

Mule Studio: una herramienta gráfica para Mule ESB

Itinerario Formativo de Doctorado 7009

Juan Boubeta Puig

Grupo UCASE de Ingeniería del Software
Departamento de Ingeniería Informática

14 de mayo de 2013



Índice

- 1 Introducción
- 2 Instalación de Mule Studio
- 3 Visión general de la herramienta
 - Creación de un proyecto Mule Studio
 - Paleta del editor
- 4 Esquema de un flujo típico de Mule
- 5 Caso de estudio



Índice

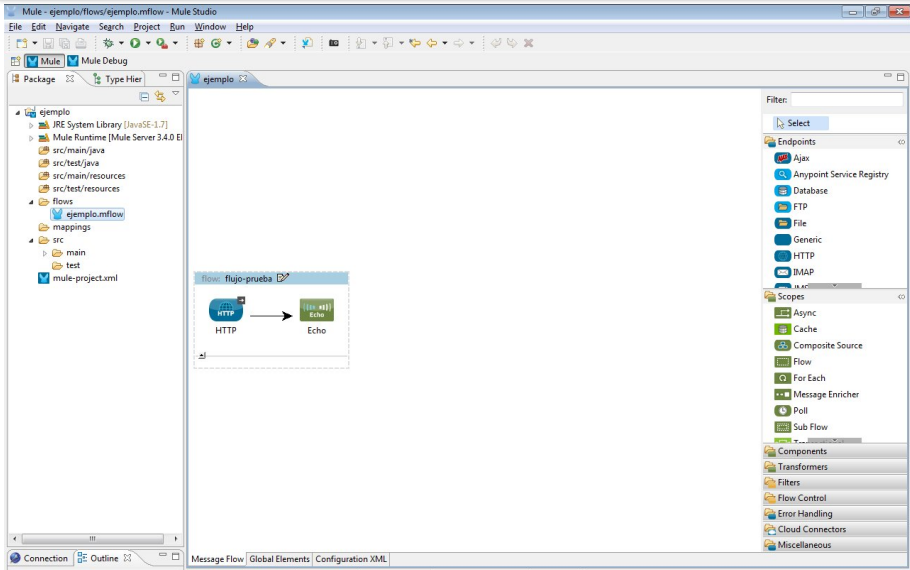
- 1 **Introducción**
- 2 Instalación de Mule Studio
- 3 Visión general de la herramienta
 - Creación de un proyecto Mule Studio
 - Paleta del editor
- 4 Esquema de un flujo típico de Mule
- 5 Caso de estudio



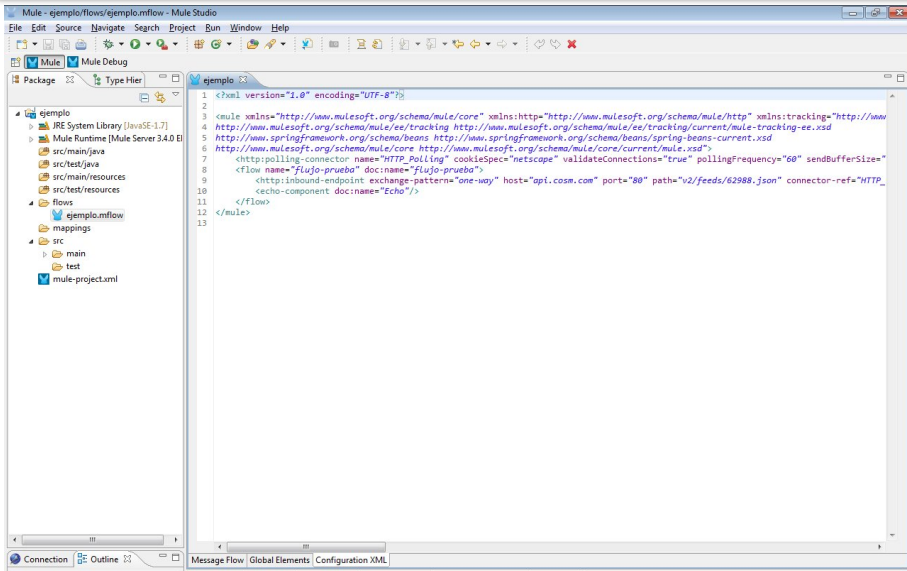
¿Qué es Mule Studio?

- Interfaz gráfica que abstrae al usuario de los detalles más técnicos de Mule ESB.
- En lugar de tener que escribir “a mano” el código XML para crear aplicaciones Mule; Mule Studio se encarga de ello.
- Los elementos necesarios para modelar y configurar aplicaciones Mule se incorporan al canvas del editor mediante *drag and drop*.
- Una aplicación Studio puede ser incluso desplegada en la nube (véase *CloudHub* para más información).
- Está basado en Eclipse y proporciona dos entornos de desarrollo que pueden utilizarse para crear aplicaciones Mule:
 - Un editor *drag and drop* visual.
 - Un editor XML.
- Lo que se desarrolle o configure en uno de los editores se actualizará automáticamente en el otro.

Mule Studio - Editor visual



Mule Studio - Editor XML



Índice

- 1 Introducción
- 2 Instalación de Mule Studio
- 3 Visión general de la herramienta
 - Creación de un proyecto Mule Studio
 - Paleta del editor
- 4 Esquema de un flujo típico de Mule
- 5 Caso de estudio



Requisitos de software y hardware

Hardware

- 3GB de RAM
- 2GHz de CPU
- 4GB libres de espacio de disco

Software

- *Java Runtime Environments:*
 - Oracle Java 1.6
 - Oracle Java 1.7
 - IBM Java 1.6
- *Sistemas operativos:*
 - Windows (32 o 64 bit)
 - Mac OS (32 o 64 bit)
 - Linux (32 o 64 bit)

Pasos para instalar y ejecutar Mule Studio

- 1 Descargar la versión Mule Studio v3.4 para Windows, Linux o Mac:
<http://www.mulesoft.org/download-mule-esb-community-edition>.
- 2 Descomprimir el archivo *MuleStudio-for-*.zip* en un directorio cuya ruta no sea muy extensa.
- 3 Una vez descomprimido, ejecutar el fichero MuleStudio:
MuleStudio.exe (Windows), MuleStudio.app (Mac OSX) o MuleStudio (Linux).

También puede descargarse e instalarse Mule Studio como plugin de Eclipse. Más información en: <http://www.mulesoft.org/documentation/display/current/Studio+in+Eclipse>

Uso de Mule Studio con sistema de control de versiones

- **Subclipse:** <http://www.mulesoft.org/documentation/display/33X/Using+Subversion+with+Studio>.
- **Git:** <http://www.mulesoft.org/documentation/display/33X/Using+Git+with+Studio>.

Es necesario registrarse en la web para poder acceder a dicha información.

Índice

- 1 Introducción
- 2 Instalación de Mule Studio
- 3 Visión general de la herramienta**
 - Creación de un proyecto Mule Studio
 - Paleta del editor
- 4 Esquema de un flujo típico de Mule
- 5 Caso de estudio



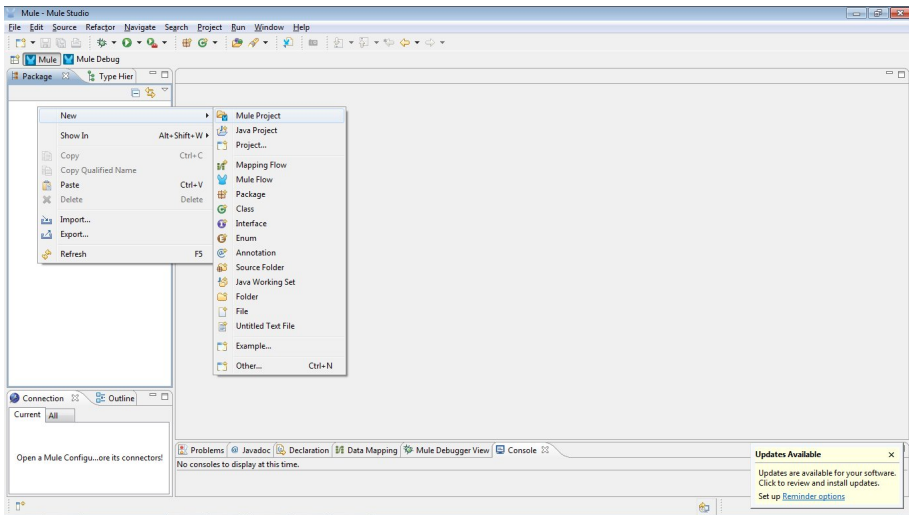
Índice

- 1 Introducción
- 2 Instalación de Mule Studio
- 3 **Visión general de la herramienta**
 - Creación de un proyecto Mule Studio
 - Paleta del editor
- 4 Esquema de un flujo típico de Mule
- 5 Caso de estudio

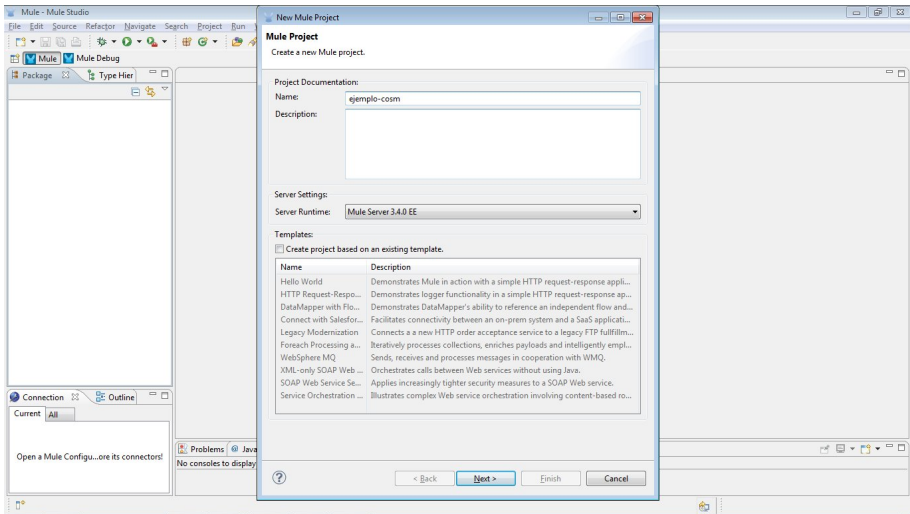


Creación de un proyecto Mule Studio

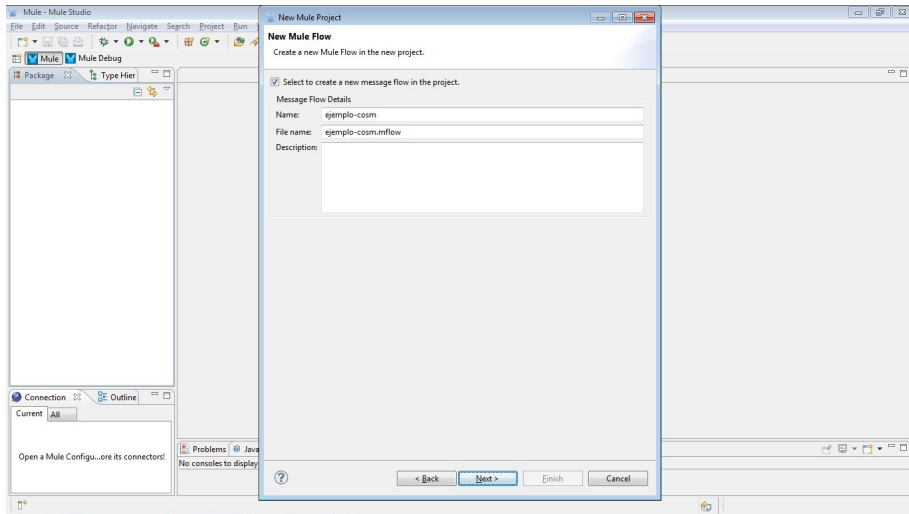
New > Mule Project



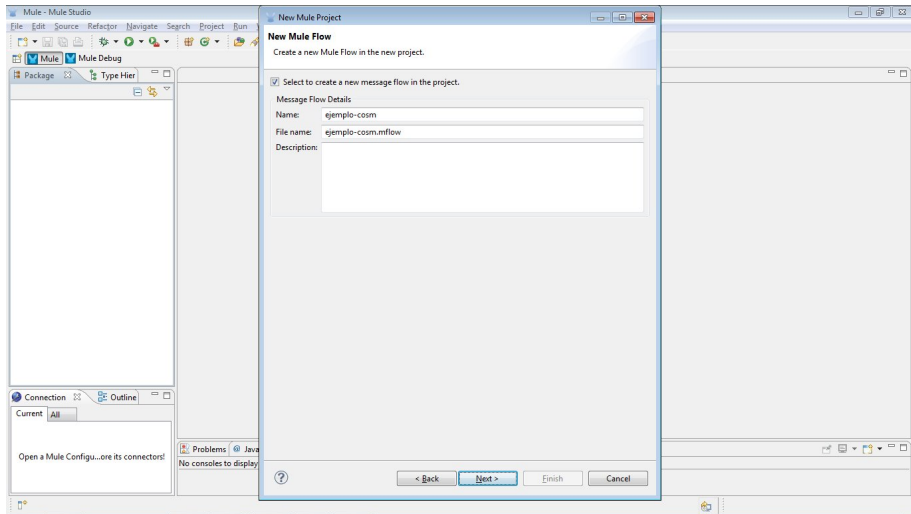
Creación de un proyecto Mule Studio (cont.)



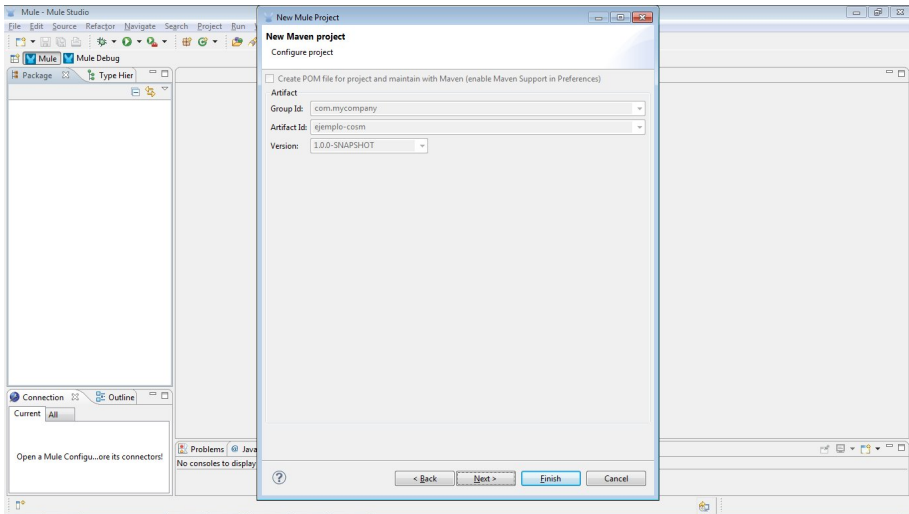
Creación de un proyecto Mule Studio (cont.)



Creación de un proyecto Mule Studio (cont.)

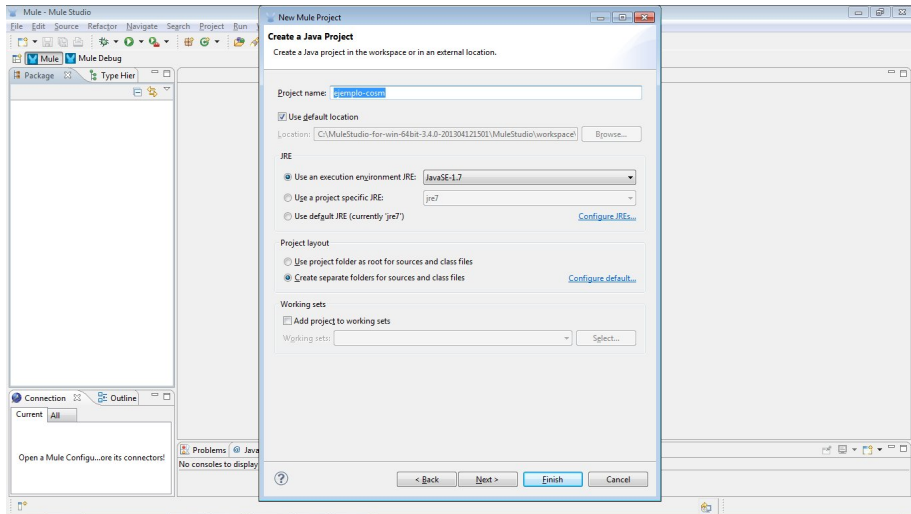


Creación de un proyecto Mule Studio (cont.)



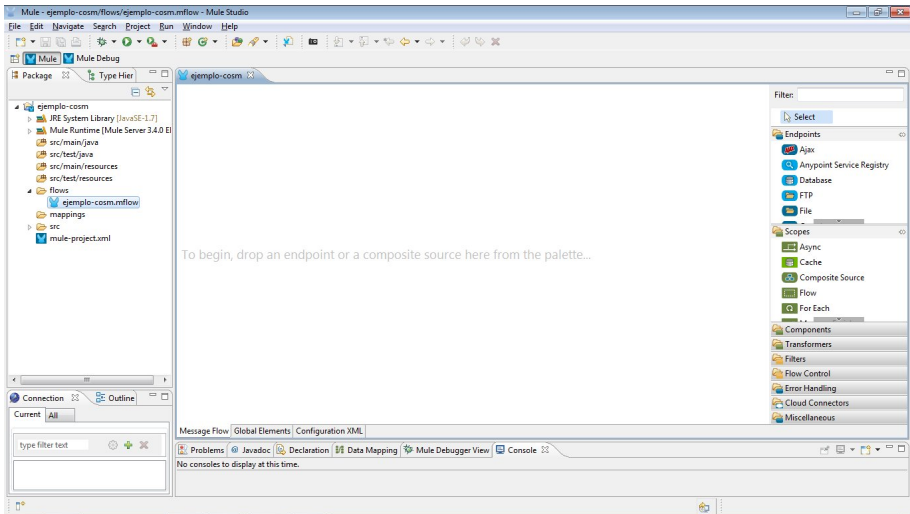
Creación de un proyecto Mule Studio (cont.)

Hacer clic en *Finish*



Creación de un proyecto Mule Studio (cont.)

Hacer clic en *Finish*



Índice

- 1 Introducción
- 2 Instalación de Mule Studio
- 3 **Visión general de la herramienta**
 - Creación de un proyecto Mule Studio
 - Paleta del editor
- 4 Esquema de un flujo típico de Mule
- 5 Caso de estudio

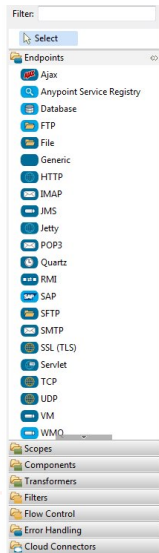


Endpoints

Permiten que las aplicaciones Mule puedan comunicarse con el “mundo” exterior. Se clasifican en:

Inbound La aplicación recibirá información del exterior.

Outbound La aplicación enviará información al exterior.

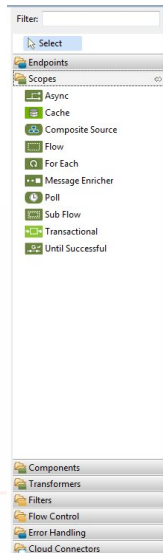


Scopes

Proporcionan diferentes formas de combinar (agrupar) varios procesadores de mensajes con el objetivo de:

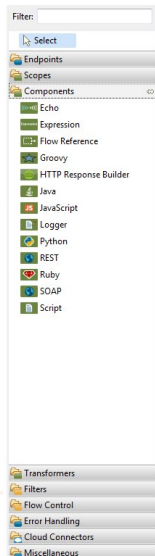
- Mejorar la legibilidad del código XML.
- Implementar procesamiento paralelo.
- Crear secuencias de bloques reusables.

Denominaremos “procesadores de mensajes” a los bloques que permiten filtrar, enriquecer, encaminar o validar los mensajes.



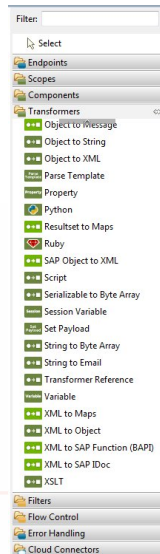
Components

- Añaden funcionalidad a un flujo como *logging* e impresión por pantalla.
- Además, también facilitan la integración *Software as a Service* (SaaS) proporcionando “shells” específicos de lenguaje que permiten definir una lógica de negocio con código personalizado para las aplicaciones Mule.
- Un componente recibe, procesa y devuelve mensajes.
- Es un objeto en el que uno de sus métodos será invocado cuando reciba un mensaje.



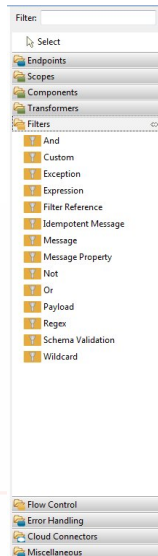
Transformers

Se encargan de transformar o enriquecer los mensajes (cabecera y cuerpo del mensaje).



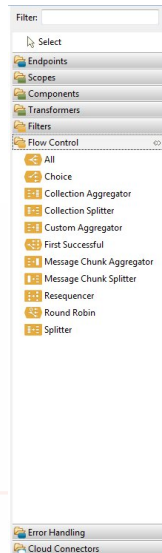
Filters

Determinan si un mensaje puede continuar a través del flujo de la aplicación, o si debe rechazarse.



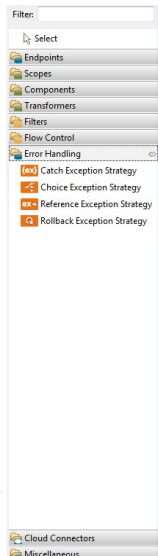
Flow controls

- Especifican cómo los mensajes serán encaminados hacia distintos procesadores de mensajes dentro de un flujo.
- También pueden procesar mensajes (agregación, separación...) antes de encaminarlos a otros procesadores de mensajes.



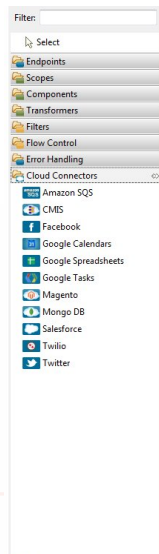
Error handlers

Ofrecen varios procedimientos para manejar excepciones bajo ciertas circunstancias.



Cloud connectors

- Permiten la integración de aplicaciones Mule con aplicaciones web y API de terceros.
- Ejemplos: Salesforce (*Customer Relationship Management*) y Mongo DB (SGBD NoSQL).



Índice

- 1 Introducción
- 2 Instalación de Mule Studio
- 3 Visión general de la herramienta
 - Creación de un proyecto Mule Studio
 - Paleta del editor
- 4 Esquema de un flujo típico de Mule
- 5 Caso de estudio



Esquema de un flujo típico de Mule I

- 1 Una fuente de mensajes:** uno o más *endpoints* activan el flujo cada vez que llega un mensaje.
- 2 Un filtro:** puede ser embebido en la fuente de mensajes o conectado a esta fuente; debe identificar mensajes inválidos y rechazar su paso al resto del flujo.
- 3 Un transformador:** puede convertir los mensajes de entrada en un formato de datos consumible por otros procesadores de mensajes del flujo.
- 4 Un enriquecedor de mensajes:** puede añadir información relevante en un mensaje. Por ejemplo, si el mensaje llega solamente con el DNI de una persona, podría añadirse al mensaje su nombre y apellidos.

Esquema de un flujo típico de Mule II

- 5 Un componente:** una vez preparado el mensaje para ser procesado, normalmente será enviado a un componente que se encargará de procesarlo de una determinada forma según su contenido. A veces también se utilizan BD externas o API (ej. Salesforce) como *cloud connectors*.
- 6** Los últimos “pasos” de un flujo pueden ser muy distintos, por ejemplo:
 - Se devuelve una respuesta al emisor original del mensaje.
 - Los resultados del procesamiento son almacenados en una base de datos o enviados a terceros (ej. correo electrónico).

Índice

- 1 Introducción
- 2 Instalación de Mule Studio
- 3 Visión general de la herramienta
 - Creación de un proyecto Mule Studio
 - Paleta del editor
- 4 Esquema de un flujo típico de Mule
- 5 Caso de estudio



Conceptos previos

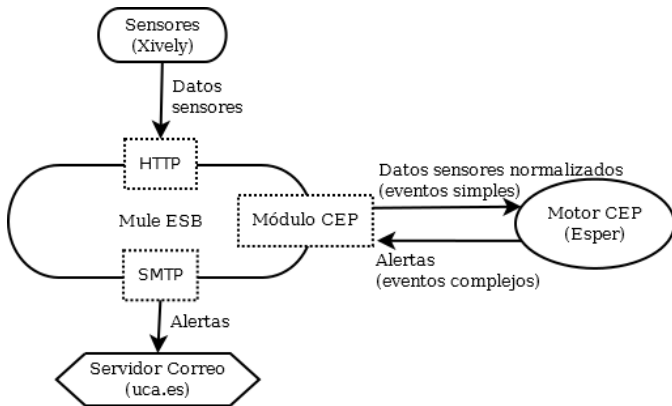
Internet de las cosas o *Internet of Things* (IoT)

- Paradigma emergente que propone el uso de una red de **cosas** u objetos, como sensores y actuadores, interconectados a nivel mundial e identificados unívocamente a través de un esquema de direcciones.
- Cada objeto puede interactuar y cooperar con los demás para alcanzar un objetivo común [Atzori *et al.*].

Procesamiento de eventos complejos o *Complex Event Processing* (CEP)

- Tecnología emergente que permite procesar, analizar y correlacionar grandes cantidades de eventos [Luckham].
- Para detectar y responder en *tiempo real* y de forma automática a las situaciones que son críticas o relevantes para los procesos de negocio.
- Se utilizan unos **patrones de eventos** que inferirán nuevos eventos más complejos y con un mayor significado semántico.

Arquitectura a implementar con Mule



Fuente: arquitectura adaptada de [Boubeta Puig *et al.*, 2011]

Fuente de mensajes (I)

Flujos de Xively sobre domótica utilizados en este caso de estudio

Nº	Nombre del flujo	País	URL	Actualiz.
F1	<i>Residential information</i>	Holanda	https://xively.com/feeds/62988/	1 min
F2	<i>HAC Center</i>	Polonia	https://xively.com/feeds/103216	1 min

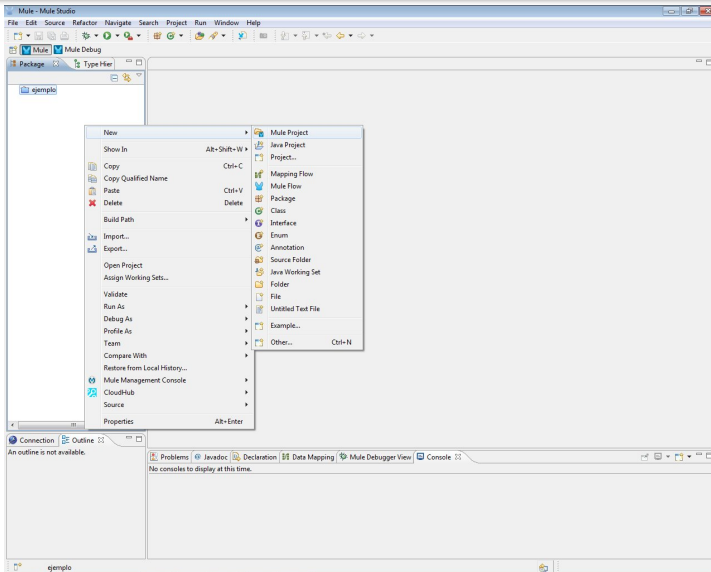


Fuente de mensajes (II)

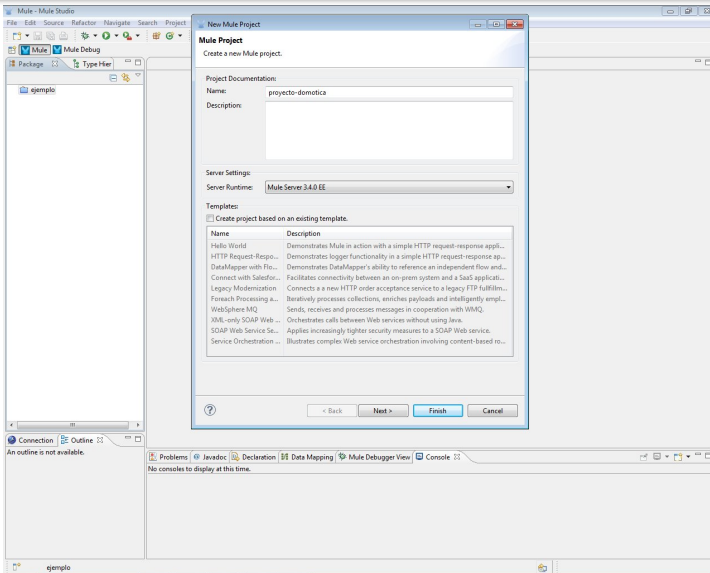
Formato de los datos de flujos normalizados

Dato	Tipo	Descripción	F1	F2
Hogar	<i>String</i>	Nombre del flujo.	x	x
Sensor	<i>String</i>	URL del flujo.	x	x
Localización	<i>String</i>	Nombre de la ciudad, país.	x	x
Latitud	<i>float</i>	Latitud de la localización.	x	x
Longitud	<i>float</i>	Longitud de la localización.	x	x
TiempoRegistro	<i>String</i>	Fecha y hora de registro del dato.	x	x
ConsumoEnergético	<i>float</i>	Consumo energético (W)	x	x
TemperaturaInterior	<i>float</i>	Temperatura interior del hogar (°C)	x	x
TemperaturaExterior	<i>float</i>	Temperatura exterior (°C)		x

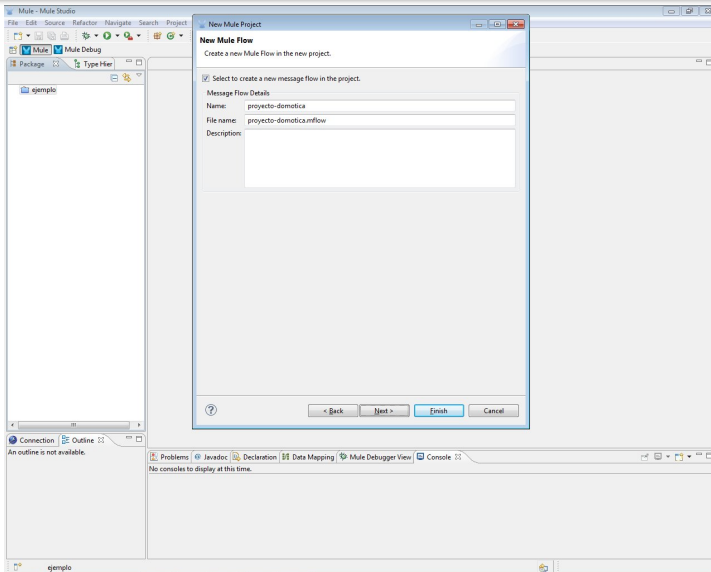
Crear un proyecto Mule



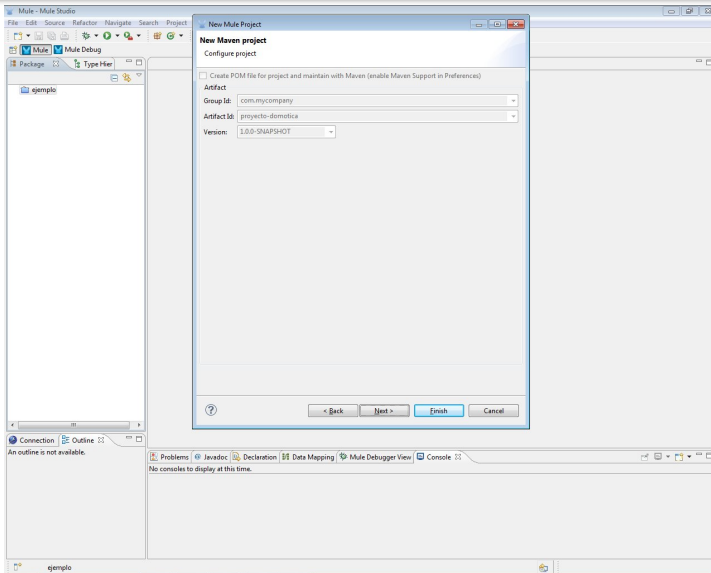
Crear un proyecto Mule (cont.)



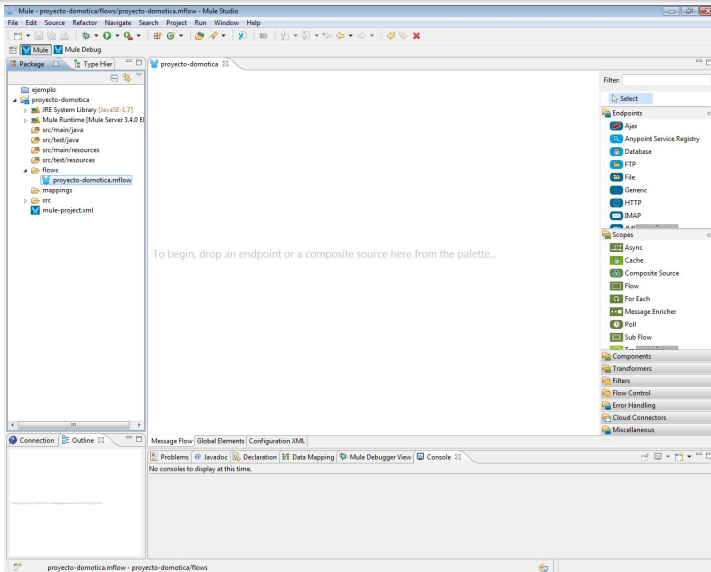
Crear un proyecto Mule (cont.)



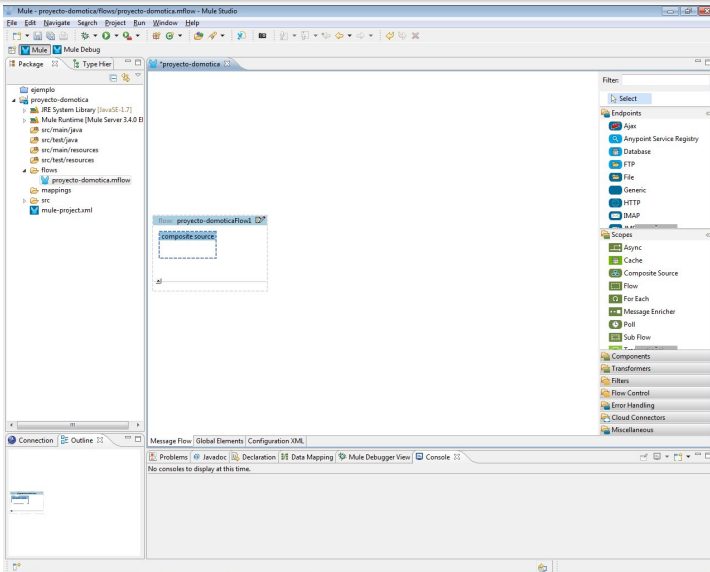
Crear un proyecto Mule (cont.)



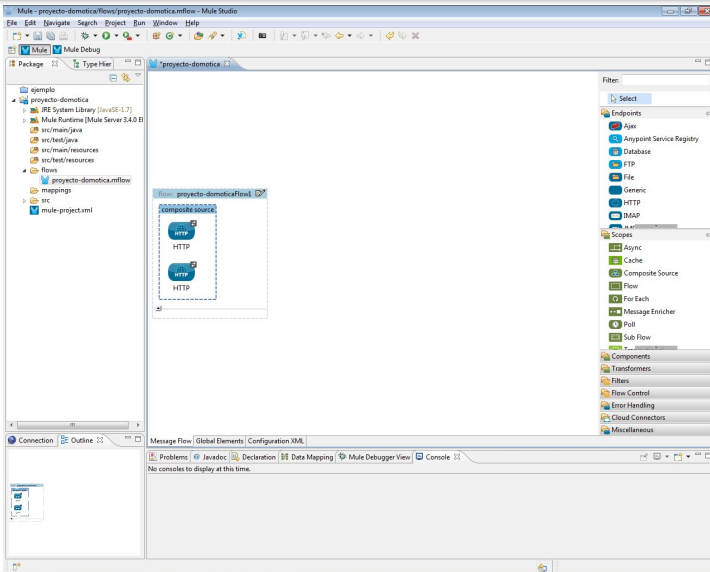
Crear un proyecto Mule (cont.)



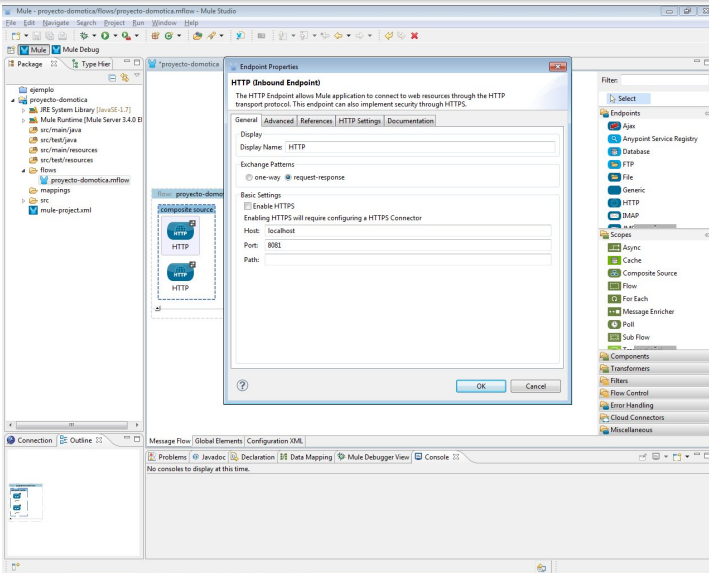
Añadir los *endpoints* de entrada (cont.)



Añadir los *endpoints* de entrada (cont.)



Añadir los endpoints de entrada (cont.)



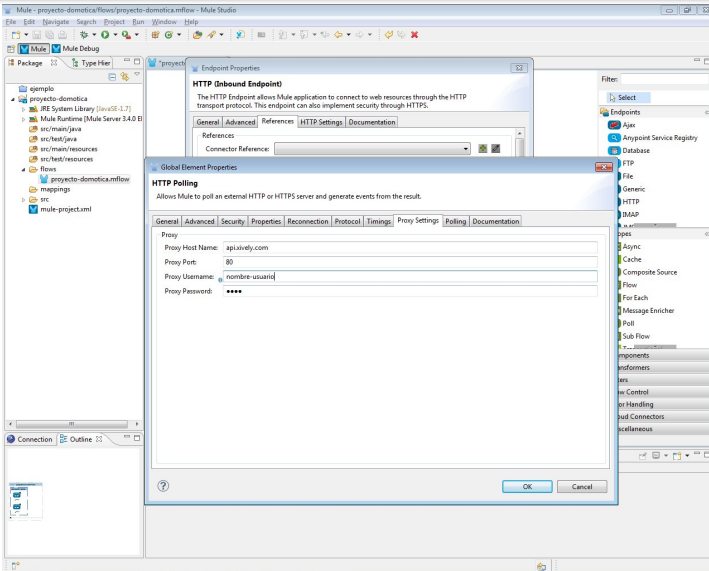
Añadir los endpoints de entrada (cont.)

The screenshot displays the Mule Studio interface with the 'Endpoint Properties' dialog box open. The dialog is titled 'HTTP (Inbound Endpoint)' and contains the following sections:

- General:** Includes a 'Display Name' field with the value 'Residential Information'.
- Exchange Patterns:** Features radio buttons for 'one-way' (selected) and 'request-response'.
- Basic Settings:**
 - Enable HTTPS: Enabling HTTPS will require configuring a HTTPS Connector.
 - Host: api.ively.com
 - Port: 80
 - Path: v2/feeds/62988.json

On the right side of the interface, the 'Component Palette' is visible, showing a list of endpoints under the 'Endpoints' category, including Ajax, Atypoint Service Registry, Database, FTP, File, Generic, HTTP, IMAP, and others.

Añadir los endpoints de entrada (cont.)



Añadir los endpoints de entrada (cont.)

The screenshot displays the Mule Studio interface with the 'Endpoint Properties' dialog box open for an 'HTTP (Inbound Endpoint)'. The dialog has several tabs: 'General', 'Advanced', 'References', 'HTTP Settings', and 'Documentation'. The 'References' tab is active, showing the following configuration:

- Connector Reference:** HTTP_Polling_1min
- Endpoint Reference:** (empty)
- Transformers References: Request**
 - Global Transformers:** (empty)
 - Transformers to be applied:** (empty)
- Transformers References: Response**
 - Global Transformers:** (empty)
 - Transformers to be applied:** (empty)

The background shows the Mule Studio workspace with a project tree on the left, a message flow diagram in the center, and a component palette on the right. The component palette includes categories like Endpoints, Scopes, Components, Transformers, Filters, Flow Control, Error Handling, and Cloud Connectors. The 'HTTP' component is visible in the palette.

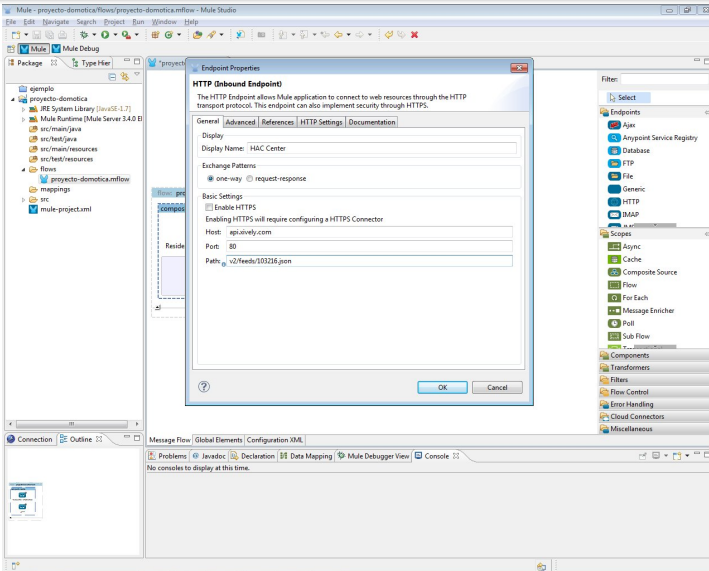
Añadir los endpoints de entrada (cont.)

The screenshot displays the Mule Studio interface with the 'Endpoint Properties' dialog box open for an 'HTTP (Inbound Endpoint)'. The dialog is divided into several tabs: General, Advanced, References, HTTP Settings, and Documentation. The 'HTTP Settings' tab is currently selected, showing the following configuration:

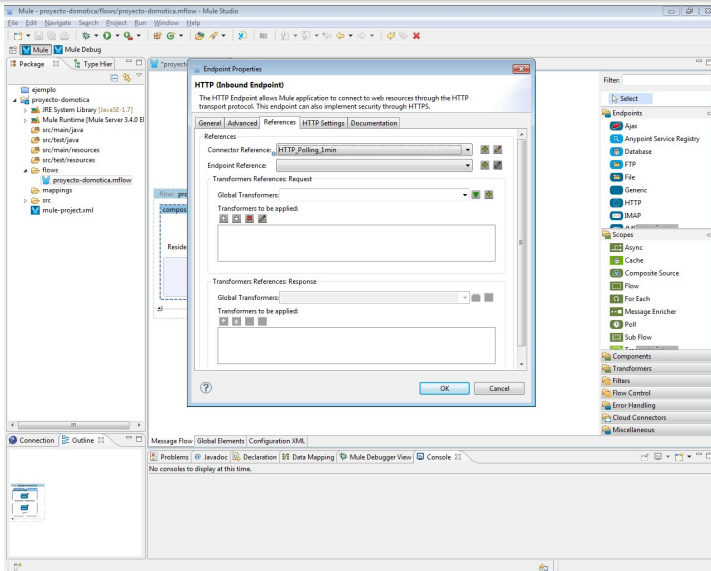
- Basic Authentication:**
 - User: nombre-usuario
 - Password: ****
- HTTP Settings:**
 - Content Type: application/json
 - Keep Alive

The background shows the Mule Studio workspace with a project tree on the left, a message flow diagram in the center, and a component palette on the right. The component palette is filtered to show 'Endpoints', with 'HTTP' selected. The bottom status bar indicates 'No consoles to display at this time.'

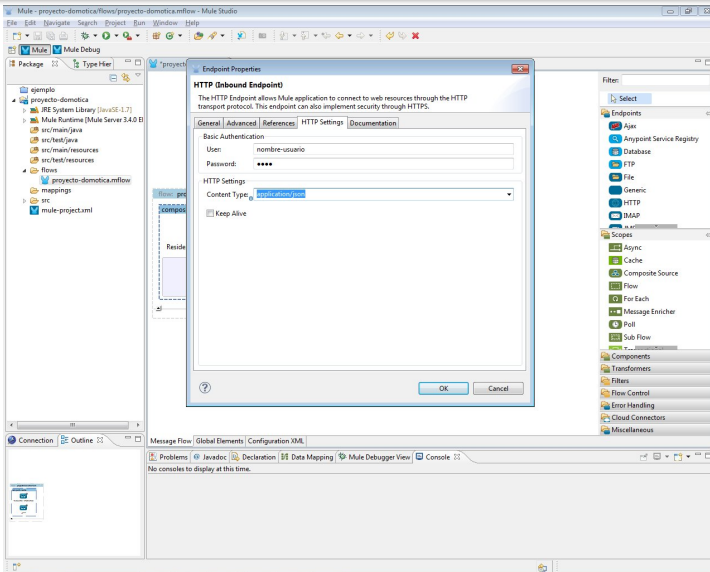
Añadir los endpoints de entrada (cont.)



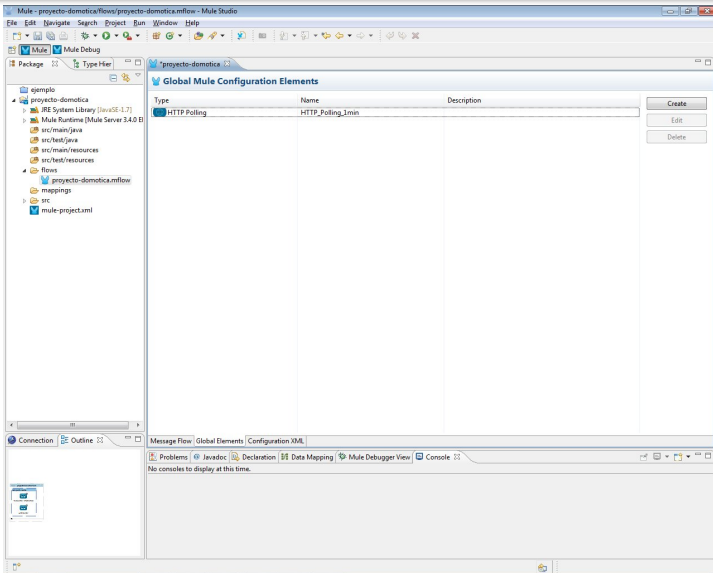
Añadir los endpoints de entrada (cont.)



Añadir los endpoints de entrada (cont.)



Añadir los *endpoints* de entrada (cont.)



Añadir los *endpoints* de entrada (cont.)

The screenshot displays the Mule Studio IDE interface. The main editor shows the configuration of an HTTP Polling In endpoint within a Mule flow. The configuration is as follows:

```

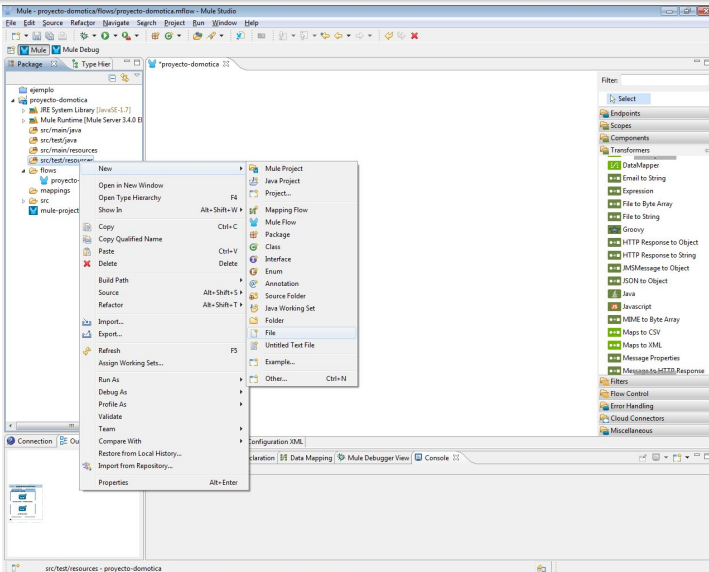
1
2
3 <ft.org/schema/mule/documentation" xmlns:spring="http://www.springframework.org/schema/beans" version="EE-3.4.0" xmlns: xsi="http://www.w
4
5
6 Size="0" receiveBufferSize="0" receiveBacklog="0" clientSoTimeout="10000" serverSoTimeout="10000" socketslinger="0" proxyHostname="api
7
8
9
10 HTTP_Polling_In" user="nombre_usuario" password="1234" contentType="application/json" doc:name="Residential Information"/>
11
12 HTTP_Polling_In" user="nombre_usuario" password="1234" contentType="application/json" doc:name="MAC Center"/>
13
14

```

The left sidebar shows the project structure, including the package explorer with folders like 'ejemplo', 'projecto-domotica', and 'flows'. The bottom status bar indicates the current file is 'mule/flow/composite-source/httpbound-endpoint'.

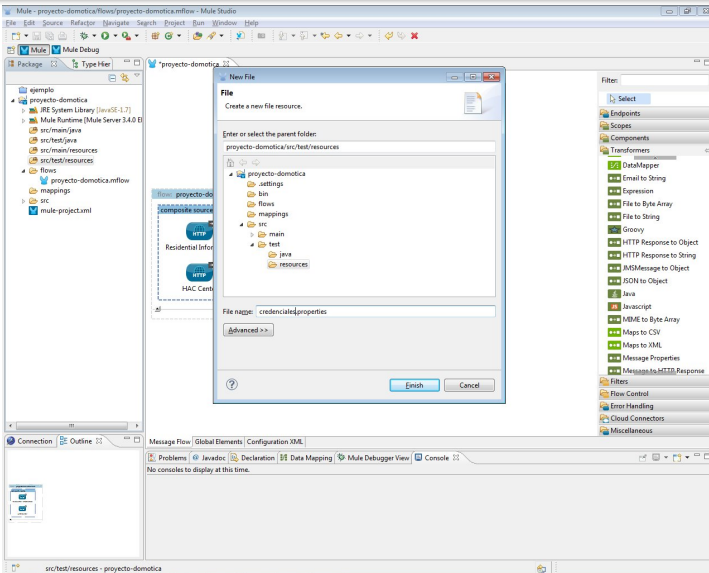
Crear un fichero `.properties` con las contraseñas

Las contraseñas no serán visibles en el fichero XML



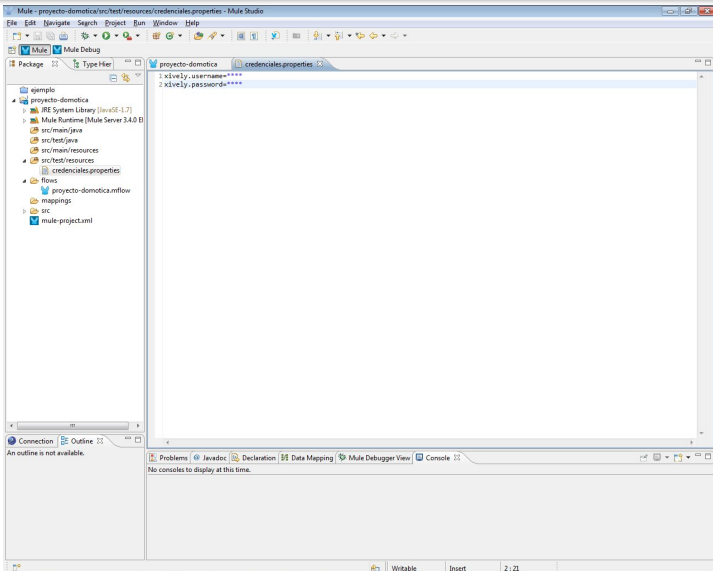
Crear un fichero .properties con las contraseñas (cont.)

Las contraseñas no serán visibles en el fichero XML



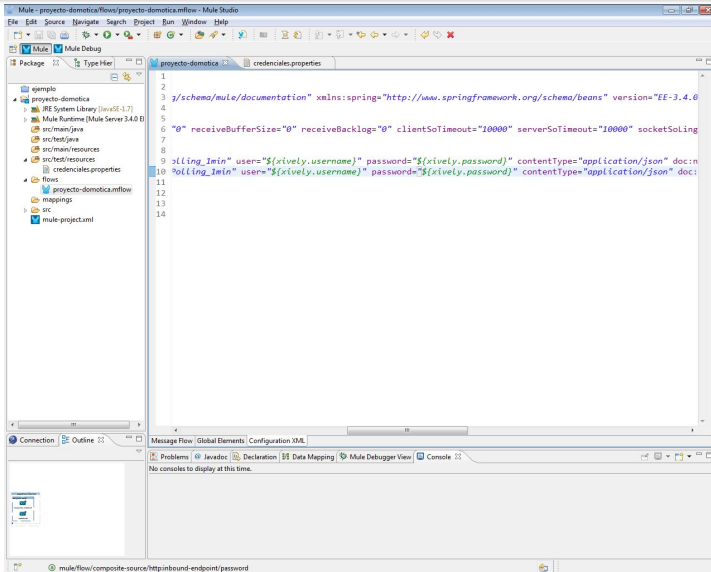
Crear un fichero `.properties` con las contraseñas (cont.)

Las contraseñas no serán visibles en el fichero XML



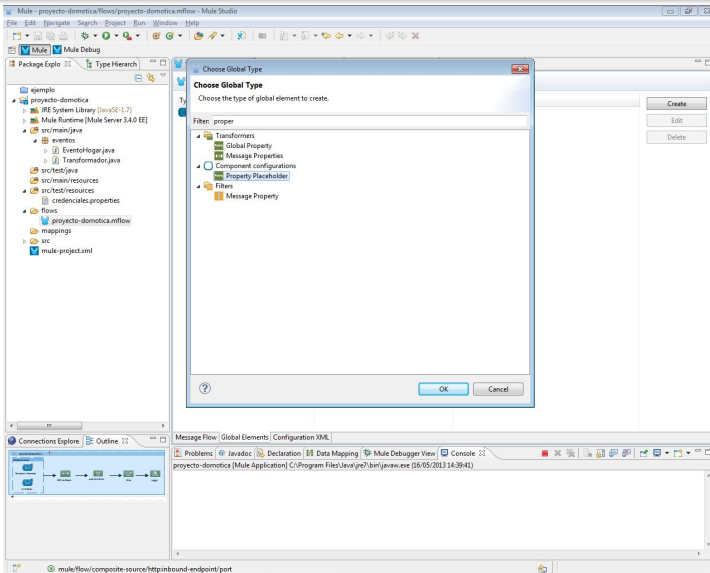
Crear un fichero `.properties` con las contraseñas (cont.)

Las contraseñas no serán visibles en el fichero XML



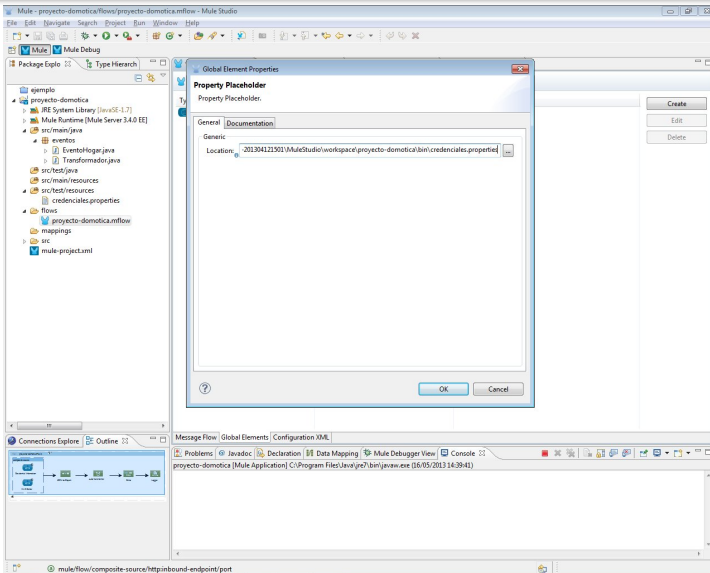
Crear un fichero `.properties` con las contraseñas (cont.)

Las contraseñas no serán visibles en el fichero XML



Crear un fichero `.properties` con las contraseñas (cont.)

Las contraseñas no serán visibles en el fichero XML



Añadir un transformador JSON-To-Object

The screenshot displays the Mule Studio IDE interface. The main workspace shows a flow diagram for 'projecto-domoticaFlow1'. The flow starts with a 'Composite source' containing two 'HTTP' connectors: 'Residential Information' and 'HAC Center'. An arrow points from this composite source to a 'JSON to Object' transformer.

The 'Components' palette on the right is open, showing a list of available transformers. The 'JSON to Object' transformer is highlighted, indicating it is being added to the flow.

The bottom status bar shows the current flow configuration: 'mule/flow/composite-source/httpinbound-endpoint'.

Añadir un transformador personalizado

Convertirá los objetos JsonData al tipo *EventoHogar*

The screenshot shows the Mule Studio IDE interface. On the left, a project tree displays the structure of 'proyecto-domotica', including source files and resources. The main editor window shows the XML configuration for a Mule flow named 'proyecto-domoticaFlow1'. The configuration includes a polling connector, an inbound endpoint, and a custom transformer. The custom transformer is a 'JSON to Object' transformer that converts 'JsonData' objects into 'Eventos.EventoHogar' objects.

```

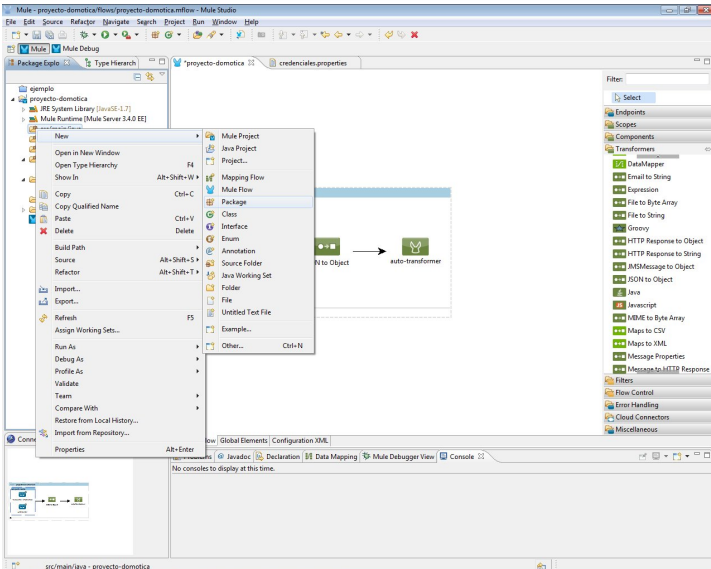
1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <mule xmlns:json="http://www.mulesoft.org/schema/mule/json" xmlns:http="http://www.mulesoft.org/schema/mule/http"
4 http://www.mulesoft.org/schema/mule/core http://www.mulesoft.org/schema/mule/core/current/mule.xsd
5 http://www.mulesoft.org/schema/mule/http http://www.mulesoft.org/schema/mule/http/current/mule-http.xsd
6 http://www.mulesoft.org/schema/mule/json http://www.mulesoft.org/schema/mule/json/current/mule-json.xsd">
7   <http:polling-connector name="HTTP_Polling_1min" cookieSpec="netscape" validateConnections="true" polling
8   <flow name="proyecto-domoticaFlow1" doc:name="proyecto-domoticaFlow1">
9     <composite-source doc:name="Composite Source">
10      <http:inbound-endpoint exchange-pattern="one-way" host="api.xively.com" port="80" path="v2/feeds/
11      <http:inbound-endpoint exchange-pattern="one-way" host="api.xively.com" port="80" path="v2/feeds/
12    </composite-source>
13    <json:json-to-object-transformer doc:name="JSON to Object"/>
14    <auto-transformer returnClass="eventos.EventoHogar" name="JsonDataToEventoHogar"/></auto-transformer>
15  </flow>
16 </mule>
17

```

Más información: <http://www.mulesoft.org/docs/site/current3/apidocs/org/mule/module/json/JsonData.html>

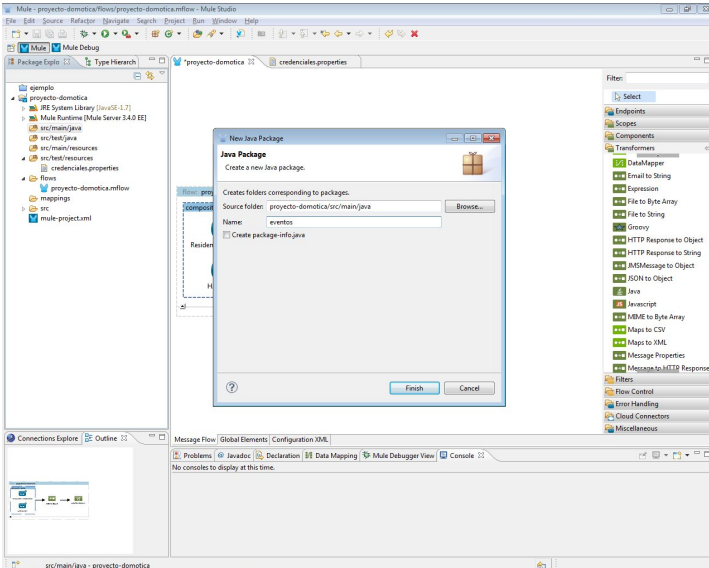
Añadir un transformador personalizado (cont.)

Convertirá los objetos JsonData al tipo *EventoHogar*



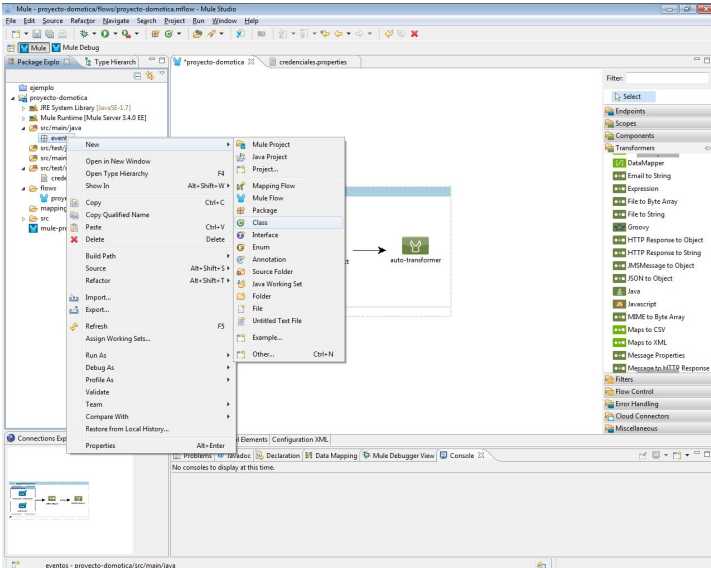
Añadir un transformador personalizado (cont.)

Convertirá los objetos JsonData al tipo *EventoHogar*



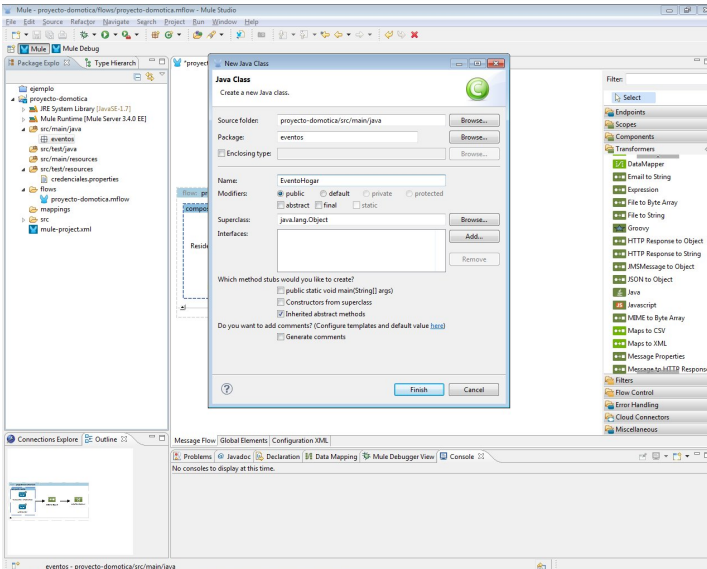
Añadir un transformador personalizado (cont.)

Convertirá los objetos JsonData al tipo *EventoHogar*



Añadir un transformador personalizado (cont.)

Convertirá los objetos JsonData al tipo *EventoHogar*



Añadir un transformador personalizado (cont.)

Convertirá los objetos JsonData al tipo *EventoHogar*

```

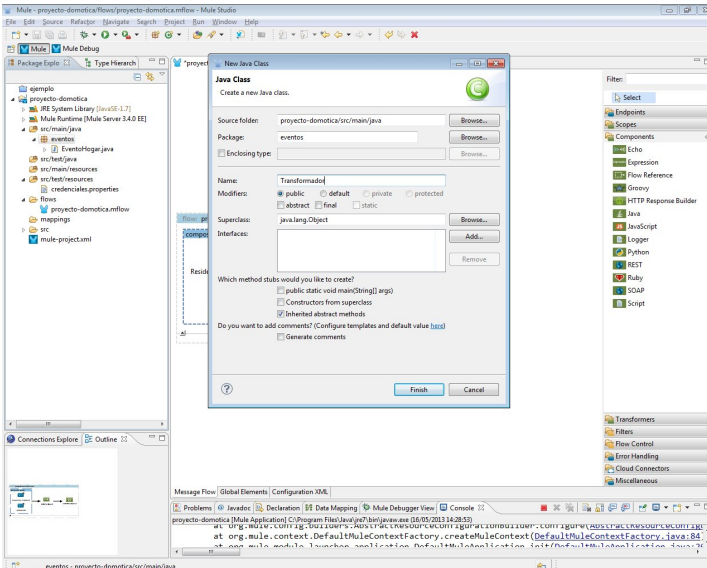
1 // .....
2 Juan Boubeta Puig <juan.boubeta@uca.es>
3 Universidad de Cádiz (Spain)
4 .....
5
6 package eventos;
7
8 public class EventoHogar {
9
10     private String hogar;
11     private String sensor;
12     private String localizacion;
13     private float latitud;
14     private float longitud;
15     private String tiempoRegistro;
16     private float consumoEnergetico;
17     private float temperaturaInt; // temperatura interior
18     private float temperaturaExt; // temperatura exterior
19
20     public EventoHogar(String ho, String s, String l, float la, float lo, String t,
21         float e, float ti, float te) {
22         hogar = ho;
23         sensor = s;
24         localizacion = l;
25         latitud = la;
26         longitud = lo;
27         tiempoRegistro = t;
28         consumoEnergetico = e;
29         temperaturaInt = ti;
30         temperaturaExt = te;
31     }
32
33     public EventoHogar() {
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

Pulse aquí para ver fichero *EventoHogar.java*

Añadir un transformador personalizado (cont.)

Convertirá los objetos JsonData al tipo *EventoHogar*



Añadir un transformador personalizado (cont.)

Convertirá los objetos JsonData al tipo *EventoHogar*

```

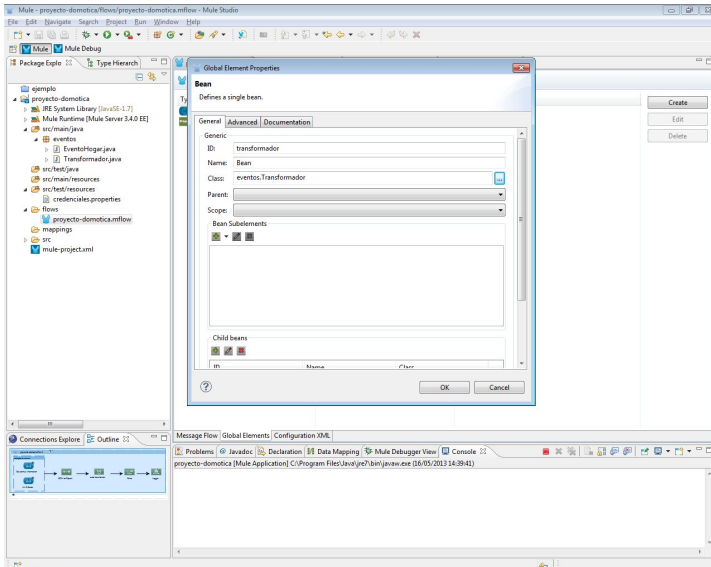
1 // EventoHogar.java
2 Juan Boubeta Puig <juan.boubeta@uca.es>
3 Universidad de Cádiz (Spain)
4
5
6 package eventos;
7
8 import org.mule.api.annotations.ContainsTransformerMethods;
9 import org.mule.api.annotations.Transformer;
10 import org.mule.module.json.JsonData;
11
12 import eventos.EventoHogar;
13
14 // ver http://www.mulesoft.org/docs/site/current3/api/docs/org/mule/module/json/JsonData.html
15
16
17 @ContainsTransformerMethods
18 public class Transformador
19 {
20     @Transformer
21     public EventoHogar JSONtoEventoHogar(JsonData obj) throws Exception
22     {
23         EventoHogar evento = new EventoHogar();
24         String nombreHogar = obj.getString("title");
25
26         evento.setHogar(nombreHogar);
27         evento.setSensor(obj.getString("feed"));
28         evento.setLocalizacion(obj.getString("location/name"));
29         evento.setLatitud(Float.parseFloat(obj.getString("location/lat")));
30         evento.setLongitud(Float.parseFloat(obj.getString("location/lon")));
31         evento.setTempoRegistro(obj.getString("updated"));
32
33         if(nombreHogar.equalsIgnoreCase("Residential information")) {
34             evento.setConsumoEnergetico(Float.parseFloat(obj.getString("datastreams[1]/current_value")));
35             evento.setTemperaturaInt(Float.parseFloat(obj.getString("datastreams[14]/current_value")));
36         }
37         else if(nombreHogar.equalsIgnoreCase("MMC Center")) {
38             evento.setTemperaturaInt(Float.parseFloat(obj.getString("datastreams[1]/current_value")));
39             evento.setTemperaturaExt(Float.parseFloat(obj.getString("datastreams[3]/current_value")));
40         }
41     }
42 }

```

Pulse aquí para ver fichero *Transformador.java*

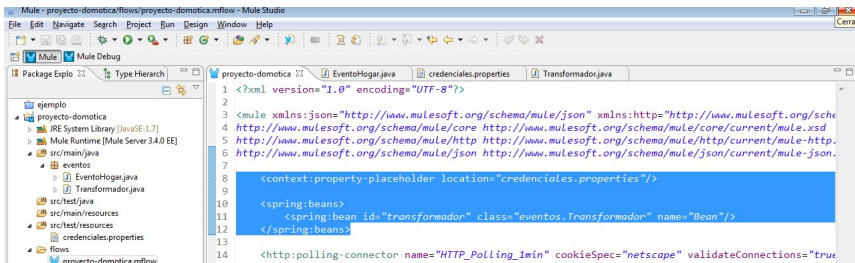
Añadir un transformador personalizado (cont.)

Convertirá los objetos JsonData al tipo *EventoHogar*



Añadir un transformador personalizado (cont.)

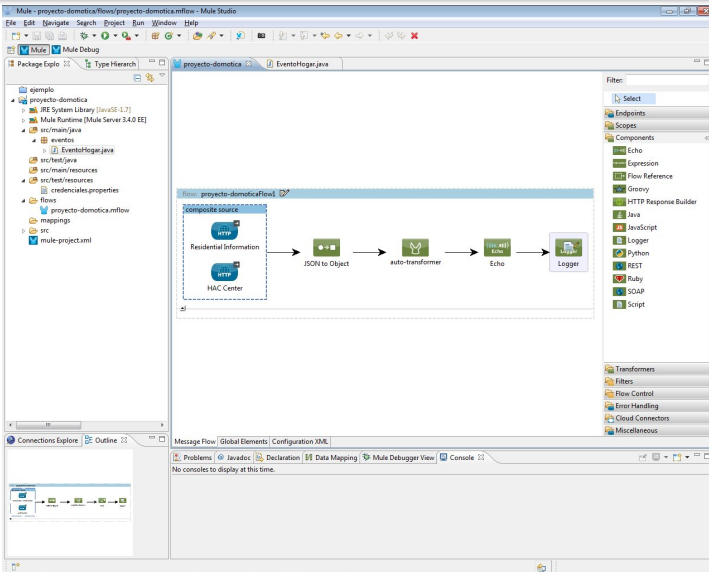
Convertirá los objetos JsonData al tipo *EventoHogar*



The screenshot shows the Mule Studio IDE interface. On the left, the Package Explorer displays the project structure for 'ejemplo', including 'projecto-domotica', 'src/main/java', and 'src/test/java'. The main editor window shows the 'Transformador.java' file with the following XML configuration:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <mule xmlns:json="http://www.mulesoft.org/schema/mule/json" xmlns:http="http://www.mulesoft.org/sche
4 http://www.mulesoft.org/schema/mule/core http://www.mulesoft.org/schema/mule/core/current/mule.xsd
5 http://www.mulesoft.org/schema/mule/http http://www.mulesoft.org/schema/mule/http/current/mule-http.
6 http://www.mulesoft.org/schema/mule/json http://www.mulesoft.org/schema/mule/json/current/mule-json.
7
8 <context:property-placeholder location="credenciales.properties"/>
9
10 <spring:beans>
11 <spring:bean id="transformador" class="eventos.Transformador" name="Bean"/>
12 </spring:beans>
13
14 <http:polling-connector name="HTTP_Polling_1min" cookieSpec="netscape" validateConnections="true
```

Añadir *Echo* y *Logger* para visualizar mensajes



The screenshot displays the Mule Studio interface for configuring a Mule flow. The main workspace shows a flow named 'projecto-domoticaFlow' with the following sequence of components:

- HTTP Inbound** (Source)
- HTTP Outbound** (Destination)
- Residential Information** (Component)
- JSON to Object** (Transformer)
- auto-transformer** (Transformer)
- Echo** (Component, highlighted with a dashed box)
- Logger** (Component)

The right-hand sidebar contains a component palette with a search filter. The 'Echo' and 'Logger' components are visible in the list. The bottom status bar shows 'No consoles to display at this time.'

Añadir *Echo* y *Logger* para visualizar mensajes (cont.)

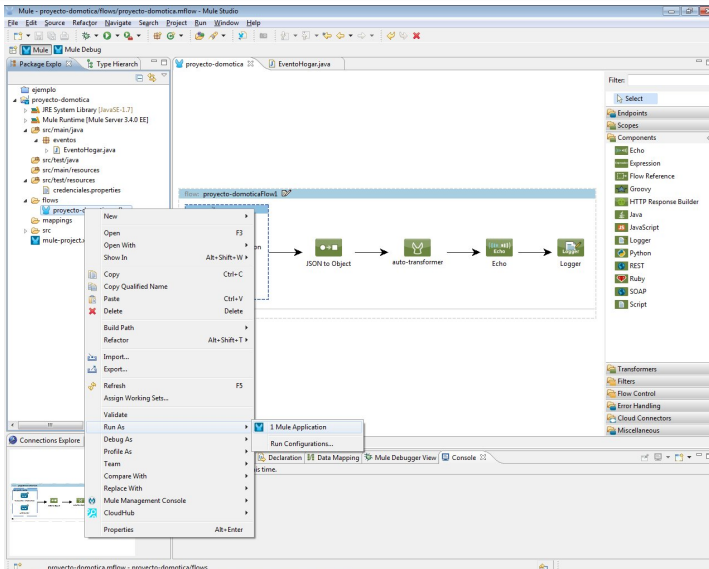
The screenshot displays the Mule Studio interface with the 'Pattern Properties' dialog box open for a 'Logger' component. The dialog is titled 'Logger' and contains the following fields:

- Display Name:** Logger
- Generic:** Contenido: #[message.payload]
- Message:** (empty)
- Level:** INFO
- Category:** (empty)

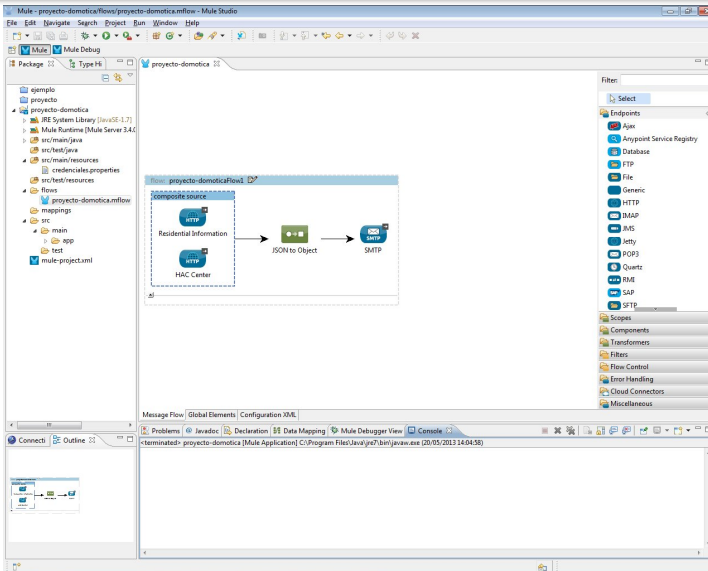
The 'General' tab is selected, and the 'Documentation' sub-tab is active. The background shows the Package Explorer on the left with a project structure including 'ejemplo', 'projecto-domotica', and 'flows'. The right sidebar shows a component palette with 'Echo' and 'Logger' selected. The bottom status bar indicates 'No consoles to display at this time.'

Ejecutar la aplicación Mule

Run As > Mule Application



Usar SMTP para enviar alertas por correo electrónico



Usar SMTP para enviar alertas por correo electrónico

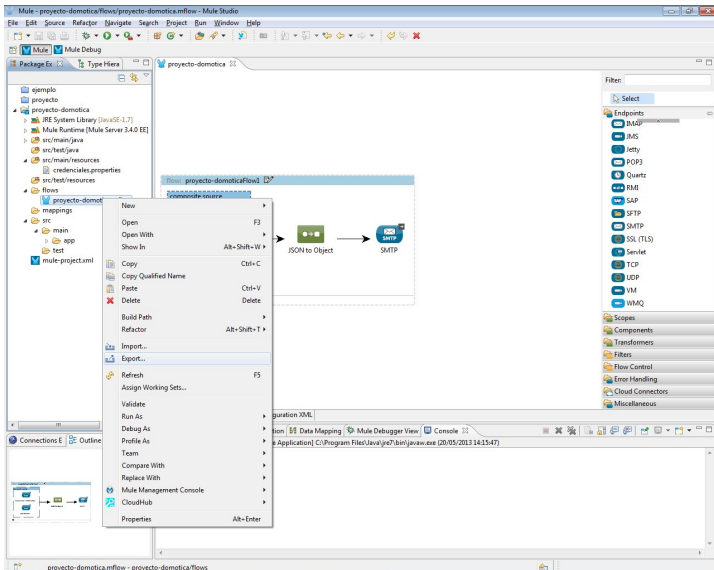
```
<smtps:outbound-endpoint host="smtp-alum.uca.es" port="465"  
  user="{uca.username}" password="{uca.password}"  
  to="****@alum.uca.es" from="****@alum.uca.es"  
  subject="Envío de alerta" responseTimeout="3000"  
  doc:name="SMTP"/>
```

A tener en cuenta

- La dirección origen (*from*) debe ser una cuenta válida de la UCA.
- Deben desactivarse los cortafuegos del sistema operativo, así como antivirus. En el caso de Avast: *Control de los escudos de avast! > Desactivar.*

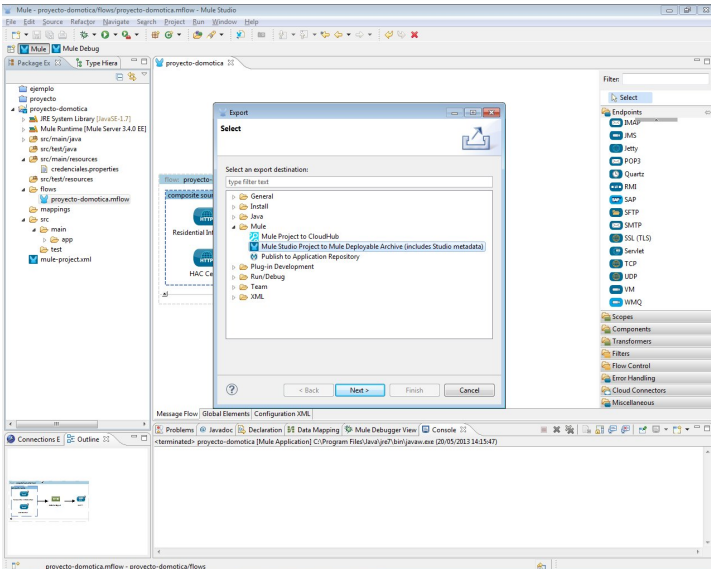
Exportar la aplicación Mule

File > Export...

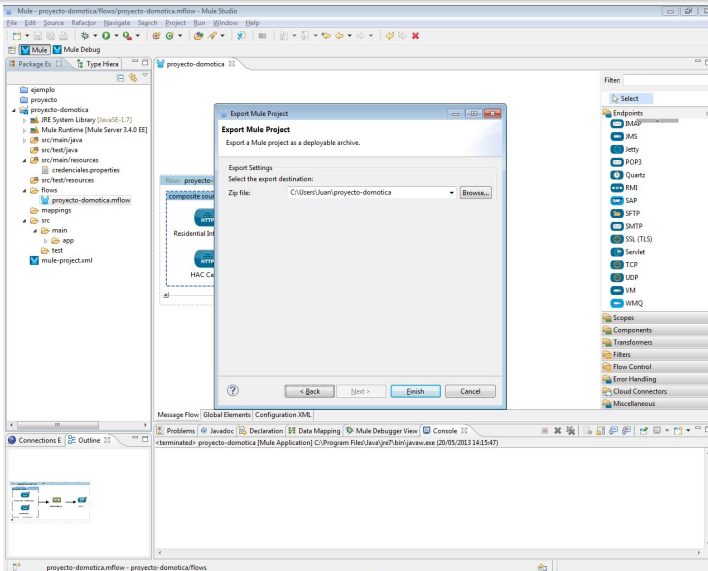


Exportar la aplicación Mule (cont.)

Mule > Mule Studio Project... (includes Studio metadata)

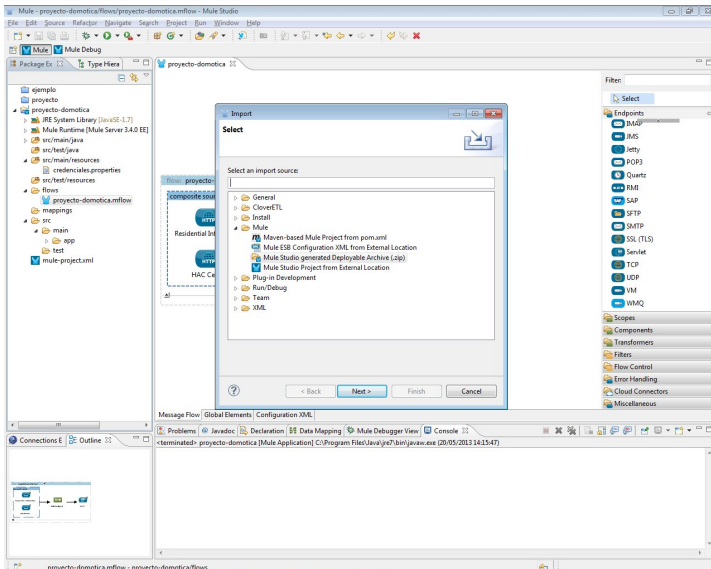


Exportar la aplicación Mule (cont.)

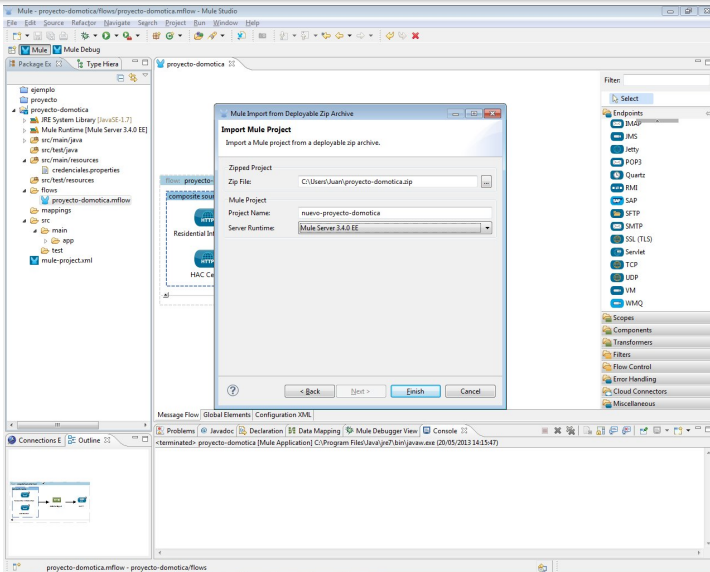


Importar la aplicación Mule

File > Import... > Mule Studio generated Deployable Archive (.zip)



Importar la aplicación Mule



Referencias bibliográficas I



MuleSoft Inc.
Mule Studio

<http://www.mulesoft.org/download-mule-esb-community-edition>,
mayo 2013.



LogMeIn, Inc.

Xively – Public Cloud for the Internet of Things

<https://xively.com/>, mayo 2013.



D. Luckham

**The Power of Events: An Introduction to Complex Event Processing in
Distributed Enterprise Systems**

Addison-Wesley, 2001.



D. Luckham

Event Processing for Business: Organizing the Real-Time Enterprise
Wiley, 2012.



Referencias bibliográficas II



EsperTech Inc.

Esper - Complex Event Processing

<http://esper.codehaus.org/>, mayo 2013.



J. Boubeta Puig; G. Ortiz; I. Medina Buló

Procesamiento de Eventos Complejos en Entornos SOA: Caso de Estudio para la Detección Temprana de Epidemias

Actas de las VII Jornadas de Ciencia e Ingeniería de Servicios A Coruña, septiembre, 2011.



L. Atzori; A. Iera; G. Morabito

The Internet of Things: A Survey

Computer Networks (15), pp. 2787-2805, octubre, 2010.



UCA

Universida