# Capítulo 10 – Desarrollo del SAR

El desarrollo del SAR se descompone en varios niveles de capas. Por un lado, existen dos esquemas muy diferenciados el lógico y el físico.

El esquema físico se compone de los dispositivos electrónicos que controlan los actuadores, efectores y sensores. Además de los microcontroladores (Arduino Mega, Arduino Nano) y la microcomputadora Raspberry Pi 3.

El esquema lógico se compone del sistema operativo Raspbian y una aplicación web desarrollada bajo la arquitectura cliente/servidor respetando el conjunto de herramientas MEAN. Además contiene un administrador de servicios para Node denominado PM2, y un controlador de cámaras de video conectadas al Sistema Operativo llamado Motion. (Enrique, 2016)

Este esquema a su vez se encuentra organizado en dos unidades funcionales llamadas Back-end y Front-end.

## Esquema general del SAR (lógico /físico)

Ilustración 1 - Esquema general del SAR

## Desarrollo de la aplicación (front-end)

El front-end se encuentra desarrollado en Angular 4+, contando con los siguientes esquemas:

(Pegar esquemas del Compodoc)

Al conectarnos al servidor, desde un cliente (Browser) nos descarga la aplicación embebida, y la comunicación con el servidor sucede a través de Json.

## Desarrollo del servidor (back-end)

El back-end, construido en Node y Express, almacena los estáticos del front-end y los suministra al conectarse un cliente. Los directorios del proyecto los podemos observar en la ilustración (Ilustración 2 - Backend)

Ilustración 2 - Backend

En el directorio Routes, podemos apreciar dos archivos: Api.js y Placas.js. El primero se encarga de gestionar las llamadas por métodos HTTP, implementando REST. Por otro lado, Placas.js se encarga de gestionar la conexión a los periféricos de la Raspberry, que son la Arduino NANO y la Arduino MEGA.

Además, en el directorio dist, se almacena el compilado (js, html, css3) de la aplicación angular y en el raíz del directorio, existe el archivo server.js encargado del despliegue del servidor en el puerto 3000, utilizando node.

La conectividad a MongoDB es realizada a través del driver MongoClient definida en el archivo api.js.

A su vez, Express filtra las rutas accedidas al host (192.168.2.1:3000) resolviendo y respondiendo las distintas consultas realizadas por la App en Angular.

Las distintas llamadas resueltas por Express, utilizando REST son:

* Temperturas 🡪 Api.js, devolviendo temperaturas almacenadas en la BD
* Monoxidos 🡪 Api.js, devolviendo monóxidos almacenadas en la BD
* Apagar 🡪 Api.js, apagando la Raspberry desde el sistema operativo
* Reiniciar 🡪 Api.js, reiniciando la Raspberry desde el sistema operativo
* Arriba, Abajo, Izquierda, Derecha, Stop 🡪 placas.js, ejecutando un orden a los motores para desplazarse o detenerse.
* Ultrasonido 🡪 placas.js, obteniendo los valores de ultrasonido detectados.
* GPS 🡪 placas.js, obteniendo los valores como altura, velocidad, curso, fecha y coordenadas.
* Monoxido 🡪 placas.js, se obtiene el valor actual de monóxido en el ambiente.

Ver anexo de códigos (Api.js, Placas.js, Server.js)

## Esquema de la arquitectura lógica

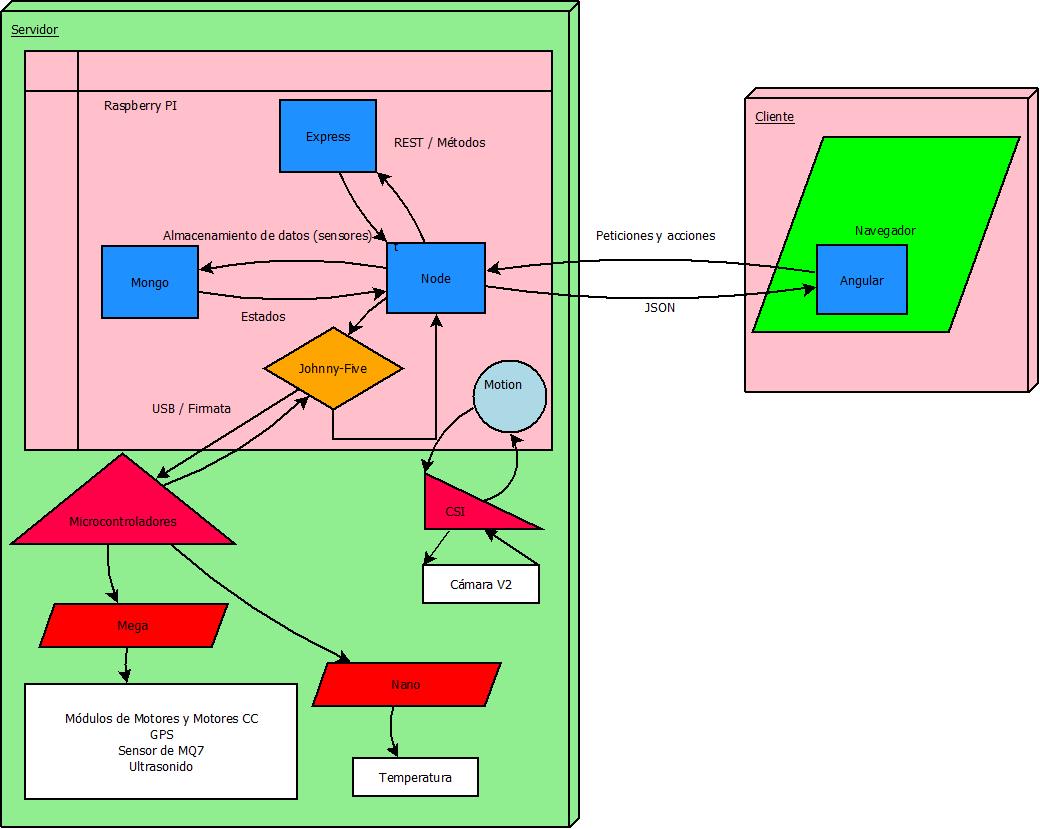


Ilustración 4 - Arquitectura lógica del SAR

## Capturas de pantalla de la aplicación

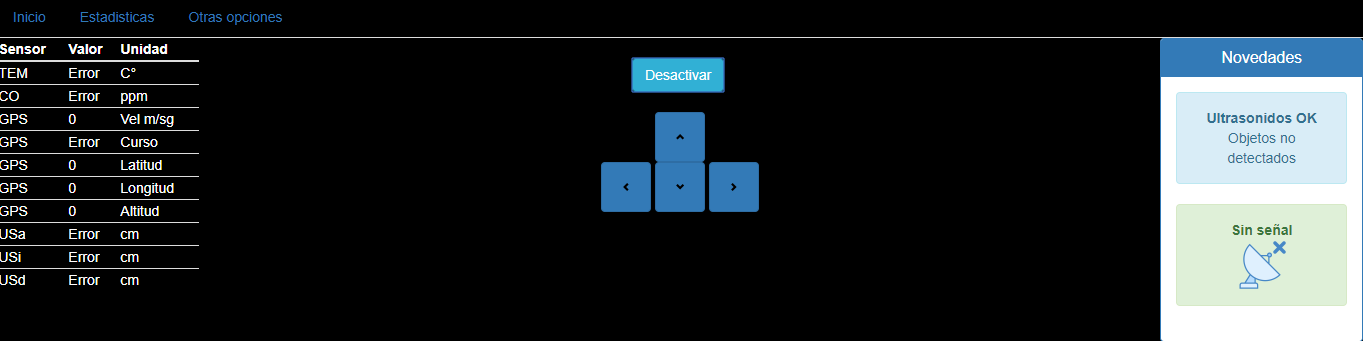


Ilustración 5 - Aplicación Web

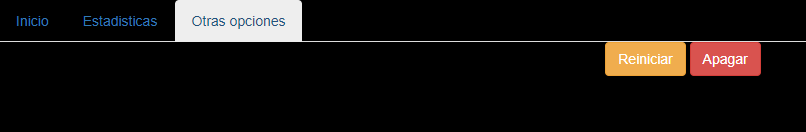


Ilustración 677 - Aplicación Web - Otras opciones

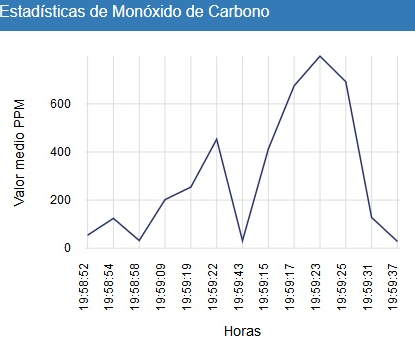


Ilustración 7 - Estadística de monóxido

## Funcionamiento de la App

Para poder acceder a la aplicación del SAR y controlar el robot móvil, se debe conectar a la red WIFI ad-hoc que genera la Raspberry denominada “SAR” con su respectiva contraseña.

Una vez conectado a la red, se debe acceder mediante un navegador web a la URL:

192.168.2.1:3000.

Automáticamente se desplegará la interfaz de la primera ilustración (). Como se puede apreciar en las ilustraciones, la interfaz cuenta con un menú, con tres opciones.

Las opciones son: Inicio, estadísticas y otras opciones.

La solapa de inicio, se encuentra dividida en tres partes. En el panel izquierdo, se observa una tabla de valores obtenidos por los sensores. Estos valores se actualizan en tiempo real. En el panel central se encuentra, un botón para activar/desactivar la visualización de video en tiempo real. Por debajo, de dicho botón, existe un conjunto de botones, que permiten controlar el movimiento y la dirección del robot móvil. En el panel derecho, veremos una sección de “novedades” la cual muestra los distintos estados de los sensores con mensajes de advertencia y/o información al usuario.

En la solapa estadísticas, se visualizan dos diagramas estadísticos. El primero es de temperatura, que a través de unos selectores podemos indicar la fecha de inicio y de fin. De esta forma se confecciona una gráfica, al cliquear en obtener, siempre y cuando existan datos almacenados en el período seleccionado. La gráfica muestra en las columnas, los días y en las filas, las horas del día desde las 00:00 hs hasta las 23:00. En cada una de las intersecciones fecha/hora se muestra el promedio de temperaturas en un rango de 1 hr. En la primera columna, se genera una media de los valores que se encuentran en la misma fila adyacente de las fechas seleccionadas.

COLOR IMAGEN

En el segundo, se toman valores en tiempo real del valor de monóxido de carbono presente en el ambiente. Los mismos se representan según el horario y el valor en PPM obtenido. Por otro lado, se permite colocar el máximo de valores a visualizar.

En la solapa otras opciones, visualizamos dos botones, que permiten apagar o reiniciar el sistema operativo, por ende, del robot móvil.

# Puesta en producción del SAR

Para desplegar el SAR se necesitó de software y configuraciones adicionales que a continuación se describen.

### Configuración Ad-hoc

### Configuración del servicio Motion

### Instalación del gestor de procesos PM2

Node es administrado por PM2 (administrador de procesos para JavaScript) el cual inicia el servidor, automáticamente al arrancar Raspbian, controlando y monitoreándolo. En la siguiente captura (COLOCAR REFERENCIA) podemos apreciar como el gestor de procesos PM2, permite monitorear, controlar y ejecutar la aplicación definida (server.js) que realiza el despliegue del servidor del SAR. En caso de fallos, PM2 re arranca la aplicación emitiendo mensajes. Además, permite monitorear cantidad de reinicios, generar un log, mostrar el tiempo de carga de la aplicación, como reiniciarla, pararla y listar todos los procesos que gestiona.

Por otro lado, permite generar el archivo de startUp(COLOCAR REFERENCIA) para iniciar con el sistema operativo de Raspbian. Este archivo y los pasos de generación se incluirán en el anexo de códigos.

Otra característica de PM2, es la integración con Keymetrics (COLOCAR REFERENCIA), donde si el servidor tiene acceso a Internet, es posible hacer el seguimiento desde internet con un token de seguridad.

