

SmartRural

Ingeniería de Sistemas de la Información

Guillermo López García

11 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	5
Referencias	6

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:

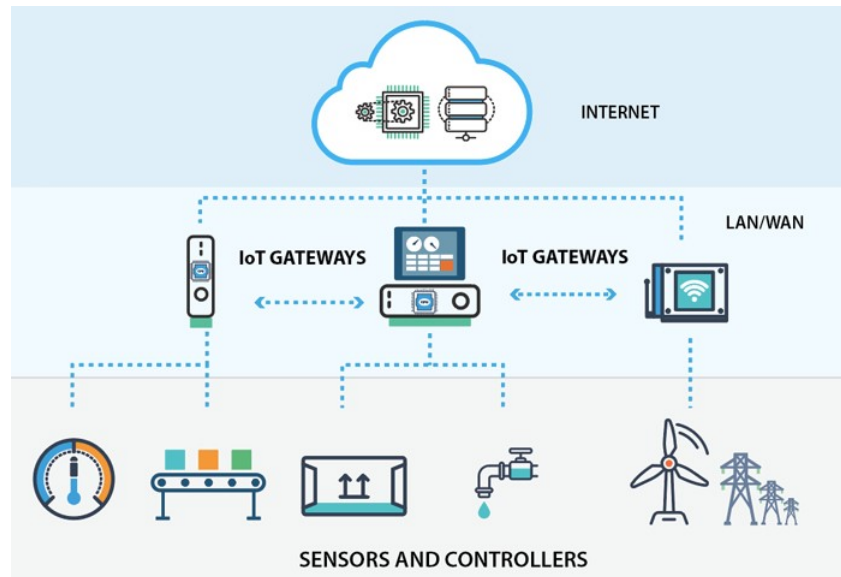


Figura 1: Representación gráfica de la arquitectura de SmartRural

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 realizarán 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.

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: 11 de enero de 2021.
- [2] “Sequelize Documentation,” <https://sequelize.org/master>, último acceso: 11 de enero de 2021.
- [3] “Ionic Documentation,” <https://ionicframework.com/docs>, último acceso: 11 de enero de 2021.
- [4] “Ionic React Chart Documentation,” <https://github.com/reactchartjs/react-chartjs-2>, último acceso: 11 de enero de 2021.
- [5] “MySQL Documentation,” <https://dev.mysql.com/doc/refman/5.7/en>, último acceso: 11 de enero de 2021.
- [6] “Docker Documentation,” <https://docs.docker.com>, último acceso: 11 de enero de 2021.
- [7] “Docker Compose Documentation,” <https://docs.docker.com/compose>, último acceso: 11 de enero de 2021.
- [8] “Esper Documentation,” <https://www.esper.tech.com/esper/esper-documentation>, último acceso: 11 de enero de 2021.