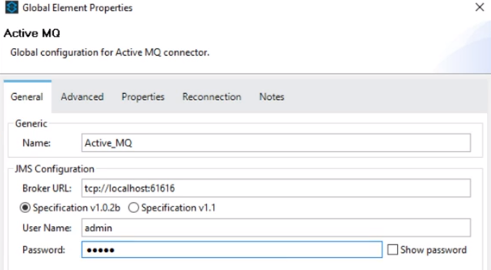
1. En el flujo principal, añada un control HTTP con su correspondiente nombre de *Path* y método permitido: *POST*.
2. Añada un componente *Logger* para mostrar la entrada
3. Añada un componente *JMS* y configúrelo con el nombre de cola (*queue*) que creó en ActiveMQ



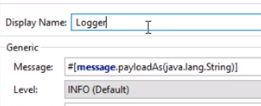
1. En la opción *Connector Configuration* del *JMS*, haga clic en el botón **+** y seleccione la opción **ActiveMQ**



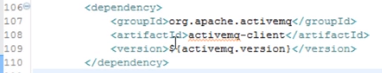
1. Termine de configurar el ActiveMQ con el usuario y clave del mismo.



1. Configure el control *Logger* para mostrar el JSON de entrada.



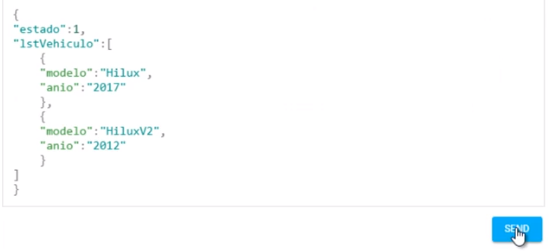
1. Agregamos la dependencia de **activemq** al archivo pom.xml.



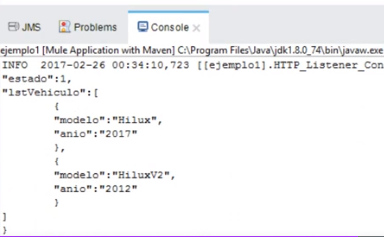
1. Agregue la versión de **activemq** en la sección properties.



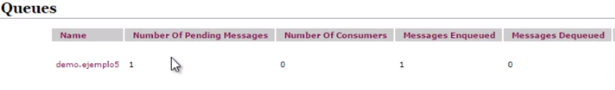
1. Ejecute su aplicación.
2. A través de Postman, envíe un mensaje en formato JSON



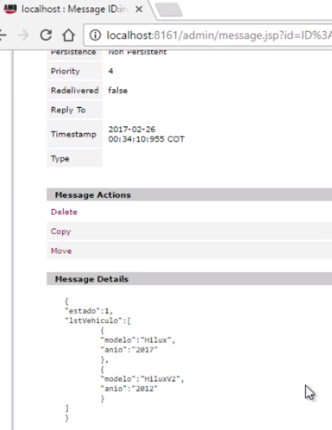
1. Verifique en la consola el mensaje enviado.



1. Verifique la cola de mensajes en **ActiveMQ**.

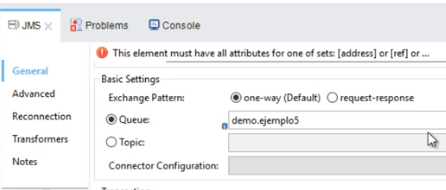


1. Seleccione la cola y verifique el mensaje.

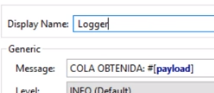


**Colas – Consumiendo mensaje**

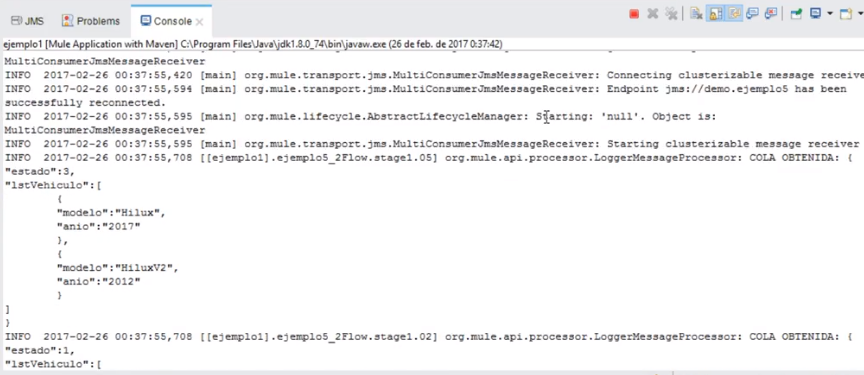
1. En *Anypoint Studio*, cree un nuevo archivo de configuración de Mule (New Mule Configuration File)
2. En el flujo principal, añada un componente JMS y configúrelo con el nombre de cola (queue) que creó en ActiveMQ y que servirá para consumir los mensajes.



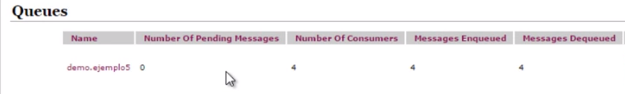
1. En la opción *Connector Configuration* del JMS, seleccione Active\_MQ.
2. Añada un componente *Logger* y configure para mostrar el *payload*.



1. Ejecute su aplicación.
2. Verifique en la consola que se hayan leído todos los mensajes de la cola.



1. Verifique en **ActiveMQ** si tiene mensajes pendientes de leer en la cola.



**Actividad:**

1. Elabore un proyecto que muestre en el navegador y en la consola un mensaje de saludo que incluya la fecha del sistema.
2. Elabore un proyecto que reciba un JSON y según el valor de un determinado campo haga lo siguiente:
   1. muestre el contenido en la consola, o
   2. modifique el contenido y lo asigne en el *payload*, o
   3. envíe el contenido como un mensaje a una cola en ActiveMQ.