

# Seminario 2.3: Anypoint Studio: The Internet of Things (IoT)

## Sistemas Distribuidos

Pablo García Sánchez  
Apuntes creados por Juan Boubeta Puig y  
Antonio Balderas Alberico

Grupo UCASE de Ingeniería del Software  
Departamento de Ingeniería Informática

Marzo de 2017



# Índice

1 The Internet of Things (IoT)

2 Implementación

# Índice

## 1 The Internet of Things (IoT)

## 2 Implementación

# The Internet of Things (IoT)

Vídeo presentación del **Internet of Things World Forum Barcelona 2013**

<https://www.youtube.com/watch?v=yeof1x1H704>

# The Internet of Things (IoT)

## 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.

# Índice

## 1 The Internet of Things (IoT)

## 2 Implementación

# Caso de estudio

## Enunciado

- En este ejemplo nuestro ESB tomará los datos recogidos por el sensor de un hogar y nos enviará un correo electrónico con la información

## Requisitos

- Tener cuenta de correo electrónico y conocer sus parámetros para la configuración vía SMTP (aquí usaremos el de la UCA).

# Fuente de mensajes (I)

Flujos de Xively sobre domótica utilizados en este caso de estudio

Nº	Nombre del flujo	País	URL	Actualiz.
F1	<i>My house</i>	EEUU	<a href="https://thingspeak.com/channels/9">https://thingspeak.com/channels/9</a>	1 min



# Fuente de mensajes (II)

## Formato de los datos de flujos normalizados

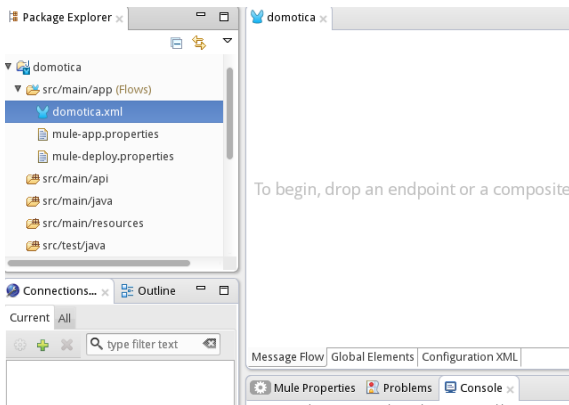
← → ↺ [api.thingspeak.com/channels/9/feed.json](http://api.thingspeak.com/channels/9/feed.json)

```
{
  - channel: {
    id: 9,
    name: "my_house",
    description: "Netduino Plus connected to sensors around the house",
    latitude: "40.44",
    longitude: "-79.9965",
    field1: "Light",
    field2: "Outside Temperature",
    created_at: "2010-12-14T01:20:06Z",
    updated_at: "2016-04-11T09:59:44Z",
    last_entry_id: 9952130
  },
  - feeds: [
    - {
      created_at: "2016-04-11T09:32:44Z",
      entry_id: 9952031,
      field1: "314",
      field2: "48.322717622080681"
    },
    - {
      created_at: "2016-04-11T09:32:59Z",
      entry_id: 9952032,
      field1: "315",
      field2: "45.774946921443735"
    }
  ]
}
```

# Crear un proyecto Mule

## Creamos un nuevo proyecto de tipo Mule

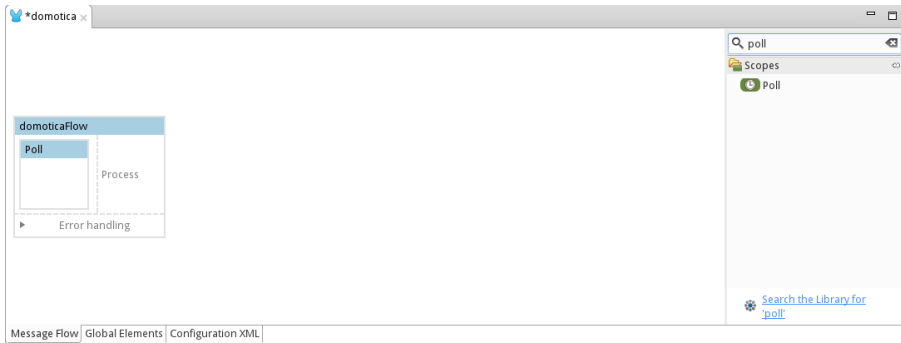
- En el ejemplo el nombre del proyecto es **domotica**



# Temporizador para captura de datos

## Creamos temporizador para consultar fuente periódicamente

- Arrastre al panel un elemento de tipo **Poll**
- Información de este componente en <http://www.mulesoft.org/documentation/display/current/Poll+Reference>



# Temporizador para captura de datos

## Ajustamos temporizador a 30 segundos

### ■ Configuración del componente Poll

The screenshot shows the configuration interface for the 'Poll' component in Anypoint Studio. The top bar includes tabs for 'Poll', 'Problems', and 'Console'. A status message indicates 'There are no errors.' The left sidebar has tabs for 'General', 'Notes', and 'Problems'. The main configuration area is divided into sections: 'Display Name' (set to 'Poll'), 'Polling Information', and 'Watermark'. Under 'Polling Information', the 'Fixed frequency scheduler' is selected, with 'Frequency' set to '30000', 'Start delay' set to '0', and 'Time unit' set to 'MILLISECONDS (Default)'. The 'Cron scheduler' is unselected. Under 'Watermark', 'Do not use watermark' is selected, and the other options are unselected. The 'Variable Name', 'Default Expression', 'Update Expression', 'Selector', and 'Selector Expression' fields are empty.

General

Display Name: Poll

Notes

Polling Information

☒ Fixed frequency scheduler

Frequency: 30000

Start delay: 0

Time unit: MILLISECONDS (Default)

☐ Cron scheduler

Expression:

Watermark

☒ Do not use watermark

☐ Enable watermark

Variable Name:

Default Expression:

☐ Update Expression:

☐ Selector:

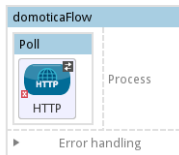
Selector Expression:

# Conector HTTP

## Añadimos conector HTTP para consultar fuente de datos

- Arrastre conector **HTTP** al interior del componente **Poll**
- Información de este componente en <http://www.mulesoft.org/documentation/display/current/HTTP+Connector>

\*domotica x



http

Connectors

HTTP

Suggestions...

HTTP

Jetty

Components

HTTP Static  
Handler[Search the  
'http'](#)

# Conector HTTP

## Configuración del conector HTTP

- Configuramos el **Path** y el **Method** como en la figura.
- Después pinchamos en el **+** del **Connector configuration**

The screenshot shows the configuration window for an HTTP connector in Anypoint Studio. The interface includes a sidebar with tabs for 'General', 'Advanced', and 'Notes'. The 'General' tab is active, showing the 'Display Name' as 'HTTP'. Below this, the 'Advanced' tab is selected, revealing 'General Settings' and 'URL Settings'. In 'General Settings', the 'Connector Configuration' dropdown is set to a default value, with a '+' icon to its right. The 'URL Settings' section contains two fields: 'Path' with a value of '/' and 'Method' with a value of 'GET'. At the bottom, there is a 'Parameters' section with an 'Add Parameter' button and a message: 'Click in the button below to add a parameter'. A red error message at the top left states: 'Attribute 'config-ref' is required'.

HTTP x Problems Console

Attribute 'config-ref' is required

**General**

Display Name: HTTP

**Advanced**

General Settings

Connector Configuration: [Dropdown] + [Icon]

URL Settings

Path: [/]

Method: [GET]

Parameters

Click in the button below to add a parameter

Add Parameter

# Conector HTTP

En la pestaña **General** definimos el **Host**, el **Port** y el **BasePath** (Éste depende de la fuente thingspeak utilizada (<https://thingspeak.com/feeds/9/>))

The screenshot shows the 'Global Element Properties' dialog box with the 'HTTP Request Configuration' tab selected. The dialog has several sections: 'General' (with tabs for General, TLS/SSL, Proxy, Authentication, Sockets, and Notes), 'URL Configuration', 'API Configuration', 'Other Settings', and 'Datasense Settings'. In the 'General' section, the 'Name' is 'HTTP\_Request\_Configuration'. In the 'URL Configuration' section, 'Protocol' is set to 'HTTP', 'Host' is 'api.thingspeak.com', 'Port' is '80', and 'Base Path' is 'channels/9/feed.json'. In the 'API Configuration' section, 'RAML Location' is empty with a 'Browse' button and a 'Search RAML Library' link. In the 'Other Settings' section, 'Use Persistent Connection' is checked, 'Connection Idle Timeout' is '30000', and 'Response Timeout' is empty. In the 'Datasense Settings' section, 'Enable DataSense' is checked. At the bottom are 'OK' and 'Cancel' buttons.

Global Element Properties

**HTTP Request Configuration**

Create reusable HTTP request manually or by adding your RAML API description

General TLS/SSL Proxy Authentication Sockets Notes

Generic

Name: HTTP\_Request\_Configuration

URL Configuration

Protocol: ☒ HTTP ☐ HTTPS

Host: api.thingspeak.com

Port: 80

Base Path: channels/9/feed.json

API Configuration

RAML Location: Type a valid RAML location [Browse](#)

[Search RAML Library](#)

Other Settings

☒ Use Persistent Connection

Connection Idle Timeout: 30000

Response Timeout:

Datasense Settings

☒ Enable DataSense

OK Cancel

# Transformer ByteArray to String

Añada un transformador de tipo **ByteArray to String** a continuación del conector HTTP

Información de este componente en <http://www.mulesoft.org/documentation/display/current/Transformers>

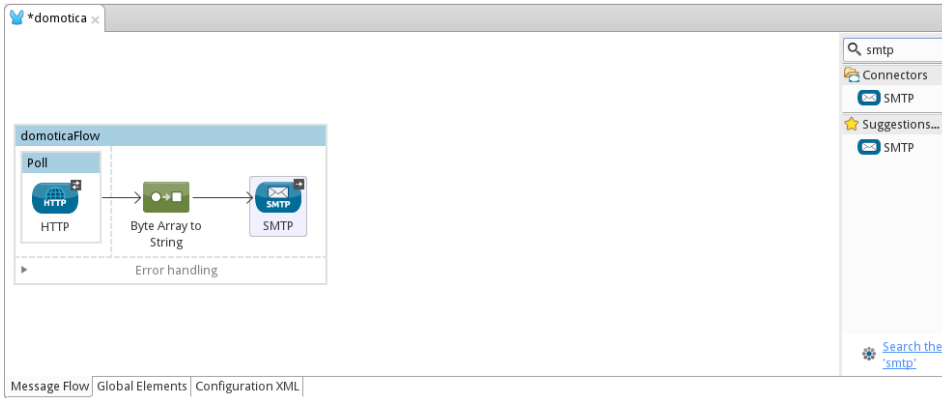
The screenshot displays the Anypoint Studio IDE. At the top, a tab labeled '\*domotica x' is visible. The main workspace shows a 'domoticaFlow' diagram. Inside this flow, there is a 'Poll' connector (represented by a blue globe icon with 'HTTP' text) connected to a 'Byte Array to String' transformer (represented by a green square icon with a plus sign). Below the main flow diagram, there is an 'Error handling' section. On the right side, a search bar contains the text 'byte array to string', and a list of 'Transformers' shows 'Byte Array to String' as the selected option. At the bottom, the 'Message Flow' tab is active, showing a status bar with 'Byte Array to String x', 'Problems', and 'Console'. The console message states 'There are no errors.' with a green checkmark icon. The bottom right corner features the UCA logo and the text 'Universidad de Cádiz'.



# Conector SMTP

## Añada un conector SMTP a continuación del conector HTTP

Información de este componente en  
<http://www.mulesoft.org/documentation/display/current/SMTP+Transport+Reference>



# Conector SMTP

Configuramos el componente SMTP con los datos de nuestro correo electrónico de la UCA (u otro que queramos siempre que conozcamos los datos para hacerlo).

Configuración UCA: <https://webmerlin.uca.es/webmerlin/ayuda.do>

The screenshot shows the configuration window for an SMTP connector in Anypoint Studio. The interface includes a sidebar with tabs for General, Advanced, Transformers, Security, and Notes. The main area is divided into sections for Basic Settings and Email Information.

**General**

Display Name: SMTP

**Basic Settings**

Host: smtp.uca.es

Port: 25

User: u12345678

Password: [masked] ☐ Show password

Connector Configuration: [dropdown menu]

**Email Information**

To: juan.lopez@gmail.com

From: juan.lopez@alum.uca.es

Subject: [SD] Alerta datos Xively

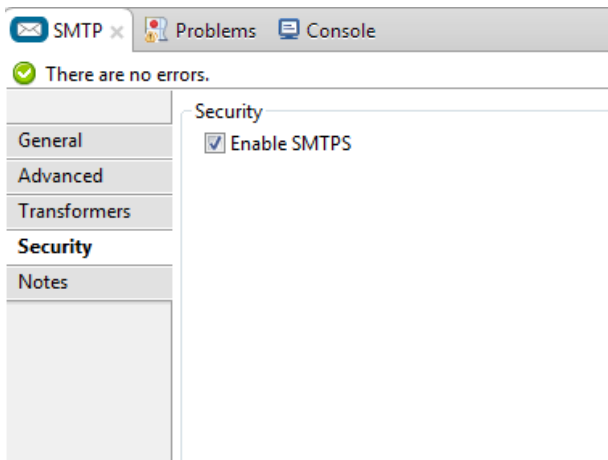
Cc: [empty field]

Bcc: [empty field]

Reply To: [empty field]

# Conector SMTP

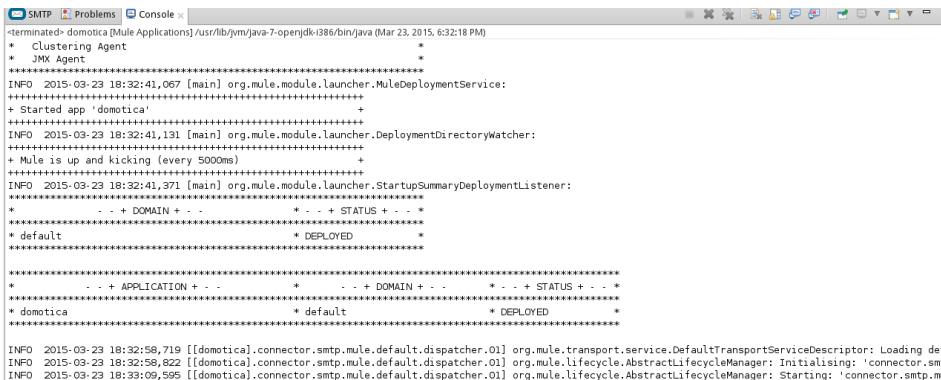
Para usar el correo de la UCA con SSL acceder a la pestaña "Security" y activar la opción SMTPS.



# Ejecución

## Consola

- Observamos el flujo, una vez aparecen las últimas líneas deberá llegarle un correo electrónico



```

SMTP Problems Console x
<terminated> domotica [Mule Applications] /usr/lib/jvm/java-7-openjdk-1386/bin/java (Mar 23, 2015, 6:32:18 PM)
* Clustering Agent *
* JMX Agent *
*****
INFO 2015-03-23 18:32:41,067 [main] org.mule.module.launcher.MuleDeploymentService:
+++++
+ Started app 'domotica' +
+++++
INFO 2015-03-23 18:32:41,131 [main] org.mule.module.launcher.DeploymentDirectoryWatcher:
+++++
+ Mule is up and kicking (every 5000ms) +
+++++
INFO 2015-03-23 18:32:41,371 [main] org.mule.module.launcher.StartupSummaryDeploymentListener:
*****
* - - + DOMAIN + - - * - - + STATUS + - - *
*****
* default * DEPLOYED *
*****

*****
* - - + APPLICATION + - - * - - + DOMAIN + - - * - - + STATUS + - - *
*****
* domotica * default * DEPLOYED *
*****

INFO 2015-03-23 18:32:58,719 [[domotica].connector.smtp.mule.default.dispatcher.01] org.mule.transport.service.DefaultTransportServiceDescriptor: Loading de
INFO 2015-03-23 18:32:58,822 [[domotica].connector.smtp.mule.default.dispatcher.01] org.mule.lifecycle.AbstractLifecycleManager: Initialising: 'connector.smtp
INFO 2015-03-23 18:33:09,595 [[domotica].connector.smtp.mule.default.dispatcher.01] org.mule.lifecycle.AbstractLifecycleManager: Starting: 'connector.smtp.m
  
```

# Ejecución

## Correo electrónico

- Cómo ve, nos ha llegado la información del flujo al correo (aunque sin procesar)

[SD] Alerta datos Xively



Recibidos x



antonio.balderas@uca.es

para mí ▾

19:33 (hace 2 minutos)

```
{
  "id": "62988",
  "title": "Residential information",
  "private": "false",
  "tags": [
    "energy",
    "PV",
    "RPI",
    "smart meter"
  ],
  "description": "Testing Raspberry Pi connected with my smartmeters (E+G+PV environmental measurements)",
  "feed": "https://api.xively.com/v2/feeds/62988.json",
  "status": "live",
  "updated": "2015-03-23T18:33:07.516392Z",
  "created": "2012-06-09T14:53:38.555781Z",
  "https://xively.com/users/ee/bakker",
  "version": "1.0.0",
  "datastreams": [
    {
      "id": "2",
      "current_value": "17.9",
      "at": "2015-03-23T18:32:36.862854Z",
      "max_value": "85.0",
      "min_value": "-999.99",
      "Attkick temperature",
      "unit": {
        "symbol": "%C",
        "label": "%C"
      },
      "id": "3",
      "current_value": "0",
      "at": "2015-03-23T18:32:59.828939Z",
      "max_value": "1220.0",
      "min_value": "0.0",
      "tags": [
        "PV current"
      ],
      "symbol": "W",
      "label": "W",
      "id": "4",
      "current_value": "4456.213",
      "at": "2015-03-23T18:32:59.828939Z",
      "max_value": "4456.213",
      "min_value": "1742.879",
      "tags": [
        "PV total"
      ],
      "unit": {
        "symbol": "kWh",
        "label": "kWh"
      },
      "id": "5",
      "current_value": "340.0",
      "at": "2015-03-23T18:33:06.662072Z",
      "max_value": "5580.0",
      "min_value": "-1140.0",
      "tags": [
        "Smartmeter Q"
      ],
      "unit": {
        "symbol": "W",
        "label": "W"
      },
      "id": "6",
      "current_value": "3632.0",
      "at": "2015-03-23T18:33:06.662072Z",
      "max_value": "3632.0",
      "min_value": "2145.0",
      "tags": [
        "Smartmeter +T1"
      ],
      "unit": {
        "symbol": "kWh",
        "label": "kWh"
      },
      "current_value": "3677.0",
      "at": "2015-03-23T18:33:06.662072Z",
      "max_value": "3677.0",
      "min_value": "0.2",
      "tags": [
        "Smartmeter +T2"
      ],
      "unit": {
        "symbol": "kWh",
        "label": "kWh"
      },
      "id": "8",
      "current_value": "753.0",
      "at": "2015-03-23T18:33:06.662072Z",
      "max_value": "753.0",
      "min_value": "315.0",
      "tags": [
        "Smartmeter -T1"
      ],
      "unit": {
        "symbol": "kWh",
        "label": "kWh"
      },
      "id": "9",
      "current_value": "1949.0",
      "at": "2015-03-23T18:33:06.662072Z",
      "max_value": "1949.0",
      "min_value": "746.0",
      "tags": [
        "Smartmeter -T2"
      ],
      "unit": {
        "symbol": "kWh",
        "label": "kWh"
      },
      "id": "10",
      "current_value": "5341.159",
      "at": "2015-03-23T18:33:06.662072Z",
      "max_value": "5341.159",
      "min_value": "3295.16",
      "tags": [
        "Gas usage"
      ],
      "unit": {
        "symbol": "m3",
        "label": "m3"
      },
      "id": "11",
      "current_value": "4.518",
      "at": "2015-03-23T18:32:59.828939Z",
      "max_value": "9.936",
      "min_value": "0.0",
      "tags": [
        "PV daily total"
      ],
      "unit": {
        "symbol": "kWh",
        "label": "kWh"
      },
      "id": "12",
      "current_value": "1011.0",
      "at": "2015-03-23T18:32:36.862854Z",
      "max_value": "1133.0",
      "min_value": "568.0",
      "tags": [
        "Pressure"
      ],
      "unit": {
        "symbol": "mBar",
        "label": "mBar"
      },
      "id": "13",
      "current_value": "36",
      "at": "2015-03-23T18:32:36.862854Z",
      "max_value": "62.0",
      "min_value": "26.0",
      "tags": [
        "Rel Humidity"
      ],
      "unit": {
        "symbol": "%RH",
        "label": "%RH"
      },
      "id": "14",
      "current_value": "40.1",
      "at": "2015-03-23T18:32:36.862854Z",
      "max_value": "54.6",
      "min_value": "27.7",
      "tags": [
        "RPI CPU temp"
      ],
      "unit": {
        "symbol": "%C",
        "label": "%C"
      },
      "id": "15",
      "current_value": "0",
      "at": "2015-03-23T18:32:36.862854Z",
      "max_value": "94.0",
      "min_value": "0.0",
      "tags": [
        "Attkick light intensity"
      ],
      "unit": {
        "symbol": "%",
        "label": "%"
      },
      "id": "16",
      "current_value": "5.7",
      "at": "2015-03-23T18:32:36.862854Z",
      "max_value": "127.7",
      "min_value": "-999.99",
      "tags": [
        "T1 temp"
      ],
      "unit": {
        "symbol": "%C",
        "label": "%C"
      },
      "id": "17",
      "current_value": "2015-03-23T18:32:36.862854Z",
      "max_value": "100.0",
      "min_value": "0.0",
      "tags": [
        "RPI CPU load"
      ],
      "unit": {
        "label": "%"
      },
      "id": "Waterverbruik dag",
      "current_value": "155",
      "at": "2015-03-23T18:32:25.515717Z",
      "max_value": "2115.0",
      "min_value": "0.0",
      "unit": {
        "symbol": "L",
        "label": "L"
      },
      "id": "Waterverbruik momentaan",
      "current_value": "0",
      "at": "2015-03-23T18:32:25.515717Z",
      "max_value": "271368.0",
      "min_value": "0.0",
      "unit": {
        "symbol": "L/min",
        "label": "L/min"
      },
      "id": "Waterverbruik totaal",
      "current_value": "750.448",
      "at": "2015-03-23T18:32:25.515717Z",
      "max_value": "751.323",
      "min_value": "0.0",
      "unit": {
        "symbol": "m3",
        "label": "m3"
      }
    },
    {
      "location": {
        "disposition": "fixed",
        "name": "Groningen",
        "exposure": "indoor",
        "domain": "physical",
        "ele": "5",
        "lat": "53.2304264353476",
        "lon": "6.593795120716099"
      }
    }
  ]
}
```

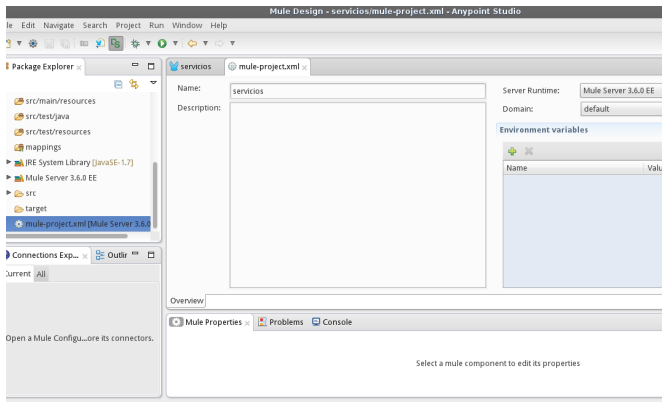
# Definición de parámetros

## Si accede a la vista XML de su flujo

- Verá que todas las claves están a la vista
- Esto es un problema porque al exportar el proyecto exportarás todos tus datos
- Vamos a definir las variables de entorno con nuestro valores

# Definimos parámetros

Vamos a definir todas nuestras variables de entorno en el fichero **mule-project.xml**



# Definimos parámetros

Añadimos una variable para cada parámetro utilizado en el ESB

## Environment variables



Name	Value	
mail.username	fulanito	
mail.password	111222555888	
mail.from	filanito@uca.es	
mail.to	fulanito@gmail.com	
mail.host	smtp.uca.es	



## Ejecutamos de nuevo el proyecto

- Volvemos a la composición y vamos a la pestaña **Configuration xml**
- Ponemos las variables donde antes estaban los valores

```

<fixed-frequency-scheduler frequency="30" timeUnit="SECONDS"/>
<http:request config-ref="HTTP_Request_Configuration" path="/" method="GET" doc:name="HTTP"/>
</poll>
<byte-array-to-string-transformer doc:name="Byte Array to String"/>
<json:json-to-object-transformer doc:name="JSON to Object"/>
<auto-transformer returnClass="eventos.EventoHogar" name="JsonDataToEventoHogar"></auto-transformer>
<echo-component doc:name="Echo"/>
<smtps:outbound-endpoint host="${mail.host}" user="${mail.username}" password="${mail.password}" to="${mail.to}" from="${mail.from}"
</flow>
</mule>

```

# Ejecución

## Ejecutamos de nuevo el proyecto

- Deberá funcionar igual que antes y podrá exportar su proyecto Mule sin revelar sus credenciales. **IMPORTANTE:** también se pueden añadir al fichero mule-app.properties.

# Enunciado

## Caso de estudio

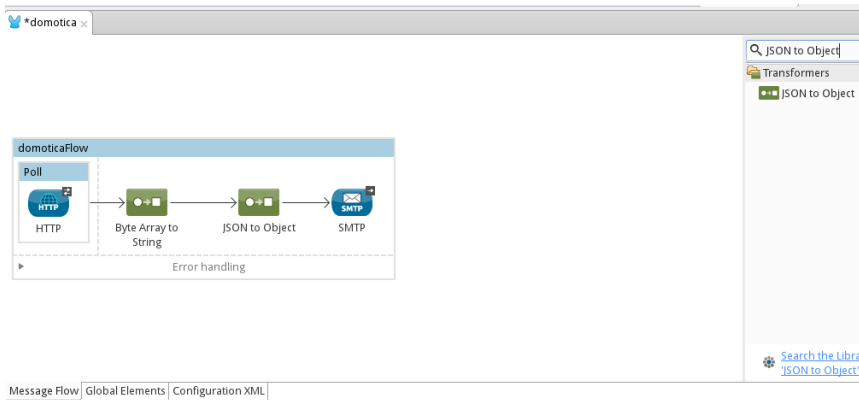
- Habrá comprobado que los emails que recibe no están bien formateados
- Vamos a crear un transformador para estos datos
- La información viene definida en un JSON, por lo que el objetivo será transformar este JSON a una clase de un objeto Java que definiremos nosotros

## Clases Java

- Accedemos al campus virtual y descargamos las clases Java que necesitamos
- **EventoHogar.java**: Clase que estancia un elemento que refleja el estado del hogar en un momento dado
- **Transformador.java**: Clase para transformar información en JSON a EventoHogar

# JSON to Object

Arrastramos al panel un elemento **JSON to Object** y lo colocamos justo antes del componente **SMTP**



# Transformador Java

Entramos en la pestaña **Configuration XML** y definimos el transformador **cómo y dónde** señala la flecha

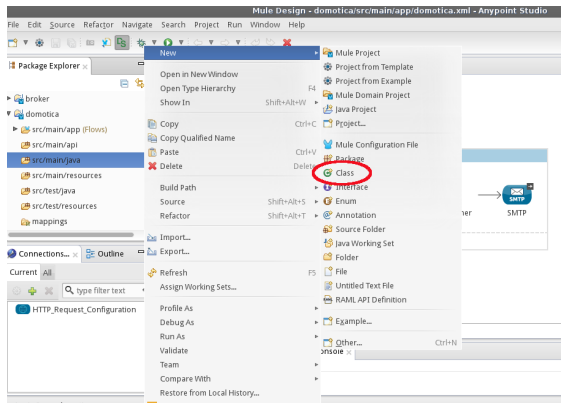
```
http://www.mulesoft.org/schema/mule/smp http://www.mulesoft.org/schema/mule/smp/current/mule-smp.xsd
http://www.mulesoft.org/schema/mule/json http://www.mulesoft.org/schema/mule/json/current/mule-json.xsd
<http:request-config name="HTTP_Request_Configuration" host="api.xively.com" port="80" basePath="/v2"
  <http:basic-authentication username="${xively.username}" password="${xively.password}"/>
</http:request-config>
<flow name="domoticaFlow">
  <poll doc:name="Poll">
    <fixed-frequency-scheduler frequency="30000"/>
    <http:request config-ref="HTTP_Request_Configuration" path="/" method="GET" doc:name="HTTP"/>
  </poll>
  <byte-array-to-string-transformer doc:name="Byte Array to String"/>
  <json:json-to-object-transformer doc:name="JSON to Object"/>
  <auto-transformer returnClass="eventos.EventoHogar" name="jsonDataToEventoHogar"/></auto-transformer>
  <smtp:outbound-endpoint host="${mail.host}" user="${mail.username}" password="${mail.password}"/>
</flow>
</mule>
```

sage Flow | Global Elements | Configuration XML

Mule Properties | Problems | Console

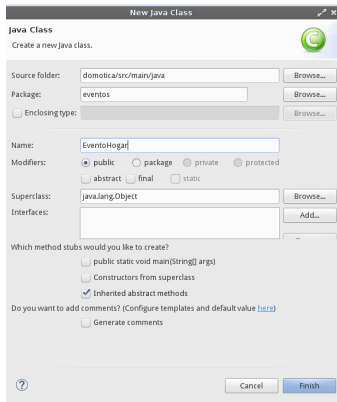
# Transformador Java

Creamos una clase Java en la carpeta **src/main/java** del explorador lateral



# Transformador Java

Definimos su **package** y su **name** tal como aparece en la imagen



**New Java Class**

Create a new Java class.

Source folder:

Package:

☐ Enclosing type:

Name:

Modifiers: ☒ public ☐ package ☐ private ☐ protected  
☐ abstract ☐ final ☐ static

Superclass:

Interfaces:

Which method stubs would you like to create?

☐ public static void main(String[] args)

☐ Constructors from superclass

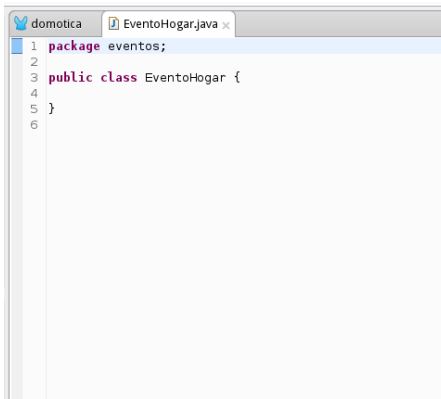
☒ Inherited abstract methods

Do you want to add comments? (Configure templates and default value [here](#))

☐ Generate comments

# Transformador Java

Así aparece una vez creado

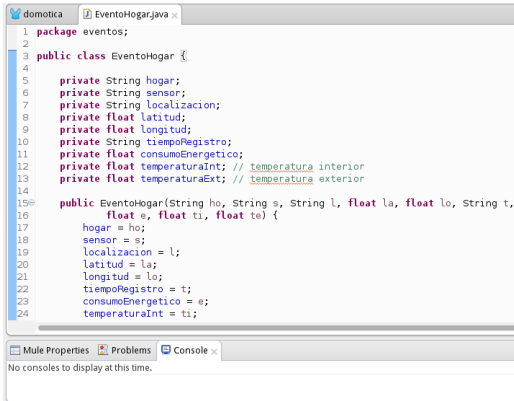


```
domotica EventoHogar.java x
1 package eventos;
2
3 public class EventoHogar {
4
5 }
6
```



# Transformador Java

Ponemos el cuerpo de la clase descargada del campus virtual



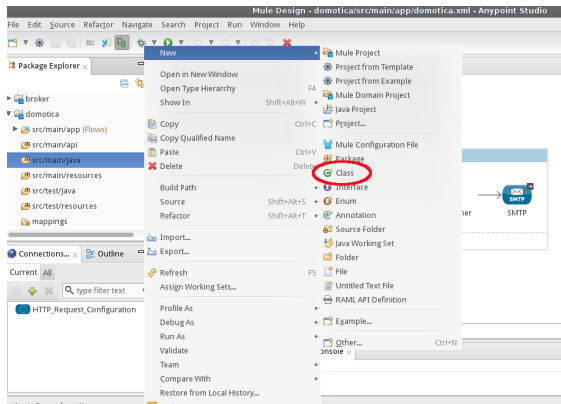
```
domotica EventoHogar.java x
1 package eventos;
2
3 public class EventoHogar {
4
5     private String hogar;
6     private String sensor;
7     private String localizacion;
8     private float latitud;
9     private float longitud;
10    private String tiempoRegistro;
11    private float consumoEnergetico;
12    private float temperaturaInt; // temperatura interior
13    private float temperaturaExt; // temperatura exterior
14
15    public EventoHogar(String ho, String s, String l, float la, float lo, String t,
16        float e, float ti, float te) {
17        hogar = ho;
18        sensor = s;
19        localizacion = l;
20        latitud = la;
21        longitud = lo;
22        tiempoRegistro = t;
23        consumoEnergetico = e;
24        temperaturaInt = ti;
```

Mule Properties Problems Console x

No consoles to display at this time.

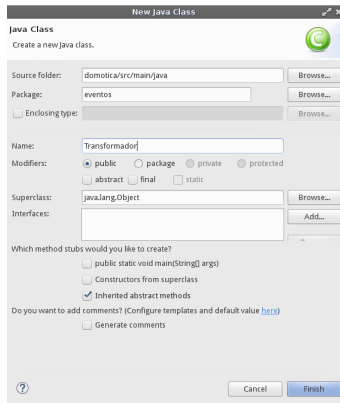
# Transformador Java

Creamos una nueva clase Java en la carpeta **src/main/java** del explorador lateral



# Transformador Java

Definimos su **package** y su **name** tal como aparece en la imagen



**New Java Class**

Create a new Java class.

Source folder:

Package:

☐ Enclosing type:

Name:

Modifiers: ☒ public ☐ package ☐ private ☐ protected  
☐ abstract ☐ final ☐ static

Superclass:

Interfaces:

Which method stubs would you like to create?

☐ public static void main(String[] args)

☐ Constructors from superclass

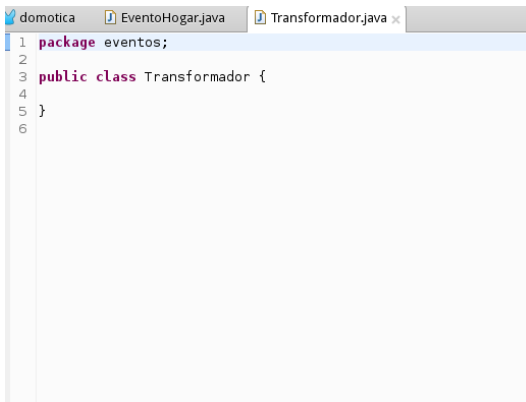
☒ Inherited abstract methods

Do you want to add comments? (Configure templates and default value [here](#))

☐ Generate comments

# Transformador Java

Así aparece una vez creado

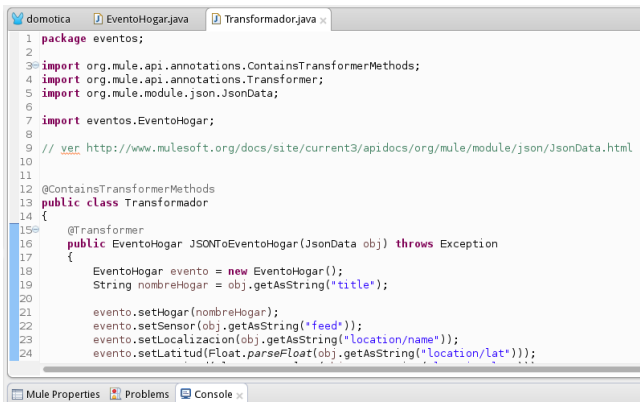


The screenshot shows an IDE window with three tabs: 'domotica', 'EventoHogar.java', and 'Transformador.java'. The 'Transformador.java' tab is active, displaying the following Java code:

```
1 package eventos;  
2  
3 public class Transformador {  
4  
5 }  
6
```

# Transformador Java

Ponemos el cuerpo de la clase descargada del campus virtual

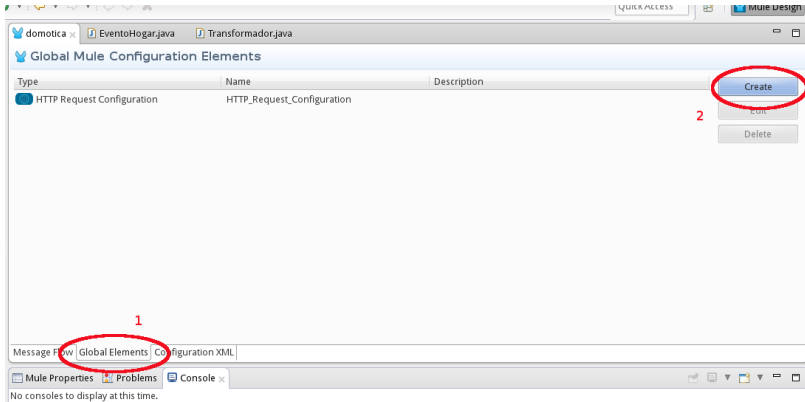


```
1 package eventos;
2
3 import org.mule.api.annotations.ContainsTransformerMethods;
4 import org.mule.api.annotations.Transformer;
5 import org.mule.module.json.JsonData;
6
7 import eventos.EventoHogar;
8
9 // ver http://www.mulesoft.org/docs/site/current3/apidocs/org/mule/module/json/JsonData.html
10
11 @ContainsTransformerMethods
12 public class Transformador
13 {
14     @Transformer
15     public EventoHogar JSONToEventoHogar(JsonData obj) throws Exception
16     {
17         EventoHogar evento = new EventoHogar();
18         String nombreHogar = obj.getString("title");
19
20         evento.setHogar(nombreHogar);
21         evento.setSensor(obj.getString("feed"));
22         evento.setLocalizacion(obj.getString("location/name"));
23         evento.setLatitud(Float.parseFloat(obj.getString("location/lat")));
24     }
25 }
```

# Global Elements

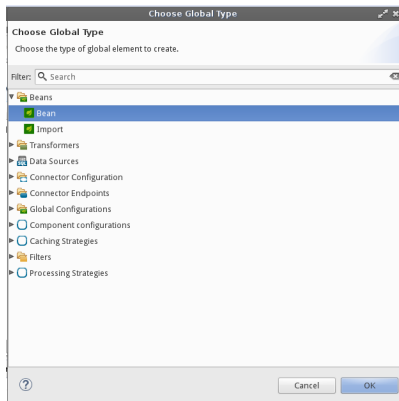
En la pantalla principal, pinchamos en la segunda pestaña: **Global Elements**

Dentro pinchamos en **Create**



# Global Elements

De tipo **Bean**



# Global Elements

Definimos **ID**, **Name** y **Class** como se ve en la figura.

The screenshot shows the 'Global Element Properties' dialog box for a 'Bean'. The dialog has three tabs: 'General', 'Advanced', and 'Notes'. The 'General' tab is selected. The 'Bean' section describes the element as 'Defines a single bean.' Below this, there are several fields for configuration:

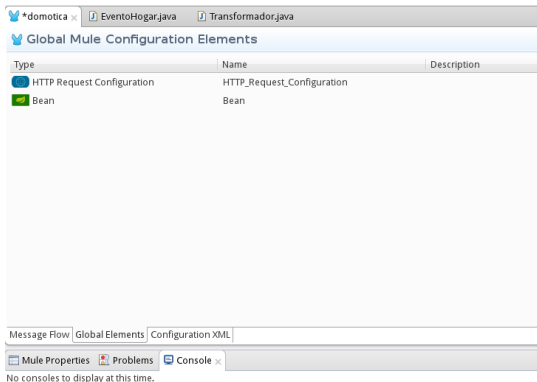
- ID:** transformador
- Name:** Bean
- Class:** eventos.Transformador
- Parent:** (empty)
- Scope:** -- Empty --

Below these fields is a section for 'Bean Subelements' with a list box and icons for adding, removing, and refreshing. At the bottom, there is a 'Child beans' section with similar icons. The dialog also includes a 'Cancel' button, an 'OK' button, and a help icon (?) in the bottom left corner.



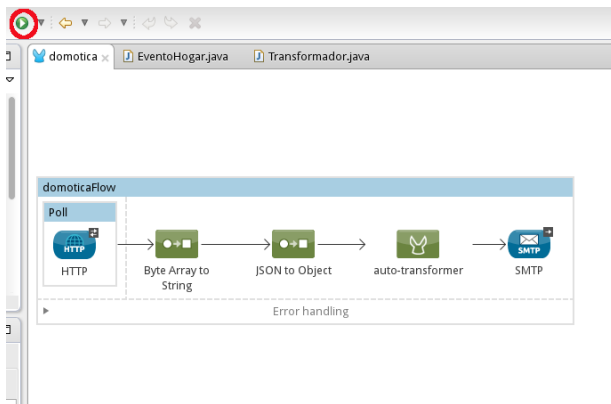
# Global Elements

Así quedará la pestaña **Global Elements**



# Ejecución

Ejecutamos de nuevo



# Ejecución

El cuerpo del email recibido sera así:

[DS] Alerta datos domotica



Recibidos x

gm.uca.es x

UCA x



antonio.balderas@uca.es

para mí ▾



inglés ▾

> español ▾

[Traducir mensaje](#)

Lugar:my\_house

- sensor=Netduino Plus connected to sensors around the house
- latitud=40.44
- longitud=-79.9965
- tiempoRegistro=null
- luz=44.50106
- temperaturaExt=318.0

# Referencias bibliográficas I



MuleSoft Inc.

Mule Studio

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