

Facultad de Ciencias  
Universidad Nacional de Ingeniería

# Diseño e Implementación de un sistema de monitoreo medioambiental

v. 1.0.0

Moreno Vera, Felipe Adrian  
`felipe.moreno.vera@gmail.com`

December 15, 2016



## Introducción

Motivación

## Estado del arte

Plataformas IoT

## Tecnologías utilizadas

web Application

rest Web Services

## Patrón de desarrollo

Model-View-Controller

## Análisis del sistema

Diagrama de clases

Actores del sistema

Diagrama de CU

## Plataforma web



- La motivación por el cual se decidió esta temática es debido a la enorme cantidad de software, librerías y frameworks que realizan este trabajo usando otras tecnologías para la obtención y actualización en un determinado tiempo (podría ser tiempo real), además de la gran necesidad que se tiene de un software que brinde información de manera amigable y de fácil uso.



- ▶ Es un software de plataforma web el cual brinda servicios sobre IoT, es decir, puedes crear conectividad con dispositivos cualesquiera, los cuales brindan alguna información que tu quieras y el ClearBlade, usando el protocolo MQTT y SDK de desarrollo, te notificará sobre cambios (encendido apagado, data, conectividad) en los dispositivos asociados, es decir, ClearBlade funciona como broker web Socket MQTT.

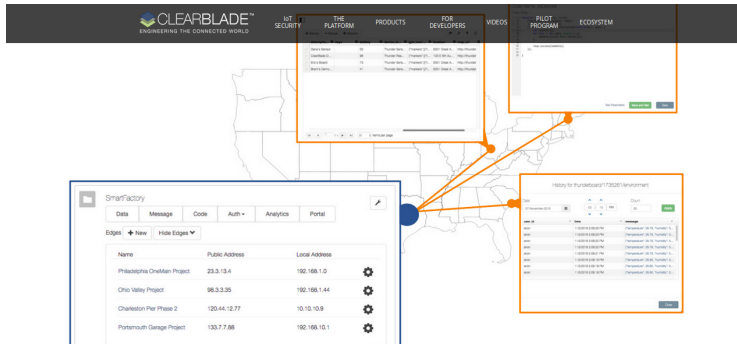


Figure: ClearBlade service



- ▶ Es un software de plataforma web el cual brinda servicios storage para cualquier dispositivo afiliado y que tenga conectividad a internet para almacenamiento, los protocolos que usa Hologram para el envío son bluetooth, wifi y GSM, los cuales lo envían a su plataforma web, almacenándola para posteriormente mostrar la información obtenida en su dashboard.

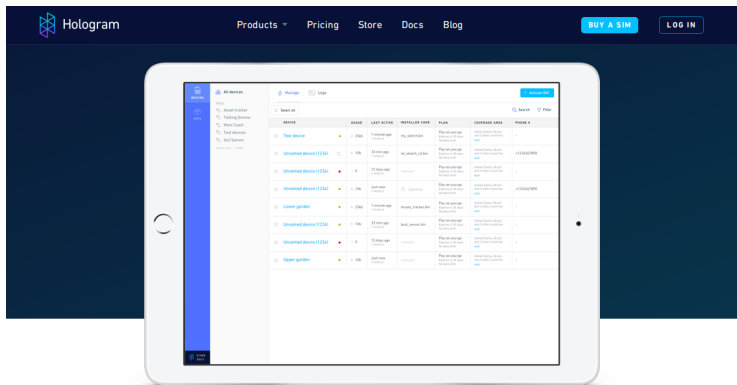


Figure: Hologram platform service



### Patrón de desarrollo

El patrón de desarrollo sobre el cual se ha planteado el diseño de la implementación de la plataforma web es MVC (model-view-controller).

### Sistema controlador de versiones

Para el desarrollo de una plataforma, una librería, un paquete o un proyecto en general, lo más recomendable es guardar las modificaciones o correcciones de errores según aparecen durante la etapa de implementación.





### Sistema de gestión de base de datos

Para la implementación de la base de datos, se tomará en cuenta el tamaño de información y desarrollo de una plataforma, una librería, un paquete o un proyecto en general, lo más recomendable es guardar las modificaciones o correcciones de errores según aparecen durante la etapa de implementación.

### Inyección de dependencias

Es un patrón de diseño orientado a objetos, en donde se asignan objetos a una clase en lugar de que la misma clase instancie el objeto, utilizado para la implementación de la aplicación web haciendo llamadas a diversos componentes necesarios para el funcionamiento.



### Definición

El MVC es un patrón de desarrollo para implementar interfaces de usuario en equipos o dispositivos. Divide una aplicación de software en tres partes interconectadas, con fin de separar las representaciones internas de información que se presenta o se acepta del usuario.

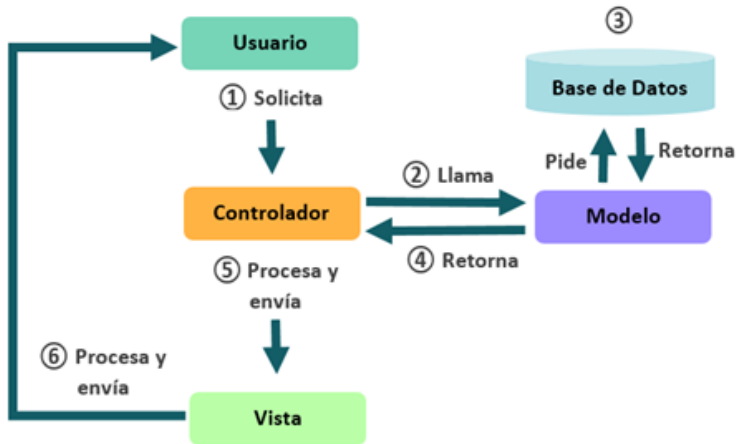


Figure: Patrón MVC



### MongoDB

- Base de dato NoSQL, es decir, no relacional, mediante la cual se pueden añadir colecciones de datos de estructura heterogénea.

```
db.usuarios.insert({  
  "id_usuario": "U0001",  
  "f_registro": "2016-05-09",  
  "h_registro": "21:44:10",  
  "password": "123456"  
})
```

Figure: documento de un usuario en MongoDB



## AngularJS

- ▶ Se utilizó el lenguaje de programación javascript, así como en framework AngularJS, mediante el cual utilizando su ingenio de desarrollo, se realiza inyección de dependencias de los diversos componentes (ya sea de bootstrap o vendor).

## Go language

- ▶ Se utilizó el lenguaje de programación go, desarrollado por google para levantar un servicio cliente en un puerto determinado, la cual será la web que interactuará con el usuario.



### Github

- ▶ Se utilizó el sistema controlador de versiones git, mediante el cual tenemos un registro de las modificaciones en base a los commits realizados, en caso de modificar y tener fallas, podremos regresar a la versión anterior (antes de cometer la falla).



## HTTP

- Se utilizó el protocolo de comunicación HTTP para las consultas entre el servicio cliente al servicio rest.

`http://localhost:1234/home/index/100`

`http://localhost:1234/home/index`

Routing in MVC

Figure: routing de los urls del servicio rest

## MQTT

- Se planteó la utilización de este protocolo para la actualización de los datos obtenidos en tiempo real.

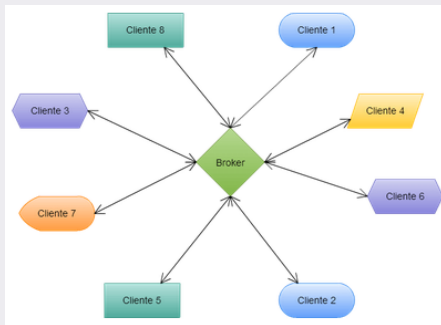


Figure: broker de mqtt



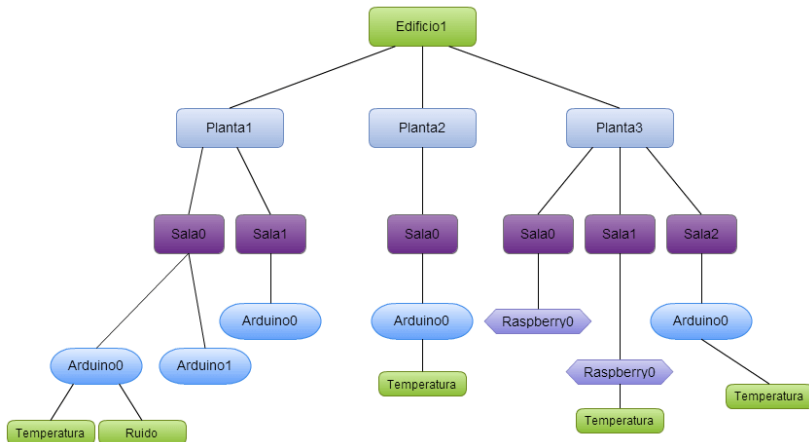


Figure: publica/suscribe

# Análisis del sistema

## Diagrama de clases

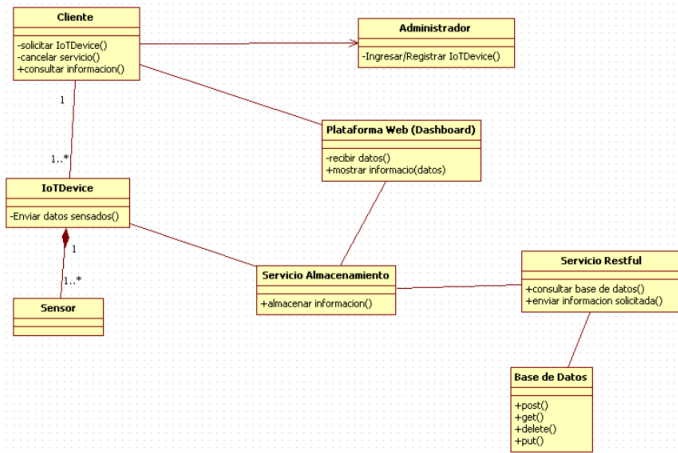


Figure: Diagrama de clases del sistema



- ▶ **Administrador** Es el actor encargado de la administración de los usuarios. Es el único usuario que no tiene IoTDevice registrados.
- ▶ **Usuario** Es el actor que ha solicitado un servicio al sistema, es decir, está al pendiente de la información sensada mostrada por el sistema dependiendo de los dispositivos que el mismo usuario haya solicitado.
- ▶ **IoTDevice** Es el actor que envía la información que capta de los sensores y los envía al actor de almacenamiento.
- ▶ **Dashboard** Es el actor encargado de mostrar la información sensada y almacenada.
- ▶ **Restful** Es el actor que se encarga de realizar y devolver respuestas según las peticiones del actor dashboard, así como también consultar el storage.
- ▶ **Storage** Es el actor encargado del almacenaje de la información

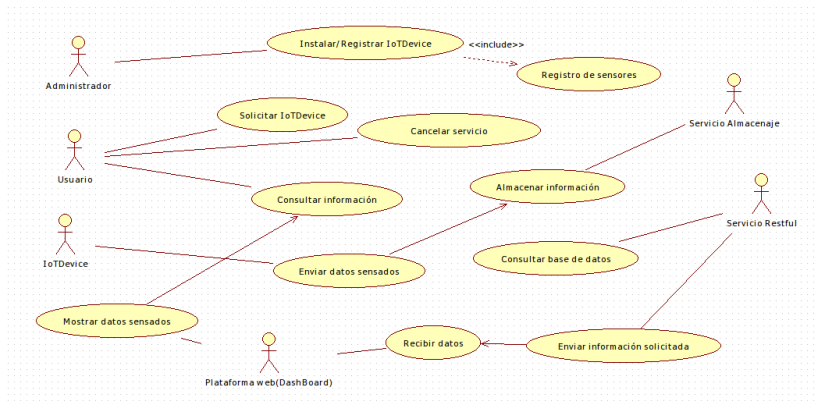


Figure: Diagrama de los actores de casos de uso



SmartCities - Dashboard - Mozilla Firefox

Tesis de ... monitor... Monitor... GPS Trac... MONITO... M2M - M... ONE ... mvc rest ... Modelo... model vi... SmartCiti... x +

localhost:9395/index\_dashboard.html#/dashboard/info/cansats

Smart Cities - Dashboard

## lista de Dispositivos

Se encuentra en: > Dispositivos Info

Datos de Dispositivos

| ID CANSAT | MODELO | UBICACION | LONGITUD    | LATITUD     | FECHA DE INSTALACION | HORA DE INSTALACION |
|-----------|--------|-----------|-------------|-------------|----------------------|---------------------|
| C1001     | veri   | Zona 1    | -77.0499641 | -12.0166427 | 2016-05-09           | 21:44:10            |
| C1003     | veri   | Zona 1    | -77.0495435 | -12.0173944 | 2016-09-20           | 13:44:10            |
| C1004     | veri   | Zona 1    | -77.0503843 | -12.0157352 | 2016-05-09           | 21:44:10            |

localhost:9395/index\_dashboard.html#/dashboard/home/

Figure: vista de la información de los IoTDevice

# Plataforma web

## Información de los sensores por dispositivo



21

SmartCities - Dashboard - Mozilla Firefox

localhost:9395/index\_dashboard.html#/dashboard/info/sensores

Smart Cities - Dashboard

### lista de Sensores

Se encuentra en: > Sensores Info

Datos de Sensores

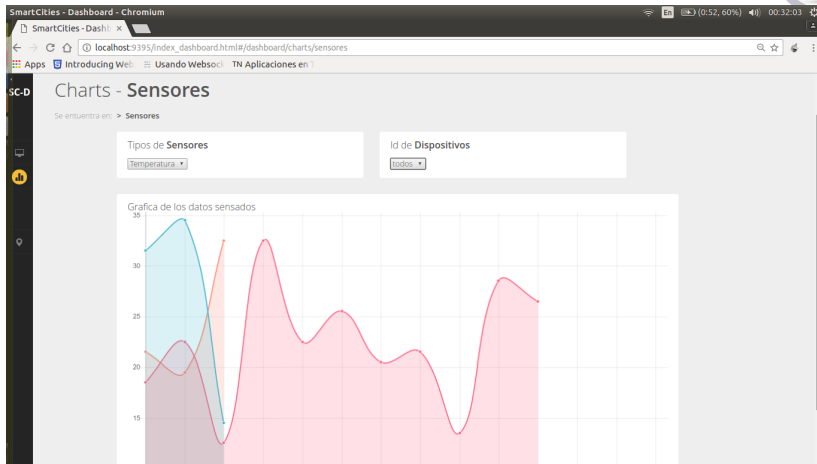
| ID CANSAT | ID SENSOR | TIPO SENSOR | PREFIJO | UNIDAD | MODELO | FECHA DE INSTALACION | HORA DE INSTALACION |
|-----------|-----------|-------------|---------|--------|--------|----------------------|---------------------|
| C1001     | S50001    | Temperatura | T       | °C     | T1     | 2016-05-09           | 21:44:10            |
| C1001     | S51001    | Presion     | P       | Pa     | PA1    | 2016-05-09           | 21:44:10            |
| C1001     | S52001    | Monoxido    | CO      | ppm    | CO1    | 2016-05-09           | 21:44:10            |
| C1003     | S50003    | Temperatura | T       | °C     | T1     | 2016-05-09           | 21:44:10            |
| C1003     | S53002    | Dioxido     | CO2     | ppm    | CO2    | 2016-05-09           | 21:44:10            |
| C1004     | S53003    | Dioxido     | CO2     | ppm    | CO2    | 2016-05-09           | 21:44:10            |

localhost:9395/index\_dashboard.html#/dashboard/home/

Figure: vista de la información de los IoTDevice y sus respectivos sensores

# Plataforma web

## Información de los sensores por tipo de sensor



**Figure:** vista de la información de los IoTDevice obtenida de un tipo de sensor determinado

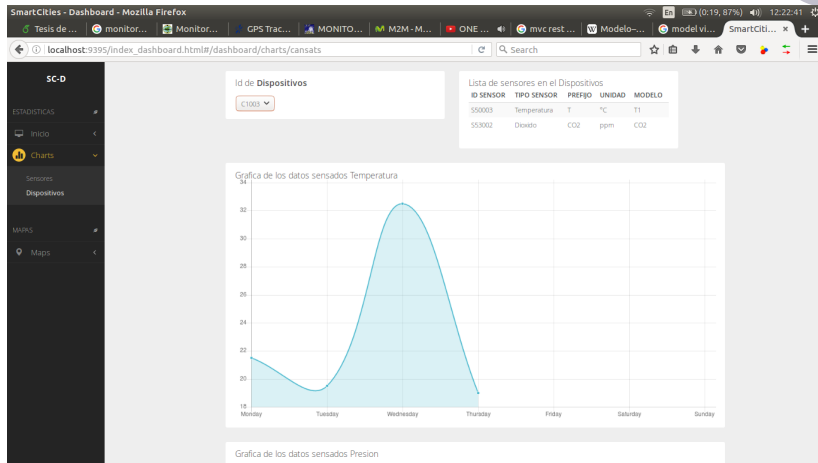


Figure: vista del muestreo de datos por dispositivo





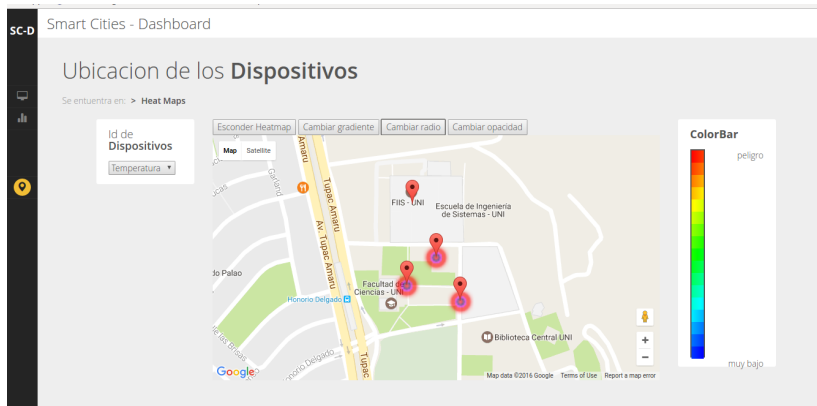


Figure: vista del mapa de calor de los IoTDevice

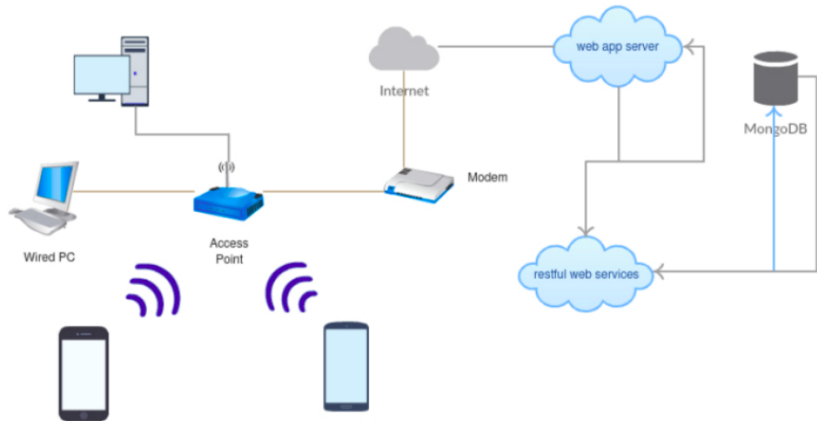


Figure: vista general de los servicios utilizados de la plataforma web

An abstract graphic consisting of multiple flowing, curved lines in shades of light blue and white. The lines originate from the left and curve towards the right, creating a sense of motion and fluidity. Some lines have small, glowing white dots or sparkles along their length. The overall shape is reminiscent of a stylized wave or a plume of smoke.

Gracias!