

SmartRural

Ingeniería de Sistemas de la Información

Guillermo López García

12 de enero de 2021



UCA

Universidad
de Cádiz

Índice

1. Introducción	3
1.1. Motivación	3
2. Objetivos	3
3. Arquitectura del sistema	3
4. Implementación	4
5. Pruebas	13
Referencias	14

1. Introducción

SmartRural es un proyecto para mezclar dos mundos: la tecnología y la agricultura. Gracias a la automatización de procesos y el procesamiento continuo de datos recogidos gracias a distintos sensores, este proyecto puede mejorar uno de los sectores mas anticuados hasta competir con otros mas modernos.

1.1. Motivación

La motivación para realizar este proyecto se basa en mi experiencia personal, dedicada al mundo rural durante mucho tiempo y siempre con la aspiración de mezclar lo que ha sido durante mucho tiempo mi medio para ganarme la vida, el campo, y lo que era en su tiempo un hobby, la informática.

2. Objetivos

Los objetivos pensados para este proyecto son 5:

- Lanzar datos relacionados con el mundo rural a un sistema de tiempo real.
- Recoger dichos datos y lanzar eventos complejos.
- Capturar dichos eventos y realizar un procesamiento de dichos datos en tiempo real.
- Crear un backend robusto para filtrar toda la información obtenida y los eventos disparados.
- Crear un frontend atractivo para mostrar la información de forma que pueda ser útil tanto para personas técnicas como para personas normales que no entiendan de forma técnica la informática.

3. Arquitectura del sistema

La arquitectura del sistema esta basada en una típica del IoT, Internet of Things, donde la información proviene de pequeños dispositivos llamados sensores que envían la información a un sistema distribuido.

Este sistema distribuido gestiona estos valores provenientes de los sensores mediante eventos complejos, los cuales al dispararse provocan distintas

acciones, las cuales se sirven para gestionar de forma automática las máquinas agrícolas.

A continuación, una imagen de la arquitectura seguida:

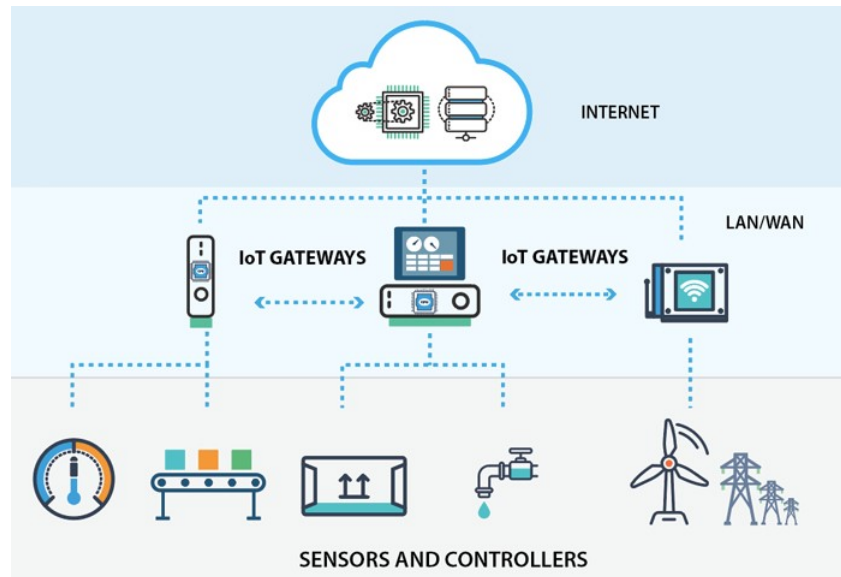


Figura 1: Representación gráfica de la arquitectura de SmartRural

4. Implementación

La implementación de este proyecto es una de las mejores definiciones que se podrían dar de un sistema distribuido encargado de gestionar eventos complejos.

Así pues, empezamos a listar:

- **Java y Maven:** para la generación aleatoria de los datos simulando la información dada por cualquier sensor.
- **ThingSpeak:** plataforma usada para almacenar brevemente la información de los sensores.
- **Anypoint Studio y Mule:** usado para obtener la información y disparar los eventos complejos que realizaran las distintas acciones de automatización.
- **Esper:** para la creación de los eventos.
- **MySQL:** para guardar la información después del procesamiento por el Mule y proveer al backend de información.

- **NodeJS:** usado para hacer el backend, acceder a la base de datos y hacer un CRUD de la misma mediante endpoints para proveer a un frontend que se encargue de mostrar de forma agradable la información.
- **Ionic con React:** por último, esta tecnología nos permite crear un frontend amigable con pocas líneas de código para hacerlo amigable, multiplataforma y bonito.

Todo este conjunto de tecnologías es lo que nos permite alcanzar la máxima eficiencia y la mayor independencia de los datos y de los resultados.

Así pues, pasaremos a listar el código de los eventos y la definición de la base de datos, junto con el código xml del simulador, que es lo que más llamativo puede ser y menos relleno es. Por motivos obvios, no vamos a listar todo el código del backend ni del frontend, ya que, es un código demasiado largo y que no aporta nada a la compresión de la memoria.

Así pues, listamos los eventos:

```

1  @Name(" OpenCeilingGreenHouse")
2  insert into OpenCeilingGreenHouse
3  select sr.sensorId as sensorId , sr.isAtDaytime as
isAtDaytime , sr.isRaining as isRaining
4  from pattern [
5      every sr = SmartRural(
6          isAtDaytime >= 0.5 ,
7          isRaining >= 0.5
8      )
9  ];

```

Código 1: OpenCeilingGreenHouse.epl

```

1  @Name(" Irrigate ")
2  insert into Irrigate
3  select sr.sensorId as sensorId , sr.isAtDaytime as
isAtDaytime , sr.isRaining as isRaining ,
4      sr.airHumidity as airHumidity , sr.roomTemperature as
roomTemperature
5  from pattern [
6      every
7          sr = SmartRural(
8              isAtDaytime >= 0.5 ,
9              isRaining < 0.5 ,
10             airHumidity >= 2.5 ,
11             roomTemperature < 25 // grade celsius
12         )
13 ];

```

Código 2: Irrigate.epl

```

1  @Name(" OpenWallGreenhouse")
2  insert into OpenWallGreenhouse
3  select sr.sensorId as sensorId , sr.windForce as windForce ,
4  sr.isRaining as isRaining ,
5  sr.roomTemperature as roomTemperature
6  from pattern [
7  every sr = SmartRural(
8  windForce < 0.5 ,
9  isRaining < 0.5 ,
10 roomTemperature > 25 // grade celcius
11 )
12 ].win:time_batch(1 minutes); // 1 min, TODO: change it to 1
13 hour
14
15 @Name(" CanOpenWallGreenhouse")
16 insert into CanOpenWallGreenhouse
17 select count(sensorId) as numEvents, (count(sensorId) > 1)
18 as canOpen
19 from OpenWallGreenhouse.win:time_batch(1 min)
20 group by sensorId
21 having count(sensorId) > 1; // change to a number more big

```

Código 3: CanOpenWallGreenhouse.epl

```

1  @Name(" Fertilizer")
2  insert into Fertilizer
3  select sr.sensorId as sensorId , sr.isAtDaytime as
4  isAtDaytime, sr.isRaining as isRaining ,
5  sr.airHumidity as airHumidity, sr.
6  canPhotosynthesisImprove as canPhotosynthesisImprove
7  from pattern [
8  every sr = SmartRural(
9  isAtDaytime >= 0.5 ,
10 isRaining >= 0.5 ,
11 airHumidity < 1 ,
12 canPhotosynthesisImprove >= 0.5
13 ) -> timer:at(1, *, *, *, *) // 1 min, TODO: change it
14 to 5 days
15 ];
16
17 @Name(" CanFertilizer")
18 insert into CanFertilizer
19 select sensorId
20 from Fertilizer.win:time_batch(1 min)
21 group by sensorId;

```

Código 4: CanFertilizer.epl

A continuación, la definición de la base de datos:

```

1  drop table if exists SmartRural;

```

```

2      create table SmartRural(
3          id                bigint auto_increment primary
key ,
4          sensorId          bigint ,
5          roomTemperature   float ,
6          airHumidity        float ,
7          groundHumidity    float ,
8          litrePerMeterWater float ,
9          windForce          float ,
10         windDirection      float ,
11         countIllumination   float ,
12         isRaining           tinyint(1) ,
13         isCeilingGreenhouseOpen tinyint(1) ,
14         isWallGreenhouseOpen tinyint(1) ,
15         isAtDaytime         tinyint(1) ,
16         canPhotosynthesisImprove tinyint(1) ,
17         'date'              timestamp not null default
current_timestamp
18     );
19
20     drop table if exists OpenCeilingGreenHouse;
21     create table OpenCeilingGreenHouse(
22         id                bigint auto_increment primary key ,
23         'date'            timestamp not null default current_timestamp ,
24         message varchar(500) not null default 'Mensaje generico
para la apertura del techo del invernadero '
25     );
26
27     drop table if exists Irrigate;
28     create table Irrigate(
29         id                bigint auto_increment primary key ,
30         'date'            timestamp not null default current_timestamp ,
31         message varchar(500) not null default 'Mensaje generico
para poder regar el invernadero '
32     );
33
34     drop table if exists CanFertilizer;
35     create table CanFertilizer(
36         id                bigint auto_increment primary key ,
37         'date'            timestamp not null default current_timestamp ,
38         message varchar(500) not null default 'Mensaje generico
para poder fertilizar el invernadero '
39     );
40
41     drop table if exists CanOpenWallGreenhouse;
42     create table CanOpenWallGreenhouse(
43         id                bigint auto_increment primary key ,
44         'date'            timestamp not null default current_timestamp ,
45         message varchar(500) not null default 'Mensaje generico

```

```

46      para la apertura de la pared del invernadero '
      );

```

Código 5: SmartRural.sql

Por último, aquí el código xml del simulador:

```

1  <?xml version="1.0" encoding="UTF-8"?>
2
3  <mule xmlns:sqs="http://www.mulesoft.org/schema/mule/sqs"
      xmlns:s3="http://www.mulesoft.org/schema/mule/s3" xmlns:db="
      http://www.mulesoft.org/schema/mule/db" xmlns:tracking="http
      ://www.mulesoft.org/schema/mule/ee/tracking" xmlns:jbossts="
      http://www.mulesoft.org/schema/mule/jbossts" xmlns:file="http
      ://www.mulesoft.org/schema/mule/file" xmlns:smtps="http://www
      .mulesoft.org/schema/mule/smtps" xmlns:smtp="http://www.
      mulesoft.org/schema/mule/smtp" xmlns:vm="http://www.mulesoft.
      org/schema/mule/vm" xmlns:http="http://www.mulesoft.org/
      schema/mule/http" xmlns:quartz="http://www.mulesoft.org/
      schema/mule/quartz" xmlns="http://www.mulesoft.org/schema/
      mule/core" xmlns:doc="http://www.mulesoft.org/schema/mule/
      documentation"
4      xmlns:spring="http://www.springframework.org/schema/beans"
5      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
6      xsi:schemaLocation="http://www.springframework.org/schema/
      beans http://www.springframework.org/schema/beans/spring-
      beans-current.xsd
7      http://www.mulesoft.org/schema/mule/core http://www.mulesoft.
      org/schema/mule/core/current/mule.xsd
8      http://www.mulesoft.org/schema/mule/quartz http://www.mulesoft
      .org/schema/mule/quartz/current/mule-quartz.xsd
9      http://www.mulesoft.org/schema/mule/http http://www.mulesoft.
      org/schema/mule/http/current/mule-http.xsd
10     http://www.mulesoft.org/schema/mule/vm http://www.mulesoft.org
      /schema/mule/vm/current/mule-vm.xsd
11     http://www.mulesoft.org/schema/mule/smtp http://www.mulesoft.
      org/schema/mule/smtp/current/mule-smtp.xsd
12     http://www.mulesoft.org/schema/mule/smtps http://www.mulesoft.
      org/schema/mule/smtps/current/mule-smtps.xsd
13     http://www.mulesoft.org/schema/mule/file http://www.mulesoft.
      org/schema/mule/file/current/mule-file.xsd
14     http://www.mulesoft.org/schema/mule/jbossts http://www.
      mulesoft.org/schema/mule/jbossts/current/mule-jbossts.xsd
15     http://www.mulesoft.org/schema/mule/ee/tracking http://www.
      mulesoft.org/schema/mule/ee/tracking/current/mule-tracking-ee
      .xsd
16     http://www.mulesoft.org/schema/mule/db http://www.mulesoft.org
      /schema/mule/db/current/mule-db.xsd
17     http://www.mulesoft.org/schema/mule/s3 http://www.mulesoft.org
      /schema/mule/s3/current/mule-s3.xsd

```



```

18 http://www.mulesoft.org/schema/mule/sqs http://www.mulesoft.
   org/schema/mule/sqs/current/mule-sqs.xsd">
19   <http:request-config name="HTTP_Request_Configuration"
   host="api.thingspeak.com" port="80" doc:name="HTTP Request
   Configuration"/>
20   <vm:connector name="VM" validateConnections="true" doc:name
   ="VM"/>
21   <vm:endpoint exchange-pattern="one-way" path="
   ComplexEventConsumerPath" name="ComplexEventConsumerGlobalVM"
   doc:name="VM"/>
22   <db:mysql-config name="MySQL_Configuration" host="127.0.0.1"
   port="3306" user="isi" password="isi" database="smartrural"
   doc:name="MySQL Configuration"/>
23   <s3:config name="Amazon_S3_Basic_Configuration" accessKey="
   AKIAIRCGR4L5ZHCTWZEA" secretKey="4jhI+d7cJCYN+A+
   NMIEI9UFNofMabWwdEtluwbVP6" doc:name="Amazon S3: Basic
   Configuration"/>
24   <sqs:config name="Amazon_SQS__Configuration" accessKey="
   AKIAIRCGR4L5ZHCTWZEA" secretKey="4jhI+d7cJCYN+A+
   NMIEI9UFNofMabWwdEtluwbVP6" defaultQueueName="smartrural-isi"
   doc:name="Amazon SQS: Configuration"/>
25   <flow name="smartruralFlow">
26     <quartz:inbound-endpoint jobName="j1" repeatInterval
   ="60000" responseTimeout="10000" doc:name="Quartz">
27       <quartz:event-generator-job/>
28     </quartz:inbound-endpoint>
29     <scatter-gather doc:name="Scatter-Gather">
30       <http:request config-ref="
   HTTP_Request_Configuration" path="/channels/1252445/feed.json
   ?results=1" method="GET" doc:name="HTTP Smart Rural 1"/>
31       <http:request config-ref="
   HTTP_Request_Configuration" path="/channels/1252453/feed.json
   ?results=1" method="GET" doc:name="HTTP Smart Rural 2"/>
32     </scatter-gather>
33     <collection-splitter doc:name="Collection Splitter"/>
34     <custom-transformer class="transformers.
   JsonToSmartRuralTransformer" doc:name="JSON to SmartRural
   Transformer"/>
35     <scatter-gather doc:name="Scatter-Gather">
36       <processor-chain>
37         <logger message="#[payload]" level="INFO" doc:name="
   Logger"/>
38         <db:insert config-ref="MySQL_Configuration" doc:name="
   Database">
39           <db:parameterized-query><![CDATA[insert into
   SmartRural(
40             sensorId, roomTemperature, airHumidity, groundHumidity,
   litrePerMeterWater, windForce, windDirection,
   countIllumination,

```

```

41     isRaining , isCeilingGreenhouseOpen , isWallGreenhouseOpen ,
    isAtDaytime , canPhotosynthesisImprove
42 )
43 values(
44     #[payload.SmartRural.sensorId] , #[payload.SmartRural.
    roomTemperature] , #[payload.SmartRural.airHumidity] ,
45     #[payload.SmartRural.groundHumidity] , #[payload.SmartRural.
    litrePerMeterWater] , #[payload.SmartRural.windForce] ,
46     #[payload.SmartRural.windDirection] , #[payload.SmartRural.
    countIllumination] , #[payload.SmartRural.isRaining] ,
47     #[payload.SmartRural.isCeilingGreenhouseOpen] , #[payload.
    SmartRural.isWallGreenhouseOpen] ,
48     #[payload.SmartRural.isAtDaytime] , #[payload.SmartRural.
    canPhotosynthesisImprove]
49 );]]></db:parameterized-query>
50     </db:insert>
51     </processor-chain>
52     <processor-chain>
53         <component doc:name="Send Event to Esper">
54             <singleton-object class="esper.
    SendEventToEsperComponent"/>
55         </component>
56         <sqs:send-message config-ref="
    Amazon_SQS__Configuration" doc:name="Amazon SQS">
57             <sqs:message messageBody="Procesando datos .... #[
    server.dateTime]"/>
58         </sqs:send-message>
59     </processor-chain>
60 </scatter-gather>
61 </flow>
62 <flow name="smartruralFlow1">
63     <vm:inbound-endpoint exchange-pattern="one-way" doc:name
    ="Complex Event Consumer" ref="ComplexEventConsumerGlobalVM">
64         <xa-transaction action="NONE"/>
65     </vm:inbound-endpoint>
66     <choice tracking:enable-default-events="true" doc:name="
    Choice">
67         <when expression="#[message.inboundProperties['
    eventPatternName'] == 'OpenCeilingGreenHouse']">
68             <db:insert config-ref="MySQL_Configuration" doc:name="
    Database">
69                 <db:parameterized-query><![CDATA[insert into
    OpenCeilingGreenHouse(
70                     date
71                 )
72                 values(
73                     now()
74                 );]]></db:parameterized-query>
75                 </db:insert>

```

```

76         </when>
77         <when expression="#[message.inboundProperties['
eventPatternName'] == 'Irrigate']" >
78             <db:insert config-ref="MySQL_Configuration" doc:name="
Database">
79                 <db:parameterized-query><![CDATA[insert into
SmartRural(
80                     date
81                 )
82                 values(
83                     now()
84                 );]]></db:parameterized-query>
85             </db:insert>
86         </when>
87         <when expression="#[message.inboundProperties['
eventPatternName'] == 'CanFertilizer']" >
88             <db:insert config-ref="MySQL_Configuration" doc:name="
Database">
89                 <db:parameterized-query><![CDATA[insert into
SmartRural(
90                     date
91                 )
92                 values(
93                     now()
94                 );]]></db:parameterized-query>
95             </db:insert>
96         </when>
97         <when expression="#[message.inboundProperties['
eventPatternName'] == 'CanOpenWallGreenhouse']" >
98             <db:insert config-ref="MySQL_Configuration" doc:name="
Database">
99                 <db:parameterized-query><![CDATA[insert into
SmartRural(
100                     sensorId, roomTemperature, airHumidity, groundHumidity,
litrePerMeterWater, windForce, windDirection,
countIllumination,
101                     isRaining, isCeilingGreenhouseOpen, isWallGreenhouseOpen,
isAtDaytime, canPhotosynthesisImprove
102                 )
103                 values(
104                     #[payload.SmartRural.sensorId], #[payload.SmartRural.
roomTemperature], #[payload.SmartRural.airHumidity],
105                     #[payload.SmartRural.groundHumidity], #[payload.SmartRural.
litrePerMeterWater], #[payload.SmartRural.windForce],
106                     #[payload.SmartRural.windDirection], #[payload.SmartRural.
countIllumination], #[payload.SmartRural.isRaining],
107                     #[payload.SmartRural.isCeilingGreenhouseOpen], #[payload.
SmartRural.isWallGreenhouseOpen],
108                     #[payload.SmartRural.isAtDaytime], #[payload.SmartRural.

```

```

109     canPhotosynthesisImprove]
110 ) ;]] > </db:parameterized-query>
111     </db:insert>
112     </when>
113     <otherwise>
114         <scatter-gather doc:name="Scatter-Gather">
115             <processor-chain>
116                 <set-payload value="Detected Alert '[message.
inboundProperties['eventPatternName']]:#[payload]'" doc:name=
="Set Payload"/>
117                 <smtps:outbound-endpoint host="smtp.gmail.com"
port="465" user="{email.username}" password="{email.
password}" to="{email.to}" from="{email.from}" subject="
Detected Alert" responseTimeout="10000" doc:name="SMTP"/>
118                 </processor-chain>
119                 <logger message="***Complex Event: #[payload]" level
="INFO" doc:name="Logger"/>
120             </scatter-gather>
121         </otherwise>
122     </choice>
123 </flow>
124 <flow name="smartruralFlow2">
125     <file:inbound-endpoint path="{patterns.path}/event-
patterns" moveToDirectory="{patterns.path}/deployed-pattern"
pollingFrequency="2000" fileAge="1000" responseTimeout
="10000" doc:name="New EPL Event Pattern">
126         <file:filename-regexp-filter pattern=".+\\.epl"
caseSensitive="true"/>
127     </file:inbound-endpoint>
128     <s3:create-object config-ref="
Amazon_S3__Basic_Configuration" bucketName="smartrural-isi"
key="#[message.inboundProperties.originalFilename]" doc:name
="Amazon S3"/>
129     <file:file-to-string-transformer doc:name="File to String
"/>
130     <component doc:name="Add Event Pattern To Esper">
131         <singleton-object class="esper.
AddEventPatternToEsperComponent"/>
132     </component>
133     <choice-exception-strategy doc:name="Choice Exception
Strategy">
134         <catch-exception-strategy when="exception.causeMatches('
com.espertech.esper.*')" enableNotifications="false"
logException="false" doc:name="Catch Exception Strategy">
135             <file:outbound-endpoint path="{patterns.path}/
incorrect-pattern" responseTimeout="10000" doc:name="File"/>
136             <sqs:send-message config-ref="
Amazon_SQS__Configuration" doc:name="Amazon SQS">
                <sqs:message messageBody="Error a la hora de

```

```

137     procesar el fichero #[message.inboundProperties.
138     originalFilename]-error"/>
139     </sqs:send-message>
140     </catch-exception-strategy>
141     </choice-exception-strategy>
142 </flow>
143 </mule>

```

Código 6: smart-rural.xml

5. Pruebas

Las únicas pruebas realizadas del entorno han sido pruebas manuales de flujo, es decir, se ha probado manualmente a lanzar el generador de datos y que el flujo del anypoint los captura con éxito y dispara los eventos.

Posteriormente, esos datos se insertan en la base de datos y por último, son tratados en el backend y mostrados en el frontend.

Referencias

- [1] “NodeJS Documentation,” <https://nodejs.org/es/docs>, último acceso: 12 de enero de 2021.
- [2] “Sequelize Documentation,” <https://sequelize.org/master>, último acceso: 12 de enero de 2021.
- [3] “Ionic Documentation,” <https://ionicframework.com/docs>, último acceso: 12 de enero de 2021.
- [4] “Ionic React Chart Documentation,” <https://github.com/reactchartjs/react-chartjs-2>, último acceso: 12 de enero de 2021.
- [5] “MySQL Documentation,” <https://dev.mysql.com/doc/refman/5.7/en>, último acceso: 12 de enero de 2021.
- [6] “Docker Documentation,” <https://docs.docker.com>, último acceso: 12 de enero de 2021.
- [7] “Docker Compose Documentation,” <https://docs.docker.com/compose>, último acceso: 12 de enero de 2021.
- [8] “Esper Documentation,” <https://www.esper.tech.com/esper/esper-documentation>, último acceso: 12 de enero de 2021.