

## Documentación de Solución SIMP

Sistema Inteligente de Monitoreo para Interiores

Camilo Rubilar, Abel Sierra, Christian Muñoz

`camilo.rubilar@mail.udp.cl`

`abel.sierra_1@mail.udp.cl`

`christian.munoz1@mail.udp.cl`

# Índice general

<b>1. Introducción</b>	<b>2</b>
<b>2. Documentación</b>	<b>3</b>
2.1. Estructura.(UML)	3
2.2. Interacciones	5
2.2.1. Arduino	5
2.2.2. API's	5
2.2.3. Cliente	5
2.2.4. Servidor	5

# 1. Introducción

En el presente documento, se entrega la completa documentación de las características y detalles a considerar del proyecto, junto con sus respectivos requerimientos tanto funcionales como no funcionales, tratados y conversados con el cliente durante la entrevista.

## 2. Documentación

### 2.1. Estructura.(UML)

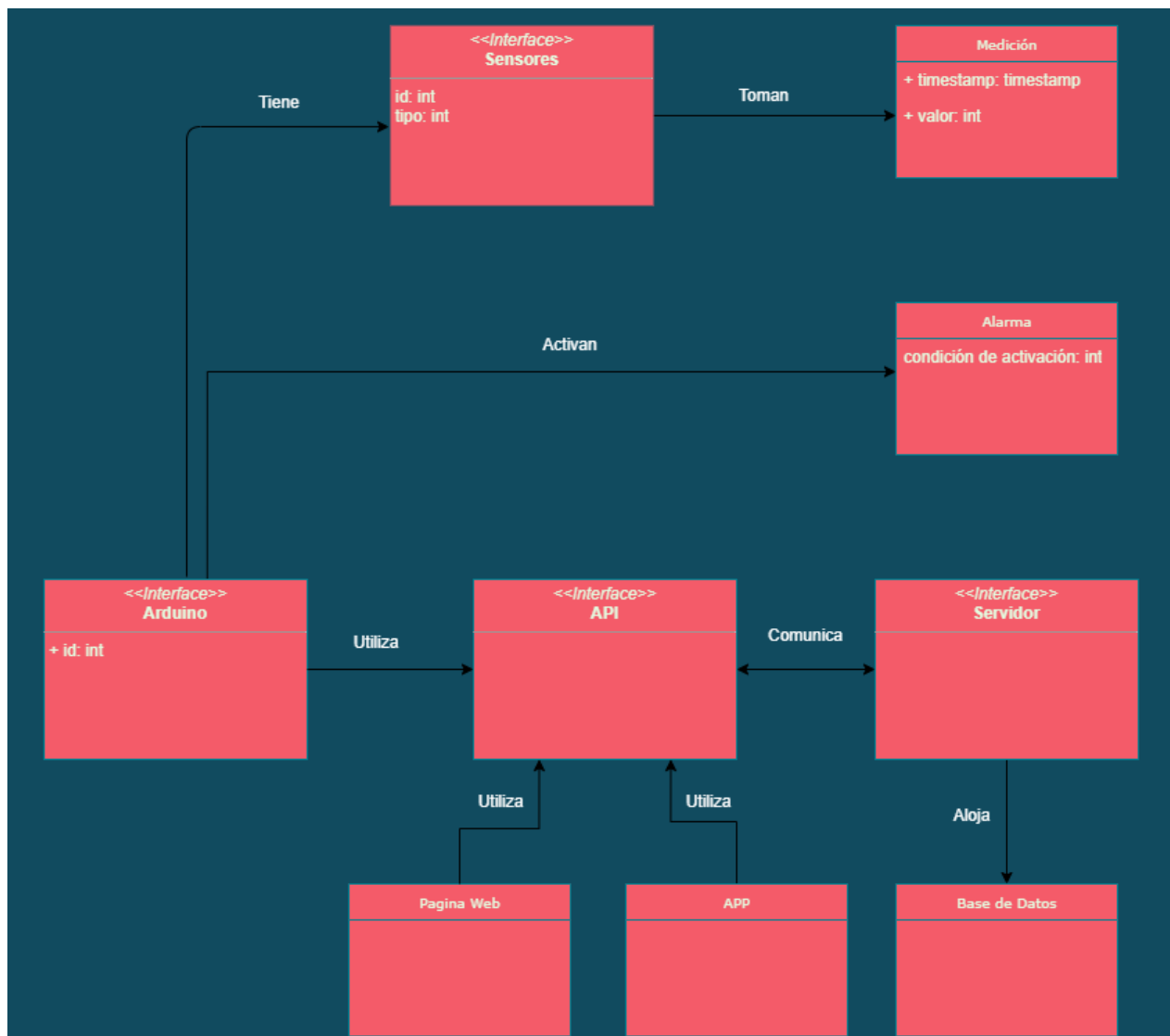


Figura 2.1: UML Estructural de la solución

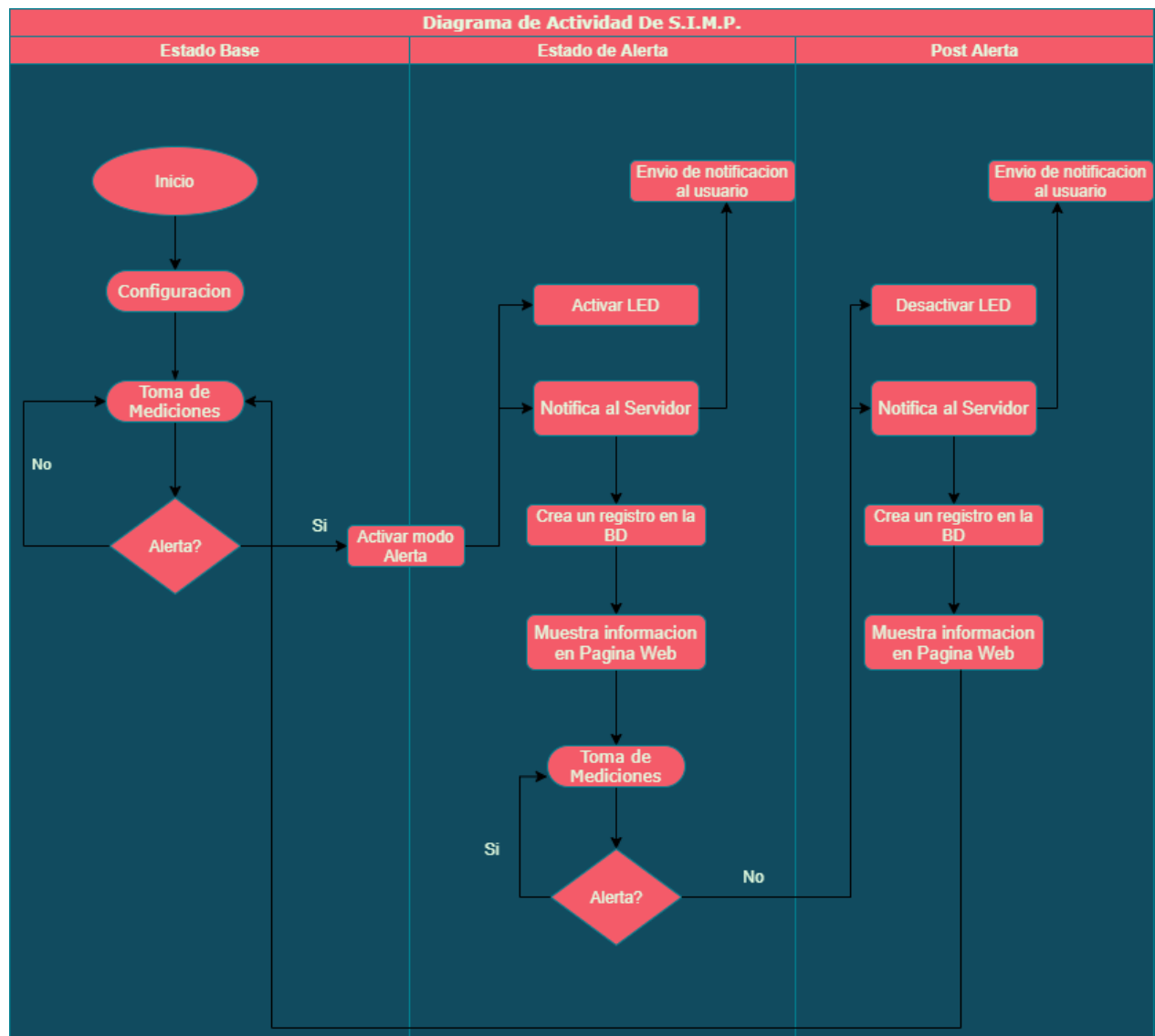


Figura 2.2: Diagrama de actividades de SIMP

## 2.2. Interacciones

### 2.2.1. Arduino

El arduino se encarga de obtener los datos de su entorno los cuales son procesados de manera local para, en caso de ser necesario, emitir alertas de la forma más pronta posible. De forma casi simultánea estos datos son enviados a la API que se encuentra en el servidor para ser registrados y almacenados en la base de datos con el fin de mostrárselos al usuario en forma de gráficas y estadísticas dinámicas, en tiempo real, según el las solicite.

### 2.2.2. API's

La interfaz de programación de aplicaciones esta encargada de procesar y gestionar la totalidad de los datos de SIMP. La interfaz se puede subdividir, según el origen y destino de los datos, en 3 partes:

- **Arduino:** Esta sección esta encargada de recibir, clasificar y almacenar los datos enviados por el arduino, además puede llegar a soportar threads para monitorear el estado de la conexión con un determinado dispositivo para manejar irregularidades en los tiempos de comunicación.
- **WEB:** Esta sección se encarga de leer los datos almacenados y procesarlos de forma apropiada para enviarlos a pagina web en base a lo que solicite el usuario. También trabaja con un sistema de usuario, registro, login, etc. Además usa cookies y sesiones para ser capas de individualizar cada cliente con el fin de responder de la mejor forma posible a sus requerimientos.
- **APP:** Es muy similar a la sección anterior pero busca optimizar el desempeño en dispositivos móviles, ya sea en relación al trafico de datos, diferencia de tamaño, recursos del dispositivo, etc. Esta API orientada a dispositivos móviles nos permite implementar un sistema de notificaciones para alertar de cualquier cambio anormal en el entorno, en base a los datos tomados. Básicamente cuando el arduino emite alguna alarma también se envía una notificación a la app.

### 2.2.3. Cliente

El cliente puede interactuar con SIMP de 3 formas diferentes, la primera sería simplemente mirar el dispositivo físico de SIMP, su pantalla, leds, etc., o estar en un entorno cercano cuando emita alguna alarma.

La segunda forma consiste en entrar a la pagina web, logearse y solicitar a la API lo que se necesite a través de esta pagina. Se espera que esta forma sea la más fácil e intuitiva para el usuario dado que puede interactuar con la base de datos histórica, a diferencia del dispositivo físico, y debido a la presencia de teclado, mouse y pantalla este método posee mejor visual e interacción usuario software en comparación con dispositivos móviles.

La tercera y ultima forma es a través de la APP, se espera que esta sea la más usada, dado que en caso de encontrarse fuera del rango de la alarma de SIMP, es decir, fuera del inmueble, llegaran notificaciones de la irregularidad en curso. Se debe mencionar que tendrá las mismas funcionalidades de la pagina web pero implementadas de otra forma o, lamentablemente, incluso de forma parcial y/o limitada dependiendo del dispositivo.

### 2.2.4. Servidor

En el servidor se guardan los datos en la base de datos, mantiene la API en ejecución, se guarda el .apk de la APP en caso de que se solicite su descarga y además aloja la pagina WEB. La interacción con el cliente es menor, ya que se busca delegar la ejecución de procesos a un sistema de API's tratando de paralelizar procesos y optimizar las respuestas a las consultas de los usuarios.