

# SISTEMA SENS CENTER

## ¿Qué es?

Es una plataforma web de monitoreo, el cual obtiene datos mediante sensores y muestra sus valores mediante gráficas y su respectiva ubicación en el mapa, así como también, un mapa de calor según el valor actual sensado, tendrá un respectivo color.

# ¿Cómo obtener un acceso al servicio?

Los usuarios (clientes), para obtener este servicio, deben comunicarse con el administrador de la empresa y especificar la siguiente información:

- ▶ Cuantos dispositivos va a solicitar (los cuales son llamados “IoTDevice”).
- ▶ Especificar los sensores que tendrá cada IoTDevice (por ejemplo, el IoTDevice1 tendrá temperatura y humedad, el IoTDevice 2 tendrá movimiento y sonido, etc.)
- ▶ Especificar los lugares o sectores en los cuales instalará los IoTDevice.

Una vez especificado la información anterior, el administrador al cual el usuario solicitó el servicio, registrará la información e instalará los IoTDevice con los sensores especificados en los lugares requeridos.

Luego de que el usuario realice el pago correspondiente, recibirá un ID usuario y un password, para poder entrar a la plataforma web (DashBoard) y poder ver la información sensada cada cierto tiempo.

# Algunas Especificaciones

- ▶ Los IoTDevice salen por kits, es decir, que ya hay IoTDevice preparados para cierto tipo de clientes. Por ejemplo, si es de agricultura, tendrá temperatura, humedad, aire, CO2 y CO.
- ▶ Un IoTDevice puede tener de 1 a más sensores específicos solicitados por el cliente, por lo cual tendrá precios diferentes según el número de sensores asociados.
- ▶ Algunos sensores especiales (como los sensores de elementos químicos) tendrán un precio especial.

# Servicio Web

- ▶ El servicio Web (según el patrón de desarrollo MVC) contiene los controllers y las vistas, las cuales siguen unas reglas de routing que se encarga de gestionar los urls y redireccionar a una vista determinada.
- ▶ Es el que interactúa con el usuario, según eventos o acciones de usuario, puede cambiar vistas (usando los routers y templates) para presentación de una pestaña, también realizar peticiones, las cuales son atendidas usando el protocolo http, gestionado por los controllers.

- Dashboard que muestra información de los dispositivos.

SmartCities - Dashboard - Mozilla Firefox

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	verl	Zona 1	-77.0499641	-12.0166427	2016-05-09	21:44:10	
C1003	verl	Zona 1	-77.0495435	-12.0173944	2016-09-20	13:44:10	
C1004	verl	Zona 1	-77.0503843	-12.0157352	2016-05-09	21:44:10	

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

- Esta vista muestra las ubicaciones en el mapa


SmartCities - Dashboard - Mozilla Firefox

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

Smart Cities - Dashboard

## Ubicación de los Dispositivos

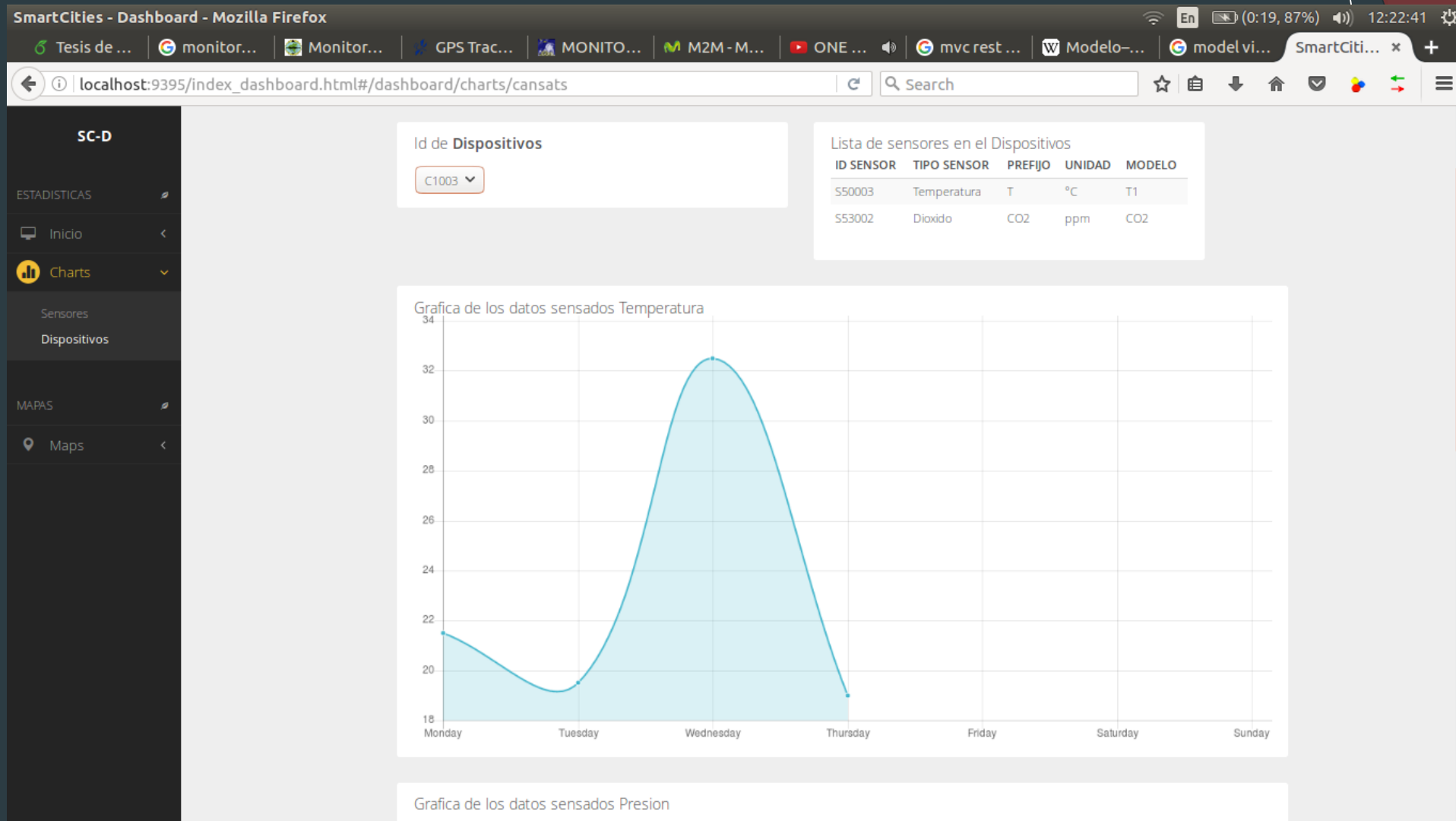
Se encuentra en: > Dispositivos Maps



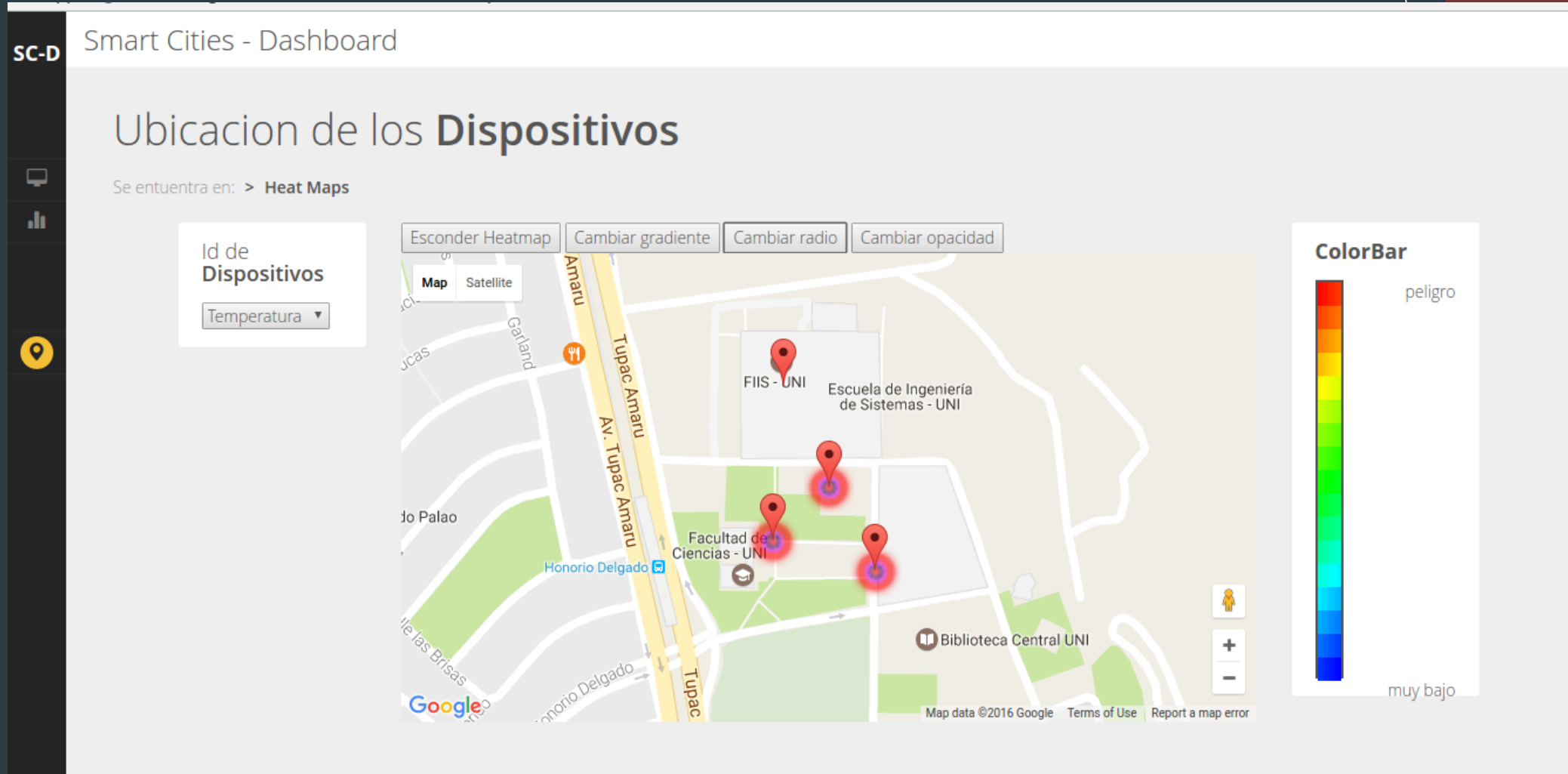
The map displays a street view of a city area, likely in Lima, Peru, centered around the University of Northern Peru (UNI). Key locations marked include the FIIS - UNI, Escuela de Ingeniería de Sistemas - UNI, Facultad de Ciencias - UNI, and Biblioteca Central UNI. The map also shows streets like Av. Tupac Amaru and Calle Las Violetas. A sidebar on the left contains navigation options: ESTADÍSTICAS, Inicio, Charts, MAPAS, Maps, Dispositivos Maps (selected), and Heat Maps. The bottom of the map includes Google logo, map data copyright 2016, and links to Terms of Use and Report a map error.



## Vista donde se muestra los datos obtenidos



# Vista de Mapas de calor



# Servicio Rest

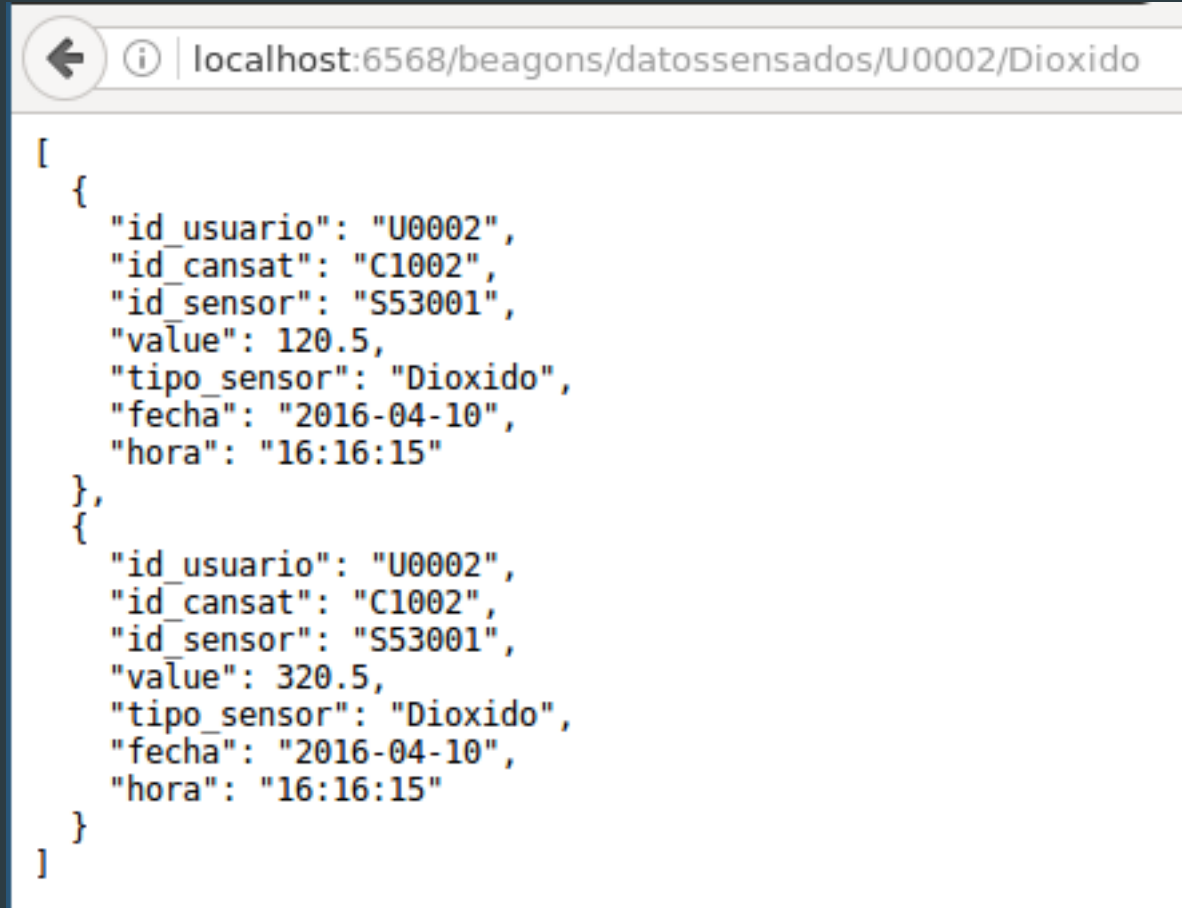
- ▶ Los servicios Rest, en sistemas de información forman parte de la capa de abstracción para el patrón de arquitectura de software Model-View-Controller, en este caso, vendría a ser el modelo del sistema.
- ▶ Son urls generados por un routing el cual se encarga de redireccionar a alguna respuesta de alguna petición según los datos ingresados.
- ▶ Usualmente retornan en formato JSON.

# Routing urls

*Controller* *Action method*  
`http://localhost:1234/home/index/100` *Id parameter value*  
*Controller* *Action method*  
`http://localhost:1234/home/index`

Routing in MVC

## Ejemplo de una consulta al servicio REST

A screenshot of a web browser window. The address bar shows the URL 'localhost:6568/beagons/datossensados/U0002/Dioxido'. The main content area displays a JSON array with two objects. Each object contains fields for user ID, sensor ID, sensor value, sensor type, date, and time.

```
[
  {
    "id_usuario": "U0002",
    "id_cansat": "C1002",
    "id_sensor": "S53001",
    "value": 120.5,
    "tipo_sensor": "Dioxido",
    "fecha": "2016-04-10",
    "hora": "16:16:15"
  },
  {
    "id_usuario": "U0002",
    "id_cansat": "C1002",
    "id_sensor": "S53001",
    "value": 320.5,
    "tipo_sensor": "Dioxido",
    "fecha": "2016-04-10",
    "hora": "16:16:15"
  }
]
```