



Universidad
de La Laguna

Sistema doméstico de monitorización del consumo eléctrico

Domestic Energy Monitoring System

Yeray Betancor Caro

Departamento de Ingeniería Informática y de Sistemas

Escuela Técnica Superior de Ingeniería Informática

Trabajo de Fin de Grado

La Laguna, 04 de Septiembre de 2014

Profesor Contratado Doctor **Jesús Miguel Torres Jorge**,
con N.I.F. 43826207-Y adscrito al Departamento de
Ingeniería Informática y de Sistemas de la Universidad
de La Laguna

C E R T I F I C A

Que la presente memoria titulada:

*"Sistema doméstico de monitorización del consumo
eléctrico."*

ha sido realizada bajo su dirección por D. Yeray
Betancor Caro, con N.I.F. 78.853.894-G.

Y para que así conste, en cumplimiento de la
legislación vigente y a los efectos oportunos firman la
presente en La Laguna a 4 de septiembre de 2014

Agradecimientos

En primer lugar, quiero mostrar mi profundo agradecimiento a mis padres y hermana, por todo el ánimo y apoyo incondicional, la finalización de esta etapa no hubiera sido posible sin ellos.

También quiero agradecer a mi novia María, que siempre ha estado a mi lado animándome a seguir y ayudándome en todas esas tardes y fines de semana de estudios.

Agradecer también a todos los compañeros y amigos de la facultad, los cuales han acompañado en este largo camino compartiendo conocimientos, apoyo, ánimos, estudio y diversión.

Y por supuesto a Jesús Miguel Torres Jorge, director de este trabajo de fin de grado, el cuál ha sido siempre un profesor ejemplar y cercano a sus alumnos, que no ha escatimado en brindarme toda su ayuda y dedicación a cualquier hora del día, por cualquier medio posible.

Resumen

Crear un mundo sostenible es uno de los grandes retos de nuestro tiempo. Para conseguirlo es importante, no sólo investigar en nuevas fuentes de energía o en el máximo aprovechamiento de las existentes, sino también en cómo hacer un consumo más responsable y eficiente.

Para lograrlo es importante tomar medidas sobre dicho consumo en condiciones lo más reales posibles, lo que no siempre resulta sencillo a grandes escalas. En ese sentido la proliferación en el mercado de pequeños SBC (Single Board Computer) y de todo tipo de accesorios para aficionados, hacen más barato y sencillo que nunca abordar el desarrollo de sistemas conectados para la monitorización del consumo eléctrico; siendo este el objetivo del presente proyecto.

Palabras clave

Consumo, energía, sostenibilidad, eficiencia, control, monitorización, Ordenador de placa reducida, microcontrolador.

Abstract

One of the great challenges of our time is to create a sustainable world. To achieve this it is important, not only to investigate in new sources of energy or the maximum use of the existing, but also on how to make a more responsible and efficient consumption.

To achieve it is important to take measures on consumption in conditions the real potential, which is not always simple to large scales. In that sense the proliferation in the market of small SBC and all kinds of accessories for amateurs, they do more cheap and simple never addressed the development of systems for monitoring of electricity consumption; being this the aim of the present project.

Keywords

Consumption, energy, sustainable, efficiency, control, monitoring, Single Board Computer, microcontroller.

Índice General

Capítulo 1. Introducción	7
1.1 Descripción del contexto	7
1.1.1 La eficiencia energética	7
1.1.2 Los microcontroladores	8
1.2 Objetivos	10
1.3 Contenido de la memoria	10
Capítulo 2. Estado del arte	12
2.1 Tecnología actual	12
2.2 Sistemas de monitorización de energía	14
2.2.1 Efergy, Engage Hub Kit	14
2.2.2 Wattio, Pack Wattio Energy	16
2.2.3 OpenEnergyMonitor	17
2.3 Tecnología aplicada	18
2.3.1 Tecnología de Hardware	18
2.3.2 Tecnología de Software	19
Capítulo 3. Diseño	21
3.1 Requisitos del proyecto	21
3.1.1 Requisitos funcionales	21
3.1.2 Requisitos no funcionales	22
3.2 Diseño del sistema	23
3.3 Comunicaciones del sistema	24
3.3.1 Comunicación física	25
3.3.2 Comunicación lógica	25
3.4 Diseño electrónico	26
3.4.1 Arduino	26
3.4.2 Transformador de corriente	27
3.4.3 Sensor de corriente	28
3.4.4 Módulo wifi	29
3.4.5 Flujo de comportamiento	30
3.5 Diseño de plataforma web	31
3.5.1 Flujo de acceso a la aplicación	31
3.5.2 Acceso y registro	32
3.5.3 Panel de control	33
3.5.4 Dispositivos	34

3.5.5	Histórico	35
3.5.6	Configuración	36
3.6	Modelo de datos	37
Capítulo 4. Implementación		39
4.1	Hardware del prototipo	39
4.1.1	Corriente generada por el sensor	39
4.1.2	Regulador de voltaje 5V a 3.3V	42
4.1.3	Pulsador de configuración de Wifly	43
4.1.4	Montaje del prototipo	44
4.1.5	Firmware del prototipo	44
4.1.6	Programación del microcontrolador	50
4.1.7	Pruebas del prototipo	50
4.2	Plataforma web	52
4.2.1	Menú	53
4.2.2	Pantalla de acceso y registro	54
4.2.3	Pantalla de panel de control	55
4.2.4	Pantalla de dispositivos	57
4.2.5	Pantalla de histórico	58
4.2.6	Pantalla de configuración	59
Capítulo 5. Conclusiones y Trabajos Futuros		60
5.1	Conclusiones	60
5.2	Trabajos futuros	61
Capítulo 6. Conclusions and future work		63
6.1	Conclusions	63
6.2	Future work	64
Capítulo 7. Presupuesto		65
7.1	Presupuesto del prototipo	65
7.2	Presupuesto de la plataforma	65
Apéndice A: Microcontrolador		67
A.1.	Código del microcontrolador	67
Apéndice B: Servicio web		67
B.1.	Código webservice.php	67

Apéndice C: Base de Datos	68
C.1. Código SQL: Tabla usuario	68
C.2. Código SQL: Tabla dispositivos	69
C.3. Código SQL: Tabla registros	69
C.4. Código SQL: Tabla zona_horaria	69
Apéndice D: Plataforma web	70
D.1.Codigo Javascript	70
Capítulo 8. Bibliografía	72

Índice de figuras

Figura 2.1. Plataforma de hardware libre Arduino Uno.	12
Figura 2.2. Raspberry Pi, PC de placa reducida.	13
Figura 2.3. Engage Hub Kit (Efergy).	15
Figura 2.4. Esquema de dispositivos Engage Hub Kit	15
Figura 2.5. Pack Wattio Energy (Wattio).	16
Figura 2.6. OpenEnergyMonitor.	17
Figura 3.1. Diseño del sistema.	23
Figura 3.2. Esquema de comunicaciones.	24
Figura 3.3. Comunicación física del sistema.	25
Figura 3.4. Comunicación lógica de los datos.	25
Figura 3.5. Arduino Mini Pro 3.3V	26
Figura 3.6. Transformador 220-5v, 2A.	27
Figura 3.7. Sensor SCT-013-000	28
Figura 3.8. Módulo WiFly RN-XV.	29
Figura 3.9. Diagrama de flujo algoritmo.	30
Figura 3.10. Diseño pantalla acceso y registro.	32
Figura 3.11. Diseño pantalla panel de control.	33
Figura 3.12. Diseño pantalla dispositivos.	34
Figura 3.13. Diseño pantalla histórico.	35
Figura 3.14. Diseño pantalla configuración	36
Figura 3.15. Diseño diagrama de base de datos.	38
Figura 4.1. Esquema resistencia de carga.	41
Figura 4.2. Esquema circuito divisor de voltaje.	42
Figura 4.3. Regulador de voltaje.	42
Figura 4.4. Imagen pulsador de configuración.	43
Figura 4.5. Imagen del prototipo.	44
Figura 4.6. Programación del microcontrolador.	50
Figura 4.7. Prueba de testeo en puerto serie.	51
Figura 4.8. Prueba de consulta a la BBDD.	51
Figura 4.9. Estructura de ficheros.	52
Figura 4.10. Pantalla menú.	53

Figura 4.11. Pantalla acceso y registro.	54
Figura 4.12. Pantalla panel de control.	55
Figura 4.13. Pantalla de panel de control Móvil.	56
Figura 4.14. Pantalla de dispositivos.	57
Figura 4.15. Pantalla de histórico.	58
Figura 4.16. Pantalla de configuración.	59

Índice de tablas

Tabla 3.1. Características Arduino Mini Pro 3.3V.	27
Tabla 3.2. Características transformador de corriente.	27
Tabla 3.3. Características sensor de corriente.	28
Tabla 3.4. Características módulo wifi.	29

Capítulo 1. Introducción

1.1 Descripción del contexto

El presente proyecto surge de la necesidad de hacer un uso eficiente de las energías, dado que las fuentes de energía no renovables tienen una duración finita, es necesario que en el presente impere la necesidad de utilizarlas correctamente, para que podamos seguir disfrutando de ellas en el futuro. En ello se basa el concepto de eficiencia energética.

1.1.1 La eficiencia energética

La sociedad y las organizaciones empresariales, consumidores directos de la energía, pueden reducir el consumo energético promoviendo la sostenibilidad económica, política y ambiental, aumentando la eficacia y maximizando, así, su beneficio. El consumo de la energía está directamente relacionado con la situación económica, por lo que es necesaria una aproximación global que permita el diseño de políticas de eficiencia energética.

El continuo desarrollo de las tecnologías es parte integral en cualquier modelo de negocio, tanto para la producción como para el consumo, ya que se trata de uno de los principales factores que afectan a la competitividad del producto final en el mercado global. Cada nueva generación de productos es, por defecto, más eficiente energéticamente que la generación anterior, debido a que la eficiencia energética es un factor de coste importante durante la vida útil del producto.

Las tecnologías de eficiencia energética se pueden encontrar en todos los procesos de transformación de la energía: desde la exploración y producción de recursos energéticos primarios, hasta la generación de energía y refinamiento del petróleo, las redes de electricidad y la utilización final en la industria, los edificios y el transporte. Pero, no es sólo el potencial técnico, el cual es crucial para la exitosa introducción de las tecnologías de energía eficiente. Para evaluar el potencial completo de estas tecnologías e identificar el camino hacia su introducción con éxito en el mercado, es necesario tener en cuenta su potencial económico, realizable y realista.

Ser más eficiente no significa renunciar a nuestro grado de bienestar y calidad de vida. Simplemente se trata de adoptar una serie de hábitos responsables, medidas e inversiones a nivel tecnológico y de gestión. Practicar un consumo más responsable e inteligente de la energía que consumimos no es tarea de una sola persona, es tarea de todos.

1.1.2 Los microcontroladores

Un microcontrolador es un circuito integrado programable, capaz de ejecutar las órdenes grabadas en su memoria. Está compuesto de varios bloques funcionales, los cuales cumplen una tarea específica.

El inicio de los microcontroladores tienen sus raíces en el desarrollo de la tecnología de los circuitos integrados. Este desarrollo ha hecho posible contener cientos de miles de transistores en un solo chip. Este era un requisito previo para la producción de microprocesadores, las primeras computadoras se construían agregando elementos externos como memoria,

contadores, dispositivos de entrada / salida, etc. El incremento posterior de la densidad de integración permitió crear un circuito integrado que contenía tanto al procesador como periféricos. Así es cómo fue desarrollada la primera microcomputadora en un solo chip, denominada más tarde microcontrolador.

El Instituto Smithsonian afirma que los ingenieros de Texas Instruments Gary Boone y Michael Cochran, lograron crear el primer microcontrolador, TMS 1000, en 1971 el cual fue comercializado en 1974. Combina memoria ROM, memoria RAM, microprocesador y reloj en un chip y estaba destinada a los sistemas embebidos.

La mayoría de los microcontroladores en ese momento tenían dos variantes. Unos tenían una memoria EPROM reprogramable, significativamente más caros que la variante PROM que sólo era programable una vez. Para borrar la EPROM se necesita exponer a la luz ultravioleta la tapa de cuarzo transparente. Los chips opacos representaban un coste menor.

En 1993 tuvo lugar el lanzamiento de la memoria EEPROM en los microcontroladores (comenzando con el Microchip PIC16x84). Estas pueden ser borradas eléctrica y rápidamente sin necesidad de asumir un coste elevado como se requiere en el proceso de borrado de las memorias EPROM, lo que permite la creación rápida de prototipos y una sencilla programación de los microcontroladores. El mismo año, Atmel lanzó el primer microcontrolador que utiliza memoria flash. Otras compañías rápidamente siguieron el ejemplo, con los dos tipos de memoria.

En la actualidad los microcontroladores son baratos y fácilmente disponibles para los aficionados,

con grandes comunidades en línea para ciertos procesadores.

1.2 Objetivos

Los objetivos generales del presente proyecto se basan principalmente en el diseño e implementación de un prototipo funcional, sencillo, de pequeño tamaño y bajo coste que permita monitorizar el consumo energético de las diferentes secciones de una instalación eléctrica y/o de diferentes dispositivos conectados a la red eléctrica en el ámbito del hogar.

Dicho sistema recoge la información de consumo proveniente de uno o varios sensores conectados al dispositivo. Esta información se procesa y se envía mediante un punto de acceso wifi a una base de datos alojada en la nube. Para la configuración del dispositivo y consulta de los datos almacenados se utiliza una plataforma web que se ha desarrollado de manera que se adapte a cualquier dispositivo, ordenador, tablet o móvil. Esta plataforma contiene toda la información estadística e histórica de los datos almacenados desde su puesta en funcionamiento.

1.3 Contenido de la memoria

En el capítulo **Introducción**, se ha descrito el contexto del proyecto y se han introducido los dos conceptos importantes sobre los que se basa el proyecto, concluyendo con los objetivos del mismo.

En el capítulo **Estado del arte**, se define la tecnología actual, los productos basados en la misma idea y la tecnología que se ha usado en el proyecto.

En el capítulo **Diseño**, se presentan los requisitos del proyecto, diseños de hardware y software, herramientas y solución adoptada.

En el capítulo **Implementación**, se especifica el trabajo realizado en detalle.

En el capítulo **Conclusiones y trabajos futuros**, se describe la conclusión del proyecto y se presentan futuras líneas de mejora.

Capítulo 2. Estado del arte

En este capítulo se analizará el estado de la tecnología actual, los diferentes productos comerciales basados en la idea de este proyecto y la tecnología usada para el desarrollo del proyecto.

2.1 Tecnología actual

Destacamos la importancia de conocer la tecnología que integra directamente este proyecto, uno de ellos, el microcontrolador, el cual será necesario para la implementación de este proyecto y programado para realizar la medición de los sensores, su procesamiento y posterior envío de información al modulo wifi el cual comunica con la base de datos.

En el año 2005, en el instituto de IVREA (Italia) nace el proyecto Arduino, una plataforma de hardware libre, basada en una placa con un microcontrolador y un entorno de desarrollo, diseñada para facilitar el uso de la electrónica en proyectos multidisciplinarios.



Figura 2.1. Plataforma de hardware libre Arduino Uno.

El hardware está formado por un microcontrolador, generalmente, Atmel AVR y puertos de entrada/salida

tanto analógicos como digitales. El microcontrolador en la placa Arduino se programa mediante el lenguaje de programación Arduino (basado en Wiring) y el entorno de desarrollo Arduino (basado en Processing).

A partir de la fecha en la que nace esta plataforma surgen otras muchas, como por ejemplo Raspberry Pi, un ordenador de placa reducida o (placa única), que revoluciona el mundo de los SBC (Single Board Computer) por su pequeño tamaño y bajo costo, desarrollado en Reino Unido por la Fundación Raspberry Pi, con el objetivo de estimular la enseñanza de ciencias de la computación en las escuelas.



Figura 2.2. Raspberry Pi, PC de placa reducida.

El hardware está formado por un "Sistem on a chip" Broadcom con procesador ARM 1176JZF-S a 700Mhz, un procesador gráfico Videocore IV, y 512Mb de memoria RAM. El sistema operativo es cargado desde una tarjeta de memoria SD que usa como dispositivo de almacenamiento. Raspberry Pi funciona mayoritariamente con sistemas operativos basados en el núcleo Linux.

2.2 Sistemas de monitorización de energía

El avance de la tecnología hace que cada vez se desarrollen nuevos sistemas que monitorizan el consumo energético, Owl, Efergy, Open Domo, Current Cost, Circutor, Wattio Energy y OpenenErgyMonitor son los productos más conocidos en el mercado.

Se ha analizado únicamente tres de ellos diferenciando los dos primeros por su desarrollo y coste del producto y el tercero por su componente Open Source (Código abierto) y DIY -Do it yourself- (Hazlo tu mismo).

2.2.1 Efergy, Engage Hub Kit

Entre ellos destaca Efergy, un fabricante global de productos de ahorro energético, cuyo objetivo es ayudar a reducir el consumo eléctrico, ahorrar dinero y conservar el medio ambiente.

Su producto más básico es el llamado "**Engage Hub Kit**" se compone de un sensor no invasivo, que se conecta al cable fase del cuadro eléctrico y en su otro extremo al transmisor de datos. Este transmisor a su vez está conectado de forma inalámbrica a un receptor llamado "hub" el cual se conecta a internet a través de un cable Ethernet. Los datos recogidos en un intervalo de 6, 12 o 18s, según la configuración, son subidos a una plataforma web y pueden ser consultados y gestionados desde cualquier dispositivo.



Figura 2.3. Engage Hub Kit (Efergy).

El precio de este dispositivo es de 79,90€, (www.efergy.com), sus ventajas y desventajas son las siguientes:

Ventajas: Su atractiva web, sus aplicaciones para los sistemas operativos móviles android e iOS y un sistema social media que permite compartir el consumo con sus amigos y es uno de los más económicos del mercado.

Desventajas: La cantidad de elementos del sistema sensor, emisor, receptor, (necesita emitir la señal a otro aparato), utiliza baterías para el emisor, su conexión a internet es por cable Ethernet, el tamaño de los elementos son considerables, sólo se puede monitorizar una señal de consumo en todo el sistema, la plataforma web y aplicaciones están adaptadas a esta única señal.

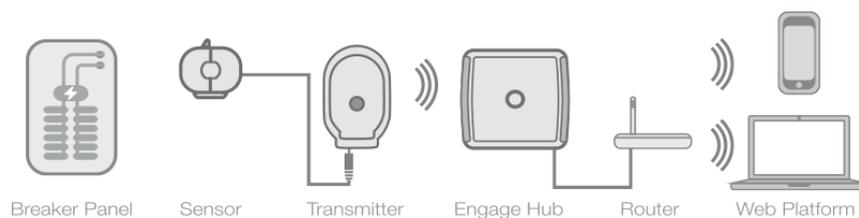


Figura 2.4. Esquema de dispositivos Engage Hub Kit

2.2.2 Wattio, Pack Wattio Energy

Otro de los productos que se ha analizado, es uno de los sistemas más completo del mercado, "**Pack Wattio Energy**", este sistema se compone de una serie de dispositivos de recogida de datos, que se gestionan desde una consola domótica con pantalla táctil o simplemente accediendo a la aplicación de Wattio desde un ordenador, o cualquier dispositivo móvil, con conexión a internet.



Figura 2.5. Pack Wattio Energy (Wattio).

Su precio es de 264,95€ (www.leroymerlin.es) y el Pack incluye centralita, enchufe inteligente, medidor de electricidad y termostato.

Ventajas: Es un sistema muy completo, con enchufe inteligente que mide consumo de aparatos individuales, medidor de electricidad de 3 circuitos para cuadro eléctrico, centralita con pantalla táctil, aplicaciones para dispositivos móviles y se le pueden agregar tantos dispositivos como se necesiten.

Desventajas: Necesita una centralita, la comunicación entre dispositivos es a lo sumo de 30m, utiliza baterías para los dispositivos enlazados.

2.2.3 OpenEnergyMonitor

OpenEnergyMonitor es un proyecto completamente Open Source (código y electrónica abierta) de herramientas de monitoreo de energía, con el objetivo de tener una relación directa con nuestro consumo energético y el desafío de la energía sostenible.

El sistema es totalmente compatible con Arduino IDE y se compone de cuatro partes principales:

- emonTx: módulo principal de conexión de sensores y procesamiento de datos.
- emonGLCD: módulo de pantalla gráfica LCD inalámbrica.
- emonBase: módulo de recepción de datos del monitoreo de energía y envío de datos a un servidor remoto.
- emoncms: software para el procesamiento y visualización de los datos almacenados en el servidor.

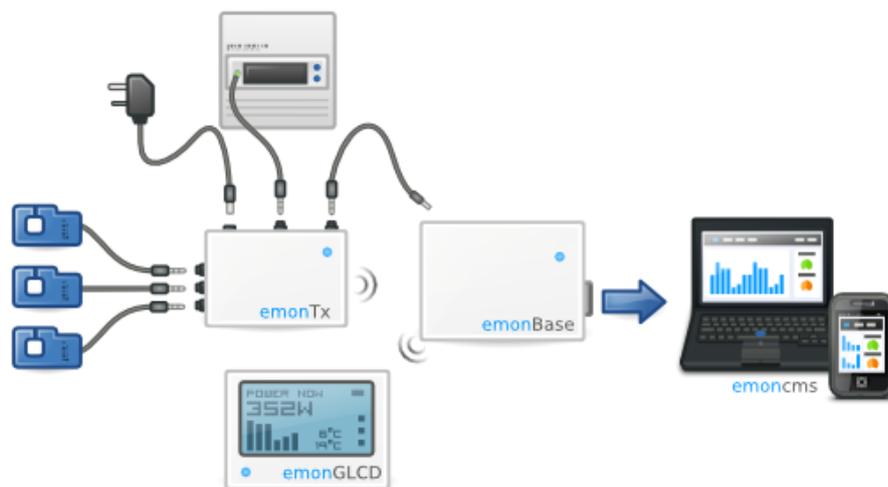


Figura 2.6. OpenEnergyMonitor.

2.3 Tecnología aplicada

Se presentará de una forma global la tecnología usada en el desarrollo el proyecto, tanto en el ámbito del hardware como en el software.

2.3.1 Tecnología de Hardware

Se ha considerado usar la plataforma Arduino como elemento central del sistema, este microcontrolador contiene entradas y salidas analógicas y digitales que permiten leer señales desde los sensores, procesarlas y comunicarlas a los diferentes elementos que integran el sistema, además posee un bajo consumo y una capacidad de cómputo que cubren nuestras necesidades.

Para la comunicación con un punto de acceso wifi, se ha utilizado un módulo llamado Wifly, el cual permite una comunicación serie con cualquier dispositivo, en este caso el Arduino establecerá la configuración y comunicación de los datos que van siendo leídos por los sensores a la base de datos del sistema, a través de un punto wifi que previamente ha configurado el usuario en el dispositivo.

Como elemento de medida, se usará uno o varios sensores de corriente no invasivos, este sensor actúa como un inductor y responde al campo magnético alrededor de un cable con corriente. Al leer la cantidad de corriente que se produce por la bobina, se puede calcular la cantidad de corriente que pasa por el cable.

2.3.2 Tecnología de Software

El software del microcontrolador se ha desarrollado en lenguaje Arduino y librerías C++, para desarrollar este software se puede usar cualquier interface de desarrollo (IDE), en este caso Arduino posee una IDE propia preparada para desarrollar, configurar y programar el dispositivo directamente.

La plataforma web de consulta y configuración está basada en el modelo vista-controlador (MVC), un patrón de arquitectura de software que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones. Este patrón de arquitectura de software se fundamenta en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento. Como herramienta para seguir este patrón se ha usado AngularJS, un framework Javascript de desarrollo de aplicaciones web del lado de cliente, mantenido por Google. Del lado de servidor se ha utilizado PHP que sirve los datos al MVC en formato JSON.

Para el diseño de la plataforma web se ha utilizado un framework llamado Bootstrap, que permite crear interfaces web con CSS y Javascript que adaptan la interfaz dependiendo del tamaño del dispositivo en el que se visualice de forma nativa, es decir, automáticamente se adapta al tamaño de un ordenador o de una Tablet sin que el usuario tenga que hacer nada, esto se denomina diseño adaptativo o Responsive. Los diseños creados con Bootstrap son simples, limpios e intuitivos, esto les da agilidad a la hora de cargar y al adaptarse a cualquier dispositivo.

Para comunicar la configuración y los registros de sensores con la base de datos, se ha desarrollado un servicio web en PHP al que se conecta el módulo wifi y envía o recibe peticiones. Este servicio web autentica en el envío de datos, comprobando que la MAC del dispositivo corresponde con la MAC dada de alta en la plataforma, para evitar que se puedan añadir registros desde cualquier otra petición.

Capítulo 3. Diseño

Una de las primeras fases de un proyecto es diseñar el proceso que se va a seguir para lograr los objetivos. En esta fase se incluyen los requisitos, diseños del hardware y software, las herramientas utilizadas, asimismo se definirá la solución adoptada.

3.1 Requisitos del proyecto

Los requisitos del proyecto representan una visión clara de las necesidades, deseos y expectativas contenidas en el proyecto, clasificados por su función.

3.1.1 Requisitos funcionales

1. Lectura del consumo energético: El usuario podrá colocar un sensor en su cuadro o dispositivo electrónico que permita medir la intensidad que circula.
2. Configuración de la comunicación: El usuario deberá configurar una conexión wifi en el dispositivo para comunicar los datos.
3. Creación de una cuenta web: El usuario podrá crear una cuenta en la plataforma web del producto.
4. Alta del dispositivo en la plataforma: El usuario podrá dar de alta tantos dispositivos como disponga.

5. Configuración de la cuenta: El usuario podrá personalizar en su cuenta el nombre de usuario, contraseña.
6. Configuración del dispositivo: El usuario podrá configurar el nombre, el intervalo de medición, fecha y hora.
7. Consulta de consumo: El usuario podrá consultar su consumo instantáneo, histórico mensual e histórico anual.
8. Notificaciones: El usuario podrá establecer notificaciones sobre límites de consumo y pérdida de conexión con los dispositivos.

3.1.2 Requisitos no funcionales

1. Seguridad en la comunicación: El sistema tendrá un procedimiento de cifrado para la comunicación con la base de datos.
2. Seguridad de acceso a la plataforma: El sistema tendrá un procedimiento de seguridad para bloquear los diferentes ataques en el acceso de usuario y contraseña.
3. Fiabilidad y disponibilidad: La base de datos y plataforma deberá tener un sistema de balanceo de carga y alta disponibilidad.
4. Usabilidad: Tanto el dispositivo como la plataforma web serán sencillos e intuitivos.
5. Soporte: El sistema deberá tener un soporte técnico para atender las dudas e incidencias de los usuarios.

6. Rendimiento y escalabilidad: El sistema tendrá un óptimo rendimiento y será escalable con nuevos firmwares y módulos.
7. Bajo costo: El sistema tendrá un bajo costo como se especifica en los objetivos del proyecto.
8. Bajo consumo: El sistema tendrá un bajo consumo como se especifica en los objetivos del proyecto.

3.2 Diseño del sistema

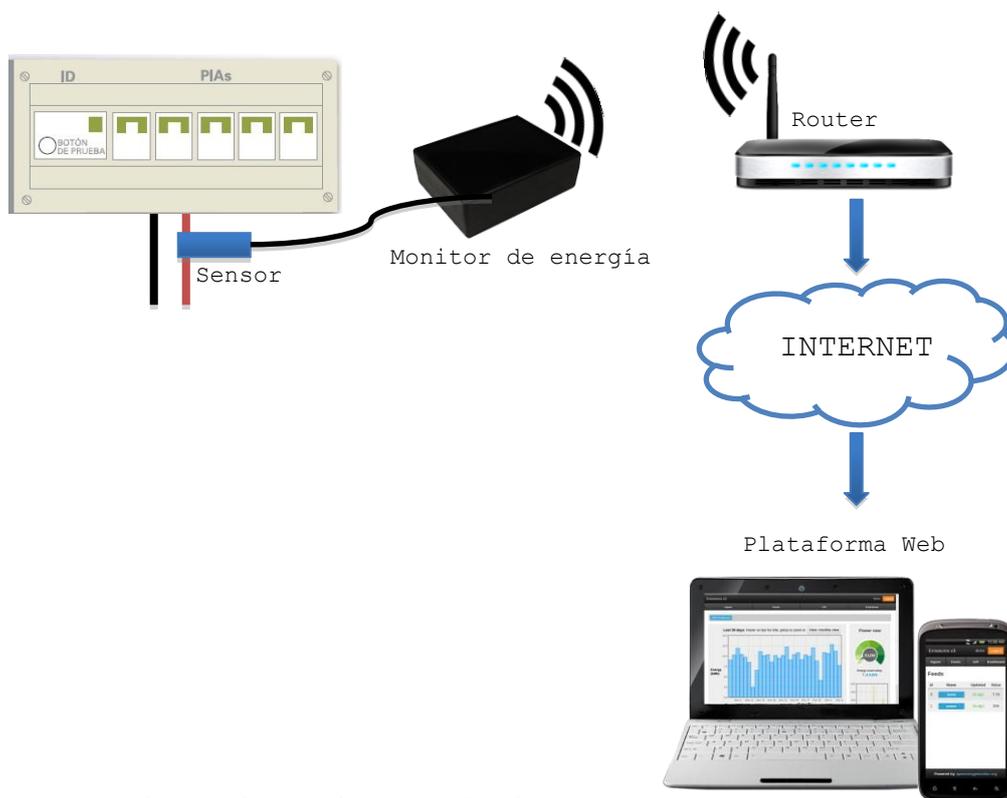


Figura 3.1. Diseño del sistema.

3.3 Comunicaciones del sistema

Esquema general del sistema con diferenciación del diseño electrónico y diseño de datos y web.

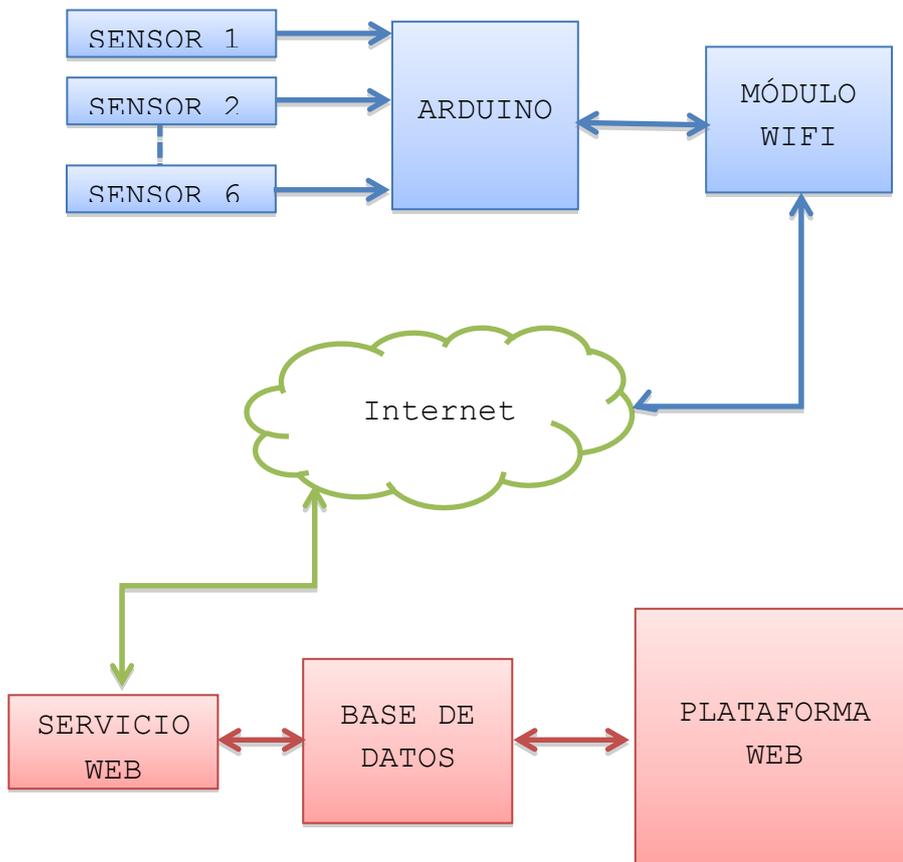


Figura 3.2. Esquema de comunicaciones.

3.3.1 Comunicación física

Esquema de comunicación física del sistema:

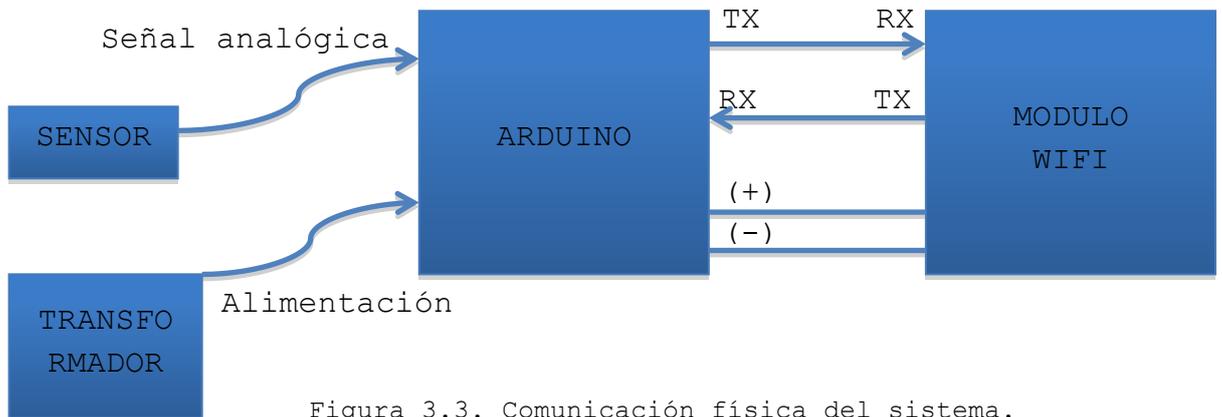


Figura 3.3. Comunicación física del sistema.

3.3.2 Comunicación lógica

Esquema de comunicación lógica del sistema, donde se identifica el flujo de información:

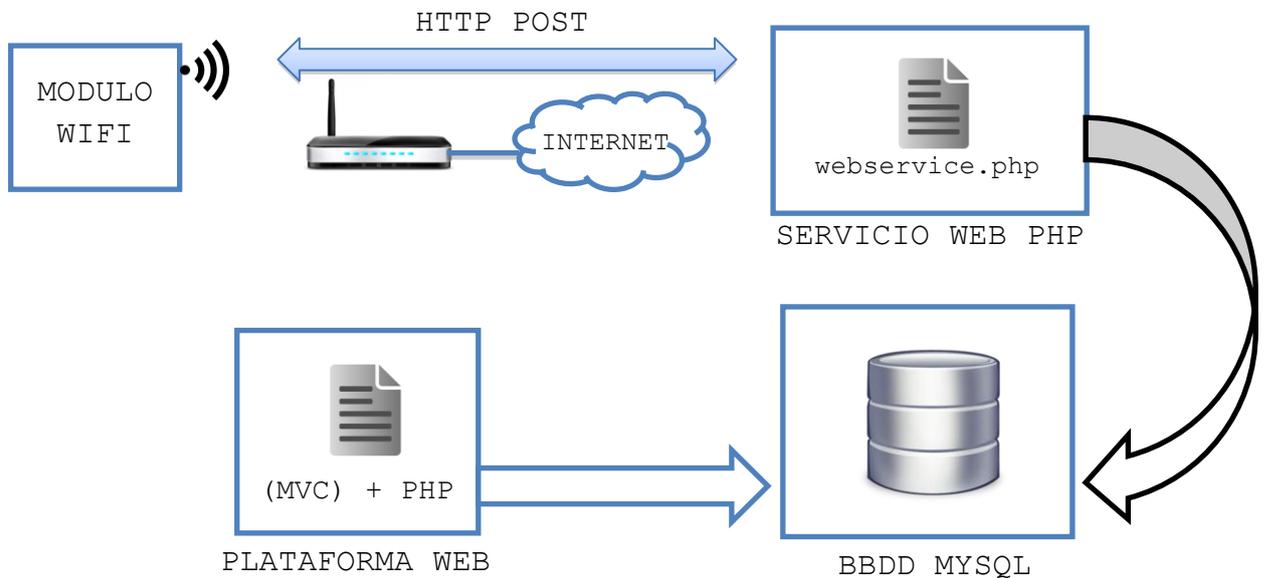


Figura 3.4. Comunicación lógica de los datos.

3.4 Diseño electrónico

Para el diseño electrónico se necesitan cuatro elementos fundamentales, la plataforma Arduino, cuyo microcontrolador se encarga de leer y comunicar los datos, transformador de corriente que alimentará al sistema, sensor de corriente que mide la corriente que atraviesa un conductor y el módulo wifi el cual se conecta a un punto de acceso para enviar los datos.

3.4.1 Arduino

Se ha elegido el Arduino Mini Pro de 3.3V y 8Mhz debido a su pequeño tamaño, ya que no integra la electrónica USB y el microcontrolador es formato SMD, además tiene un bajo consumo y trabaja al mismo voltaje que el módulo wifi.

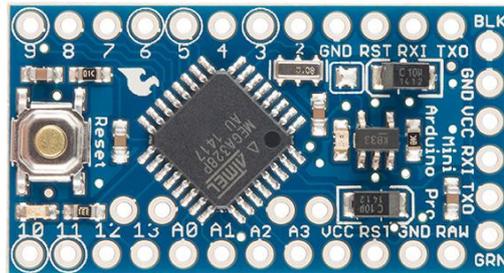


Figura 3.5. Arduino Mini Pro 3.3V

Especificaciones técnicas:

Característica	Valor
Modelo	Mini Pro 3.3V
Microcontrolador	ATmega168
Voltaje	3.3V
Voltaje entrada	3.3V - 12V
Entradas/Salidas digitales	14 (de las cuales 6 PWM)
Entradas analógicas	8
Corriente máxima por entrada/salida	40mA

Memoria flash	16Kb
SRAM	1Kb
EEPROM	512 bytes
Velocidad de reloj	8Mhz

Tabla 3.1. Características Arduino Mini Pro 3.3V.

3.4.2 Transformador de corriente

Es necesario un transformador de corriente de 220V de corriente alterna a 5V de corriente continua, con una intensidad de salida de 2A para alimentar el sistema.



Figura 3.6. Transformador 220-5v, 2A.

Especificaciones técnicas:

Característica	Valor
Tipo transformador	AC-DC
Voltaje entrada	220V
Voltaje salida	5V
Corriente de salida	2A

Tabla 3.2. Características transformador de corriente.

3.4.3 Sensor de corriente

Para medir la corriente se ha utilizado un transformador de corriente de núcleo dividido, ya que pueden ser abiertos y puestos alrededor de los cables existentes sin tener que interrumpir el circuito. El sensor tiene una terminación con conector Jack de 3,5mm que se usará como estándar de conexión de sensores al dispositivo.

Los transformadores de corriente usan la inductancia para medir la corriente, de esta forma el circuito debe pasar al menos una vez a través del sensor.



Figura 3.7. Sensor SCT-013-000

Especificaciones técnicas:

Característica	Valor
Modelo	SCT-013-000
Tensión de entrada	0-100A
Voltaje salida	0-50mV
Resistencia de carga (incluida)	62 Ohm
No linealidad	$\pm 1\%$

Tabla 3.3. Características sensor de corriente.

3.4.4 Módulo wifi

Para la comunicación de los datos a la plataforma se ha utilizado un módulo Wifly RN-XV que provee de comunicación wifi.

El módulo RN-XV se basa en el módulo RN-171 Wi-Fi de Roving Networks e incorpora 802.11 b/g de radio, procesador, pila TCP/IP de 32 bits, reloj de tiempo real, acelerador criptográfico, unidad de administración de energía y sensores analógicos. Tiene precargado un firmware Roving que simplifica la integración y tiempo de desarrollo del sistema.



Figura 3.8. Módulo WiFly RN-XV.

Especificaciones técnicas:

Característica	Valor
Modelo	WiFly RN-XV
Radio	802.11b/g
Consumo en espera	4uA
Consumo recepción	30mA
Consumo transmisión	130 - 185mA
Potencia de salida	+12dBm
Velocidad de transmisión	2.7Mbps

Tabla 3.4. Características módulo wifi.

3.4.5 Flujo de comportamiento

El siguiente diagrama de flujo representa gráficamente el algoritmo que contiene el microcontrolador para enviar los datos al servicio web.

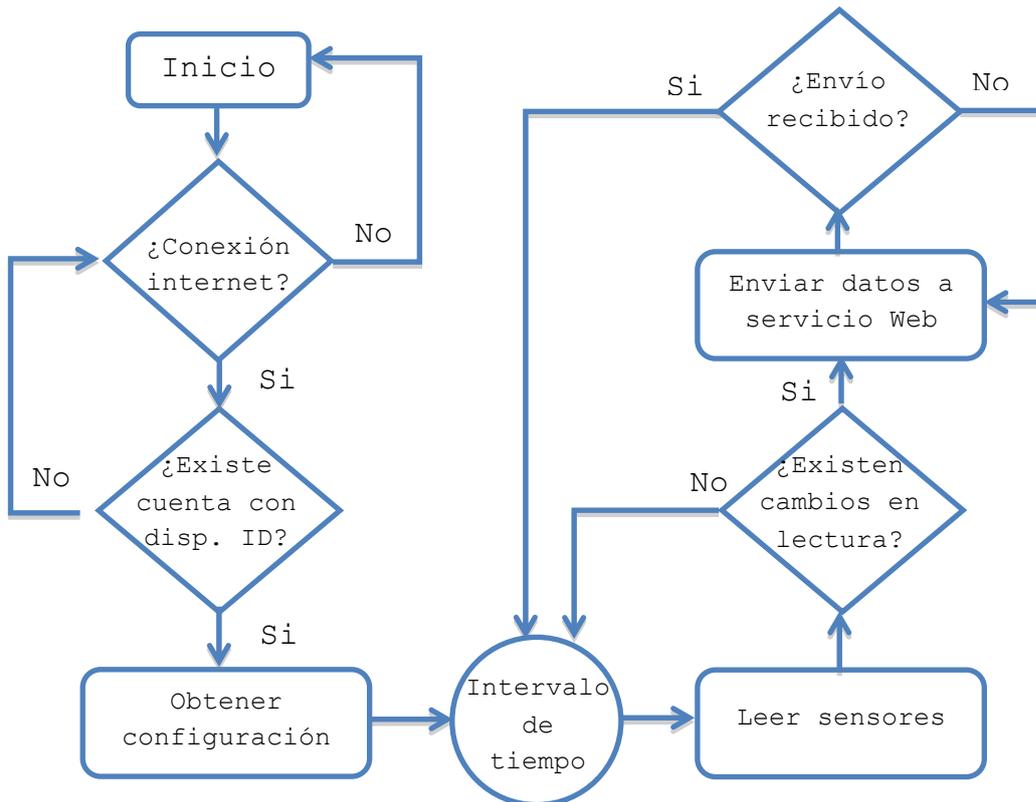


Figura 3.9. Diagrama de flujo algoritmo.

La descripción del algoritmo será la siguiente:

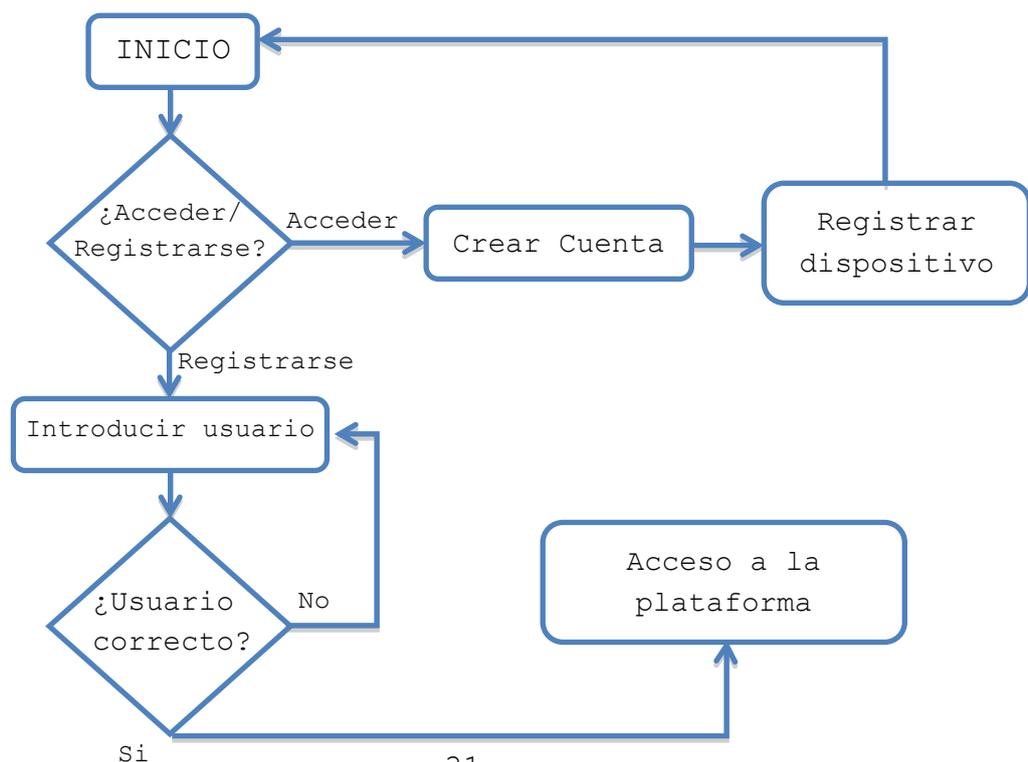
Al conectar el dispositivo, este comprueba que a través del modulo wifi, exista una conexión a internet. Si no existe la conexión, volverá al principio, si existe se comunicará con el servicio web para comprobar que el dispositivo esté en la base de datos. Si no encuentra el dispositivo en la base de datos, seguirá comprobando hasta que el usuario lo registre en su cuenta. Si el dispositivo está registrado, procederá a

obtener la configuración del intervalo de tiempo asignado por el usuario. Una vez obtenida la configuración, comenzará un bucle en el cual, cada vez que se cumple el intervalo de tiempo, lee los sensores de corriente y comprueba que existen cambios en la lectura, si no existen, volverá al intervalo de tiempo, con el fin de evitar excesivos registros en la base de datos. Si existen cambios, se conecta al servicio web y envía los registros. Si los registros no son enviados satisfactoriamente, vuelve a intentar enviarlos con el fin de evitar perder datos en una interrupción de la comunicación wifi. Si los datos se envían correctamente, vuelve a comenzar el bucle en el intervalo de tiempo.

3.5 Diseño de plataforma web

Se ha estudiado el flujo de manejo de la aplicación y a continuación se han realizado los bocetos de las pantallas principales.

3.5.1 Flujo de acceso a la aplicación



3.5.2 Acceso y registro

Para el acceso a la plataforma web se ha realizado un sencillo boceto donde el usuario se podrá identificar una vez creada su cuenta, para crear dicha cuenta existe un enlace que llevará a un formulario de registro. También existe la posibilidad de volver a generar su contraseña en caso de pérdida.

The image shows a wireframe of a web browser window. The browser's address bar contains 'http://'. The page title is 'Monitor Energía'. The main heading is 'Plataforma Monitor de energía'. Below the heading is a login form with two input fields: 'Usuario' and 'Contraseña'. Below the fields is a button labeled 'Entrar'. At the bottom of the form are two links: 'Olvidé mi contraseña' and 'Crear cuenta'.

Figura 3.10. Diseño pantalla acceso y registro.

3.5.3 Panel de control

La pantalla de Panel de control, servirá de pantalla principal de la aplicación, donde se podrá ver de una forma general el consumo actual, el gasto diario en unidades monetarias (previa configuración del precio del Kw/h), gráfica anual y diferentes marcadores de picos de consumo.

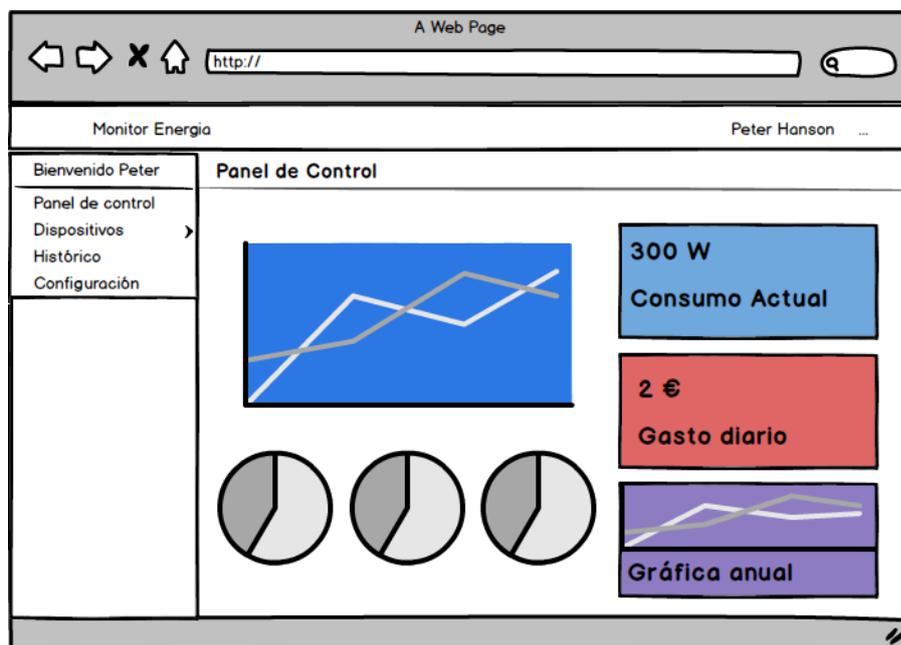


Figura 3.11. Diseño pantalla panel de control.

3.5.4 Dispositivos

La pantalla dispositivos servirá de configuración de los dispositivos instalados en la misma cuenta, de esta manera, se podrán agregar/eliminar, activar/desactivar y ver el estado de cada uno de ellos.

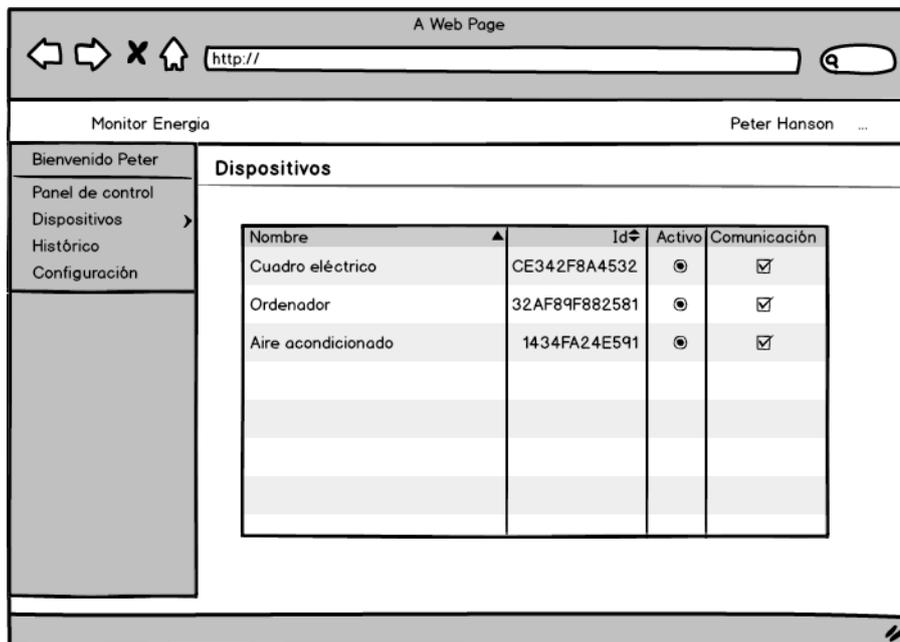


Figura 3.12. Diseño pantalla dispositivos.

3.5.5 Histórico

En la pantalla histórico, se podrá consultar todos los datos recogidos por el sistema desde su puesta en marcha o reseteo, teniendo la posibilidad de ver los datos acumulados por años, meses y días.

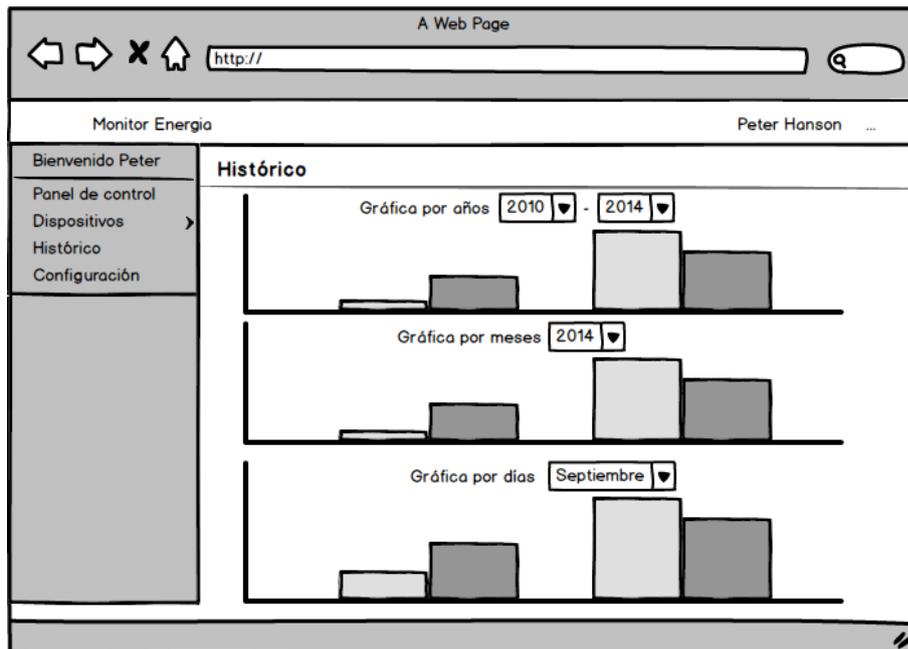


Figura 3.13. Diseño pantalla histórico.

3.5.6 Configuración

En la ventana de configuración existirán cuatro pestañas: general, cuenta, dispositivos y notificaciones.

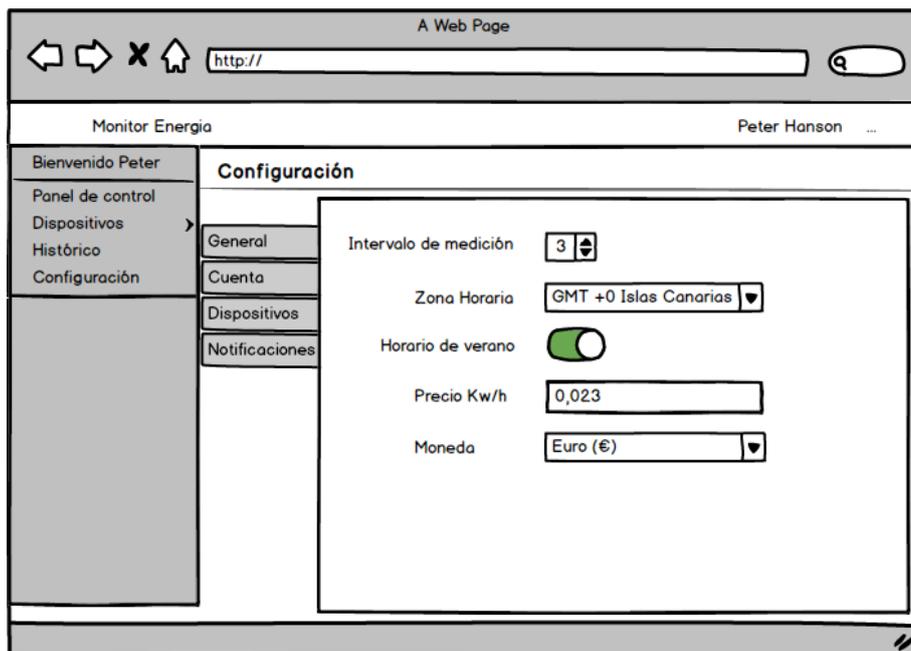


Figura 3.14. Diseño pantalla configuración

3.6 Modelo de datos

Se ha designado MySQL como el motor de base de datos de este sistema, que aunque es un sistema open-source, es necesario una licencia para el uso privativo.

La estructura de la base de datos se fundamenta en cuatro tablas:

- Tabla usuario, se almacenan los datos del usuario de la plataforma, contraseña, nombre completo, email, fecha de creación, fecha de último acceso, activo y relación uno a uno con la tabla de zona horaria.
- Tabla dispositivo, se almacena la configuración de cada uno de los dispositivos para cada usuario, dispositivoId (corresponde con la mac del dispositivo), nombre para identificar el dispositivo, el intervalo de envío de datos y si el dispositivo activo.
- Tabla de registros, contiene la información de los datos recopilados por los sensores, esta tabla va a ser actualizada directamente por el servicio web php y sus campos son el dispositivoId, la fecha y un campo para cada uno de los 6 registros de sensores que puede contener el dispositivo.
- Zona horaria, esta tabla contiene la información de todos los husos horarios del mundo con su valor con el fin de poder mostrar la hora exacta del lugar donde se configure el sistema.

Diagrama de estructura de la base de datos:

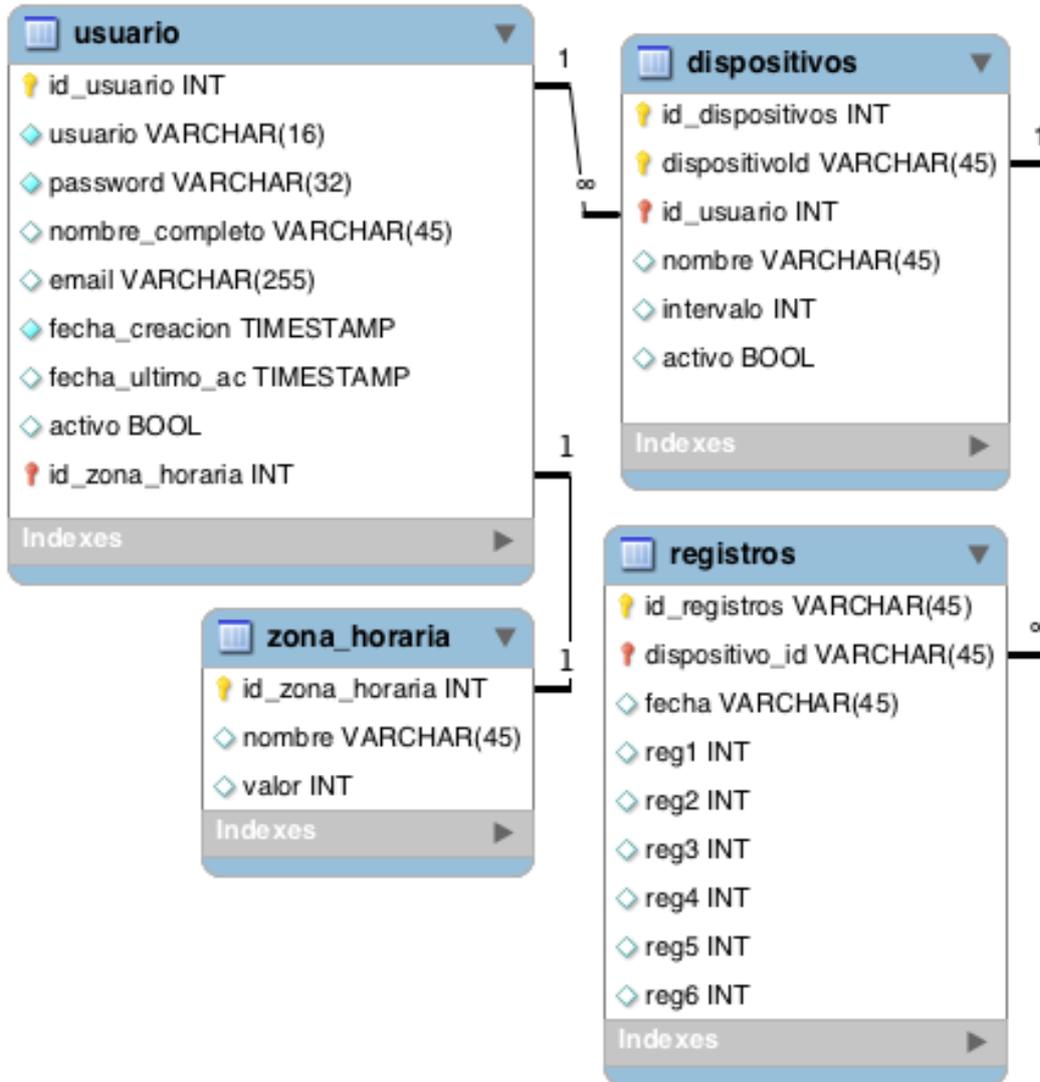


Figura 3.15. Diseño diagrama de base de datos.

Capítulo 4. Implementación

Siguiendo el diseño planteado en capítulos anteriores, se ha implementado el prototipo, que se compone de los elementos electrónicos (hardware) con su programación y de la plataforma web.

En este capítulo, se presentará como se ha construido paso a paso el prototipo del sistema, incluyendo los componentes necesarios, cálculos, conexiones, programaciones y demás detalles hasta concluir con su puesta en funcionamiento.

4.1 Hardware del prototipo

4.1.1 Corriente generada por el sensor

Los transformadores de corriente (CTs), son sensores para medir la corriente alterna, el cual tiene: un devanado primario, un núcleo magnético y un devanado secundario. Para medir la corriente, es necesario hacer pasar el cable que conduce la corriente a medir, por el orificio del sensor, que actuaría como devanado primario y el devanado secundario lo comprenden muchas vueltas de cobre fino alojado dentro de la carcasa del mismo.

La corriente alterna que fluye en el devanado primario, produce un campo magnético en el núcleo que induce una corriente en el devanado secundario. Esta corriente es proporcional a la corriente que fluye en el primario.

La corriente generada por el sensor STC-013-000 es corriente alterna y está calibrado para medir un máximo

de 100A AC, corriente máxima que puede manejar el sensor.

Para conectar un sensor CT a un Arduino, se necesita adecuar su señal para que cumpla con los requisitos de las entradas analógicas del Arduino (tensión positiva entre 0 V y la tensión de referencia AREF 3.3v).

Esto se puede lograr con el siguiente circuito que consta de dos partes principales:

1. La resistencia de carga del sensor:

La señal de salida del sensor es de tipo corriente, por lo que debe ser convertida a voltaje, para ello se colocará una resistencia de carga que se calcula a continuación:

a) Convertir la corriente máxima del sensor a corriente pico multiplicando por $\sqrt{2}$.

$$corr_pico1 = RMS(sensor) * \sqrt{2} = 100A * 1.414 = 141.4 A$$

b) Dividir la corriente pico entre el número de vueltas del sensor.

$$corr_pico2 = corr_pico1 / N^\circ_Vueltas = 141.4 / 2000 = 0.0707 A.$$

c) Para obtener el valor de la resistencia dividir el voltaje de referencia del Arduino (AREF) entre la corriente pico secundaria.

$$val_resistencia = (AREF/2) / corr_pico2 = 1.65 / 0.0707 = 23.33 Ohm$$

Puesto que 23.33 Ohm no es un valor común de resistencia se ha de elegir la más próxima, en este caso 22 Ohm.

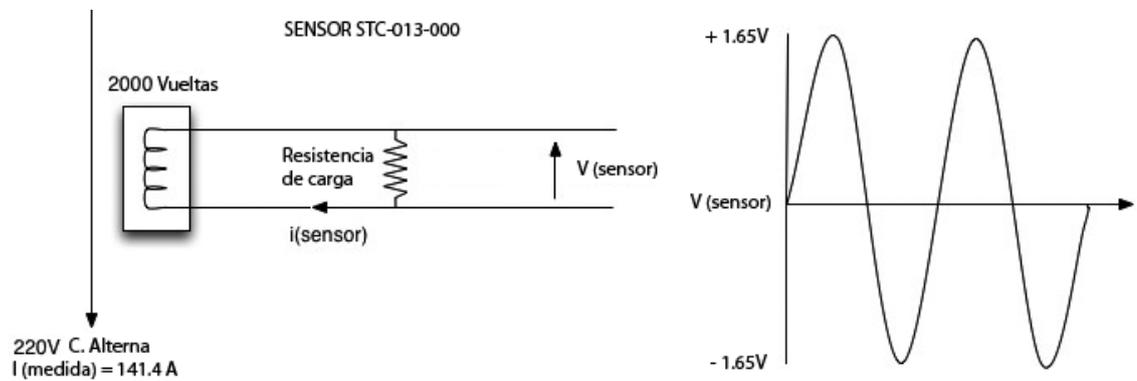


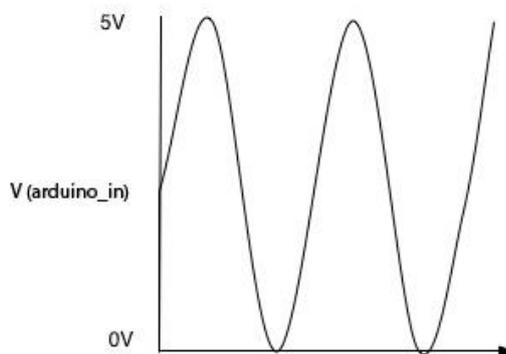
Figura 4.1. Esquema resistencia de carga.

2. El circuito divisor de voltaje:

Como se ha señalado anteriormente, las entradas del Arduino no trabajan con voltaje negativo, por lo que se adecuará la oscilación del voltaje entre 0 y 3.3V.

Para ello se implementará un circuito divisor que, necesita 3 elementos electrónicos:

- 2 Resistencias de 10Kohm.
- 1 Condensador de 10uF.



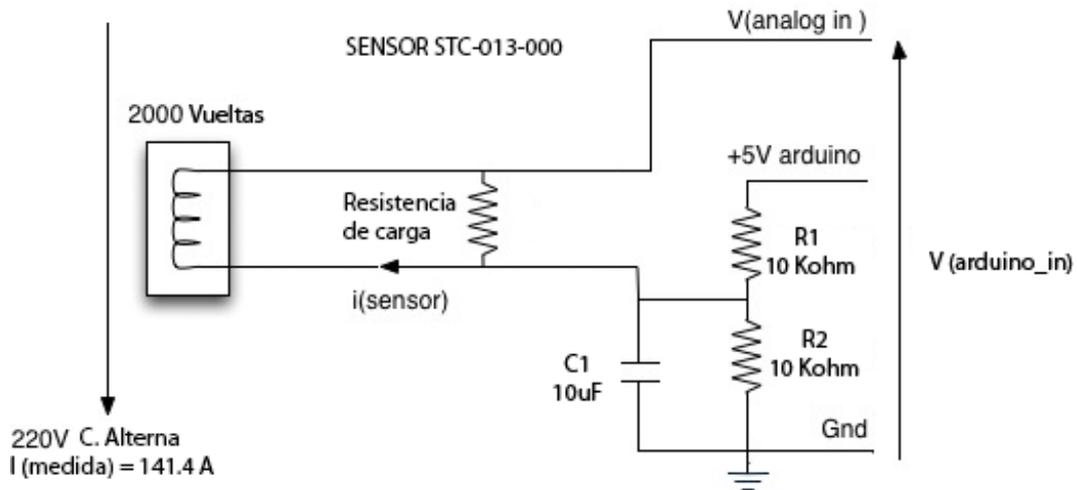


Figura 4.2. Esquema circuito divisor de voltaje.

4.1.2 Regulador de voltaje 5V a 3.3V

La corriente suministrada por el transformador es de 5V. El Arduino soporta tensiones entre 3.3V y 12V pero el módulo Wifly, solo soporta una tensión de 3.3V por lo que se deberá regular el voltaje de 5V a 3.3 con un simple transistor regulador de voltaje.

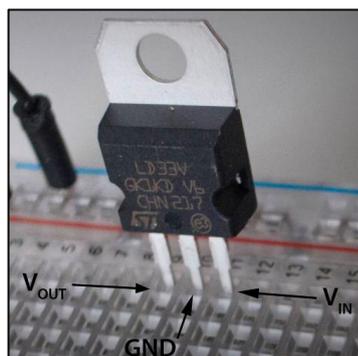


Figura 4.3. Regulador de voltaje.

4.1.3 Pulsador de configuración de Wifly

Para poder establecer una conexión entre el módulo Wifly y un punto de acceso, es necesario conectar durante al menos un segundo el pin N°8 de dicho módulo al polo positivo.

El Wifly pasara a estar en modo ad-hoc y creará un punto de acceso wifi, al que se podrá conectar con cualquier dispositivo. Una vez establecida la conexión, al abrir el navegador, se mostrará una web incluida en el firmware del dispositivo que permite configurar la conexión con cualquier punto de acceso. Simplemente con explorar los puntos de acceso disponibles, seleccionarlo e introducir la clave si fuera necesaria, el módulo volverá a su funcionamiento normal. Esta vez configurado para enviar y recibir datos desde el punto de acceso especificado.

Para facilitar el uso del sistema al usuario, se ha conectado un pulsador, entre el pin de modo ad-hoc y el polo positivo de manera que para configurar la red wifi, simplemente habrá que pulsar el botón una vez que el sistema está conectado y este entrara en modo configuración (ad-hoc).

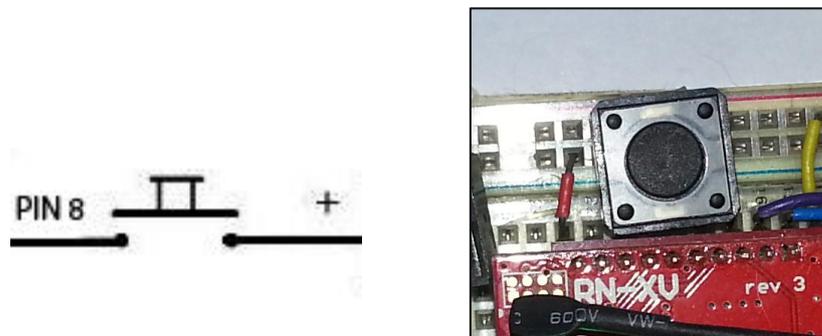


Figura 4.4. Imagen pulsador de configuración.

4.1.4 Montaje del prototipo

Para la implementación del hardware del prototipo se han conectado la resistencia de carga, el circuito divisor de voltaje, el regulador de voltaje, el módulo Wifly y el pulsador de configuración.

Se ha utilizado una pequeña protoboard que permite hacer pruebas con el hardware del prototipo.

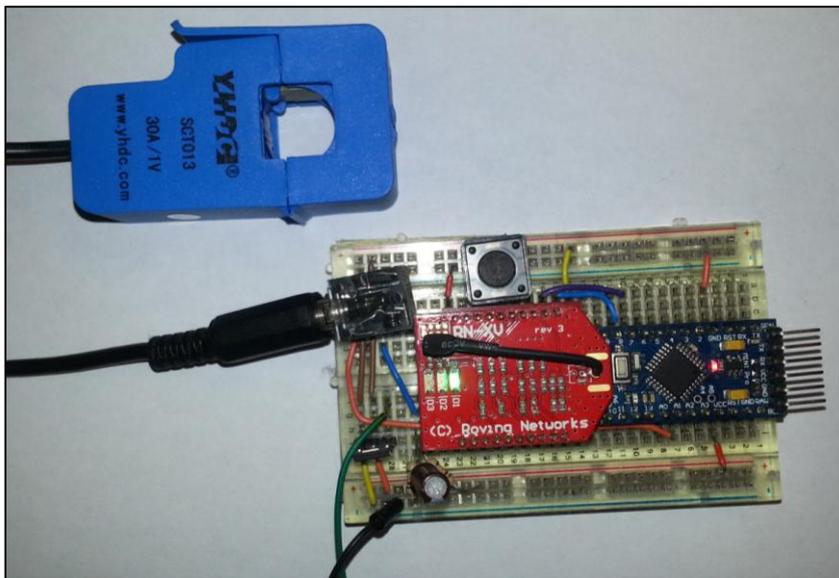


Figura 4.5. Imagen del prototipo.

4.1.5 Firmware del prototipo

Una vez alimentado el sistema, se necesita desarrollar el código que va a ser programado en el microcontrolador.

En primer lugar se han declarado las variables globales, instanciado las clases y declarado las entradas y salidas.

Se han desarrollado cuatro funciones principales, "setupemonitor" configura e inicializa el módulo Wifly para poder operar con él. "readPower" calcula la potencia con intensidad que ha sido calculada del sensor. "calcIrms" realiza la lectura y los cálculos de la intensidad. "postPower" se comunica con el Wifly para consumir el servicio web y subir los datos.

a) Función "setupwifly"

En esta función se inicializa un puerto serial virtual del Arduino a 9600 Baudios, al que esta conectado físicamente el módulo Wifly. A continuación se inicia el módulo en modo de comandos con \$\$\$ como especifica su manual. Se configurada a 0 la ruta por defecto al servicio web, y por último se obtiene la MAC.

```
void setupwifly(){
  serialFly.begin(9600);
  WiFly.setUart(&serialFly);
  delay(500);
  Serial.println("Inicializando...");
  delay(6000);
  serialFly.flush();
  Serial.println("$$$");
  serialFly.print("$$$"); delay(250); check();
  serialFly.println();
  Serial.println("set com remote 0\r");
  serialFly.print("set com remote 0\r"); check();
  Serial.println("get mac\r");
  serialFly.print("get mac\r");
  getMac();
  Serial.print("MAC -> ");
  Serial.println(cmac);
  serialFly.print("exit\r");
  delay(2000);
}
```

```
}
```

b) Función "readPower"

Esta función llama a "calcIrms" tantas veces como sensores tenga el sistema. "calcIrms" se encargará de leer los datos y hacer todo el cálculo para devolver un valor de intensidad (Irms). Este valor se multiplicará por el voltaje normal en las viviendas (230V) para calcular la potencia (Power). Este dato se almacena en la base de datos para cada uno de los sensores.

```
void readPower(){  
  
    for (int i=0;i<5;i++){  
        double Irms = calcIrms(1480, i);  
        int Power = (Irms-0.05)*230;  
        if (Power<0){  
            Power = 0;  
        }  
        powers[i] = Power;  
        Serial.print("Power -> ");  
        Serial.println(Power);  
    }  
}
```

c) Función "calcIrms"

La función "calcIrms" es la más compleja del código, en ella se hacen los cálculos, transformaciones de onda y filtro que devuelven un valor de intensidad medida.

La corriente resultante del sensor (CT) es convertida en una tensión por la resistencia de carga. Este voltaje se mide en relación a la tensión de

alimentación del procesador (en este caso 3,3 V), que se utiliza como referencia, y se escala de manera que la tensión de referencia dará el número máximo de 2^{10} (1024).

El Arduino tiene un rango de tensión de entrada de 0 a 3.3 y así cuando se muestrea una forma de onda de CA de la forma de onda debe estar sesgados a $3.3 / 2$. Esto se traduce en un desplazamiento en el dominio digital de alrededor de 512. La forma de onda muestreada en el dominio digital será entre 0 y 1024, en torno a 512.

Para los cálculos de la intensidad RMS, se deberá quitar primeramente este desplazamiento y esto se puede realizar con un filtro digital de paso alto por software:

```
filteredI=0.996*(lastFilteredI+sampleI-  
lastSampleI);
```

Una vez hecho el cálculo, se eleva al cuadrado la cantidad y se suma al valor anterior:

```
sqI = filteredI * filteredI;  
sumI += sqI;
```

Debido a que es una onda sinusoidal, se debe recoger una determinada cantidad de muestras aplicando el filtro anterior, para obtener una intensidad media. El número de muestras se especifica en la llamada a "calcIrms(NUMBER_OF_SAMPLES)" en este caso 1480, y éste es el bucle en el que se comprenden dichos cálculos.

Una vez obtenida la suma de los cuadrados de la intensidad medida, se calculará su ratio en función del

voltaje de referencia (3.3V) y el número de bits 1024 del dominio digital:

```
double I_RATIO = ICAL * ((SUPPLYVOLTAGE/1000.0) /  
(ADC_COUNTS));
```

Finalmente se calcula la raíz cuadrada de la suma de los cuadrados dividida entre el número de muestras y multiplicada por el ratio calculado:

```
Irms = I_RATIO * sqrt(sumI / NUMBER_OF_SAMPLES);
```

El algoritmo completo de los cálculos anteriores será el siguiente:

```
double calcIrms(int NUMBER_OF_SAMPLES){  
    int SUPPLYVOLTAGE=3300;  
    int ICAL = 90.9;  
  
    for (int n = 0; n < NUMBER_OF_SAMPLES; n++){  
        lastSampleI = sampleI;  
        sampleI = analogRead(inPinI);  
        lastFilteredI = filteredI;  
        filteredI=0.996*(lastFilteredI+sampleI-lastSampleI);  
        sqI = filteredI * filteredI;  
        sumI += sqI;  
    }  
    Serial.println("");  
    Serial.print("A0 -> ");  
    Serial.println(sampleI);  
    double I_RATIO = ICAL * ((SUPPLYVOLTAGE/1000.0) /  
    (ADC_COUNTS));  
    Irms = I_RATIO * sqrt(sumI / NUMBER_OF_SAMPLES);  
    sumI = 0;  
    Serial.print("Irms -> ");  
    Serial.println(Irms);  
    return Irms;  
}
```

d) Función "postPower"

A esta función se le pasan por parámetro los datos de los 6 sensores, a continuación establece una conexión cliente con el servidor y envía un post con dichos datos:

```
void post(int d1=0, int d2=0, int d3=0, int d4=0,
int d5=0){
    String data = "mac="; data += cmac;
    data += "&reg1="; data += d1;
    data += "&reg2="; data += d2;
    data += "&reg3="; data += d3;
    data += "&reg4="; data += d4;
    data += "&reg5="; data += d5;

    Serial.println("Intentando conectar...");
    Serial.println(data);
    if (client.connect()) {
        client.println("POST    /data/set_registro.php
HTTP/1.1");
        client.println("Host: 192.168.1.34");
        client.println("User-Agent: Arduino/1.0");
        client.println("Connection: close");
        client.println("Content-Type: application/x-
www-form-urlencoded;");
        client.print("Content-Length: ");
        client.println(data.length());
        client.println();
        client.println(data);
        Serial.println("-> Datos enviados <-");
        client.stop();
    } else {
        Serial.println("--> FALLO CONEXION <----");
    }
}
```

4.1.6 Programación del microcontrolador

Una vez desarrollado el firmware, se necesita programar el microcontrolador, que consiste en compilar el código del firmware y grabarlo en la memoria del microcontrolador para que sea ejecutado. El modelo Arduino mini pro, no tiene conector de USB para reducir el tamaño de su circuito, por lo que se grabara el firmware con un programador USB to TTL, conectado a los pines TX, RX, Vcc y Ground.

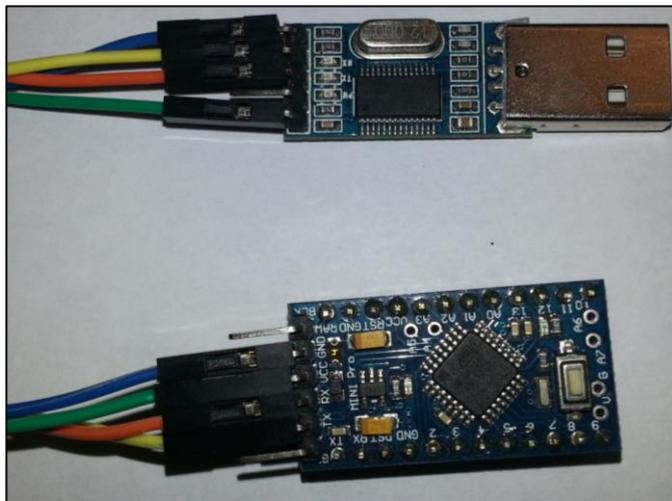


Figura 4.6. Programación del microcontrolador.

4.1.7 Pruebas del prototipo

Tras haber programado el microcontrolador, se configura el módulo Wifly con la conexión wifi del punto de acceso. A continuación se abre un puerto serie configurado con 9600 baudios para observar si los datos y la conexión funcionan correctamente.

La siguiente imagen, muestra la salida que imprime en el puerto serie a modo de testeo.

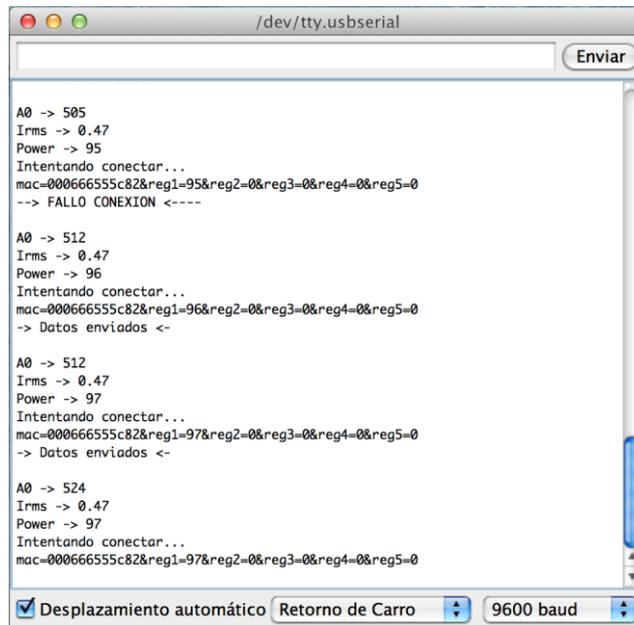


Figura 4.7. Prueba de testeo en puerto serie.

Una vez comprobado que mide la corriente del sensor y se comunica exitosamente con el servicio web para introducir los datos en la base de datos, se hace una consulta al servidor de base de datos y se comprueba que son insertados los registros correctamente.

id	mac	time	reg1	reg2	reg3	reg4	reg5
1384	000666555c82	2014-09-01 23:55:20	99	0	0	0	0
1383	000666555c82	2014-09-01 23:54:48	102	0	0	0	0
1382	000666555c82	2014-09-01 23:54:28	101	0	0	0	0
1381	000666555c82	2014-09-01 23:54:08	97	0	0	0	0
1380	000666555c82	2014-09-01 23:53:47	97	0	0	0	0
1379	000666555c82	2014-09-01 23:53:27	96	0	0	0	0
1378	000666555c82	2014-09-01 23:52:54	95	0	0	0	0
1377	000666555c82	2014-09-01 23:52:34	92	0	0	0	0
1376	000666555c82	2014-09-01 23:52:14	91	0	0	0	0
1375	000666555c82	2014-09-01 23:51:54	102	0	0	0	0
1374	000666555c82	2014-09-01 23:51:35	55	0	0	0	0
1373	000666555c82	2014-09-01 23:51:14	7	0	0	0	0
1372	000666555c82	2014-09-01 23:50:54	0	0	0	0	0
1371	000666555c82	2014-09-01 23:50:34	4	0	0	0	0

Figura 4.8. Prueba de consulta a la BBDD.

4.2 Plataforma web

La plataforma se ha desarrollado en HTML5, CSS3, JavaScript y PHP con los frameworks Bootstrap y AngularJS, siguiendo un modelo vista-controlador. Los datos se obtienen mediante métodos POST a servicios web en PHP. Esta comunicación entre cliente y servidor se realiza en formato JSON.

Se ha definido la siguiente estructura de ficheros: En la carpeta "assets" se agrupan los contenidos de fuentes, imágenes, librerías JavaScript y hojas de estilo. Los ficheros de los diferentes módulos de JavaScript están ubicados en la carpeta "js". Los ficheros secundarios de HTML en la carpeta "pages". El fichero HTML principal "index.html" está ubicado en la raíz. Por último, a carpeta "data" contiene los ficheros PHP que permiten el acceso a la base de datos.

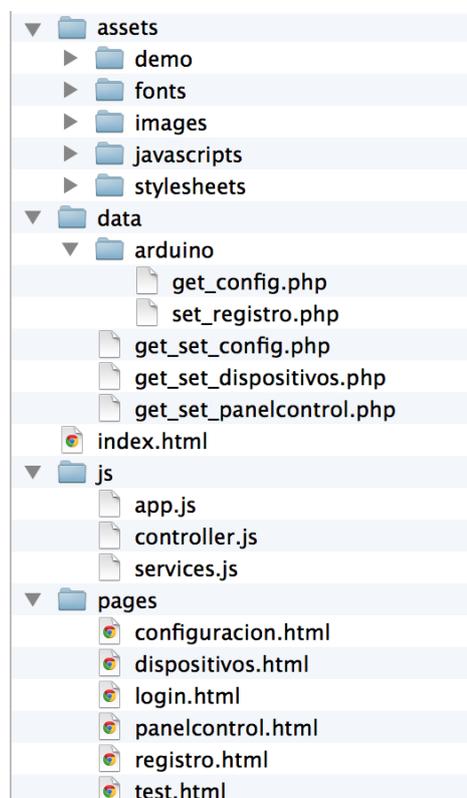


Figura 4.9. Estructura de ficheros.

4.2.1 Menú

Una vez se ha realizado el acceso a la plataforma, se puede visualizar un menú lateral en color oscuro con los datos del usuario registrado y una serie de acciones: salir, configurar, notificaciones y configuración de la cuenta.

A continuación se encuentran las diferentes pantallas por las que se puede navegar en la plataforma: panel de control, dispositivos y configuración.

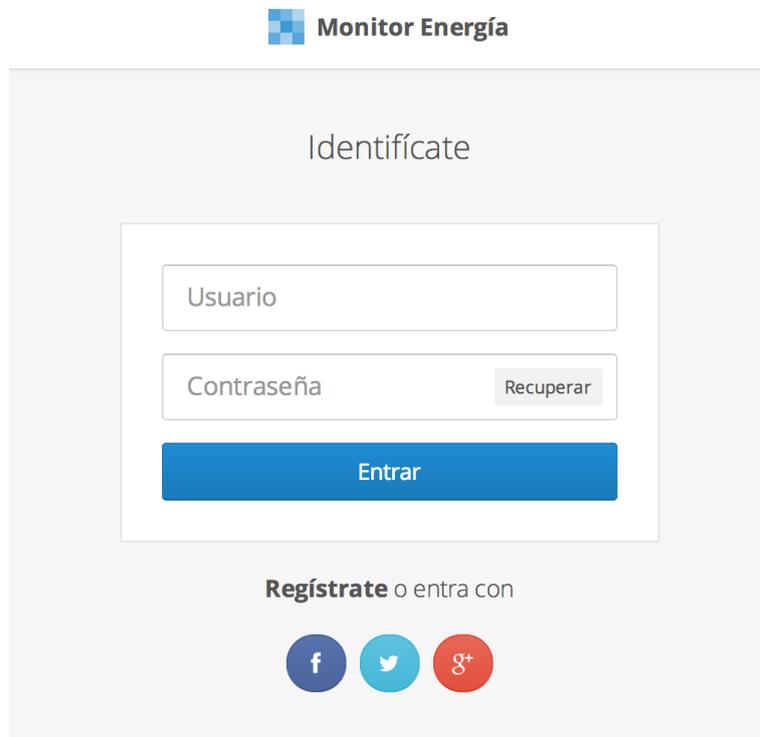
Si existe un fallo con algún dispositivo, se muestra a la derecha una alerta con el texto "Error".



Figura 4.10. Pantalla menú.

4.2.2 Pantalla de acceso y registro

En la siguiente pantalla, el usuario podrá identificarse o registrarse si es su primera vez, también podrá recuperar su contraseña en caso de olvido. Además existe la posibilidad de acceder con una cuenta de Facebook, Twitter o Google+



The image shows a login and registration interface for 'Monitor Energía'. At the top, there is a logo consisting of a blue square with a white grid pattern, followed by the text 'Monitor Energía'. Below this, the word 'Identifícate' is centered. The main form is a white box containing three input fields: 'Usuario', 'Contraseña', and a 'Recuperar' button next to the password field. A blue 'Entrar' button is positioned below the password field. Underneath the form, the text 'Regístrate o entra con' is displayed, followed by three circular icons for social media: Facebook (blue with 'f'), Twitter (light blue with a bird), and Google+ (red with 'g+').

Figura 4.11. Pantalla acceso y registro.

4.2.3 Pantalla de panel de control

La pantalla principal de la plataforma se ha diseñado como un panel de control, que permite visualizar los datos más importantes recogidos por el sistema sin necesidad de navegar por diferentes páginas. En ella se puede ver principalmente, una gráfica con los consumos de los últimos 4 días. En el lado derecho existen 3 bloques que informan de la potencia de consumo actual, la cantidad de consumo en unidades monetarias y una gráfica de medias de consumo, donde especifica la media semanal, mensual y anual.

En la parte inferior de la pantalla se muestran tres relojes con los porcentajes de consumo diario, semanal y mensual según los límites que el usuario establezca en la configuración.

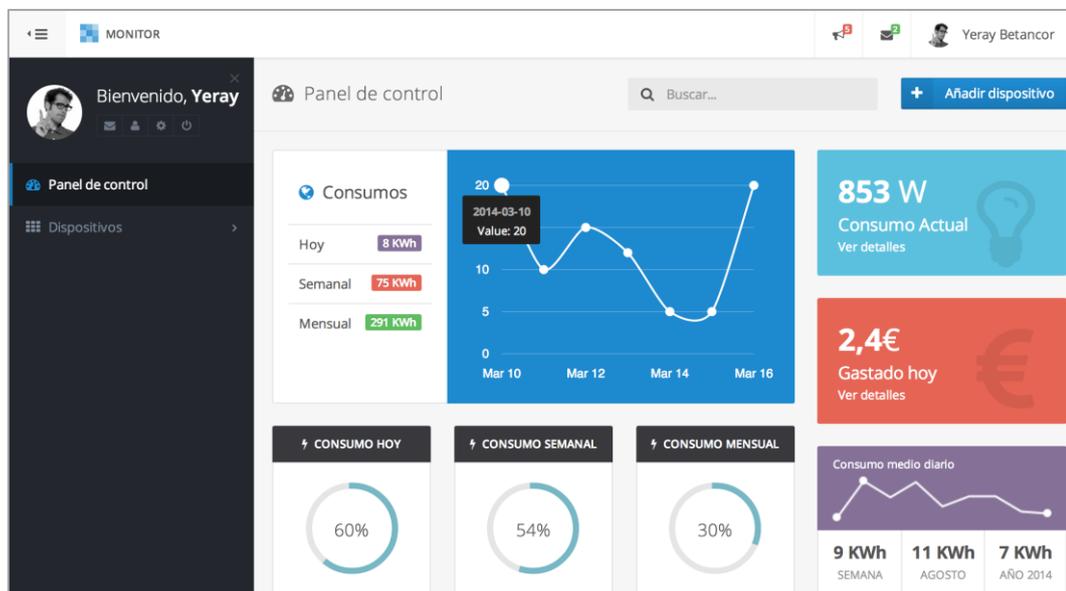


Figura 4.12. Pantalla panel de control.

El framework Bootstrap hace que los diseños sean adaptativos de manera se pueda ver la plataforma en cualquier dispositivo, adaptándose a la resolución de pantalla.

La siguiente imagen muestra la plataforma web desde un dispositivo móvil con resolución de 480x800.

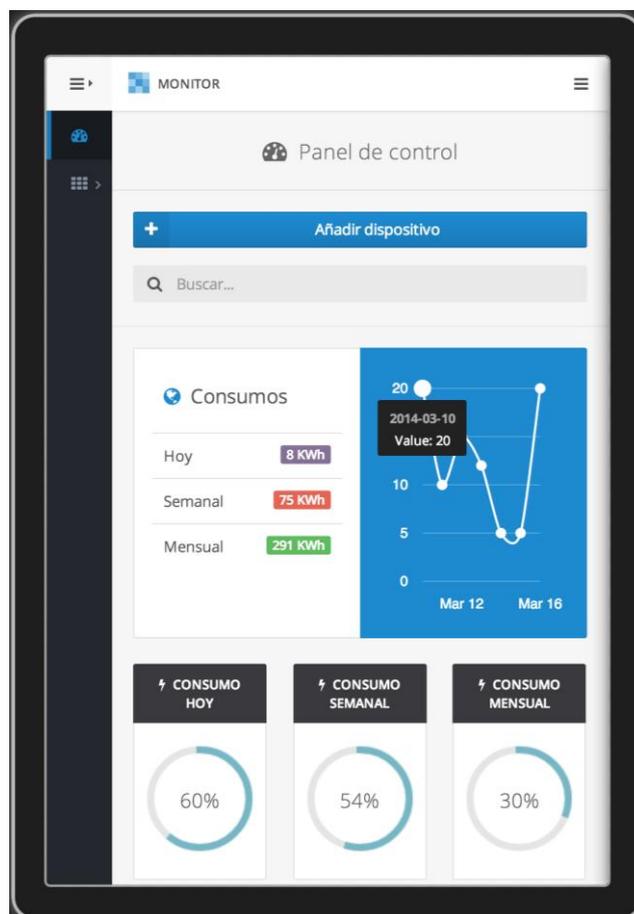
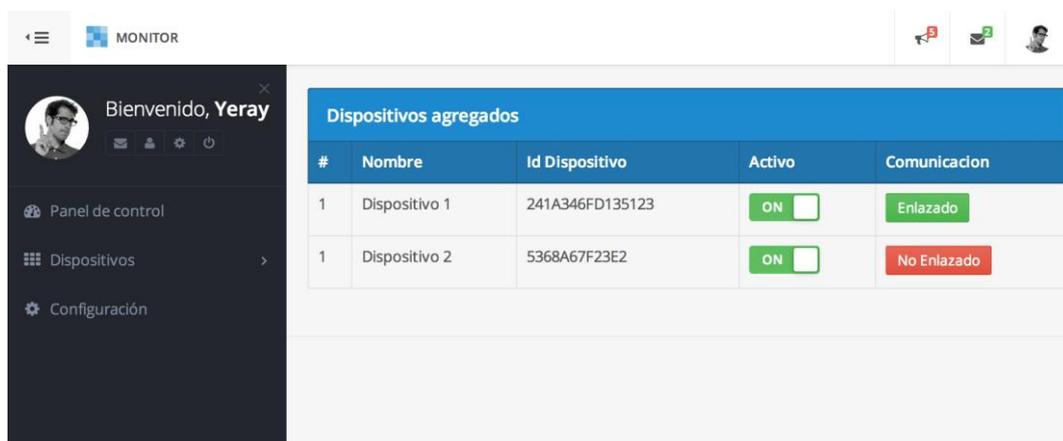


Figura 4.13. Pantalla de panel de control Móvil.

4.2.4 Pantalla de dispositivos

La pantalla de dispositivos se utiliza para agregar o quitar dispositivos, activarlos o desactivarlos, consultar su id, establecer un nombre y comprobar si el dispositivo además de agregado a la cuenta de la plataforma, esta enlazada su comunicación, es decir, si el dispositivo está enviando los datos a la plataforma estará en modo enlazado, en caso contrario, modo no enlazado.



#	Nombre	Id Dispositivo	Activo	Comunicacion
1	Dispositivo 1	241A346FD135123	ON <input type="checkbox"/>	Enlazado
1	Dispositivo 2	5368A67F23E2	ON <input type="checkbox"/>	No Enlazado

Figura 4.14. Pantalla de dispositivos.

4.2.5 Pantalla de histórico

La pantalla histórico recoge todos los datos de los diferentes sensores y dispositivos, estos se ven en una gráfica y en una tabla que puede filtrar y ordenar datos de consumo.

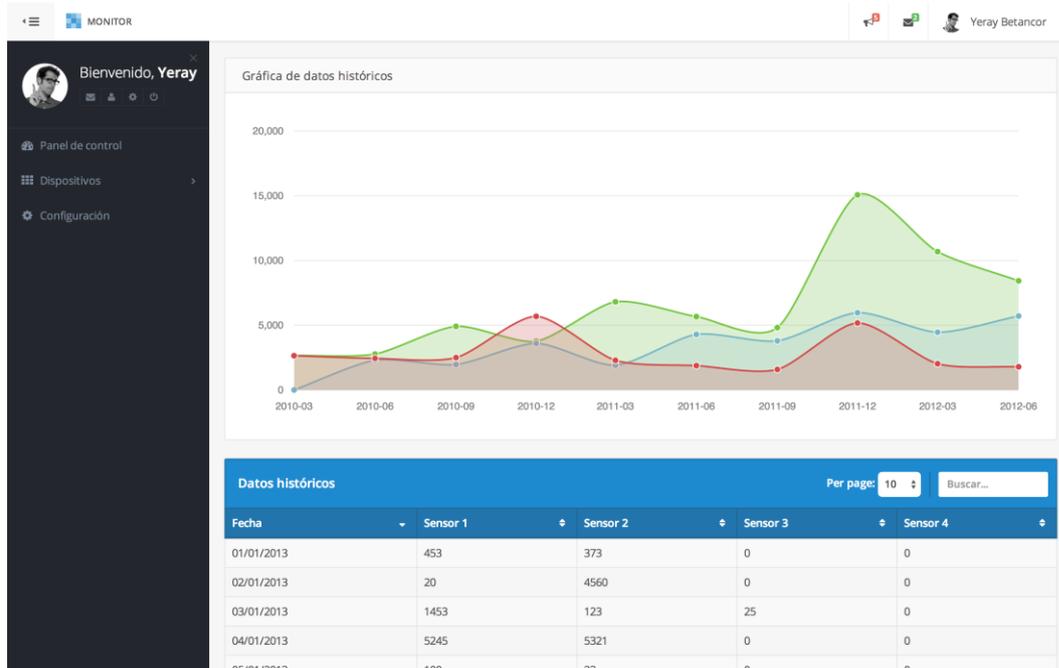


Figura 4.15. Pantalla de histórico.

4.2.6 Pantalla de configuración

La pantalla de configuración se ha dividido en cuatro pestañas según el tipo de configuración:

En la pestaña general se podrá modificar el intervalo de medición, la zona horaria, activar/desactivar el horario de verano, configurar precio de la energía en (€ Kh/h).

En la pestaña cuenta se podrá configurar el nombre de usuario, contraseña, idioma, etc.

En la pestaña dispositivos, configuración detallada de los dispositivos.

En la pestaña notificaciones, se podrán configurar los avisos como sobreconsumo, pérdida de conexión de dispositivos etc.

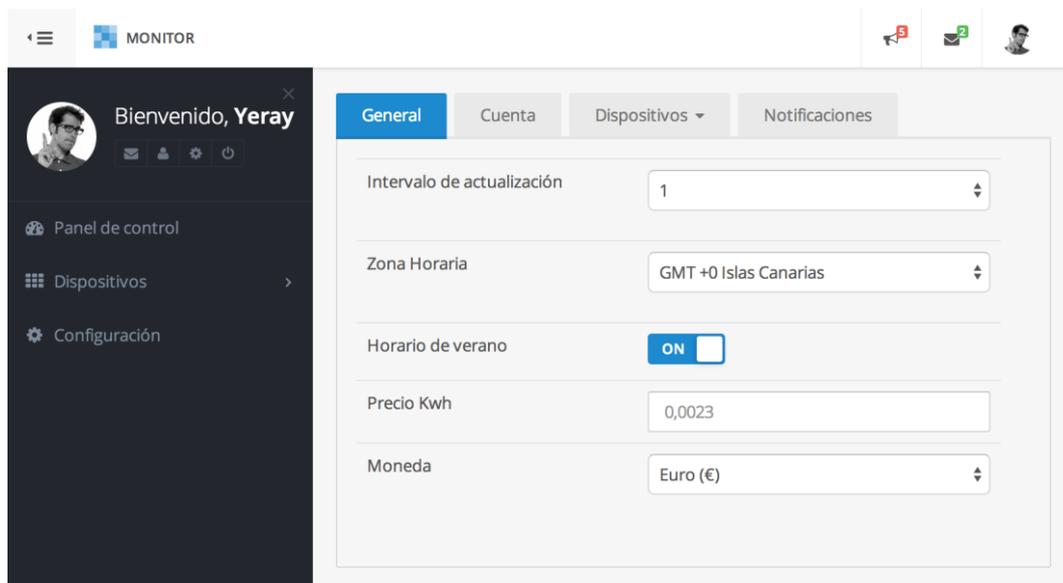


Figura 4.16. Pantalla de configuración.

Capítulo 5. Conclusiones y Trabajos Futuros

5.1 Conclusiones

Durante el desarrollo de este proyecto se ha realizado un análisis completo de las tecnologías relacionadas con el objetivo definido en el primer capítulo del documento. Unas tecnologías que avanzan diariamente en materia de funcionalidad, pero no al mismo tiempo que los avances en el campo de la simplicidad, economía y facilidad de uso e instalación, estos sin embargo lo hacen muy lentamente.

El uso de las tecnologías Arduino y Wifly, permiten abstraerse de la microelectrónica y la programación de bajo nivel, al igual que el uso de Angular y Bootstrap, aportan una rápida integración en el proyecto reduciendo considerablemente su costo. Si el costo del proyecto no fuera uno de los objetivos ni limitaciones, se hubiera optado por el uso de otros sistemas que permitan ejecutar un sistema operativo, aportando más funcionalidades para el usuario y una mayor robustez y flexibilidad en computación. Una de las posibles tecnologías a usar será la placa ODROID-W. Esta placa, está compuesta por una Raspberry Pi y diferentes módulos que aportan un extra de funcionalidad y seguridad.

Siendo la simplicidad, el bajo costo y la facilidad de uso, además del prototipo, los principales objetivos del proyecto podemos determinar que se han cumplido con éxito dichos objetivos con un prototipo funcional como resultado final.

Destacamos que como todo proyecto, ha tenido sus dificultades, las cuales han sido solventadas por medio de la investigación en el campo que las compete siendo esto, un éxito personal en el avance de la formación académica y profesional.

Personalmente, este proyecto ha supuesto todo un reto académico, donde el desarrollo desde sus fases iniciales hasta su fases finales implica poner en práctica las diferentes competencias que componen el conjunto de las asignaturas del Grado en Ingeniería Informática, objetivo que aporta una gran experiencia en la práctica docente del alumno.

5.2 Trabajos futuros

Una vez implementado y probado el prototipo, se abren diferentes líneas de trabajos futuros, la principal línea de trabajo será transformar dicho prototipo en un producto comercial, esto comprende el diseño y fabricación de una placa electrónica que contenga el principal microcontrolador, los circuitos de regulación de voltaje, divisor de voltaje y conectores para cada una de las entradas de sensores.

En la línea de la plataforma web, será necesario aplicar cálculos estadísticos, que contengan diferentes gráficas de consumos, además de tener una línea Social Media, donde se puedan compartir consumos y diferentes datos con otros usuarios. Otra línea importante será la de notificaciones y alertas que se pueden recibir tanto en la aplicación como en los dispositivos móviles.

Capítulo 6. Conclusions and future work

6.1 Conclusions

During the development of this project has made a comprehensive analysis of the technologies related to the objective outlined in the first chapter of the document, advancing technologies daily in functionality but not at the same time that advances in the field of simplicity , economy and easy to use and installation, however these only very slowly.

Using the Arduino and Wifly technologies allow abstract from microelectronics and low-level programming, like using Angle and Bootstrap, provide rapid integration into the project greatly reducing its cost. If the cost of the project was not one of the objectives and constraints, would have opted for the use of other systems allowing run an operating system, providing more user functionality and greater flexibility and robustness in computation. One of the possible technologies to use will be the ODROID-W module. This module is composed of a Raspberry Pi and different modules which provide an extra security functionality.

Being the simplicity, low cost and availability of use, in addition to the prototype, the main objectives of the project can determine which have successfully met these objectives with a working prototype as a final result.

We note that like any project, it has had its difficulties, which have been resolved through research

in the field that falls this being, personal success in the advancement of academic and vocational training.

Personally, this project has all academically challenging course where development from its early stages to its final phase involves implementing the different skills that make all the subjects of Computer Engineering, a target that gives experience student teaching practice.

6.2 Future work

Once implemented and tested the prototype, different lines of future work are opened, the main line of work will transform the prototype into a commercial product, this includes the design and manufacture of a circuit board that contains the main microcontroller, control circuits voltage, voltage divider and connectors for each of the sensor inputs.

In line with the web platform, it is necessary to apply statistical calculations, graphs containing different consumption, in addition to a line Social Media, which can be shared and different consumption data with other users. Another important line is the notifications and alerts that can be received in both the application and mobile devices.

Capítulo 7. Presupuesto

Se ha elaborado un presupuesto diferenciando los costes relacionados con el prototipo y los costes relacionados con la plataforma.

7.1 Presupuesto del prototipo

El siguiente presupuesto recoge los costes para producir una unidad del prototipo con un solo sensor aunque opcionalmente se le pueden agregar hasta cinco.

Componente	Cantidad	Precio	Total
Arduino Mini Pro	1	15,65	15,65
Módulo Wifly	1	31,95	31,95
Conector Jack	5	0,30	1,50
Resistencia	3	0,05	0,15
Condensador	1	0,10	0,10
Transistor	1	0,90	0,90
Sensor	1	9,70	9,70
Mano de obra montaje	2	10	20
TOTAL			79,95

7.2 Presupuesto de la plataforma

El siguiente presupuesto está dividido en un coste de desarrollo y coste anual de hosting y dominio.

Componente	Cantidad	Precio	Total
Desarrollo de la plataforma web	120	10	1200
Servicio de dominio + <u>hosting</u> anual	1		300

Apéndice A: Microcontrolador

A.1. Código del microcontrolador

Código que se ejecuta en el microcontrolador llamando a las funciones explicadas en los anteriores capítulos.

```
void loop() {  
  
    readPower();  
    postPower(powers[0],powers[1],powers[2],powers[3],powers[4]);  
    delay(2000);  
}
```

Apéndice B: Servicio web

B.1. Código webservice.php

```
<?php  
  
if (!empty($_POST)) {  
    $con = mysqli_connect("localhost","root","123456","test");  
    // Se comprueba la conexión con la BBDD  
    if (mysqli_connect_errno())  
        echo "Failed to connect to MySQL: " .  
mysqli_connect_error();  
    else  
        if (auth($con)){  
            insert_reg($con);  
        }  
  
    // Función que inserta registros pasados por POST a la base de  
datos.  
    function insert_reg($con){  
        $sql="INSERT INTO regmonitor VALUES "  
            ."(NULL, '".$_POST["mac"]."',NOW(), '"
```

```

        ".$_POST["reg1"]."', '.$_POST["reg2"]."', '.$_POST["reg3"]."',
        ".$_POST["reg4"]."', '.$_POST["reg5"]."', '.$register[6]."', '.$re
        gister[7]."', '.$register[8].'");

        if (!mysqli_query($con, $sql)){

                echo('Error insert: ' . mysqli_error());
        }
        else{
                echo "1 record added";
                mysqli_close($con);
        }
}

// Funcion que comprueba la autenticación.

function auth($con){
        $sql_auth="SELECT * FROM `dispositivos` WHERE `
        `dispositivoId` = ' ".$_POST["mac"]."'";
        if ($result = mysqli_query($con, $sql_auth)){
                if (mysqli_num_rows($result)>0){
                        echo "Authenticated";
                        return true;
                }
        }
        else{
                die('Error auth: ' . mysqli_error());
                return false;
        }
}

```

Apéndice C: Base de Datos

C.1. Código SQL: Tabla usuario

```

CREATE TABLE IF NOT EXISTS `mydb`.`usuario` (
  `id_usuario` INT NOT NULL,
  `usuario` VARCHAR(16) NOT NULL,
  `password` VARCHAR(32) NOT NULL,
  `nombre_completo` VARCHAR(45) NULL,
  `email` VARCHAR(255) NULL,

```

```
`fecha_creacion` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,  
`fecha_ultimo_ac` TIMESTAMP NULL,  
`activo` TINYINT(1) NULL,  
`id_zona_horaria` INT NOT NULL)
```

C.2. Código SQL: Tabla dispositivos

```
CREATE TABLE IF NOT EXISTS `mydb`.`dispositivos` (  
  `id_dispositivos` INT NOT NULL,  
  `dispositivoId` VARCHAR(45) NOT NULL,  
  `id_usuario` INT NOT NULL,  
  `nombre` VARCHAR(45) NULL,  
  `intervalo` INT NULL,  
  `activo` TINYINT(1) NULL)
```

C.3. Código SQL: Tabla registros

```
CREATE TABLE IF NOT EXISTS `mydb`.`registros` (  
  `id_registros` VARCHAR(45) NOT NULL,  
  `dispositivoId` VARCHAR(45) NOT NULL,  
  `fecha` VARCHAR(45) NULL DEFAULT 'CURRENT_TIMESTAMP',  
  `reg1` INT NULL,  
  `reg2` INT NULL,  
  `reg3` INT NULL,  
  `reg4` INT NULL,  
  `reg5` INT NULL,  
  `reg6` INT NULL)
```

C.4. Código SQL: Tabla zona_horaria

```
CREATE TABLE IF NOT EXISTS `mydb`.`zona_horaria` (  
  `id_zona_horaria` INT NOT NULL,  
  `nombre` VARCHAR(45) NULL,  
  `valor` INT NULL)
```

Apéndice D: Plataforma web

D.1.Codigo Javascript

```
var energyApp = angular.module('energyapp',
    ['ngRoute', 'energyControllers']);

energyApp.config(['$routeProvider',
    function($routeProvider) {
        $routeProvider.
            when('/panel', {
                templateUrl: 'pages/panelcontrol.html',
                controller: 'TestCtrl'
            }).
            when('/config', {
                templateUrl: 'pages/configuracion.html',
                controller: 'ConfigCtrl'
            }).
            when('/login', {
                templateUrl: 'pages/login.html'
            }).
            when('/dispositivo', {
                templateUrl: 'pages/dispositivos.html',
                controller: 'DispositivosCtrl'
            }).
            when('/registro', {
                templateUrl: 'pages/registro.html'
            }).
            when('/historico', {
                templateUrl: 'pages/historico.html',
                controller: 'HistoCtrl'
            }).
            otherwise({
                redirectTo: '/pages/login'
            });
    }]);
```


Capítulo 8. Bibliografía

- [1] PLAN DE AHORRO Y EFICIENCIA ENERGÉTICA 2012 - 2020. Instituto para la diversificación y ahorro de la energía. www.idae.es.
- [2] Microprocesadores y Microcontroladores. Hector Uriel Villamil Gonzalez. Universidad Nacional Abierta y a Distancia. 2009.
- [3] Microcontroladores Microchip. Gerardo Silva. <https://sites.google.com/site/microcontroladoresmicrochip/>
- [4] Boylestad, Robert L. Electrónica: teoría de circuitos. 968-88080-5-9, 1997.
- [5] Arduino Language Reference. <http://arduino.cc/en/Reference/HomePage>.
- [6] OpenEnergyMonitor Documentation <http://openenergymonitor.org/emon/>
- [7] RN-171-XV 802.11 b/g Datasheet. Roving Networks. <http://ww1.microchip.com/downloads/en/DeviceDoc/rn-171-xv-ds-v1.04r.pdf>.
- [8] WiFly Command Reference, Advanced Features & Applications User's Guide. Roving Networks. <http://ww1.microchip.com/downloads/en/DeviceDoc/rn-wiflycr-ug-v1.2r.pdf>.
- [9] Guide to AngularJS Documentation. Open-source tools for energy monitoring and analysis. <http://openenergymonitor.org/>.