2017

Informe Final de PPS

Empresa: Arbit Ingeniería

Alumno: Guido Tebes
Carrera: Ingeniería en Sistemas
Tutor Institucional: Leandro Alesso
Tutor Académico: Luis Olsina

Tabla de Contenidos

1. Descripción de la Empresa: Arbit Ingeniería 2
2. Tareas Desarrolladas 3
3. Conclusiones y Trabajos Futuros 13
Bibliografía y referencias 14
ANEXO I: Requerimientos del proyecto 16
ANEXO II: DER final del sistema 21
ANEXO III: Capturas de pantalla de la última versión del sistema 22

1. Descripción de la Empresa: Arbit Ingeniería

Arbit Ingeniería es una empresa perteneciente a la Incubadora de Empresas denominada “INCUBATEC”, situada en calle 21 N° 1354 primer piso, en General Pico, La Pampa. El equipo de la empresa está constituido por personas altamente capacitadas para el desarrollo en el área de la automatización. Entre ellos se encuentran estudiantes avanzados de las carreras de Ingeniería Electromecánica, Ingeniería en Sistemas y Analista Programador con amplia experiencia en el desarrollo a medida.

Arbit se especializa, a nivel general, en *automatización*, *control* y *telemetría*. Según [4], la *automatización* consiste en “*el uso de sistemas o elementos computarizados y electromecánicos para controlar maquinarias o procesos industriales. Como una disciplina de la ingeniería más amplia que un sistema de control, abarca la instrumentación industrial, que incluye los sensores, los transmisores de campo, los sistemas de control y supervisión, los sistemas de transmisión y recolección de datos y las aplicaciones de software en tiempo real para supervisar y controlar las operaciones de plantas o procesos industriales*”. Por otra parte, el *control* consiste en un conjunto de dispositivos encargados de administrar, ordenar, dirigir o regular el comportamiento de otro sistema, con el fin de reducir las probabilidades de fallo y obtener los resultados teóricamente verdaderos [13].

Con respecto a la *telemetría*, la misma permite monitorear remotamente cualquier magnitud física (por ejemplo presión, temperatura, humedad) y visualizarla desde cualquier lugar del mundo en tiempo real a través de Internet. En otras palabras, el servicio de telemetría permite transmitir datos de sensores y contactos en forma remota a través de diferentes tecnologías, ya sea una página web, ethernet, GPRS, entre otras. Diferentes sistemas de telemetría se pueden encontrar documentados en [1, 2, 14, 15].

En Arbit se realizan diferentes actividades relacionadas con la automatización, el control y la telemetría. Entre ellas se encuentran las siguientes actividades:

- Dar soluciones a las industrias para afrontar la problemática de la automatización de los procesos.
- Asesorar a sus clientes para tomar mejores decisiones de compra.
- Diseñar equipos y soluciones con las tecnologías más adecuadas.
- Automatizar maquinaria industrial, procesos y líneas de producción con equipos diseñados a la medida de las necesidades de los clientes.
- Fabricar equipos electrónicos de control especializados.
- Recopilar datos de diferentes sistemas para su visualización remota.

La empresa se encarga de diseñar/desarrollar diferentes productos, entre ellos tenemos:

- **Telemetría por Internet (ETE)**: Tiene como objetivo relevar datos digitales y analógicos para mostrarlos en una página web almacenada en el mismo equipo. A su vez cuenta con la posibilidad de programar secuencias lógicas para darle inteligencia al equipo.
- **Contador Temporizador (CT20)**: Equipo diseñado para contabilizar eventos y tomar acciones a intervalos periódicos de tiempo. Es un equipo flexible que se puede adecuar a las necesidades del cliente.
- **Sistema Scada**: Permite al cliente observar el estado de su planta industrial desde una interfaz web. Además, permite controlar y supervisar procesos industriales a distancia.
- **Control PID (ganancia Proporcional, Integral y Derivativa)**: Un controlador PID es un mecanismo de control por realimentación ampliamente usado en sistemas de control industrial. Este calcula la desviación o error entre un valor medido y un valor deseado. Se puede utilizar para controlar temperatura, corriente, presión, etc.
- **Datalogger de termocuplas tipo K**: Equipo diseñado para tomar muestras de 10 termocuplas tipo K y almacenar los datos en una tarjeta de memoria SD cada cierto intervalo de tiempo.

Para más información relacionada con “Arbit Ingeniería” se puede acceder a su sitio web: *http://arbitingenieria.com.ar.*

2. Tareas Desarrolladas

La empresa propuso como proyecto la realización de un sistema *SCADA* (*Supervisory Control And Data Acquisition*). En pocas palabras, un sistema *SCADA* es un software que permite controlar y supervisar procesos industriales a distancia [12]. La elaboración del proyecto se realizó en las siguientes etapas con sus correspondientes actividades, con una duración total estimada de 200 horas. La carga horaria diaria realizada fue de 4 horas y semanalmente de 5 días. La mayoría de las tareas se realizaron en conjunto con el estudiante Denis Peppino ya que se aplicaron metodologías agiles para llevar a cabo el proyecto [10]. A continuación se describen las etapas planteadas en el anteproyecto de la correspondiente PPS:

- Etapa 1: Introducción a las nuevas Tecnologías
  - Actividad 1.1: Estudio y Compresión de Tecnologías Necesarias para la realización del Proyecto. A realizar en conjunto con Denis Peppino.
- Etapa 2: Introducción a la temática de los Proyectos a realizar

- Actividad 2.1: Contextualización de la problemática planteada en cada proyecto a realizar. A realizar en conjunto con Denis Peppino.

- Etapa 3: Diseño de Proyecto
  - Actividad 3.1: Diseño general del Sistema. A realizar en conjunto con Denis Peppino.
  - Actividad 3.2: Diseño de Módulos Individuales del Sistema.
  - Actividad 3.3: Integración y correspondencia de los distintos Diseños. A realizar en conjunto con Denis Peppino.
- Etapa 4: Desarrollo de Proyecto
  - Actividad 4.1: Desarrollo de los diferentes módulos funcionales/no funcionales del sistema con su correspondiente prueba. A realizar en conjunto con Denis Peppino.
  - Actividad 4.2: Integración de módulos con su correspondiente prueba. A realizar en conjunto con Denis Peppino.

Etapa/Semana	I	II	III	IV	V	VI	VII	VIII	IX	X
1										
2										
3										
4										

Figura 1: Diagrama GANT propuesto para el proyecto

Siguiendo el plan de actividades anteriormente descripto, se detallaran cada una de las actividades realizadas:

Para comenzar, en la actividad 1.1, se seleccionaron las tecnologías que se utilizaron en el proyecto. Dado que el proyecto consistió en la realización de un sistema *SCADA*, el cual permitía gestionar a través de una interfaz web algún tipo de nodo (ya sea un invernadero, una planta industrial, una planta de reciclaje, una impresora, entre otros), se decidió utilizar tecnologías como HTML5 [5], JavaScript [6], PHP [9], JQuery [7] y el marco de trabajo responsivo de Bootstrap [3]. Además, para la gestión de los datos se utilizó como motor de base de datos a MySQL [8].Vale la pena aclarar que tanto yo como mi compañero Denis Peppino teníamos conocimiento y experiencia en dichas tecnologías, los cuales fueron adquiridos durante el transcurso de la carrera. Por mi parte, tuve que investigar y estudiar el lenguaje de programación Python [11], ya que no tenía experiencia alguna con dicho lenguaje.

Siguiendo con la actividad 2.1, se realizó una introducción a la problemática a resolver. Para ello se efectuaron varias reuniones con el tutor institucional, con el fin de dejar en claro los requerimientos que deseaba la empresa. A su vez dicho tutor nos entregó un documento semi-formal donde estaban especificados los requerimientos del sistema (ver Anexo I). Por otra parte, se observaron varios ejemplos de sistemas *SCADA* (para así tener una idea más clara de lo que se solicitaba) entre estos tenemos a RRDTools (*https://oss.oetiker.ch/rrdtool/doc/rrdtool.en.html*), Observium (*http://www.observium.org/*), Cacti (*https://www.cacti.net/*), y Bootstrap SB Admin 2 (*https://startbootstrap.com/template-overviews/sb-admin-2/*). En resumen, se solicitó un sistema *SCADA* Web que permitiera monitorear parámetros de un nodo, con intervalos de tiempo configurables, como así también debía permitir el ingreso de distintos usuarios con acceso a determinados nodos. Además, el sistema debería de contar con un sub-sistema de gestión de alarmas para los parámetros monitoreados. Por último, también se pidió la realización de los ABM (Altas, Bajas y Modificaciones) necesarios para las demás funciones del sistema. Para más detalles respecto del sistema a realizar ver Anexo I. Notar que nodo es un término genérico que sirve para representar diferentes entidades, por ejemplo un invernadero, un motor, etc.

Una vez discutidos, analizados y aclarados los requerimientos del proyecto, se procedió con la actividad 3.1. La misma consistió en realizar un diseño de alto nivel del sistema. Como se puede ver en la Figura 2, el diseño del sistema de monitoreo consiste en cuatro grandes módulos a desarrollar: Gráficos, Gestión de Alarmas, Gestión de Usuarios y Gestión de Nodos.

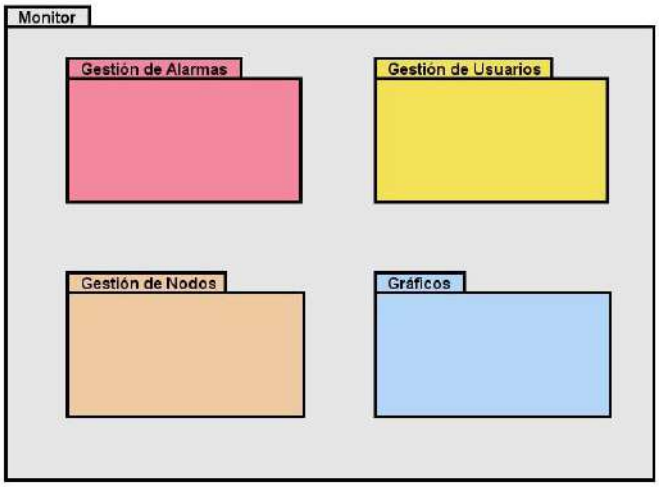


Figura 2: Esquema general del sistema de monitoreo

Para continuar, en la actividad 3.2, realice el diseño de cada uno de los módulos anteriormente nombrados (ver Figura 2) pero con más baja granularidad, es decir, un diseño donde se puedan apreciar los diferentes sub-módulos del sistema. Se decidió separar todos los módulos en dos grandes sub-módulos: “Interfaz Gráfica” y “ABM (Altas, Bajas y Modificaciones)”. Además, se diseñaron los módulos pertenecientes a cada uno de dichos sub-módulos a desarrollar. En la Figura 3 se puede apreciar el diseño del módulo Gestión de Alarmas con sus respectivos sub-módulos.

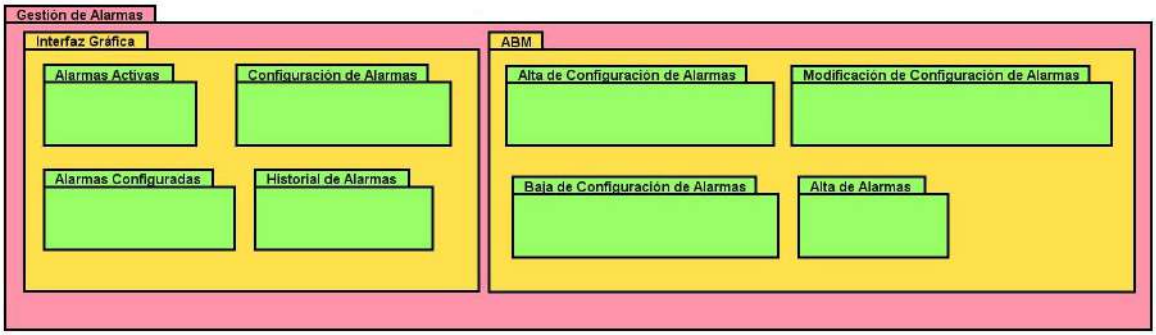


Figura 3: Diseño del módulo “Gestión de Alarmas”

Con respecto a “Gestión de Usuarios” y “Gestión de Nodos”, en las Figuras 4 y 5, respectivamente, se puede ver los correspondientes diseños de más bajo nivel.

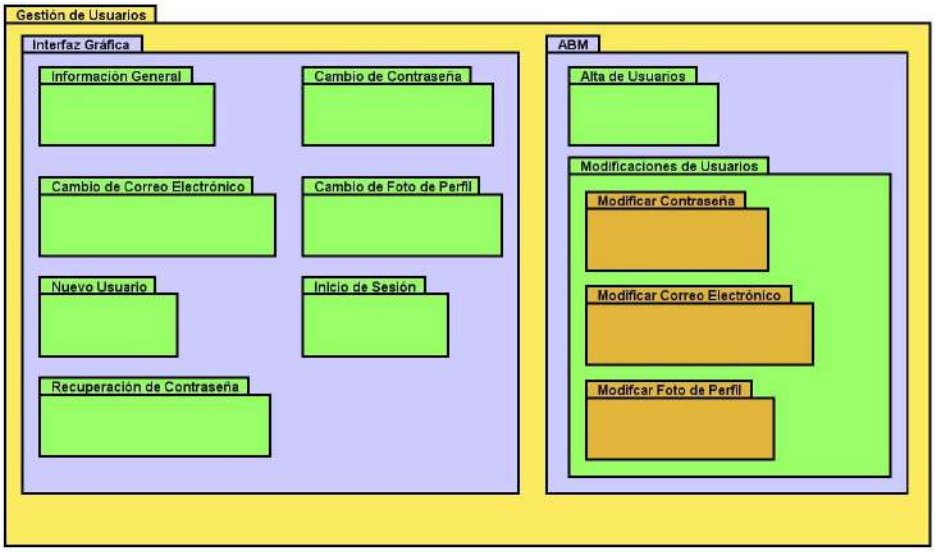


Figura 4: Diseño del módulo “Gestión de Usuarios”

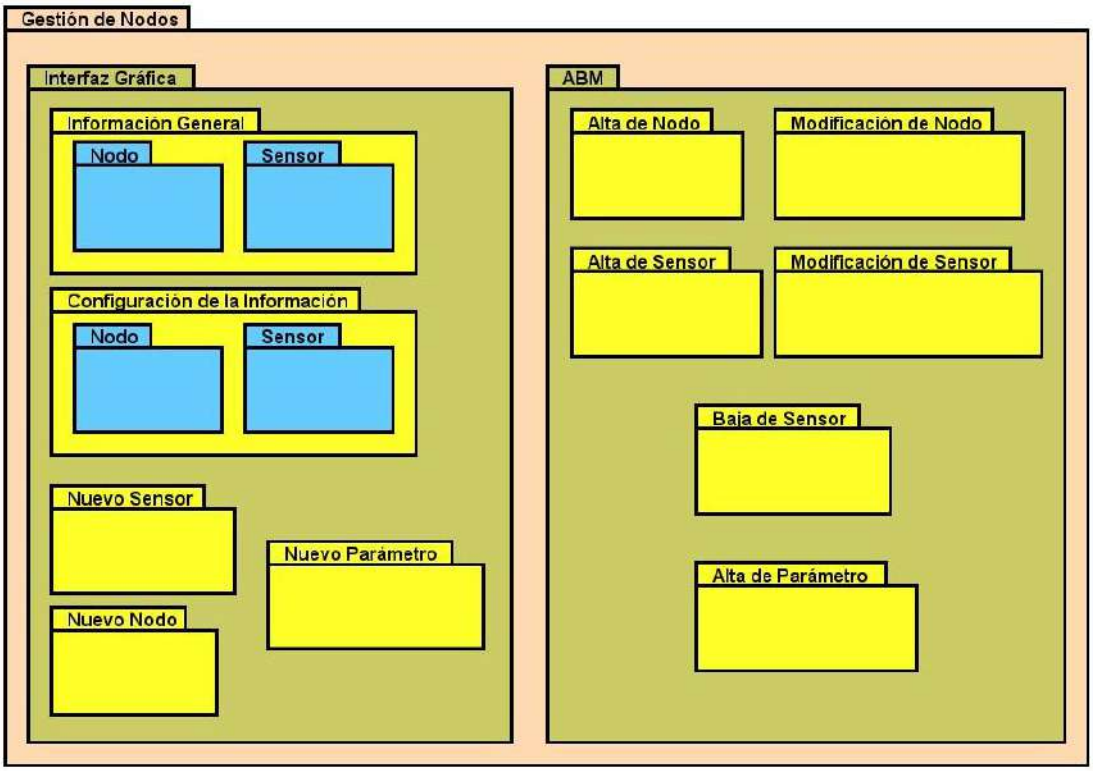


Figura 5: Diseño del módulo "Gestión de Nodos"

Por último, en la Figura 6, se puede ver el diseño del módulo de Gráficos. Para este caso particular, además de los sub-módulos de "Interfaz Gráfica" y "ABM", se diseñó un sub-módulo más para representar la funcionalidad de exportar datos de un determinado gráfico a un documento Excel.

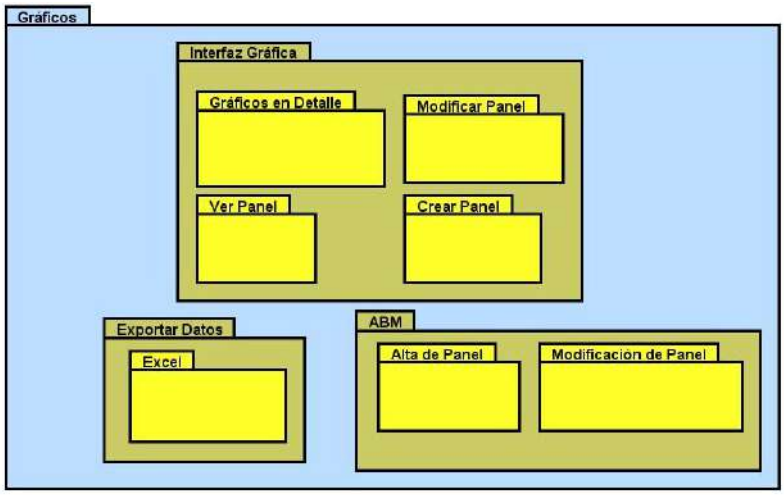


Figura 6: Diseño del módulo "Gráficos"

Para finalizar la etapa 3, se realizó la actividad 3.3. La instancia de modelado dio como resultado tres tipos de modelos: i) Diseño general del sistema (Figura 2); ii)

Diseño de datos o DER – Diagrama de Entidad-Relación- (ver Anexo II); y iii) Diseño de módulos individuales del sistema (Figuras 3, 4, 5 y 6). Una vez finalizados dichos modelos se revisó la consistencia entre los mismos con el alumno Denis Peppino. Llegamos a la conclusión que no habían problemas de inconsistencia.

Por último queda la etapa 4, donde la misma inicia con la actividad 4.1. Antes de empezar a desarrollar los distintos módulos del sistema se realizó una división del trabajo con el alumno Denis Peppino. Por mi parte, desarrollé el módulo completo de "Gestión de Alarmas". Además implemente casi la totalidad del módulo de "Gestión de Usuarios", ya que mi compañero realizó los sub-módulos que están relacionados con la carga de usuarios (paquetes "ABM→Alta de Usuarios" e "Interfaz Gráfica→Nuevo Usuario" en Figura 4). Vale la pena aclarar que luego contribuí en la mejora de la interfaz gráfica del "nuevo usuario" en término de requerimientos no funcionales (haciendo la interfaz más amigable con el usuario). A su vez desarrolle la totalidad del paquete "Gestión de Nodos" a excepción de los sub-módulos "Interfaz Gráfica→Información General→Nodo", "Interfaz Gráfica→Configuración de la Información→Nodo", "Interfaz Gráfica→Nuevo Nodo", "ABM→Alta de Nodo" y "ABM→Modificación de Nodo" (ver Figura 5). Una vez más contribuí en la mejora de la interfaz gráfica de los módulos mencionados en términos de requerimientos no funcionales. Por último vale la pena agregar que participé en la mayoría de las funcionalidades, colaborando con mi compañero en los momentos que surgían problemas, como también él hizo lo mismo conmigo. Lo detallado anteriormente, son los sub-módulos en los que me aboque principalmente, mientras que en todos los demás realice pequeñas colaboraciones. Una vez aclarado todo esto, empezare a especificar cada uno de los módulos que realice.

1. Módulo: "Gestión de Usuarios"

1.1 Sub-módulo: "Interfaz Gráfica"

1.1.1 Sub-módulo "Información General": consiste en desarrollar una interfaz amigable que permita visualizar los datos del usuario. Entre los datos a mostrar tenemos: nombre de usuario, nombre, apellido, correo electrónico y una foto de perfil.

1.1.2 Sub-módulo "Cambio de Contraseña": consiste en desarrollar una interfaz amigable que permita visualizar un formulario para el cambio de la contraseña del usuario.

1.1.3 Sub-módulo "Cambio de Correo Electrónico": consiste en desarrollar una interfaz amigable que permita visualizar un formulario para el cambio del correo electrónico del usuario.

1.1.4 Sub-módulo "Cambio de Foto de Perfil": consiste en desarrollar una interfaz amigable que permita visualizar un formulario para el cambio de la foto de perfil del usuario.

1.1.5 Sub-módulo "Nuevo Usuario": consiste en desarrollar una interfaz amigable que permita visualizar un formulario poder cargar nuevos usuarios al sistema. Esta tarea fue realizada por mi compañero pero yo realice mejoras en términos de requerimientos no funcionales.

1.1.6 Sub-módulo "Inicio de Sesión": consiste en desarrollar una interfaz amigable que permita visualizar un formulario para poder ingresar al sistema con los datos de un usuario.

1.1.7 Sub-módulo "Recuperación de Contraseña": consiste en desarrollar una interfaz amigable que permita visualizar un formulario para poder recuperar los datos de un usuario a través de su correo electrónico.

1.2 Sub-módulo: "ABM"

1.2.1 Sub-módulo: "Modificaciones de Usuarios"

1.2.1.1 Sub-módulo "Modificar Contraseña": consiste en implementar la lógica necesaria para poder modificar la contraseña de un usuario del sistema.

1.2.1.2 Sub-módulo "Modificar Correo Electrónico": consiste en implementar la lógica necesaria para poder modificar el correo electrónico de un usuario del sistema.

1.2.1.3 Sub-módulo "Modificar Foto de Perfil": consiste en implementar la lógica necesaria para poder modificar la foto de perfil de un usuario del sistema.

2. Módulo: "Gestión de Alarmas"

2.1 Sub-módulo: "Interfaz Gráfica"

2.1.1 Sub-módulo "Alarmas Activas": consiste en desarrollar una interfaz amigable que permita visualizar una lista de alarmas activas (aquellas que el usuario no confirmo haberla vista). Entre los datos a mostrar en la lista tenemos: nodo, parámetro, tipo de alarma, valor de inicio de alarma del parámetro, valor actual del parámetro, fecha de inicio, fecha de fin y un "*checkbox*" para poder dar el "visto".

2.1.2 Sub-módulo "Configuración de Alarmas": consiste en desarrollar una interfaz amigable que permita visualizar un formulario para la configuración de los valores de las alarmas por nodo, por parámetro y por tipo de alarma.

2.1.3 Sub-módulo "Alarmas Configuradas": consiste en desarrollar una interfaz amigable que permita visualizar la lista de alarmas configuradas. Los datos a mostrar son: nodo, parámetro, tipo de alarma, valor y dos opciones (entre ellas tenemos a editar alarma y borrar alarma).

2.1.4 Sub-módulo "Historial de Alarmas": consiste en desarrollar una interfaz amigable que permita visualizar la lista de alarmas ya vistas por el usuario. Los datos a mostrar son: nodo, parámetro, tipo de alarma, valor de inicio de la alarma, fecha de inicio y fecha de fin.

2.2 Sub-módulo: "ABM"

2.2.1 Sub-módulo "Alta de Configuración de Alarmas": consiste en implementar la lógica necesaria para poder cargar distintas configuraciones de alarmas por nodo, por parámetro y por tipo de alarma. Entre los tipos de alarmas a cargar tenemos: MaxMax, Max, Min y MinMin con sus respectivos valores.

2.2.2 Sub-módulo "Modificación de Configuración de Alarmas": consiste en implementar la lógica necesaria para poder modificar las distintas configuraciones de alarmas ya cargadas, por nodo, por parámetro y por tipo de alarma. Entre los tipos de alarmas a modificar tenemos: MaxMax, Max, Min y MinMin con sus respectivos valores.

2.2.3 Sub-módulo "Baja de Configuración de Alarmas": consiste en implementar la lógica necesaria para poder borrar las distintas configuraciones de alarmas ya cargadas, por nodo, por parámetro y por tipo de alarma. Entre los tipos de alarmas a borrar tenemos: MaxMax, Max, Min y MinMin.

2.2.4 Sub-módulo "Alta de Alarmas": consiste en desarrollar un código (o *script*) en Python que se ejecuta en el servidor periódicamente, el cual controla por parámetro, por nodo y por tipo de alarma cargada si hay algún valor detectado por algún sensor de algún nodo que corresponda con el disparo de una alarma. En caso de haber alarma, la misma se carga en la base de datos. Además de la carga se notifica a los usuarios correspondientes vía e-mail de la alarma.

3. Módulo: "Gestión de Nodos"

3.1 Sub-módulo: "Interfaz Gráfica"

3.1.1 Sub-módulo: "Información General"

3.1.1.1 Sub-módulo "Nodo": consiste en desarrollar una interfaz amigable que permita visualizar los datos de un nodo. Entre los datos a mostrar tenemos:

categoría, nombre, foto de nodo, una descripción, localidad, dirección, latitud, longitud y una dirección IP. Esta tarea fue realizada por mi compañero pero yo realice mejoras en términos de requerimientos no funcionales.

3.1.1.2 Sub-módulo "Sensor": consiste en desarrollar una interfaz amigable que permita visualizar los datos de un sensor. Entre los datos a mostrar tenemos: nombre, modelo, un teléfono de contacto, latitud, longitud, tiempo del último dato recibido, un registro y el parámetro a medir por el sensor.

3.1.2 Sub-módulo: "Configuración de la Información"

3.1.2.1 Sub-módulo "Nodo": consiste en desarrollar una interfaz amigable que permita modificar los datos de un nodo. Entre los datos a modificar tenemos: categoría, nombre, foto de nodo, una descripción, localidad, dirección, latitud, longitud y una dirección IP. Esta tarea fue realizada por mi compañero pero yo realice mejoras en términos de requerimientos no funcionales.

3.1.2.2 Sub-módulo "Sensor": consiste en desarrollar una interfaz amigable que permita modificar los datos de un sensor. Entre los datos a modificar tenemos: nombre, modelo, un teléfono de contacto, latitud, longitud, un registro y el parámetro a medir por el sensor.

3.1.3 Sub-módulo "Nuevo Sensor": consiste en desarrollar una interfaz amigable que permita cargar un nuevo sensor al nodo seleccionado. Entre los datos a introducir del sensor tenemos: nombre, modelo, un teléfono de contacto, latitud, longitud, un registro y el parámetro a medir por el sensor.

3.1.4 Sub-módulo "Nuevo Nodo": consiste en desarrollar una interfaz amigable que permita cargar un nuevo nodo al sistema. Entre los datos a introducir del nodo tenemos: categoría, nombre, una descripción, la organización a la cual pertenecerá, localidad, dirección, latitud, longitud y una dirección IP. Esta tarea fue realizada por mi compañero pero yo realice mejoras en términos de requerimientos no funcionales.

3.1.5 Sub-módulo "Nuevo Parámetro": consiste en desarrollar una interfaz amigable que permita cargar un nuevo parámetro del nodo seleccionado. El único dato a cargar del parámetro es el nombre.

3.2 Sub-módulo: "ABM"

3.2.1 Sub-módulo "Alta de Sensor": consiste en implementar la lógica necesaria para poder cargar un sensor del nodo seleccionado. Entre los datos a introducir del

sensor tenemos: nombre, modelo, un teléfono de contacto, latitud, longitud, un registro y el parámetro a medir por el sensor.

3.2.2 Sub-módulo "Modificación de Sensor": consiste en implementar la lógica necesaria para poder modificar los datos de un sensor del nodo seleccionado. Entre los datos a modificar del sensor tenemos: nombre, modelo, un teléfono de contacto, latitud, longitud, un registro y el parámetro a medir por el sensor.

3.2.3 Sub-módulo "Baja de Sensor": consiste en implementar la lógica necesaria para poder borrar un sensor del nodo seleccionado. Hay que tener en cuenta que si el sensor contiene datos cargados en el sistema entonces no puede ser borrado.

3.2.4 Sub-módulo "Alta de Parámetro": consiste en implementar la lógica necesaria para poder cargar un parámetro del nodo seleccionado. El único dato a cargar del parámetro es el nombre.

4. Otras funcionalidades realizadas:

4.1. Además de la interfaz de inicio de sesión implemente toda la lógica correspondiente que hacen funcionar una sesión de usuario. También realice la funcionalidad de "Mantener Sesión Iniciada" que consiste en que una vez se elimine la sesión del usuario, se pueda recuperar dicha sesión sin tener que iniciar sesión nuevamente. Esta última funcionalidad se realizó utilizando *cookies*.

4.2. Además de la interfaz de recuperación de contraseña implemente toda la lógica correspondiente para dicha funcionalidad. La misma consiste en enviar a través de un correo electrónico los datos de un usuario, si el mismo existe y el correo es válido.

4.3. Desarrollé, para el historial de alarmas, una búsqueda amigable por fecha de inicio y fecha de fin de alarmas ya vistas por el usuario.

Nota 1: La prueba de los distintos módulos se fue realizando a medida que se iban desarrollando con la mayor cantidad de casos de pruebas posibles. El desarrollo de cada módulo consistió entonces en su implementación, prueba y modificación (en caso de encontrar algún error) hasta que funcionara con todos los casos de pruebas posibles. Una vez se tenía el módulo desarrollado y sin posibles errores, el mismo era validado por el cliente. En caso afirmativo se daba por cerrado la implementación del módulo individualmente. Por otra parte, si el cliente no validaba dicho módulo se planteaban las cuestiones a cambiar y se procedía a modificarlo como mencionamos anteriormente (implementación, prueba, modificación y validación).

Nota 2: En el Anexo III se pueden ver capturas de pantallas de la última versión del sistema.

Para finalizar esta sección nos queda la actividad 4.2. A medida que se iban terminando el desarrollo de cada módulo individualmente se iban integrando con los demás módulos de los cuales se relacionaban. Por ejemplo los módulos que implicaban el alta de un sensor y su modificación eran módulos que se relacionaban. Una vez finalizados estos se integraban y se realizan pruebas en ambos para verificar que todo funcione correctamente. En caso de encontrar algún error se realizaban las modificaciones necesarias para poder solucionarlos.

3. Conclusiones y Trabajos Futuros

Como conclusión de la PPS realizada en Arbit Ingeniería, pude observar que es de suma importancia el trabajo en equipo, que exista una buena comunicación con las partes interesadas (por ejemplo, el cliente y los compañeros de trabajo), ya que el intercambio de ideas y puntos de vista ayudan a que el producto final sea de buena calidad. Además, la constante comunicaciones ayuda a detectar problemas de requerimientos en etapas tempranas de desarrollo de *software* y así disminuir los costos del proyecto. También resulta importante tener en cuenta que el uso de ciertas tecnologías simplifica el trabajo, permitiendo acelerar el proceso de desarrollo, ya que puede brindar soluciones a muchos problemas. En este caso, el uso de la librería JMorris, por ejemplo, evitó que se tuviera que desarrollar un módulo extra que sea el encargado de realizar los gráficos, o como en el caso de la librería JQuery, que amplía y simplifica las funcionalidades de JavaScript. Además fue evidente como el hecho de tener un diseño claro y ser organizados para trabajar, facilitó la integración de los módulos, como también que ambos practicantes en distintos momentos trabajemos en la misma funcionalidad, sin generar grandes conflictos. Por otra parte, resultó enriquecedora la experiencia del trabajo en una empresa real con un superior.

En resumen, la experiencia adquirida tras realizar la PPS en Arbit Ingeniería resultó ser muy grata. Me agrado el ámbito del problema a resolver, como también la fluidez de la comunicación con el tutor institucional, facilitando la comprensión del proyecto en sí. También me entusiasmó ver que el proyecto iba cumpliendo con sus expectativas, y que en un futuro será utilizado en diferentes organizaciones.

Por otra parte, con respecto a los trabajos futuros, el proyecto realizado se puede seguir expandiendo en términos de funcionalidades. Por ejemplo se podría agregar la función de dar de baja un nodo, dar de baja un usuario con el usuario administrador, dar de alta una organización, entre otras. A su vez se puede seguir

mejorando el sistema haciéndolo más amigable con el usuario, más interactivo, hacer mejoras al código para que sea más mantenible, entre otras mejoras.

Bibliografía y referencias

1. Cárdenas Valencia Á., Echeverry Giraldo A.: Prototipo de un Sistema de Telemetría y Control para Seguridad en Vehículos, Soportado en Redes Móviles, 2010.
2. Castellano N., Gázquez Parra J., López Rodríguez J., Manzano-Agugliaro F.: Sistema de Telemetría para la Transmisión de Datos desde Ambulancia, 2012.
3. http://getbootstrap.com/. Enlace Visitado el 03/08/2017
4. https://es.wikipedia.org/wiki/Automatización_industrial. Enlace Visitado el 28/07/2017
5. https://es.wikipedia.org/wiki/HTML5. Enlace Visitado el 03/08/2017
6. https://es.wikipedia.org/wiki/JavaScript. Enlace Visitado el 03/08/2017
7. https://es.wikipedia.org/wiki/JQuery. Enlace Visitado el 03/08/2017
8. https://es.wikipedia.org/wiki/MySQL. Enlace Visitado el 03/08/2017
9. https://es.wikipedia.org/wiki/PHP. Enlace Visitado el 03/08/2017
10. https://es.wikipedia.org/wiki/Programación_extrema. Enlace Visitado el 03/08/2017
11. https://es.wikipedia.org/wiki/Python. Enlace Visitado el 03/08/2017
12. https://es.wikipedia.org/wiki/SCADA. Enlace Visitado el 03/08/2017
13. https://es.wikipedia.org/wiki/Sistema_de_control. Enlace Visitado el 28/07/2017
14. Pérez Tovar F., Huerta Aguilar A., Quezada J.: Sistema de Telemetría y Radio Control para PLC, 2009.
15. Solarte Varney P., Agredo Mendez G.: Implementación de un Sistema de Telemetría para la Detección Temprana de Eventos Tsunamigénicos asociados con la variación extrema del nivel de la marea en la costa Pacífica Colombiana, 2011.

ANEXOS

ANEXO I: Requerimientos del proyecto

Sistema web de monitoreo

Se desea diseñar e implementar un sistema para el monitoreo de las diferentes variables de un nodo (ya sea un invernadero, una planta industrial, etc.). El sistema debe permitir agregar diferentes parámetros (estado de una planta industrial, monitoreo de caudal, potencia consumida, temperatura de equipos, etc.). Estas variables son medidas por sensores dentro del nodo, y cada sensor puede relevar una única variable. La cantidad de sensores en el nodo es variable y no tienen una distribución específica.

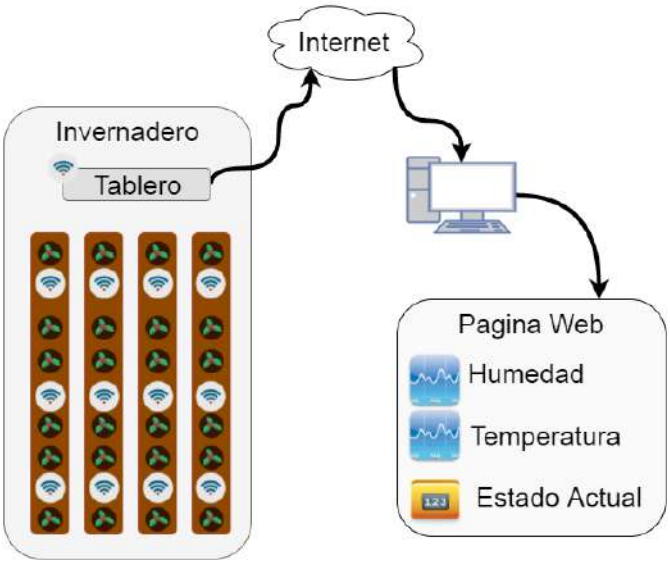


Figura 7: Ejemplo de nodo (Invernadero)

El sistema web es accedido por diferentes usuarios, los cuales poseen acceso a determinados nodos. Un usuario puede tener uno o más nodos y un nodo puede ser accedido por más de un usuario.

En la aplicación web se deben mostrar los diferentes parámetros en un periodo de tiempo configurable. Además de una función que pueda comparar diferentes periodos de tiempos y diferentes parámetros.

El sistema debe tener un sistema de alarmas y registro de eventos programables (log), en los casos donde los parámetros superen ciertos valores configurables se debe guardar la fecha y hora en que ocurrió y el valor. En caso de alarmas, debe contar con una opción

donde el usuario marque que la vio (Acknowledge, ACK). Para este módulo se deberá desarrollar un script en Python que se ejecute periódicamente en el servidor, el cual se va a encargar a su vez de notificar a los usuarios vía e-mail.

El sistema debe gestionar información del sensor: tener registro del modelo, nombre, contacto, localización, tiempo del último dato recibido y número de sensor. Cada sensor debe tener un ID para en un futuro poder emplear consultas Modbus TCP/IP y SNMP. Para realizar una consulta Modbus TCP/IP se necesita la IP del equipo y el registro de la variable a consultar. Para las consultas SNMP se debe tener IP y el identificador del objeto, oid. Además, guardar la posición geo-referencial para mostrarla en un Mapa de Google todos los nodos. (La API de Google permite fácilmente hacer click en una parte del mapa y obtener la latitud y longitud).

La actualización de los datos en la base de datos varía entre 1 minuto a 1 hora. La página web debe actualizar el estado de los gráficos en un tiempo razonable.

Debe tener una opción de exportar datos de un determinado parámetro y periodo de tiempo en planillas Excel.

Para datos antiguos, promediar varios valores y almacenar el dato promediado en otra tabla de la base de datos. Ejemplo: Los datos del mes pasado se deben agrupar cada media hora y sacar un promedio, guardando cual fue el valor máximo y mínimo de ese grupo. Luego borrar la tabla del mes pasado. Debe ser configurable la condición en que los datos se consideran antiguos, como así también el periodo en que se agrupan para sacar el promedio.

Requerimientos no funcionales y otros:

- Ser responsivo.
- Optimizar las consultas a la base de datos. Diseñar la base de datos para evitar trabajar con tablas muy grandes.
- Expandible y versátil para usarlo en otra aplicación (Sistemas industriales para medición de caudal, temperatura de calderas, nivel de agua en tanques y cisternas).
- Capacidad para integrar protocolos Modbus TCP/IP y SNMP (para estos protocolos se debe tener en cuenta que en la base de datos debe haber un campo con el registro de la variable en el caso de Modbus, y para SNMP un identificador de objeto, OID).
- Correr en Linux.
- Ser amigable con el usuario.

Para desarrollar este sistema se recomienda analizar:

- RRDTools: Base de datos Round Robin, se caracteriza en que cantidad de datos a trabajar es fija y tiene la opción de generar una imagen con un gráfico con los datos almacenados. Muy utilizado en sistemas SCADA web comerciales, también en los sistemas Observium y Cacti.
- Observium: Sistema SCADA para administrar diferentes nodos de una red. Prestar atención a la organización de los gráficos de las variables y sus periodos de tiempo.
- Cacti: Similar a Observium, potente pero menos estético.
- Bootstrap SB Admin 2: Framework para el diseño de páginas web con un template con diversos gráficos.
- Rapid Scada: Sistema SCADA web con drivers para encuestar a equipos mediante diferentes protocolos.

La visualización que se requiere es similar al sistema de software libre Observium. Ejemplo de gráficos:

Figura 8: Gráficos Observim. Notar que debajo de cada grafico se muestran los valores máximos, mínimos y el promedio.

Ejemplo de información del sensor:

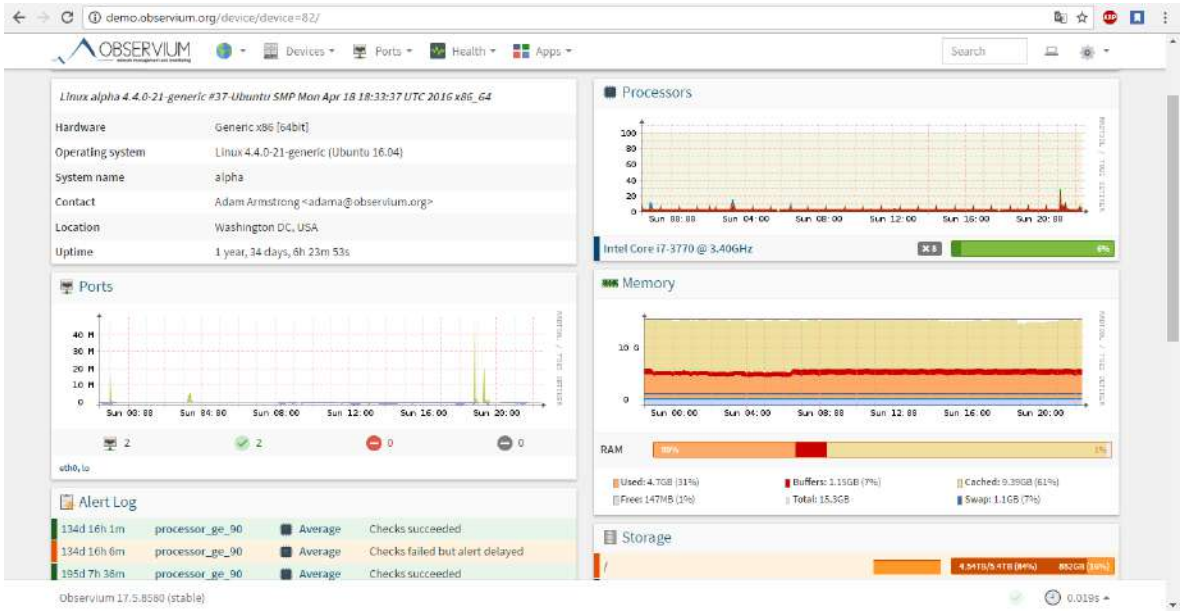


Figura 9: Información de un sensor, es decir, hardware, nombre, contacto, localización y tiempo del último dato recibido.

Vista del estado actual y visualización de alarmas:

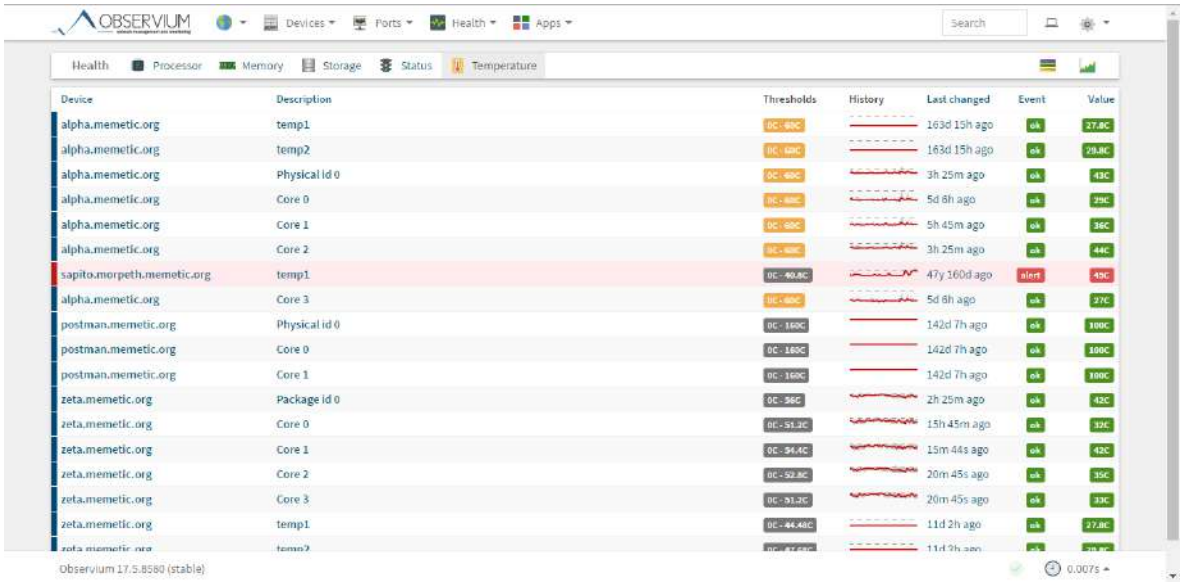


Figura 10: Sistema de Alarmas de Observium.

ANEXO II: DER final del sistema

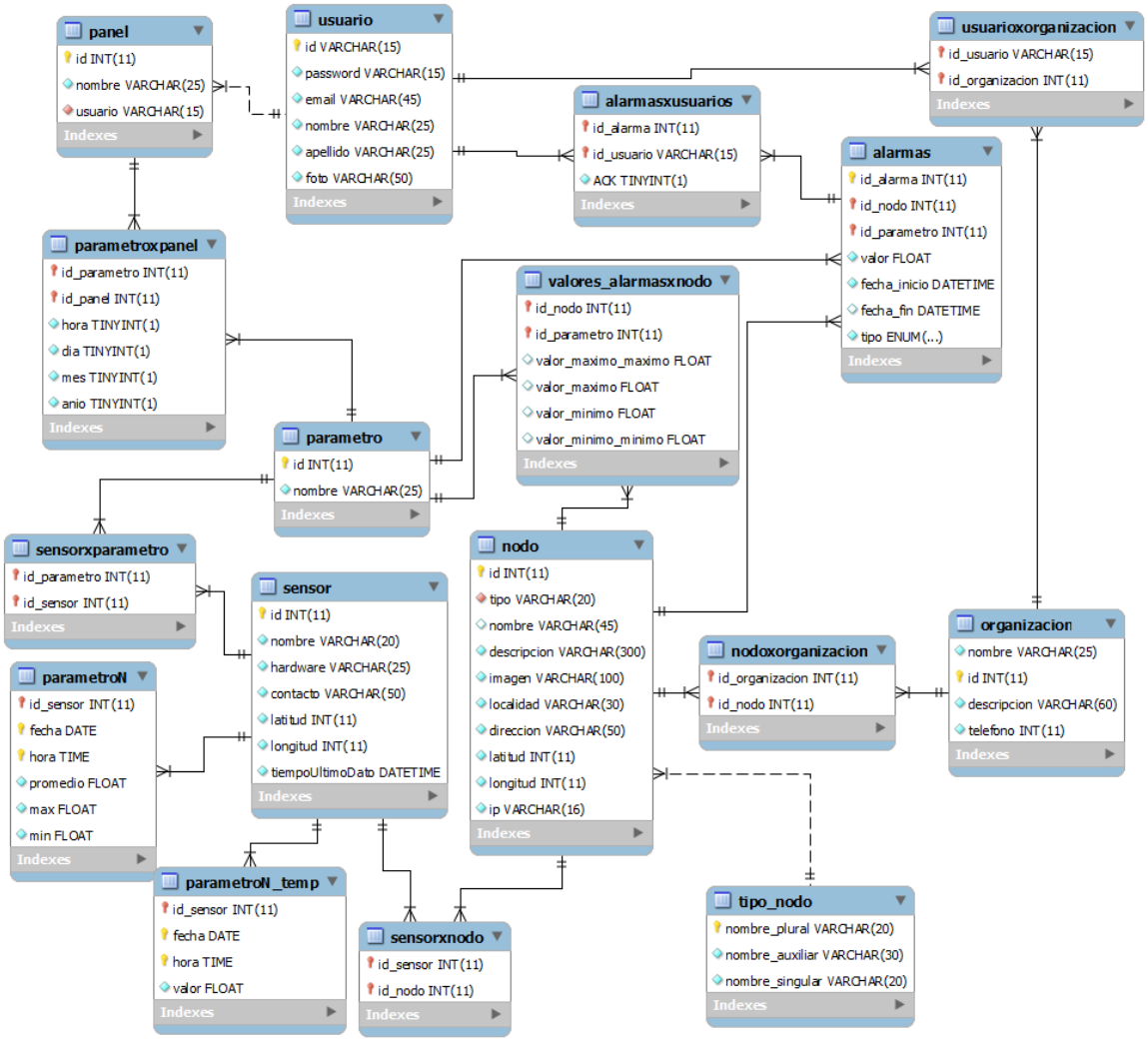


Figura 11: DER utilizado en el proyecto

ANEXO III: Capturas de pantalla de la última versión del sistema

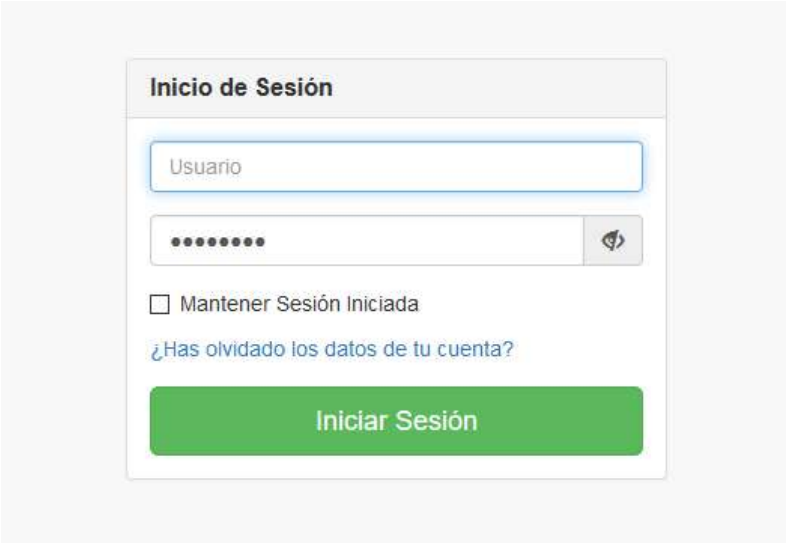


Figura 12: captura de la pantalla "inicio de sesión"

Figura 13: captura de la pantalla "home"

Figura 14: captura de la pantalla "ver gráficos en detalle" del nodo "I1", promedio diario

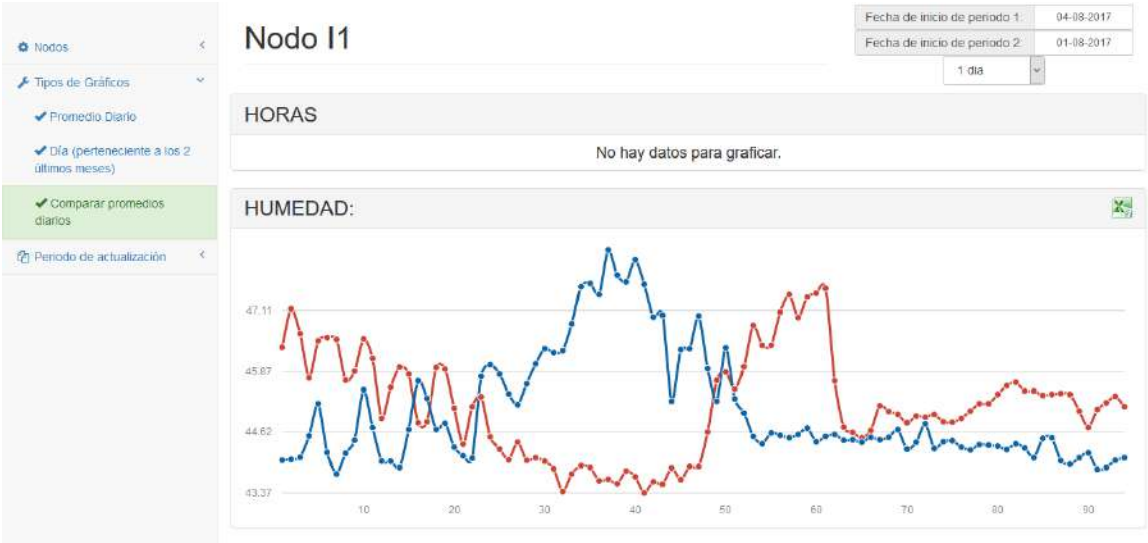


Figura 15: captura de la pantalla "ver gráficos en detalle" del nodo "I1", comparación de dos periodos

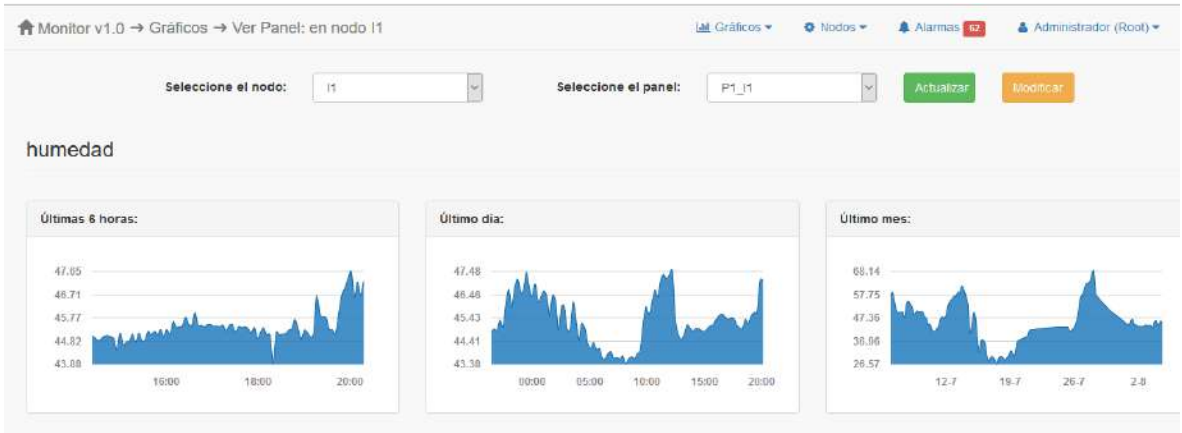


Figura 16: captura de la pantalla "ver paneles" para el nodo "I1"

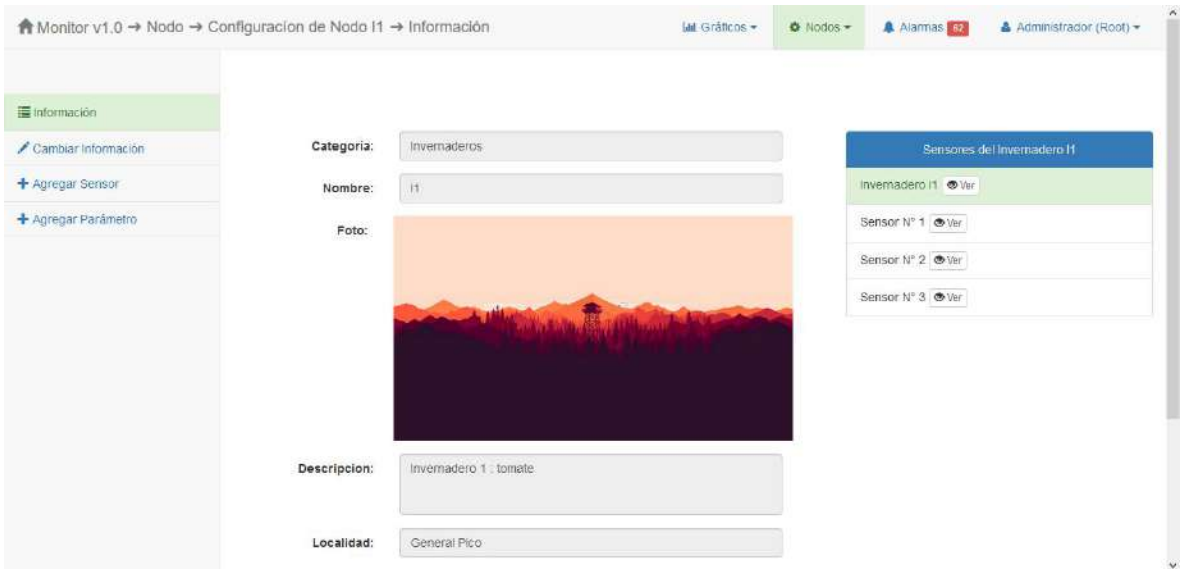
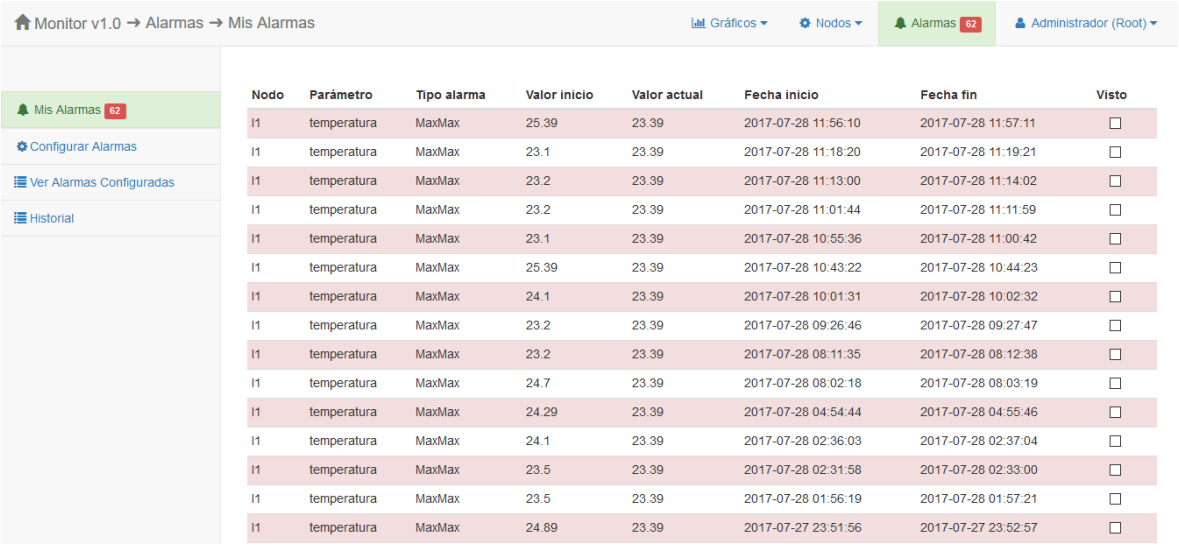


Figura 17: captura de la pantalla "ver nodo" para el nodo "I1"



Nodo	Parámetro	Tipo alarma	Valor inicio	Valor actual	Fecha inicio	Fecha fin	Visto
I1	temperatura	MaxMax	25.39	23.39	2017-07-28 11:56:10	2017-07-28 11:57:11	☐
I1	temperatura	MaxMax	23.1	23.39	2017-07-28 11:18:20	2017-07-28 11:19:21	☐
I1	temperatura	MaxMax	23.2	23.39	2017-07-28 11:13:00	2017-07-28 11:14:02	☐
I1	temperatura	MaxMax	23.2	23.39	2017-07-28 11:01:44	2017-07-28 11:11:59	☐
I1	temperatura	MaxMax	23.1	23.39	2017-07-28 10:55:36	2017-07-28 11:00:42	☐
I1	temperatura	MaxMax	25.39	23.39	2017-07-28 10:43:22	2017-07-28 10:44:23	☐
I1	temperatura	MaxMax	24.1	23.39	2017-07-28 10:01:31	2017-07-28 10:02:32	☐
I1	temperatura	MaxMax	23.2	23.39	2017-07-28 09:26:46	2017-07-28 09:27:47	☐
I1	temperatura	MaxMax	23.2	23.39	2017-07-28 08:11:35	2017-07-28 08:12:38	☐
I1	temperatura	MaxMax	24.7	23.39	2017-07-28 08:02:18	2017-07-28 08:03:19	☐
I1	temperatura	MaxMax	24.29	23.39	2017-07-28 04:54:44	2017-07-28 04:55:46	☐
I1	temperatura	MaxMax	24.1	23.39	2017-07-28 02:36:03	2017-07-28 02:37:04	☐
I1	temperatura	MaxMax	23.5	23.39	2017-07-28 02:31:58	2017-07-28 02:33:00	☐
I1	temperatura	MaxMax	23.5	23.39	2017-07-28 01:56:19	2017-07-28 01:57:21	☐
I1	temperatura	MaxMax	24.89	23.39	2017-07-27 23:51:56	2017-07-27 23:52:57	☐

Figura 18: captura de la pantalla "alarmas"

Figura 19: captura de la pantalla "Configuración de la cuenta de usuario"

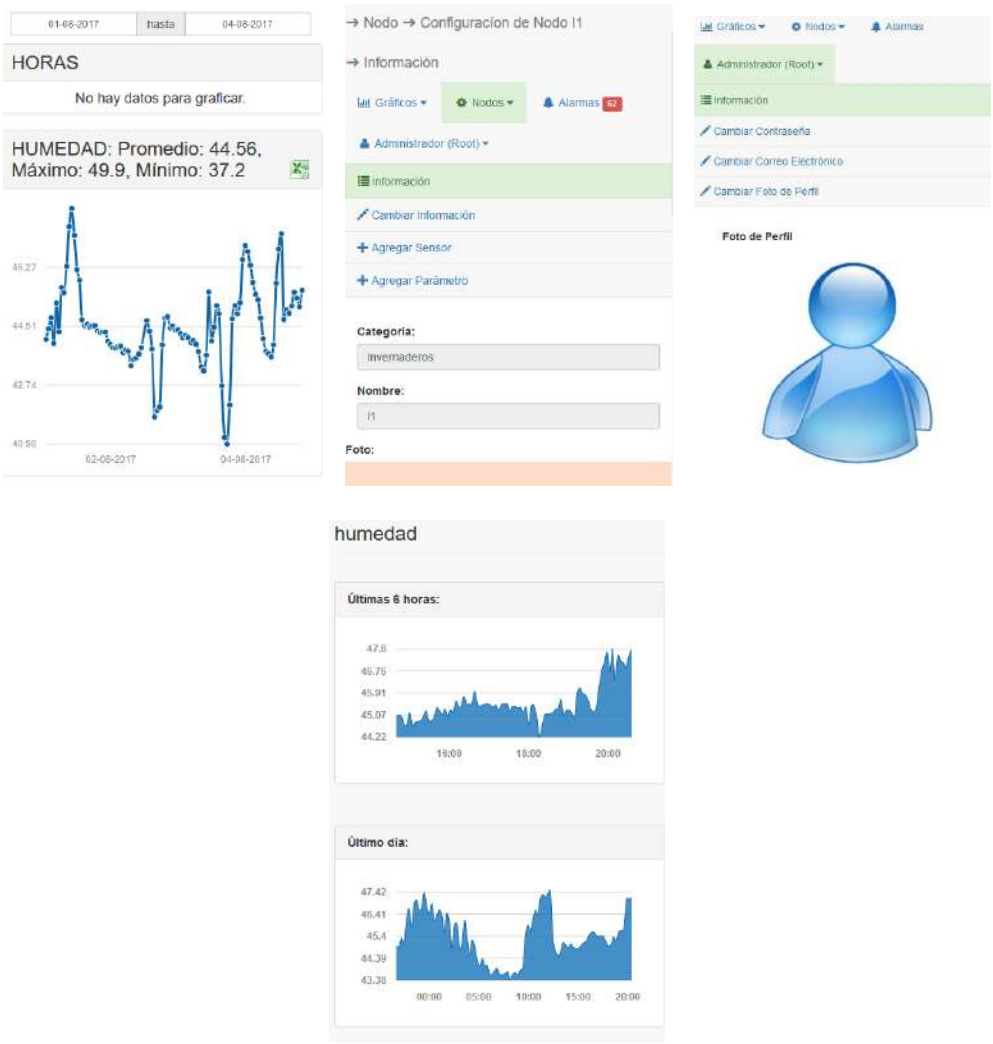


Figura 20: captura de distintas pantallas vistas desde un Smartphone