



VNiVERSIDAD
D SALAMANCA

CAMPUS DE EXCELENCIA INTERNACIONAL

Trabajo fin de Master

SISTEMA DE CONTROL CON MICROCONTROLADORES

Autor:

Miguel-Ángel Rabanillo de la Fuente

Tutores:

Vidal Moreno Rodilla, José Andrés Vicente Lober

Junio,2021

D. Vidal Moreno Rodilla y D. José Andrés Vicente Lober, Profesores del Departamento de Informática y Automática de la Universidad de Salamanca,

CERTIFICAN:

Que el trabajo titulado “**SISTEMA DE CONTROL CON MICROCONTROLADORES**” ha sido realizado por D. Miguel-Ángel Rabanillo de la Fuente, con DNI 11958529R y constituye la memoria del trabajo realizado para la superación de la asignatura Trabajo de Fin de Máster de la Titulación Máster Universitario en Ingeniería Informática de esta Universidad.

Y para que conste a todos los efectos oportunos

Salamanca, a 3 de septiembre de 2021

D. Vidal Moreno Rodilla

D. José Andrés Vicente Lober



Resumen

En el presente trabajo se expone el diseño de un sistema cuyo hardware está formado por una placa Arduino a la cual están conectados varios sensores como son, sensores ambientales, sensores de calidad del aire, GPS,... que toman medidas en determinados intervalos de tiempo, dichas medidas son enviadas a través de internet a un servidor para ser almacenados y procesados posteriormente.

El conjunto de Arduino y sensores forman un sistema autónomo, el cual está conectado a internet, recibe órdenes desde internet y envía valores leídos de los sensores para su tratamiento. Pueden estar conectados varios sistemas al sistema general, siempre se pueden activar/desactivar para que funcionen o no, se trata de poder colocar un sistema en cualquier lugar que haya internet para tomar valores y ser estudiados.

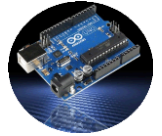
Forma para de este trabajo, el diseño de una aplicación Android, que es la encargada de activar/desactivar sistemas autónomos, así como sus respectivos sensores, puede darse el caso de que en un sistema estén todos, algunos o ningún sensor activo, con lo cual no enviará datos para almacenar.

Esta aplicación Android, se comunica con cada sistema Arduino a través del servidor, de esta forma se le pueden enviar ordenes de ejecución, recibir información sobre su funcionamiento y procesar los datos leídos.

Para el desarrollo del proyecto se contemplan tres bloques bien diferenciados en cuanto al diseño de software, por un lado, está la parte de diseño y programación del sistema Arduino, por otra la programación en la parte del servidor y por último a la aplicación móvil, se han utilizado varias herramientas y lenguajes de programación para enlazar las diferentes fases que se describen en diferentes apartados de esta documentación.

Palabras Clave

Arduino, Microcontrolador, Sensores, Android, Java, Servidor, PHP.



Abstract

In the present work, the design of a system formed by an Arduino board to which several sensors are connected, such as environmental sensors, air quality sensors, GPS, ... which take measurements at certain time intervals, are exposed. They are sent over the internet to a server to be stored and processed later.

The set of Arduino and sensors form an autonomous system, which is connected to the internet, receives orders from the internet and sends values read from the sensors for processing. Several systems can be connected to the general system, they can always be activated / deactivated so that they work or not, it is about being able to place a system anywhere there is internet to take values and be studied.

Form for this work, the design of an Android application, which is responsible for activating / deactivating autonomous systems, as well as their respective sensors, it may be the case that all, some or no sensors are active in a system, thus which will not send data to store.

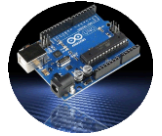
This Android application communicates with each Arduino system through the server, in this way execution orders can be sent, receive information about its operation and process the data read.

For the development of the project, three well differentiated blocks are considered in terms of software design, on the one hand, there is the design and programming part of the Arduino system, on the other the programming in the server part and finally the mobile application, Several tools and programming languages have been used to link the different phases that are described in different sections of this documentation.



TABLA DE CONTENIDO

Resumen	ii
Palabras Clave	ii
Abstract	iii
TABLA DE CONTENIDO	iv
Listado de figuras	vi
Índice de tablas	vii
1 INTRODUCCION	1
1.1 Objetivos	1
1.2 Documentación técnica.....	1
1.3 Organización de la memoria	2
1.4 Metodología	3
1.4.1 Etapas	3
2 DISEÑO DEL PROYECTO	6
2.1 Metodología de desarrollo.....	6
2.2 Especificación de requisitos	7
2.3 Diseño.....	11
2.3.1 Diseño de los datos	12
2.3.2 Diseño arquitectónico	13
2.4 Diseño de la Interfaz.....	16
3 MATERIAL	17
3.1 Hardware.....	17
3.1.1 Arduino Mega 2560.....	17
3.1.2 Ethernet shield	18
3.1.3 Módulos sensores	19
3.1.3.1 Sensor de intensidad lumínica	20
3.1.3.2 Sensor ambiental SPI BME280	20
3.1.3.3 Sensor de calidad del aire CCS811	21
3.1.3.4 Sensor GPS GY-NEO6MV2	21
3.1.3.5 Relé.....	22



3.1.3.6	Pantalla Oled	23
3.1.3.7	Conjunto de leds	23
3.1.3.8	Protoboard	24
3.1.3.9	Fuentes de alimentación	24
3.1.3.10	Cables y resistencias.....	25
3.2	Software	26
3.2.1	IDE Arduino	26
3.2.2	Notepad ++.....	27
3.2.3	Desarrollo para Android	28
3.2.3.1	Android Studio.....	28
3.2.3.2	Librerías Jetpack.....	31
3.2.4	Filezilla.....	33
3.2.5	Dominio en internet	34
3.2.6	Servidor	35
3.2.7	Gestión dominio y hosting	36
3.2.8	PHP	40
3.2.9	MySQL	41
4	INSTALACION Y MONTAJE	42
4.1	Montaje del hardware.....	42
4.2	Instalación del software	45
5	PRUEBAS Y EJECUCIÓN.....	46
5.1	Pruebas.....	46
5.2	Ejecución	48
6	PRESUPUESTO	51
7	LÍNEAS DE TRABAJO FUTURAS	52
8	REFERENCIAS.....	53



Listado de figuras

Figura 1. Diagrama de Gantt	5
Figura 2. Fases del Proceso Unificado	7
Figura 3. REM diseño Especificación de Requisitos	8
Figura 4. Tabla Participante.....	9
Figura 5. Tabla Objetivo Aplicación dispositivo móvil.....	9
Figura 6. Tabla requisito no funcional, diseño de la Interfaz	9
Figura 7. Tabla caso de uso mostrar gráficas	10
Figura 8. Visual Paradigm	11
Figura 9. Diagrama Entidad-Relación	13
Figura 10. Arquitectura de tres capas	14
Figura 11. Diseño pantalla de login	16
Figura 12. Boceto pantalla de login.....	16
Figura 13. Arduino Mega 2560.....	17
Figura 14. Ethernet shield.	19
Figura 15. LDR.....	20
Figura 16. Sensor ambiental BME280.	20
Figura 17. Sensor calidad del aire CCCS811	21
Figura 18. Sensor GPS GY-NEO6MV2	22
Figura 19. Módulo de 4 relés.....	22
Figura 20. Pantalla Oled	23
Figura 21. Leds.....	23
Figura 22. Protoboard	24
Figura 23. Fuente de alimentación BM12	25
Figura 24. Resistencia.....	25
Figura 25. Cables	25
Figura 26. IDE Arduino.....	26
Figura 27. Compilación Sketch de Arduino	27
Figura 28. Editor de texto Notepad ++.....	28
Figura 29. Android Studio versión.....	29
Figura 30. Logo Android Studio	29
Figura 31. Configuración del fichero AndroidManifest.xml	30
Figura 32. Configuración build.gradle (App)	31
Figura 33. Configuración build.gradle (Project)	31
Figura 34. Android Jetpack.....	32
Figura 35. Configuración sitio en Filezilla.....	33
Figura 36. Logo filezilla	33
Figura 37. Progreso de subida de archivos con Filezilla	34
Figura 38. Web domiteca.com	34
Figura 39. Buscador de dominio.....	35
Figura 40. Gestión DNS del dominio	37



Figura 41. Plesk - gestión del sitio	38
Figura 42. Sistema de archivos.....	39
Figura 43. Ejemplo de PHP con HTML.....	40
Figura 44. Gestión PHP desde panel PLESK.....	41
Figura 45. Logo MySQL.....	41
Figura 46. Diseño completo componentes hardware.....	43
Figura 47. Diseño real del sistema	44
Figura 48. Virtual device.....	46
Figura 49. Opciones de desarrollador, móvil	47
Figura 50. Elegir lugar de ejecución.	48
Figura 51. Depuración USB.....	48
Figura 52. Bloque sql inserción microcontrolador	48
Figura 54. Pantalla login	49
Figura 53. Listado de Mcus.....	49
Figura 55. Pantalla lecturas y estado	50
Figura 56. Gráficas.....	50

Índice de tablas

Tabla 1. Base de datos relacional, Entidades	12
Tabla 2. Tabla de características del Hosting	35
Tabla 3. Tabla de presupuesto	51



1 INTRODUCCION

1.1 Objetivos

En la actualidad son cada día más las viviendas calificadas como inteligentes, hacen uso de la domótica para hacer la vida más confortable a sus moradores, no solo en viviendas, sino en edificaciones de todo tipo. Con el avance de las nuevas tecnologías y el coste cada vez menor para su implantación, todo ello propicia que cada día más entornos dispongan de instalaciones que realicen tareas de forma automática sin ser necesaria la presencia física de personas para llevarlas a cabo, en este trabajo se pretende implantar un sistema de toma de medias a través de sensores para su posterior tratamiento, todo ello utilizando tecnologías cada vez más presentes en nuestras vidas. Los objetivos que se pretender conseguir en este trabajo son los siguientes:

- **Objetivo principal:**

El objetivo principal de este “**Trabajo fin de Master**”, es el desarrollo de un sistema compuesto por un microcontrolador y sensores, cuya función es la de tomar mediciones por parte de los sensores para su posterior procesado en remoto.

- **Objetivos secundarios:**

1. Diseñar el sistema completo hardware.
2. Programar el microcontrolador Arduino.
3. Diseñar una aplicación Android para la gestión del sistema.
4. Programar en servidor todo lo necesario para el almacenamiento de los datos enviados desde Arduino.
5. Diseñar la base de datos en la cual se almacenan los datos de gestión.

1.2 Documentación técnica

Se adjunta esta memoria y los Anexos que se describen a continuación:

1. **Plan de Software:** Se recoge la información correspondiente a la planificación temporal del desarrollo del proyecto, consistente en organizar una serie de tareas principales, que a su vez se subdividen otras tareas secundarias.



2. **Especificación de requisitos:** Se muestra información relativa a los requisitos software correspondientes al proyecto a desarrollar.
3. **Especificación de diseño:** Se recoge información relativa al diseño del sistema software, se trata de buscar una solución al problema planteado en forma de software, se definen los primeros pasos para el desarrollo del sistema.
4. **Guía de programación:** Se muestra información sobre el desarrollo del software para este proyecto, se mostrará información sobre las herramientas utilizadas, los ficheros generados e información sobre el hardware donde se instalará.
5. **Componentes hardware:** En este anexo se tratarán los diferentes componentes hardware utilizados en el sistema, se muestra información al detalle de los mismos, se describen el microcontrolador y todos los sensores utilizados.
6. **Montaje e instalación:** En este anexo se divide en dos partes, el montaje del hardware, explicando paso a paso como se conectan los diferentes componentes que forma parte del sistema, en la segunda parte se detalla la instalación del Software para que se ejecutó en hardware y en el servidor.
7. **Manual de Usuario:** En este Anexo se muestra al lector la forma de poner en funcionamiento el sistema completo, una vez instalados el hardware y el software.

1.3 Organización de la memoria

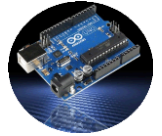
La memoria esta dividida en siete capítulos, más referencias, en el primer capítulo, en el cual nos encontramos, se encuentra la introducción donde se indica los objetivos del proyecto, metodología y etapas, en el segundo capítulo se comentan los puntos principales en cuanto al diseño del sistema.

En el tercer capítulo se describe el material utilizado, en primer lugar, el hardware, mostrando descripción de cada componente e indicando el microcontrolador utilizado, en el segundo apartado dentro de este capítulo se describe el software utilizado para el desarrollo del sistema.

El cuarto capítulo, se describen el montaje e instalación de todas las piezas que componen el sistema.

En el quinto capítulo se indican varias de las pruebas en ejecución realizadas con capturas de pantalla y mensajes, también se indican la forma de utilizar el sistema.

En el sexto capítulo se muestra información sobre los detalles presupuestarios para la implantación de este sistema.



En el séptimo capítulo se mencionan varias líneas de mejora en futuras para que el sistema siga un progreso de mejoría.

Por último, en el octavo capítulo se muestran las referencias externas utilizadas en este documento.

En los Anexos que acompañan a esta memoria se tratan todos y cada uno de los puntos anterior en profundidad, en esta memoria se quiere dar una visión general del trabajo realizado.

1.4 Metodología

La primera etapa llevada a cabo en el desarrollo de este proyecto consistió en el estudio de los requisitos que deben cumplir el sistema a desarrollar, a continuación, se recopiló información sobre el hardware y software necesarios, todo ello tras la celebración de las reuniones pertinentes con los tutores del proyecto, el trabajo fue dividido en diferentes etapas que se muestran a continuación.

1.4.1 Etapas

El presente proyecto se desarrolló en varias etapas sucesivas y coordinadas, se describen a continuación, algunas de las etapas se superponen en el tiempo, en el diagrama de **Gantt** de más abajo se pueden apreciar los tiempos de desarrollo, todos los detalles se describen con detalle en los diferentes Anexos que acompaña esta documentación.

Etapas 1.- Especificación de requisitos, una vez planteada la idea del sistema que se quiere desarrollar, se llevan a cabo varias reuniones con los tutores del proyecto y se conforman los requisitos del sistema.

Etapas 2.- Estudios previos y búsqueda de información en la red sobre microcontroladores, sensores y software. En esta primera etapa, se buscó información en la red sobre cuál de los microcontroladores disponibles en el mercado vendría mejor para realizar este trabajo, el elegido fue Arduino.

Una vez seleccionada la placa, se buscaron los sensores y demás componentes hardware necesarios, buscando los que mejor relación calidad precio tenían. Por último, habiendo decidido el hardware se descargó de la red el software necesario para diseñar todos los módulos, fueron el IDE de Arduino y Android Studio, se buscó información sobre la programación de los diferentes módulos sensores y las novedades de programación de Android Studio incluyendo las librerías JetPack.

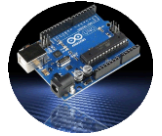


Etapa 3.- Análisis de requisitos, en esta fase fueron elaborados diferentes diagramas entre ellos el diagrama de casos de uso del sistema que ayuda al diseño del mismo.

Etapa 4.- Diseño del sistema, en esta etapa, se definen y diseñan los diferentes componentes que conforman el sistema, incluyendo información y diagramas que ayudan a desarrollar el sistema de manera eficiente.

Etapa 5.- Implementación del software y pruebas, esta etapa se subdivide en varias etapas diferentes

- **Desarrollo del sistema Arduino.** Los sensores y shields adquiridos, se fueron insertando en las placas protoboard y probando cada uno de ellos, comprobando el correcto funcionamiento y mostrando los valores leídos por cada sensor en la pantalla Oled, también comprobó la conexión a internet a través del módulo Ethernet shield. Una vez finalizada esta etapa el hardware se encuentra en funcionamiento, falta optimizarlo y seguir avanzando en el diseño según se vaya avanzando en las etapas siguientes.
- **Desarrollo del sistema PHP en servidor.** Para poder almacenar los datos leídos por los sensores en una base de datos instalada en el servidor web, se desarrolló el software para gestionar dichos datos. El código PHP fue subido a un servidor Web, en una subcarpeta de un dominio personal existente que sirvió para la correcta ejecución de todo el proyecto.
- **Desarrollo de la comunicación entre los sistemas Arduino y Servidor.** Una vez terminadas las etapas 2 y 3, se estableció la comunicación entre Arduino y el servidor Web, comprobando las rutinas de inserción de datos leídos por los sensores, la respuesta por parte del servidor al recibir datos por parte de la placa y control de errores. El diálogo entre las dos partes sirvió para comenzar la siguiente etapa.
- **Desarrollo del sistema Android.** Con Android Studio se diseñó la aplicación Android que sirve para gestionar los sistemas Arduino conectados, los sensores que tienen cada uno de ellos conectado y su estado.
- **Desarrollo de la comunicación entre los sistemas Android y Servidor.** En esta etapa se comprobó el correcto funcionamiento de la aplicación Android, las tareas de leer información de los sistemas conectados, los datos de lectura de cada sistema, mostrar dichos datos de forma textual y gráfica.
- **Integración de las tres plataformas Arduino, Servidor y Android.** La etapa final en cuanto a software consistió en relacionar los tres servicios, aplicación Android, Arduino y servidor apache, se comprobó que funcionan correctamente todos los módulos del proyecto.
- **Realización de pruebas de software de cada uno de los bloques desarrollados.**



- **Documentación**, en esta etapa se fue solapando con el desarrollo de etapas anteriores, se documentó toda la información del proyecto.

Nombre	Fecha de inicio	Fecha de fin
☐ ● Especificación de requisitos	15/1/21	11/2/21
● Reunión con los tutores del proyecto	15/1/21	15/1/21
● Estudio de requisitos del sistema funcionales y no funcionales	18/1/21	21/1/21
● Estudio de los casos de uso	22/1/21	29/1/21
● Documentar los requisitos	1/2/21	2/2/21
● Documentar los casos de uso	3/2/21	10/2/21
● Revisión por parte de los tutores	11/2/21	11/2/21
☐ ● Búsqueda de información	12/2/21	11/3/21
● Búsqueda de información sobre microcontroladores	12/2/21	16/2/21
● Búsqueda de información sobre sensores	17/2/21	26/2/21
● Búsqueda de información sobre hosting web	1/3/21	3/3/21
● Búsqueda de información sobre herramientas de desarrollo	4/3/21	11/3/21
☐ ● Análisis de requisitos	12/3/21	22/4/21
● Dominio	12/3/21	17/3/21
● Casos de uso	18/3/21	31/3/21
● Diagrama de actividad	1/4/21	16/4/21
● Diagrama de clases	19/4/21	20/4/21
● Documentación del análisis	22/4/21	22/4/21
☐ ● Diseño del sistema	23/4/21	29/4/21
● Modelo de diseño de software	23/4/21	23/4/21
● Arquitectura	26/4/21	26/4/21
● Diseño de casos de uso	27/4/21	27/4/21
● Diseño de la interfaz	28/4/21	28/4/21
● Documentación del diseño	29/4/21	29/4/21
☐ ● Implementación y pruebas	30/4/21	31/8/21
● Programación del microcontrolador	30/4/21	17/6/21
● Programación del servidor	18/6/21	22/7/21
● Creación de la base de datos	23/7/21	23/7/21
● Programación del dispositivo móvil	26/7/21	20/8/21
● Pruebas	23/8/21	25/8/21
● Documentación	26/8/21	31/8/21

Figura 1. Diagrama de Gantt



2 DISEÑO DEL PROYECTO

En este apartado se muestra información sobre los aspectos más importantes del desarrollo del proyecto, únicamente se muestra información relevante, en detalle se encuentra en los **Anexos I, II y III** que acompañan esta memoria.

2.1 Metodología de desarrollo

La metodología utilizada en el desarrollo del software fue la que está basada en denominado Proceso Unificado, el cual es un marco de trabajo en el cual se basan muchos proyectos software, las principales características son las siguientes:

Iterativo e incremental:

El Proceso Unificado se encuentra dividido en cuatro fases, *Inicio, Elaboración, Construcción y Transición*, a su vez subdivididas en otras fases.

Dirigido por casos de uso:

Los casos de uso son utilizados para recopilar los requisitos funcionales del sistema, cada iteración toma un conjunto de casos de uso.

Centrado en la arquitectura:

En el Proceso Unificado no existe un modelo único de arquitectura que cubra todos los aspectos del sistema, por lo tanto, existen varios modelos que definen la arquitectura del sistema software.

Enfocado a riesgos:

Se deben identificar los riesgos críticos de forma temprana en cada fase del ciclo de vida, asegurando que los riesgos principales sean tratados en orden preferente.

Las fases del proceso Unificado se pueden observar en la figura siguiente.

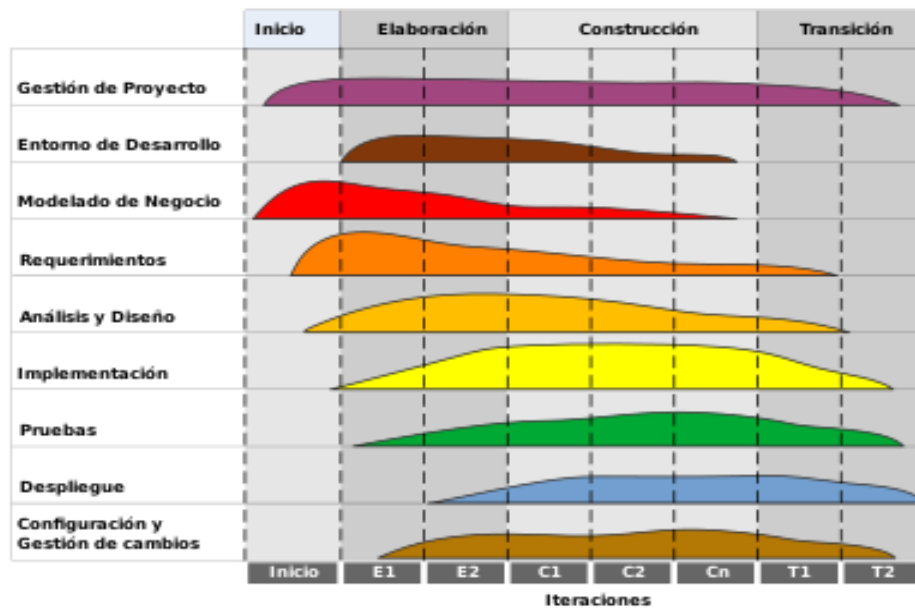


Figura 2. Fases del Proceso Unificado

2.2 Especificación de requisitos

La especificación de requisitos del sistema recoge las necesidades de los usuarios, clientes y los requisitos que debe cumplir el sistema para satisfacer las necesidades. Para realizar el documento de requisitos, se han seguido los pasos de la siguiente publicación “*Metodología para la Elicitación de Requisitos de Sistemas Software*”.

Se trata de una metodología que nos ayuda a definir las tareas a llevar a cabo, productos a obtener y técnicas a utilizar, cuyos autores son *Amador Durán Toro* y *Beatriz Bernárdez Jiménez*, este método define las tareas a realizar, sirve como base para la creación del documento requisitos, entre otras cosas figuran una serie de tablas en las cuales recogen toda la información del proyecto. Las distintas fases que describe esta metodología son las siguientes:

- Participantes en el proyecto
- Objetivos a cumplir por el sistema
- Requisitos de información
- Requisitos de restricción
- Requisitos funcionales
- Requisitos no funcionales



Las herramientas utilizadas para esta fase de “Especificación de Requisitos”, fueron **REM**, que es una herramienta de gestión de requisitos y **Visual Paradigm**, herramienta utilizada para diseño en UML, en esta imagen de abajo se observa la captura de la herramienta REM.

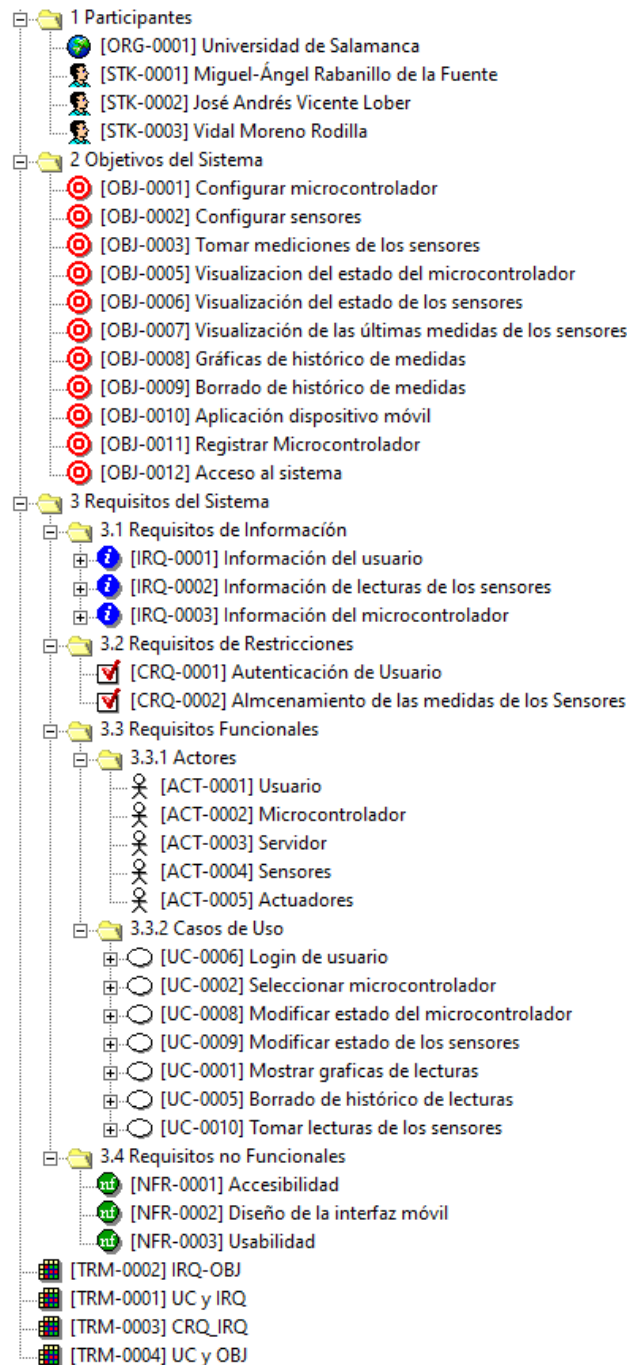


Figura 3. REM diseño Especificación de Requisitos



A continuación, se muestran algunas de las tablas diseñadas en este proyecto, el documento de Especificación de Requisitos completo se encuentra en el **Anexo II**.

En esta tabla se refleja el objetivo del sistema “Aplicación dispositivo móvil” (OBJ-0010) que indica, el autor, las fuentes, la descripción del objetivo, la importancia y demás características. En las dos tablas siguientes se muestra la de un participante, tutor en este caso y la de un requisito no funcional

OBJ-0010	Aplicación dispositivo móvil
Versión	1.0 (10/08/2021)
Autores	<ul style="list-style-type: none"> • Miguel-Ángel Rabanillo de la Fuente
Fuentes	<ul style="list-style-type: none"> • José Andrés Vicente Lober • Vidal Moreno Rodilla
Descripción	El sistema deberá <i>disponer de una Aplicación Movil, la cual gestiona todo el sistema completo, permitirá activar/desactivar sensores y microcontroladores, así como leer lecturas y borrar las mismas, el usuario mediante login accede a todas estas funcionalidades</i>
Subobjetivos	Ninguno
Importancia	vital
Urgencia	inmediatamente
Estado	PD
Estabilidad	alta
Comentarios	Ninguno

Figura 5. Tabla Objetivo Aplicación dispositivo móvil

Participante	José Andrés Vicente Lober
Organización	Universidad de Salamanca
Rol	Tutor
Es desarrollador	No
Es cliente	Sí
Es usuario	No
Comentarios	Ninguno

Figura 4. Tabla Participante

NFR-0002	Diseño de la interfaz móvil
Versión	1.0 (13/08/2021)
Autores	<ul style="list-style-type: none"> • Miguel-Ángel Rabanillo de la Fuente
Fuentes	<ul style="list-style-type: none"> • José Andrés Vicente Lober • Vidal Moreno Rodilla
Dependencias	Ninguno
Descripción	El sistema deberá <i>disponer de una interfaz para dispositivos móviles con el mayor porcentaje compatibilidad</i>
Importancia	importante
Urgencia	inmediatamente
Estado	PD
Estabilidad	alta
Comentarios	Ninguno

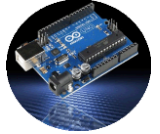
Figura 6. Tabla requisito no funcional, diseño de la Interfaz



En esta última tabla, se refleja un caso de uso, “mostrar gráficas de lecturas”, la información que contiene dicha tabla figuran el autor, las fuentes de las cuales se obtuvo la información, las dependencias, la descripción, precondition en este caso si la hay y las secuencias, en ellas se observan los pasos y las acciones a llevar a cabo.

UC-0001	Mostrar graficas de lecturas	
Versión	1.0 (10/08/2021)	
Autores	<ul style="list-style-type: none"> • Miguel-Ángel Rabanillo de la Fuente 	
Fuentes	<ul style="list-style-type: none"> • José Andrés Vicente Lober • Vidal Moreno Rodilla 	
Dependencias	<ul style="list-style-type: none"> • [IRQ-0002] Información de lecturas de los sensores • [IRQ-0003] Información del microcontrolador • [OBJ-0010] Aplicación dispositivo móvil 	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando se pulse el botón para ver las gráficas de las lecturas	
Precondición	Haber seleccionado un microcontrolador entre el listado de ellos	
Secuencia normal	Paso	Acción
	1	El sistema muestra a través de la Aplicación Móvil, un botón, que tras pulsarlo visualizar las las lecturas de los diferentes sensores en modo gráfica
	2	El actor Usuario (ACT-0001) pulsa el botón para visualizar las gráficas
	3	Si pulsado el botón para ver las gráficas, el actor Servidor (ACT-0003) muestra los datos graficados para cada uno de los sensores
	4	El actor Usuario (ACT-0001) pulsando diferentes botones se pueden visualizar las graficas correspondientes a cada uno de los sensores
Postcondición	PD	
Excepciones	Paso	Acción
	-	-
Rendimiento	Paso	Tiempo máximo
	-	-
Frecuencia esperada	PD	
Importancia	importante	
Urgencia	inmediatamente	
Estado	PD	
Estabilidad	alta	
Comentarios	Ninguno	

Figura 7. Tabla caso de uso mostrar gráficas



2.3 Diseño

En esta fase comienza al término de la anterior, una vez obtenidos todos los requisitos del sistema software se pasa a la fase de diseño, se encuentra dividida en varios apartados, el grueso de la información se encuentra en el **Anexo III**, acto seguido se muestran las siguientes fases dentro del diseño de manera simplificada, se utilizó la herramienta de diseño UML, **Visual Paradigm**, la siguiente imagen muestra una captura del esquema generado con la herramienta.

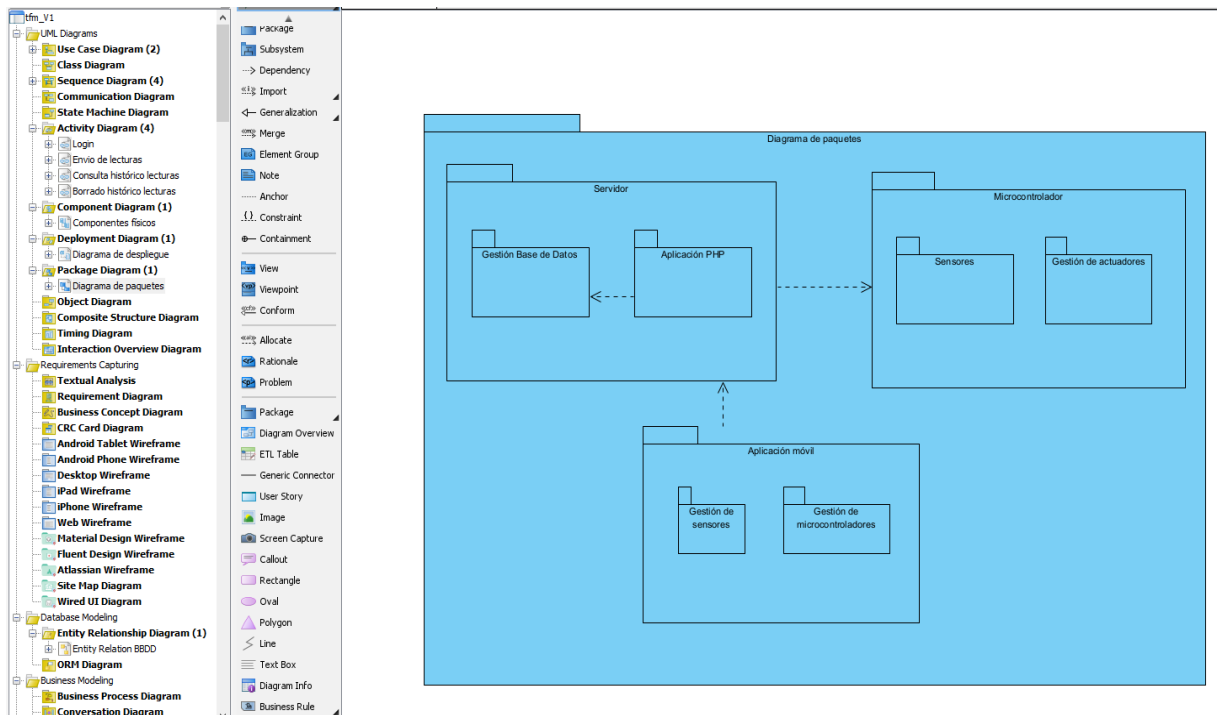


Figura 8. Visual Paradigm



2.3.1 Diseño de los datos

En este apartado se explica la creación de la base de datos relacional utilizada en el sistema, en primer lugar, La base de datos relacional está formada por un conjunto de tablas (relaciones), compuestas por filas (registros) y por columnas (campos), cada registro dispone de un identificador único denominado clave. Las entidades que forman la base de datos son las siguientes:

ENTIDAD	DESCRIPCIÓN
lecturas	En esta tabla se almacenan las lecturas de los sensores
mcu	En esta tabla se almacena la información relacionada con los microcontroladores
log_error	En esta tabla se almacenan los errores producidos en la ejecución de la aplicación
usuarios	En esta tabla están almacenados los datos de acceso del usuario

Tabla 1. Base de datos relacional, Entidades

El modelo Entidad-Relación, es una herramienta a utilizar para modelar datos y que ayuda a representar las entidades en la base de datos y sus relaciones.

El modelo de datos consta de una colección de objetos relacionados entre sí a través de unos atributos, los componentes que lo conforman son los siguientes:

Entidades: representa un objeto del mundo real, se representa por sus atributos o características, puede representar un objeto físico, como lo es una persona, o no, como puede ser un puesto de trabajo, en un diagrama entidad-relación se representan mediante un rectángulo.

Atributos: son las características o propiedades que definen a una entidad, por ejemplo, a una entidad persona la caracteriza su dni, edad, nombre..., en un diagrama entidad-relación se representan mediante una elipse.

Relación: representa el cómo las entidades se relacionan o asocian entre sí, en un diagrama entidad-relación se representan mediante un rombo.



En la siguiente imagen se muestran las tablas, las relaciones entre ellas y la cardinalidad de las mismas dentro del diagrama Entidad-Relación.

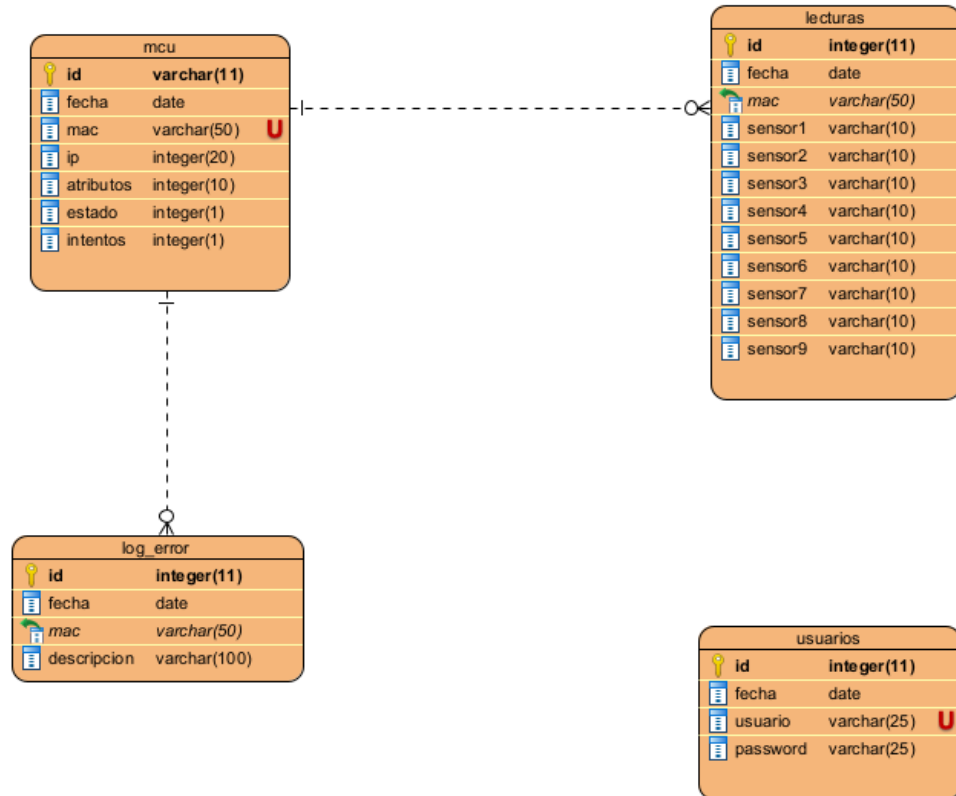
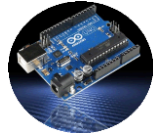


Figura 9. Diagrama Entidad-Relación

Todas las relaciones y demás información de las diferentes tablas se muestran en el **Anexo III** que acompaña esta memoria.

2.3.2 Diseño arquitectónico

El diseño arquitectónico incluye los aspectos más significativos del proyecto que se desea diseñar, estos son tanto aspectos estáticos como dinámicos, la arquitectura del software es una descripción de los sistemas divididos en subsistemas más pequeños y las relaciones existentes entre ellos. En el presente proyecto se utiliza la arquitectura de tres capas que se describe en los puntos siguientes.



Arquitectura de tres capas

La arquitectura de tres capas se encuentra dividida en tres capas como se observa en la figura de abajo, la primera de ellas es la capa de presentación, la cual se ocupa de que el usuario interactúe con el sistema, representa la interfaz de la aplicación para dispositivo móvil con la cual el usuario interactúa con el sistema.

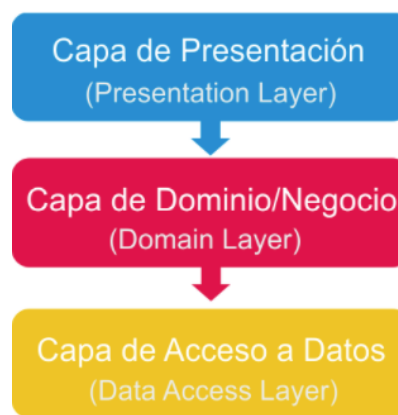


Figura 10. Arquitectura de tres capas

La segunda capa es la de negocio realiza las operaciones que deben cumplirse tras haberlas solicitado el usuario a través de la capa de presentación, como partes principales figuran las siguientes:

- **Código PHP:** Este código se ejecuta en el servidor apache, carecen de interfaz de usuario, se ocupan de recibir y enviar órdenes desde la aplicación móvil.
- **Código en el microcontrolador:** El software instalado en el hardware del sistema, el microcontrolador, es el encargado de tomar lecturas de los sensores, enviarlas para almacenarlas en la base de datos, también se encarga de activar los actuadores.

La última capa, la de datos, es la encargada de almacenar los datos de la aplicación en la base de datos de la aplicación, entre otros datos almacena las lecturas de los sensores y datos de acceso a la aplicación.



Arquitectura física

- **Diagrama de componentes**: El diagrama de componentes es uno de los diagramas más importantes dentro del UML (*lenguaje unificado de modelado*), utilizado para representar de forma estática el sistema de información, proporciona una vista de alto nivel de los componentes que forman parte del sistema.
- **Diagrama de despliegue**: El diagrama de despliegue es otro de los diagramas utilizados en UML utilizados para mostrar las relaciones físicas entre el hardware y el software que conforman el sistema.

Arquitectura vista estática

Describe qué componentes tiene la arquitectura.

- **Diagrama de paquetes**: El objetivo de los diagramas de paquetes es obtener una visión más clara del sistema a desarrollar, se agrupan los elementos análisis y diseño formando paquetes y estableciendo relaciones entre ellos.

Arquitectura vista dinámica

Describe cómo se comportan los componentes a lo largo del tiempo y como interactúan entre sí.

- **Diagrama de secuencia**: Muestra la interacción de un grupo de objetos pertenecientes a un sistema software a través del tiempo.
- **Diagrama de actividad**: Un diagrama de actividad es utilizado para mostrar el punto de vista dinámico de los sistemas software modelando el flujo entre actividades.



2.4 Diseño de la Interfaz

Para el diseño de la interfaz se comenzó pintando simples bocetos como el de la imagen de abajo, de forma manual para después con la herramienta software, Android Studio, llevarles al diseño final.

Este diseño únicamente se utiliza para la aplicación móvil de usuario, que realmente es la única que requiere de dicha interfaz, se realizaron diseños claros, simples y concisos, abajo se muestra una imagen de un boceto y la misma imagen diseñada con la herramienta software.

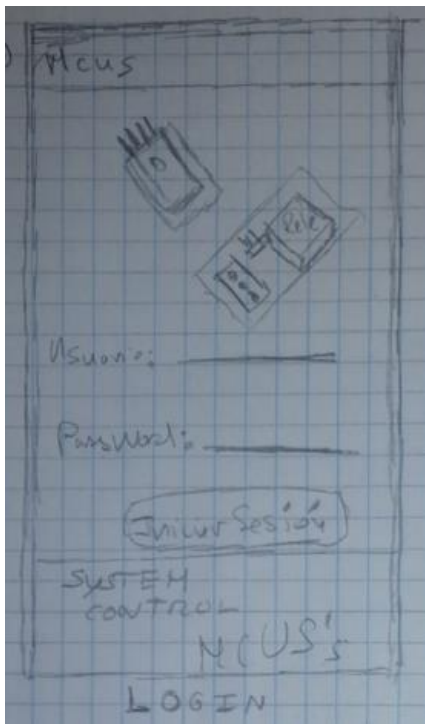


Figura 12. Boceto pantalla de login

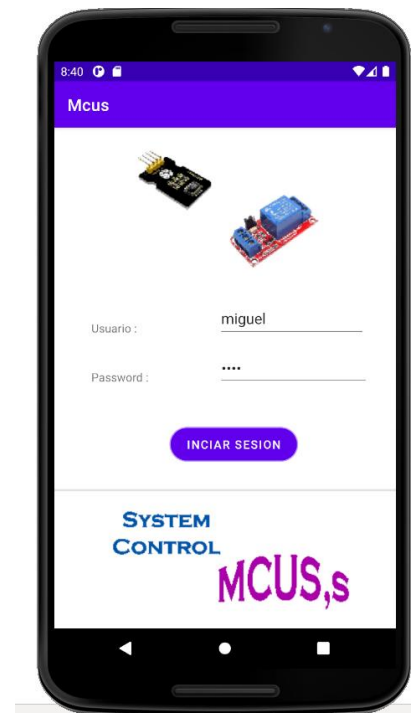


Figura 11. Diseño pantalla de login



3 MATERIAL

En los próximos apartados se describen todos los componentes hardware utilizados en el presente proyecto, así como el software desarrollado para el correcto funcionamiento de las diferentes partes que conforman el proyecto.

3.1 Hardware

Se describen a continuación las características de los dispositivos utilizados, comenzando por el microcontrolador y a continuación los sensores y actuadores. En las primeras etapas del presente trabajo, se llevó a cabo un estudio del hardware y software a utilizar. En el **ANEXO V** se describen al detalle cada uno de los elementos hardware utilizados y sus conexiones.

Dentro de la familia Arduino existen diferentes modelos, se eligió el Mega 2560 debido a varias razones, entre ellas, el número de pines que dispone tanto analógicos como digitales, la capacidad de la memoria Flash y EEPROM es superior a la de versiones anteriores, también debido a otras funcionalidades que en este proyecto no son utilizadas, como es la comunicación con impresoras 3D.

3.1.1 Arduino Mega 2560

La placa Arduino utilizada para el proyecto es modelo Mega 2560 [1], basado en el microcontrolador ATmega2560, probablemente es una de las mejores placas Arduino del mercado, el modelo corresponde al de la figura.

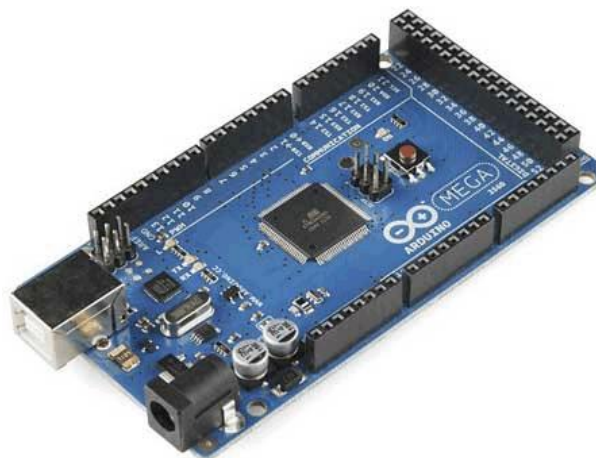


Figura 13. Arduino Mega 2560



Las principales características son las siguientes:

- Velocidad en su reloj: 16 MHz
- Voltaje de Trabajo: 5V
- Voltaje de Entrada: 7 a 12 V
- Voltaje máximo de entrada: 6 a 20 V
- Número de pines digitales: 54
- Numero de pines para PWM: 15
- Pines analógicos: 16
- Corriente aceptada en cada pin: 20 mA
- Corriente en el pin 3.3V: 50 mA
- Memoria Flash: 256 Kb
- Memoria EEPROM: 4 Kb
- 4 puertos serial
- Puerto SPI para conexiones con otros dispositivos como Ethernet shield
- Bus I²C para conectar diferentes dispositivos
- Microcontrolador ATmega2560

Dentro de la placa tenemos la memoria en la cual se almacena el código Arduino para su ejecución, la CPU ejecuta las instrucciones del código del programa almacenado y los pines de entrada/salida envían órdenes a los sensores y reciben los valores obtenidos.

3.1.2 Ethernet shield

El módulo Arduino Ethernet shield [2] que se aprecia en la figura, es utilizado por la placa Arduino para establecer conexión con Servidor Web a través de internet.

Este módulo está basado en el chip Wiznet W5100, para obtener una dirección IP para establecer conexiones TCP y UDP.

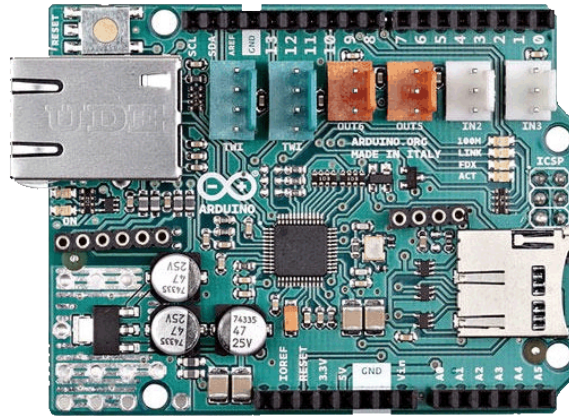


Figura 14. Ethernet shield.

- Dispone de una conexión RJ-45 para conectar la clavija a la red.
- Utiliza los pines digitales 10, 11, 12, y 13 (SPI) para comunicarse con el W5100 en la Ethernet shield
- También este módulo dispone de un lector de tarjeta micro-SD, la cual ser utilizada para enviar y recibir datos, configurar conexión...
- Arduino utiliza la biblioteca Ethernet para establecer conexiones y la biblioteca SD para utilizar la micro-tarjeta
- Sistema de encriptación soportados WEP y WPA2
- leds de estado
- Velocidad de conexión de 10 a 100MB
- Voltaje de alimentación 5V

3.1.3 Módulos sensores

A continuación, se detallan los diferentes sensores utilizados en el proyecto, estos módulos trabajan para obtener lecturas, las ordenes para tomar dichas lecturas on enviadas por el Arduino.



3.1.3.1 Sensor de intensidad lumínica

LDR es una fotorresistencia que es sensible a la luz y ofrece más o menos resistencia dependiendo de la intensidad de luz recibida.

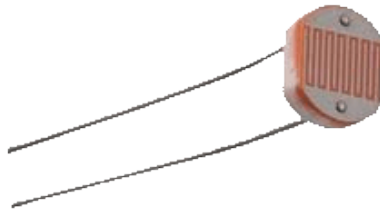


Figura 15. LDR

3.1.3.2 Sensor ambiental SPI BME280

El módulo BME280 es un chip que se utiliza para obtener el valor actual de la presión atmosférica, humedad, temperatura y altitud [3], este tipo de sensores pueden ser utilizados para calcular la altitud con gran precisión, por lo que es un sensor muy utilizado en sistemas de Autopiloto para Drones. Este sensor puede entregar medidas de altitud con una precisión de hasta 1m, el modelo utilizado es el que se muestra en esta figura, se puede observar los 6 pines de los que consta.

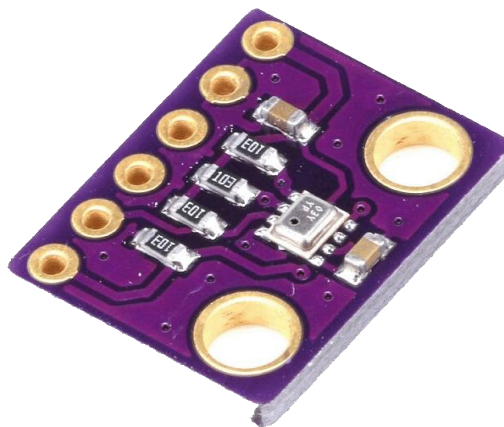


Figura 16. Sensor ambiental BME280.



3.1.3.3 Sensor de calidad del aire CCS811

El sensor de calidad del aire CCS811 obtiene mediciones de Compuesto Orgánico Volátil Total (TVOC) y una lectura de Dióxido de Carbono Equivalente (eCO₂) [4].

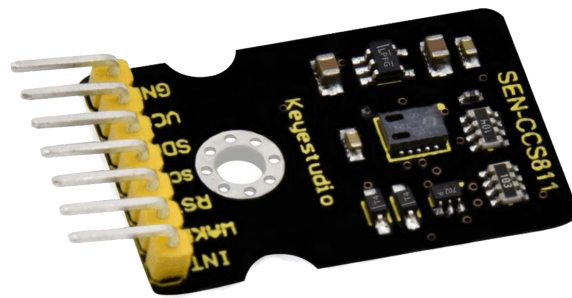


Figura 17. Sensor calidad del aire CCS811

El dióxido de carbono equivalente (eCO₂) se mide en el rango de 400 a 8192 ppm (partes por millón), y varios compuestos orgánicos volátiles (TVOC) varían de 0 a 1187 ppb (partes por billón), también dispone de un termistor para medir la temperatura.

3.1.3.4 Sensor GPS GY-NEO6MV2

Este módulo localizador GPS [5], encapsula el transceptor GPS NEO-6, Incluye una memoria Eprom para guardar la configuración cuando se apaga, una batería de backup y la antena es cerámica. Los GPS NEO-6 disponen de interface de comunicación Uart, Spi, i2c y USB.



Figura 18. Sensor GPS GY-NEO6MV2

En la figura 10, se aprecian dos componentes bien diferenciados, el propio módulo gps y la placa cerámica que corresponde a la antena del sensor.

3.1.3.5 Relé

Los actuadores son dispositivos activados por el sistema Arduino, cada actuador lleva a cabo una determinada función, en este proyecto se utilizar un grupo de 4 relés.

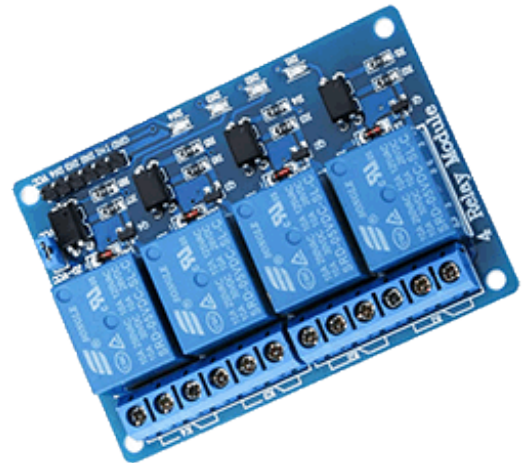


Figura 19. Módulo de 4 relés

Un relé es un dispositivo electrónico que funciona como un interruptor, es activado mediante una señal eléctrica enviada a través de una línea de salida del sistema Arduino. Consta de una bobina y un electroimán que activa un contacto para abrir o cerrar el circuito eléctrico.

En este proyecto se disponer de 4 relés, cada relé activa otro sistema como puede ser un motor que lleve a cabo una determinada función.



3.1.3.6 Pantalla Oled

El tamaño es de 128 x 64 pixeles oled de colores amarillos y azules, funciona en ambientes oscuros, sin luz de fondo [6], la pantalla se puede conectar a cualquier dispositivo que admita comunicación I²C. En la figura de abajo, se observan a simple vista los 4 pines de los que dispone.



Figura 20. Pantalla Oled

3.1.3.7 Conjunto de leds

Un led o diodo led, es un dispositivo que transforma la corriente eléctrica en luz al ser polarizado, dicha luz puede ser de diferente color, no emite calor por lo tanto no sube la temperatura.

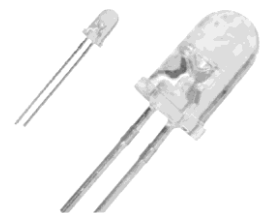


Figura 21. Leds

El color depende del material inorgánico utilizado en la fabricación del led.



3.1.3.8 Protoboard

Las placas protoboard o placa de pruebas son unos objetos utilizados en electrónica para conectar los diferentes dispositivos. A los lados y con los símbolos (+ -) se pinchan los cables que llevan la alimentación al circuito, estas líneas tienen la misma tensión de forma horizontal, están conectados internamente, mientras que en el centro de la placa la tensión circula en cada línea de forma vertical.

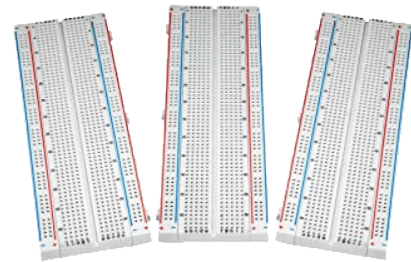


Figura 22. Protoboard

3.1.3.9 Fuentes de alimentación

En el proyecto se utilizan varias fuentes de alimentación, en primer lugar, para realizar pruebas se utiliza el cable USB conectado entre el ordenador y la placa Arduino.

Esta fuente es suficiente para alimentar todos los componentes, pero únicamente en modo de pruebas y programación.

Fuente de alimentación MB12:

Proporciona 5v y 3.3v respectivamente, dispone de un interruptor, como se puede apreciar en la figura siguiente, para apagar y encender la fuente, dispone de varios conectores para seleccionar el voltaje de salida, se puede alimentar mediante USB o mediante una batería cuyo voltaje de entrada varía entre 6.5v y 12v de corriente continua.

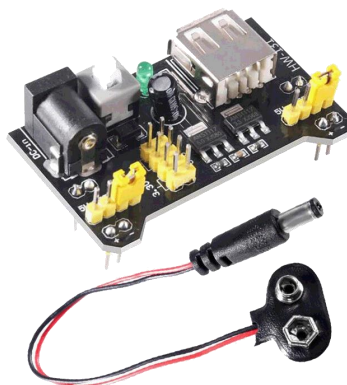


Figura 23. Fuente de alimentación BM12

3.1.3.10 Cables y resistencias

Los cables tienen 15 cm de longitud, no necesitan soldaduras entre la placa protoboard, los componentes y Arduino.

Los hay macho-hembra, macho-macho, hembra-hembra y de diferentes colores.

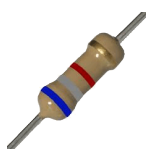


Figura 24. Resistencia

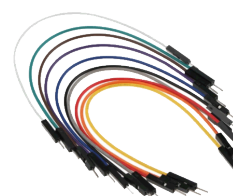


Figura 25. Cables

Las resistencias son componentes pasivos utilizados en electrónica a modo de protección de diferentes partes del sistema para evitar sean dañados los diferentes módulos debido a sobrecargas eléctricas no deseadas, se oponen al paso de los electrones, eso supone una transformación de la energía en calor.



3.2 Software

En este apartado, se describen las diferentes componentes y herramientas software utilizadas en el sistema. Existen tres bloques de software bien diferenciados, son el bloque correspondiente a la programación del microcontrolador Arduino en primer lugar, el bloque de programación en el servidor apache y por último la programación del dispositivo móvil para controlar toda la operativa. Para cada uno de estos bloques se utilizar herramientas para desarrollar software y para su prueba e instalación en el destino final.

3.2.1 IDE Arduino

Entorno de Desarrollo Integrado (IDE) [7] para Arduino, se aprecia en la figura siguiente, es una aplicación informática desarrollada en lenguaje JAVA utilizada para el microcontrolador Arduino.

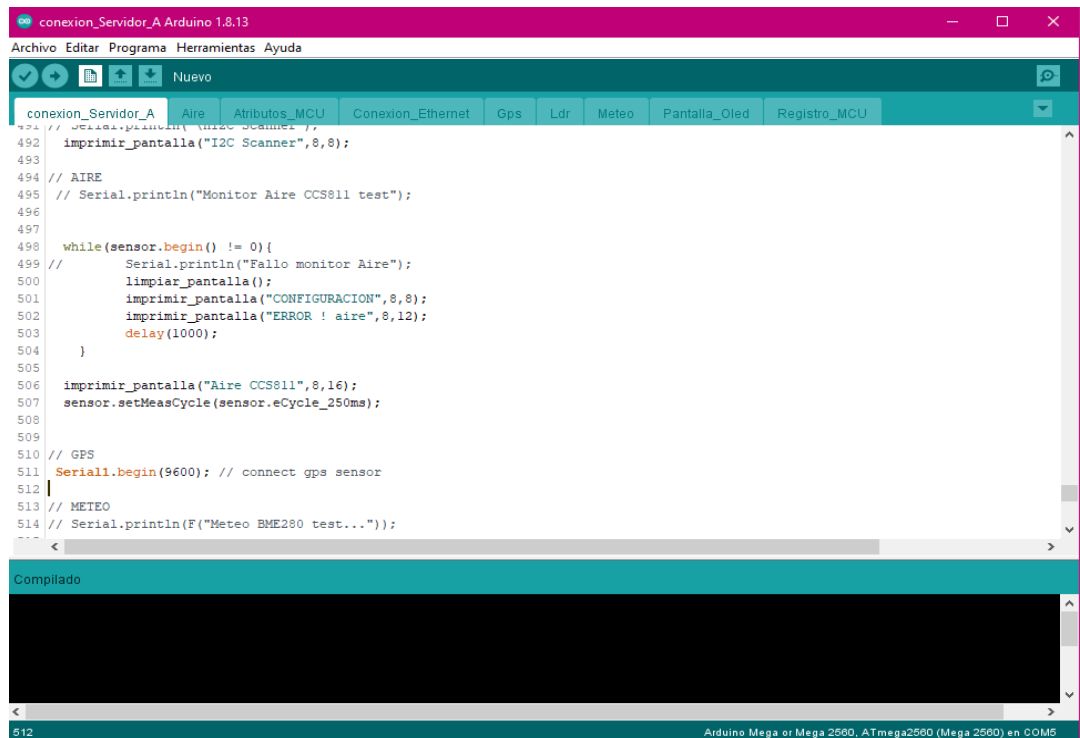
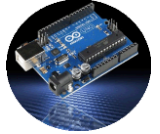


Figura 26. IDE Arduino

El programa escrito en C o C++, es traducido a lenguaje hexadecimal por parte del programa **avr-gcc**, que es un compilador para microcontroladores AVR, una vez compilado, es enviado a la placa a través del programa **avrdude**, a la vez que se compila se guarda; El código fuente del IDE está bajo Licencia Pública General de GNU, todo este



proceso se puede observar en la figura de má abajo. Entre las características principales figuran las siguientes:

- Detección automática de la placa Arduino conectada.
- Muestra información de la memoria Ram y Flash ocupada por el proyecto.
- Multiplataforma

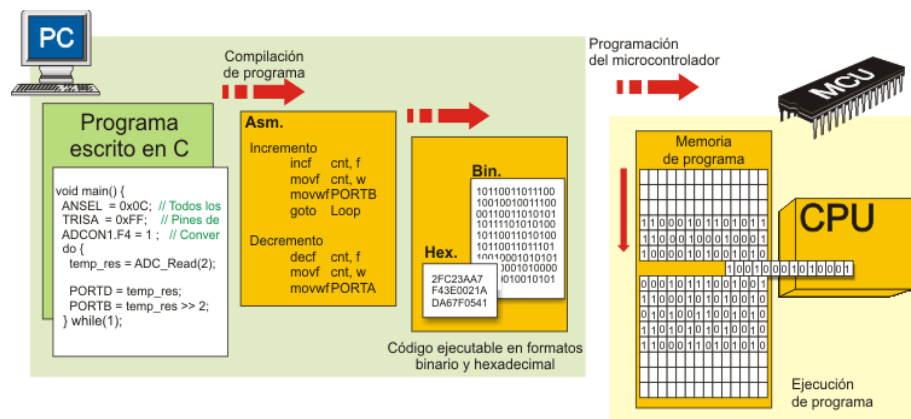


Figura 27. Compilación Sketch de Arduino

3.2.2 Notepad ++

NotePad Plus Plus [8] es un completo editor de texto plano para suplir en todos los aspectos al Bloc de notas tradicional, además de trabajar con texto sin formato, puede ser utilizado como editor de código para programadores compatible con más de 50 lenguajes.



```
1 |;Copyright (C) 2004-2009 John T. Haller
2 |
3 |;Website: http://PortableApps.com/Notepad++Portable
4 |
5 |;This software is OSI Certified Open Source Software.
6 |;OSI Certified is a certification mark of the Open Source Initiative.
7 |
8 |;This program is free software; you can redistribute it and/or
9 |;modify it under the terms of the GNU General Public License
10 |;as published by the Free Software Foundation; either version 2
11 |;of the License, or (at your option) any later version.
12 |
13 |;This program is distributed in the hope that it will be useful,
14 |;but WITHOUT ANY WARRANTY; without even the implied warranty of
15 |;MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
16 |;GNU General Public License for more details.
17 |
18 |;You should have received a copy of the GNU General Public License
19 |;along with this program; if not, write to the Free Software
20 |;Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.
21 |
22 |!define NAME "Notepad++Portable"
23 |!define PORTABLEAPPNAME "Notepad++ Portable"
24 |!define APPNAME "Notepad++"
25 |!define VER "1.6.8.0"
```

Figura 28. Editor de texto Notepad ++

3.2.3 Desarrollo para Android

Para seleccionar la plataforma con la cual desarrollar la para el dispositivo móvil, se estudiaron dos plataformas, **Android Studio** y **Xamarin**, que en la actualidad son muy utilizadas para diseñar software en dispositivos móviles que es lo que se requiere para este proyecto.

3.2.3.1 Android Studio

Para el desarrollo de la aplicación para el dispositivo móvil, se utilizó Android Studio [9], es un IDE enfocado a implementar software para Android, pertenece a Google desde el año 2005, basado en el software IntelliJ IDEA de JetBrains [10].

Existen varias versiones del IDE, la utilizada para este proyecto es la Android Studio Arctic Fox 2020.3.1 Canary 15, como se puede observar en la imagen siguiente.



Figura 29. Android Studio versión

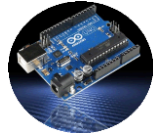
Para desarrollar los proyectos con Android Studio, se pueden utilizar varios lenguajes de programación, el último en ser incorporado es del Kotlin, en el año 2019, admite lenguajes como C++ o JAVA entre otros, el utilizado para el presente proyecto fue JAVA.

Características:

- Diseño de interfaz de usuario rápido con Jetpack Compose.
- Compilación automática de código basado en Gradle.
- ProGuard utilizado para ocultar el código y reducir el tamaño de la APK.
- Integra herramienta Lint para detectar problemas de rendimiento, usabilidad, compatibilidad de versiones.
- Soporte para programar aplicaciones para Android Wear.
- Un dispositivo virtual de Android que se utiliza para ejecutar y probar aplicaciones para diferentes modelos de dispositivos.
- Posibilidad de emparejar dispositivos físicos mediante Wifi.
- Renderizado de imágenes en tiempo real.



Figura 30. Logo Android Studio



Cada proyecto desarrollado con Android Studio tiene tres bloques principales, considerando un proyecto desarrollado en el lenguaje Java:

- **Carpeta Java:** todo el código fuente de la aplicación se encuentra en la carpeta Java, pueden ser clases creadas con un propósito específico, fragment, activity, etc. Dentro de la carpeta Java puede haber otras subcarpetas para la buena organización de la App en general.
- **Carpeta Res:** en esta carpeta se encuentran los recursos utilizados en el proyecto como pueden ser, documentos xml, imágenes, sonidos, textos, videos, estilos,...

Los documentos xml se encuentran situados en la subcarpeta Layout, están asociados por lo general a los fragment y a las activities, contienen todos los elementos del diseño de la aplicación, definen la interfaz con la cual interactúa el usuario. Cuando se crea un activity o un fragment, automáticamente se crea un fichero xml en blanco.

- **AndroidManifest.xml:** en este documento se incluyen las características principales características del proyecto, como permisos de acceso a internet, nombre del proyecto, imagen de icono de la aplicación, activity principal de inicio...etc.

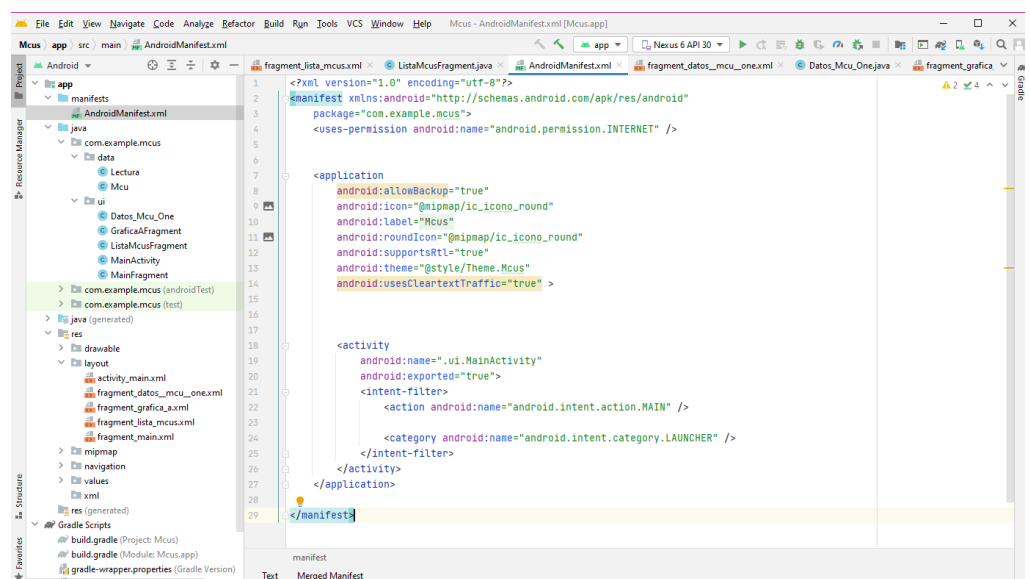


Figura 31. Configuración del fichero AndroidManifest.xml



3.2.3.2 Librerías Jetpack

Jetpack [11] es un conjunto de librerías que ayudan a desarrollar aplicaciones más eficientes, que aprovechan más eficientemente la potencia del lenguaje y que dichas aplicaciones corran en una mayor gama de dispositivos físicos.

Para usar la biblioteca Jetpack en cualquier proyecto debemos incluir todos los componentes que encuentran en el repositorio *Maven de Google*. En la carpeta Gradle Scripts hay dos archivos `build.gradle`(Project) y se añaden las líneas que figuran en la imagen.

```

1 // Top-level build file where you can add configuration options common to all sub-projects/module
2 buildscript {
3     repositories {
4         google()
5         mavenCentral()
6         maven { url 'https://maven.google.com' }
7         maven { url "https://jitpack.io" }
8     }
9 }
10
11
12 dependencies {
13     classpath "com.android.tools.build:gradle:7.0.0-alpha15"
14     classpath "org.jetbrains.kotlin:kotlin-gradle-plugin:1.5.0"
15     def nav_version = "2.3.5"
16     classpath "androidx.navigation:navigation-safe-args-gradle-plugin:$nav_version"
17 }
18
19

```

Figura 32. Configuración `build.gradle` (App)

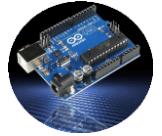
Para agregar dependencias de Jetpack Compose se edita el fichero `build.gradle` (App) y se añaden las líneas que se requieran para poder utilizar las librerías en el diseño del proyecto. Las líneas con `androidx` son las pertenecientes a JectPack.

```

dependencies {
    implementation 'androidx.appcompat:appcompat:1.2.0'
    implementation 'com.google.android.material:material:1.3.0'
    implementation 'androidx.constraintlayout:constraintlayout:2.0.4'
    implementation 'androidx.navigation:navigation-fragment:2.3.5'
    implementation 'androidx.navigation:navigation-ui:2.3.5'
    implementation 'androidx.legacy:legacy-support-v4:1.0.0'
    testImplementation 'junit:junit:4.+'
    androidTestImplementation 'androidx.test.ext:junit:1.1.2'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.3.0'
}

```

Figura 33. Configuración `build.gradle` (Project)



Las principales ventajas al utilizarlas librerías Jecpack Compose son las siguientes:

- Menos código en los proyectos y realizando más tareas, el código generado contiene menos errores internos es más simple y fácil de mantener.
- Acelera el desarrollo, ya que es compatible con cualquier código ya existente, se puede adaptar fácilmente.
- Posibilidad utilizar diseños interactivos, ágiles y fácilmente mantenible. Se reducir al mínimo el número de activitys, con una solamente y utilizando bloques fragmet se pueden diseñar aplicaciones agilizando los ciclos de vida que tienen las activity.

Jetpack se involucra en cuatro áreas de suma importancia en Android, la arquitectura, la interfaz de usuario, el comportamiento y el fundamento.



Figura 34. Android Jetpack



3.2.4 Filezilla

Filezilla [12] es un cliente FTP gratuito y fácil de usar, utilizado para subir los archivos PHP al servidor apache. El primer es descarga el programa desde la web oficial de Filezilla, una configurado el programase debe añadir un sitio nuevo para configurar los datos de acceso al servidor apache.

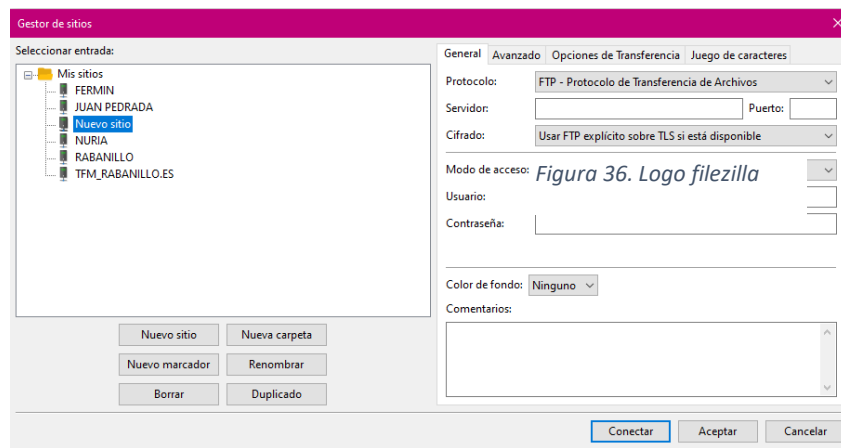


Figura 35. Configuración sitio en Filezilla

Funcionalidades destacadas:

- **Administrar sitios:**

Sección en la cual se gestionan los diferentes sitios FTP, se pueden añadir, modificar y suprimir sitios conectarse/desconectarse según se desee sin necesidad de hacer login.

- **Registro de mensajes:**

Sección que tiene Filezilla para mostrar mediante mensajes las transacciones llevadas a cabo con cada sitio ftp, mostrando las respuestas recibidas por parte del servidor. Ej. éxito, error de login, transferencia completa, etc.



- Cola de transferencia:

En la parte de debajo de la pantalla se sitúa una sección que muestra información en tiempo real de las tareas que está realizando el programa en un momento determinado, indicando los ficheros subidos y los que quedan por subir, tiempos y demás. Se pueden pausar las subidas y reanudarlas más tarde.

Estado:	Conexión TLS establecida.
Estado:	Registrado en
Estado:	Comenzando la subida de C:\Users\MiguelAngel\Documents\TRABAJO\UNIVERSIDAD_DE_SALAMANCA\MASTER_ING_INFORMATICA\TFM\PHP\sensores\login.php
Estado:	Transferencia correcta, transferidos 407 bytes en 1 segundo
Estado:	Recuperando el listado del directorio "/httpdocs/sensores"...
Estado:	Directorio "/httpdocs/sensores" listado correctamente

Figura 37. Progreso de subida de archivos con Filezilla

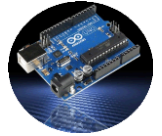
3.2.5 Dominio en internet

El primer paso para poder alojar los archivos necesarios para el proyecto en internet, fue adquirir un dominio en internet, el nombre del dominio es “**rabanillo.es**”, se adquirió a una empresa que ofrece este tipo de servicios a través de su página web, la empresa es Domiteca [13] y la web es **www.domiteca.com**.

The screenshot shows the Domiteca website interface. At the top, there is a navigation menu with links: INICIO, BUSCAR, MI PANEL, TRANSFERIR, RESELLERS, ayuda, empresa, and contacto. Below the navigation is a search bar with the text 'WWW.dominio.es' and a 'Buscar' button. A large banner advertises 'OFERTA DE LANZAMIENTO DOMINIO.CO EL DOMINIO MÁS EXCLUSIVO' with a price tag of '24.90€'. On the right side, there is a section for 'ACCESO A MI PANEL' and a list of 'Dominios TOP' including .com, .net, .org, .tv, .co, and .me with their respective prices.

Dominio	Precio
.com, .net y .org...	11.90 €
.es	11.90 €
.org	11.90 €
.tv	50 €
.co	24.95€
.me	24.95€

Figura 38. Web domiteca.com



El primer paso fue buscar el dominio para comprobar su disponibilidad a través del buscador de dominios, una vez comprobado que estaba disponible, se procedió a comprar dicho dominio previo registro en la página web de Domiteca.



Figura 39. Buscador de dominio

3.2.6 Servidor

El servicio de hosting se utiliza para alojar los archivos utilizados por la aplicación Android y por el sistema Arduino para la gestión de todo el proyecto, aparte la base datos en la cual se almacenan los datos. El hosting es contratado a la empresa Interdominios [14], se trata de un hosting sin límites con sistema operativo Linux, las principales características son las siguientes:

Espacio en disco	Características
Alojamiento dominios	Sin límites
Cuentas de email	Sin límites
Redirección cuentas de email	Sin límites
Alias de email	Sin límites
Parking de dominios	Sin límites
Subdominios	Sin límites
Usuarios web	Sin límites
Soporte IMAP	
Privacidad de dominios	
Cuentas FTP	
Base de Datos MySQL	
Librerías CGI-BIN	
Acceso FTP	
Servidor Apache	
Soporte .htaccess	
Webmail	
Protección AntiSpan	
PHP, Javascript, Asp	

Tabla 2. Tabla de características del Hosting



El hosting ofrece servicio de alojamiento web principalmente, en nuestro caso no se necesita ninguna página web para visualizar en ningún navegador, requiere de apache como servicio, únicamente se creará un espacio en el cual almacenamos los archivos de aplicación utilizando el dominio *rabanillo.es*.

El lenguaje de programación utilizado es, PHP ejecutándose bajo el servidor apache que se ejecuta en el servidor para que puedan ser interpretados archivos PHP

, que nos permite programar las tareas necesarias en el sistema, otro servicio que ofrece el hosting utilizado es FTP, protocolo de transporte de ficheros, que nos permite subir los archivos de programa. Para almacenar los datos se utiliza el servidor de base de datos MySQL.

3.2.7 Gestión dominio y hosting

Para poder utilizar el dominio en el presente proyecto, y en la aplicación Android o a través de Arduino, debemos de gestionar las **DNS** del dominio.

Los equipos conectados a internet se comunican unos con otros a través de la IP que es un número que identifica a cada, pero es más fácil acordarse de un nombre que de un número para llamar a un determinado equipo, para ello se creó el sistema de nombre de dominios, de esta forma cada nombre de dominio se traduce en un IP.

En este proyecto, Interdominios facilita una IP para poderla asignar al dominio y **Domiteca**, da la opción de dar al dominio “**rabanillo.es**” unas **DNS** para poder ser utilizado en internet y acceder a los documentos almacenados en el espacio de dominio, pues únicamente lo que hacer es asignar IP al dominio según se indica en la figura de la siguiente página.

Para alojar los archivos utilizados por el proyecto, se utilizó un sitio web creado para el dominio, *rabanillo.es*, para este propósito, fue creada una carpeta de archivos mediante el panel de gestión que ofrece el servidor llamado Plesk [14] , con el panel de gestión se puede configurar todo lo relacionado con el dominio, se pueden crear “sitios” nuevos para otros dominios, se pueden crear carpetas de archivos, subir archivos, crear cuentas de usuario, de ftp y también crear la base de datos con sus respectivas tablas, recordemos que la base de datos es MySQL y se gestiona mediante su propio panel de gestión denominado phpMyAdmin.



Para nuestro trabajo, se creó una carpeta de archivos llamada “**sensores**”, en la cual se alojan los archivos de la aplicación, dicha carpeta se encuentra situada dentro del directorio principal llamado httpdocs, todos los documentos fueron subidos mediante el programa de FTP Filezilla.

> SERVIDORES DNS: rabanillo.es

Usar nuestros DNS

Si deseas utilizar nuestros servicios de redireccionamiento y la gestión avanzada DNS, deberás de elegir esta opción. De otro modo, dichos servicios no funcionarán.

Usar los siguientes DNS:

Introduce los NOMBRES de los servidores DNS que deseas usar para este dominio. Escribe un mínimo de DOS servidores DNS.

NO INTRODUZCAS LAS IPs, sólo los nombres.

1.-

2.-

3.-

4.-

5.-

6.-

 * Nota: los dominios .ES podrían tardar hasta 1 hora en actualizarse desde que envíes este formulario

Figura 40. Gestión DNS del dominio

Toda la gestión del dominio realiza a través del panel Plesk facilitado por el servicio de hosting, Plesk no es más que una aplicación instalada en el servidor que nos permite administrar nuestro dominio en internet, nos permite realizar entre otras las siguientes tareas:

- Crear usuarios web
- Crear alojamientos web para dominios
- Crear bases de datos
- Gestionar usuarios FTP
- Gestionar directorios
- Control de niveles de seguridad en directorios

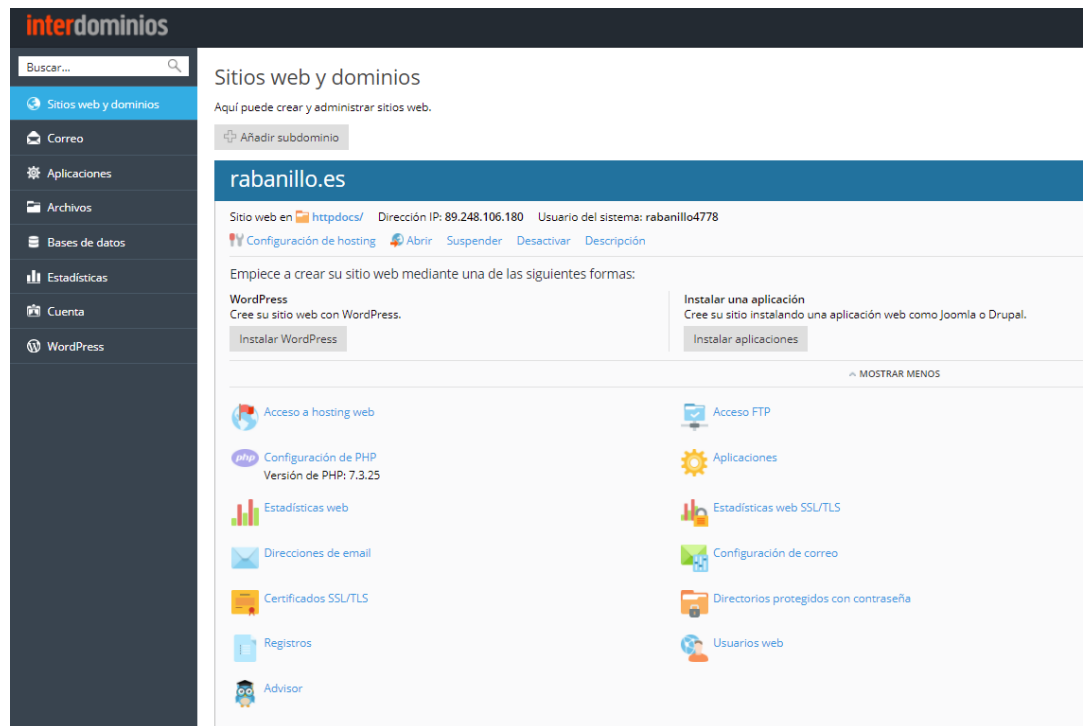


Figura 41. Plesk - gestión del sitio

Los archivos PHP del proyecto, son utilizados por el software ejecutado en la placa Arduino conectada al sistema y por la aplicación Android instalada en el un dispositivo móvil.

La placa Arduino utiliza el código PHP para las siguientes funciones:

- Envío de datos leídos por los sensores.
- Control de permisos de estado On/Off de cada sistema Arduino.
- Gestión para cada sensor conectado a la placa de su estado On/Off.
- Envío de errores de lectura/escritura.

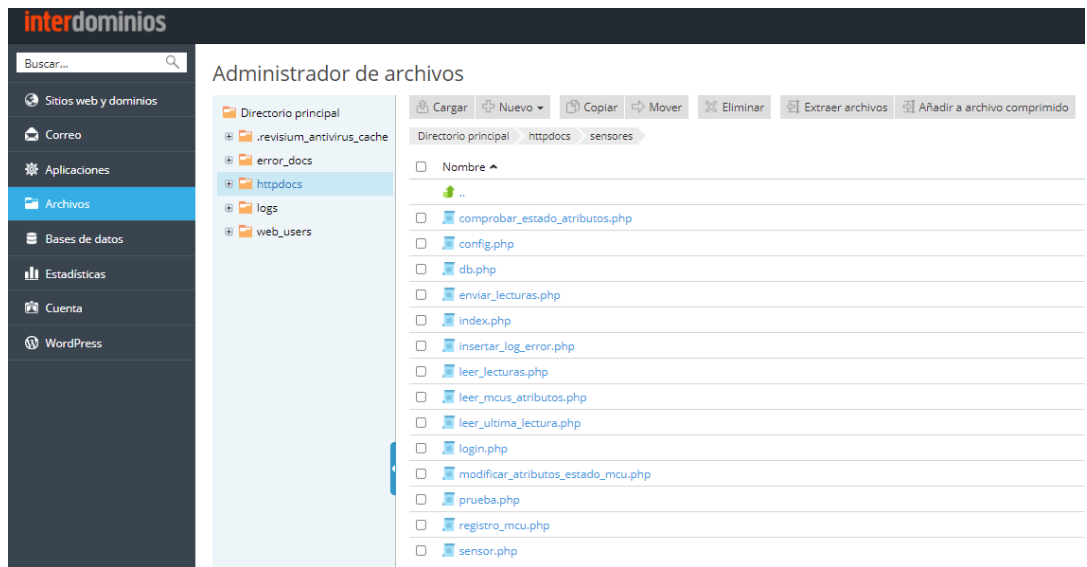


Figura 42. Sistema de archivos

La aplicación Android se conecta al servidor apache haciendo uso del código PHP para las siguientes tareas:

- Hacer login de conexión.
- Ver estados de los diferentes sistemas Arduino conectados.
- Activar desactivar los sistemas Arduino conectados.
- Activar/desactivar sensores de los diferentes sistemas Arduino
- Obtener últimas lecturas y gráficas de lecturas de sensores.



3.2.8 PHP

¿Qué es PHP (Hypertext Preprocessor)?

Es un lenguaje de programación, que se ejecuta en el lado del servidor para ser enviado a la petición por parte del cliente, se puede incrustar dentro del código HTML de las páginas, dicho código se ejecuta al solicitar una llamada a una determinada página HTML.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Ejemplo</title>
  </head>
  <body>

    <?php
      echo "¡Hola, soy un script de PHP!";
    ?>

  </body>
</html>
```

Figura 43. Ejemplo de PHP con HTML

El código PHP se encuentra delimitado por las etiquetas `<?php >` insertadas en un lugar cualquiera del código html.

Puede correr bajo cualquiera de los sistemas operativos como Windows, Linux, MacOS, entre otros muchos, permite el acceso a varios motores de base de datos como puede ser MySQL y servidores web como Apache.

¿Para qué se utiliza?

Este lenguaje está principalmente enfocado a la ejecución de scripts del lado del servidor, PHP es utilizado para ejecutar tres funcionales principales como las siguientes:

- Scripts del lado del servidor: acceso vía web a determinados servicios como puede ser acceso a una base de datos o cualquier otro tipo de proceso.
- Scripts ejecutados en línea de comandos: para hacer uso de esta modalidad no hace falta ningún navegado web, simplemente ejecutar el script.
- Aplicaciones de escritorio: para escribir este tipo de aplicaciones se suele utilizar PHP-GTK, que es una extensión para diseño de aplicaciones, orientada a la programación orientada a objetos, dispone de una interfaz gráfica.



¿Qué versión se utilizar?

La versión proporcionada por el servicio de hosting de Interdominios [18], es la 7.3.25

interdominios

Conectado como rabanillo.es
Suscripción rabanillo.es

Advisor Ayuda plesk

Buscar...

Sitios web y dominios

Configuración PHP para rabanillo.es

En esta página puede ver la configuración PHP para el sitio web. Estos parámetros de configuración son específicos para el sitio web y no afectan a los demás sitios web presentes en la suscripción. Es posible modificar la configuración PHP siempre que el proveedor de hosting le conceda el permiso correspondiente. Asimismo, puede personalizar dicha configuración por ejemplo para limitar el consumo de los recursos del sistema por parte de los scripts PHP o para cumplir los requisitos de una determinada aplicación web. Si define el valor predeterminado para un determinado parámetro, PHP utilizará el valor usado en la configuración PHP a nivel del servidor.

Usted puede usar los siguientes marcadores en los valores del parámetro:

- {DOCRROOT} para el directorio raíz de documentos de este sitio web. El valor actual es /var/www/vhosts/rabanillo.es/httpdocs. El directorio de la raíz de documentos HTTPS es el mismo que el de http.
- {WEBSPECROOT} para el directorio raíz de documentos del sitio web principal de la suscripción.
- {TMP} para el directorio que almacena archivos temporales.

Soporte para PHP (Versión de PHP 7.3.25) ejecutar PHP como Aplicación FastCGI Ver la página phpinfo()

A continuación puede ver la configuración PHP del sitio web. Puede cambiar la configuración PHP siempre que el proveedor de hosting le conceda el permiso correspondiente.

Descubra cómo escoger la configuración PHP más adecuada

Configuración de seguridad y rendimiento

memory_limit	128M (Predeterminado)
max_execution_time	30 (Predeterminado)
max_input_time	60 (Predeterminado)
post_max_size	8M (Predeterminado)
upload_max_filesize	2M (Predeterminado)
opcache.enable	on (Predeterminado)
disable_functions	opcache_get_status

Figura 44. Gestión PHP desde panel PLESK

3.2.9 MySQL

Gestor de base de datos relacionales, dispone de dos licencias una comercial y otra de código abierto (RBMS), utiliza el modelo cliente-servidor, propiedad de Oracle desde 2008.



Figura 45. Logo MySQL

MySQL es el motor de base datos más utilizado, utilizado principalmente en servicios Web, también se puede utilizar desde la consola o línea de comandos, esta forma de utilización suele ocurrir cuando no existe interfaz gráfica para utilizar phpMyAdmin a través de un navegador o cuando se accede a un servidor a través de un terminal remoto.

Sirve para almacenar información en tablas pertenecientes a base de datos relacionales, gestionadas por aplicaciones informáticas. MySQL tiene completa compatibilidad con el lenguaje de manejo de datos SQL, esto significa que cualquier otra base de datos que utilice este mismo lenguaje puede ser tratado con MySQL.



4 INSTALACION Y MONTAJE

La instalación y montaje se divide en dos partes, el montaje del hardware, con la conexión de los diferentes componentes que forma parte del sistema, en la segunda parte la instalación del software, en detalle se describen en el **Anexo VI**, aquí únicamente se muestran los detalles principales.

4.1 Montaje del hardware

Los componentes hardware utilizados en el proyecto son los que se nombran a continuación, para ver más en detalle en el **Anexo V** se describen todos y cada uno de los detalles de cada componente:

- Placas protoboard
- Microcontrolador Arduino
- Ethernet shield
- Diodos led
- Sensor GPS
- Sensor de CO₂
- Sensor del clima
- Pantalla Oled
- Fuente de alimentación
- Cables
- Resistencias
- Sensor LDR

La conexión de todos los componentes se describen en el anexo correspondiente, para mostrar todos los pasos se utilizó la herramienta **fritzing**, utilizada para mostrar todos los diseños, el diseño completo se observa en esta imagen de más abajo.

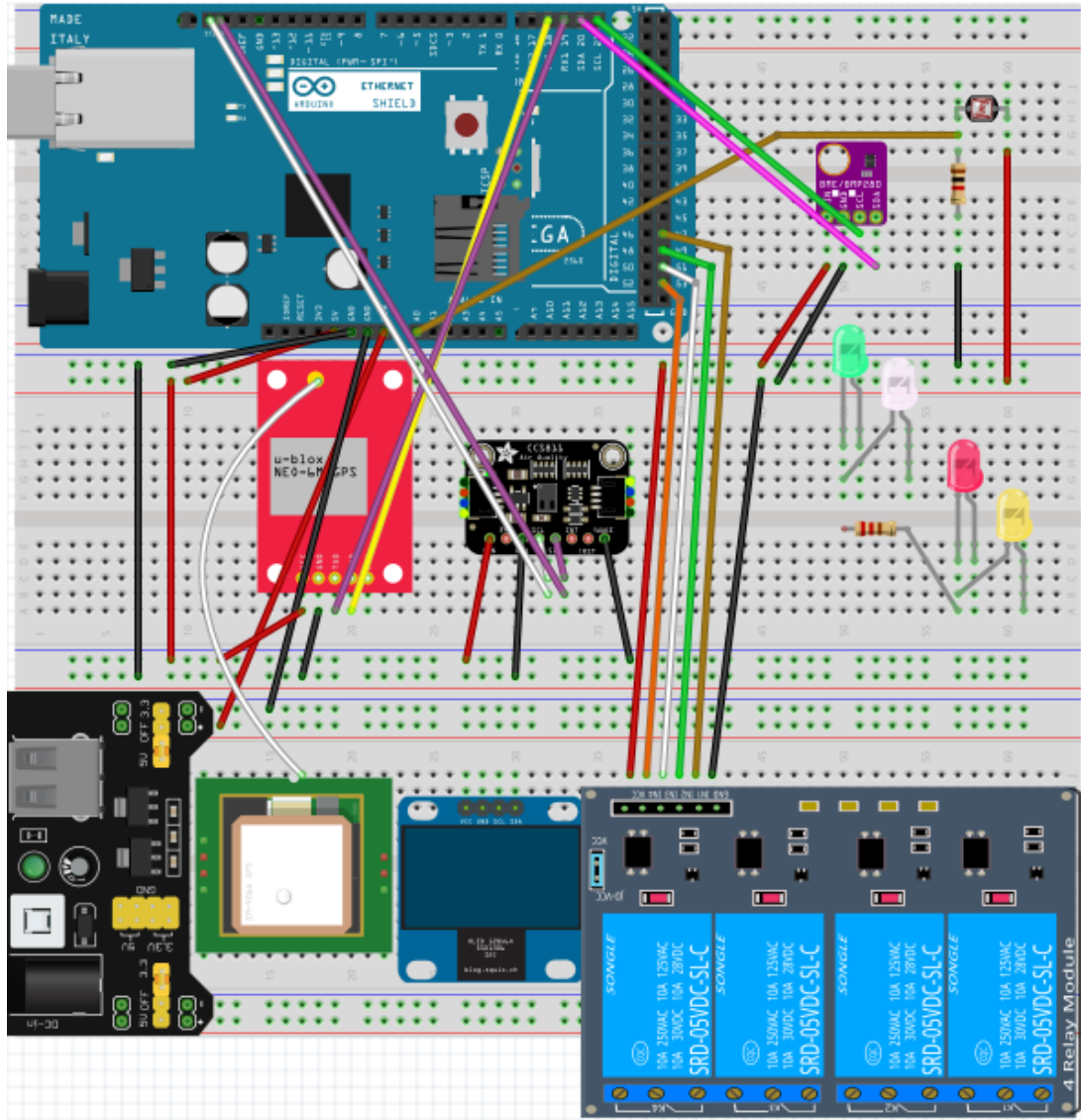


Figura 46. Diseño completo componentes hardware



La imagen real del diseño se observa en esta imagen de abajo

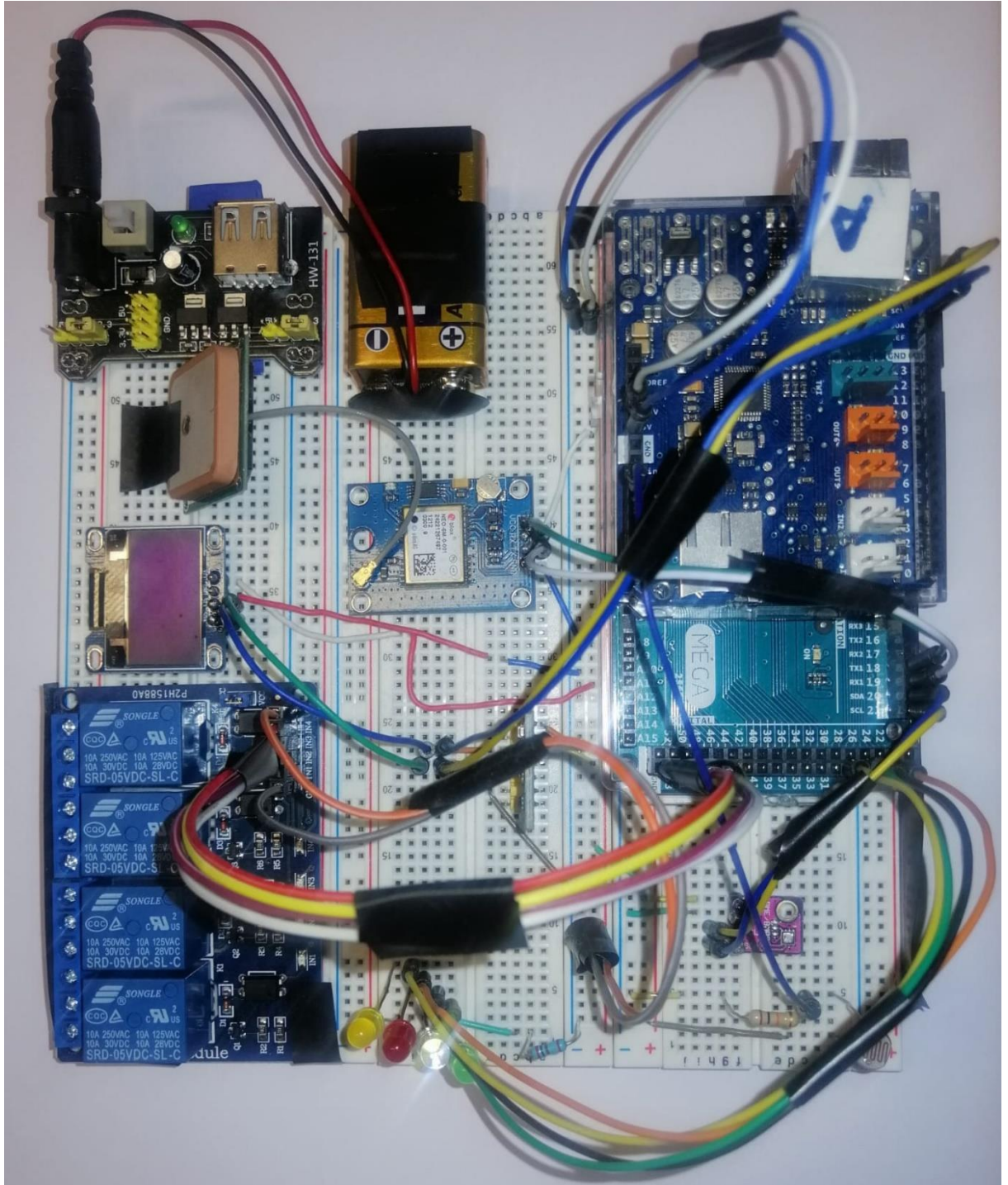


Figura 47. Diseño real del sistema



4.2 Instalación del software

Estos son los cuatro bloques principales de instalación de software, para ver en con más detalle la instalación ver el **Anexo VI**.

- Código PHP en servidor apache: Para instalar el código php se deben seguir varios pasos previos, entre ellos prepara el dominio utilizado, en este caso rabanillo.es, una vez configuradas las DNS del dominio y configurado el servidor, estaremos en disposición de subir los archivos PHP al servidor mediante una herramienta FTP, Filezilla es la herramienta utilizada.
- MySQL en servidor apache: Para crear la base de datos en el servidor, a través de la web se accede a la herramienta phpMyAdmin, con la cual se gestiona las tablas y demás.
- Código Arduino en el microcontrolador: Con el IDE Arduino se creó el código utilizado en el proyecto, para grabarlo en la memoria de la placa Arduino se utiliza la conexión USB entre el pc donde se desarrolla el software y la placa.
- Aplicación en dispositivo móvil: Para grabar en el teléfono móvil la aplicación diseñada se puede hacer de diferentes formas, todas son desde la herramienta de desarrollo utilizada como es Androdi Studio, la utilizada en este proyecto, pues bien a la hora de del desarrollo se puede ejecutar directamente en el móvil conectado mediante USB, o través de Wifi, todos estos pasos se describen al detalle en el Anexo correspondiente la instalación.



5 PRUEBAS Y EJECUCIÓN

5.1 Pruebas

Para ejecutar la aplicación mientras se está desarrollando se puede hacer de dos formas diferentes:

➤ Ejecución en una unidad virtual:

Sobre todo, en fase de pruebas, se suele utilizar la opción, crear una unidad virtual para ejecutar nuestra aplicación en ella.

El primer paso es buscar la opción “**AVD Manager**” que se encuentra en la opción de herramientas Tools, se mostrará un cuadro de dialogo como el de la imagen de más abajo.

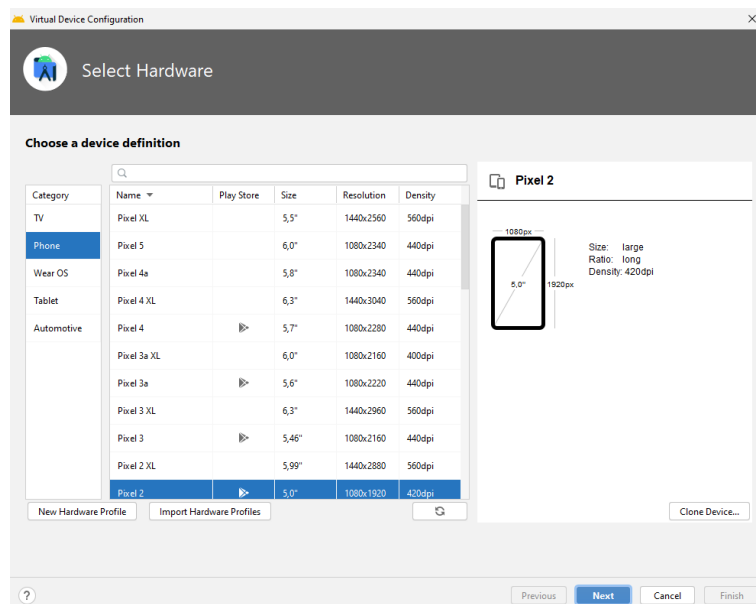


Figura 48. Virtual device

Hay varios modelos, entre teléfonos, tv, tablet, ... el modelo elegido está en función de las necesidades de la aplicación, cada opción nos da información de las medidas que ocupará la aplicación.



➤ Ejecución en dispositivo físico:

Hay varios pasos a seguir para la ejecución de la aplicación en el dispositivo móvil y son los siguientes:

- 1.- Dirigirse a ajustes, a cerca del teléfono.
- 2.- Pulsar al menos 7 veces en nº de compilación hasta ver el mensaje “modo de desarrollador activado”.
- 3.- Se nos muestra una imagen del estilo a esta, depende del modelo de dispositivo.



Figura 49. Opciones de desarrollador, móvil

- 4.- Una vez visualizada la imagen anterior, pulsar “Opciones de desarrollador”, acto seguido, habilitamos “Depuración USB”.



5.- Finalizados estos pasos, conectaremos el móvil mediante USB a la computadora, se nos mostrará un mensaje para elegir donde ejecutar la aplicación, y entre las opciones saldrá el móvil.

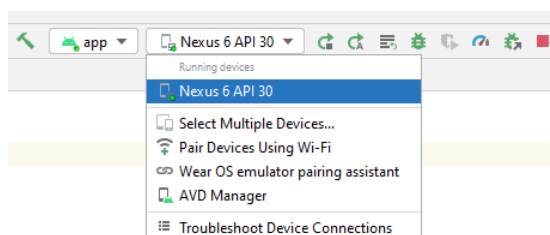


Figura 50. Elegir lugar de ejecución.



Figura 51. Depuración USB

5.2 Ejecución

Para la ejecución del sistema, basta con suministrar alimentación al montaje de Arduino y el resto de los componentes y conectarlo a Internet, la placa Arduino automáticamente comprueba el correcto funcionamiento de los sensores, mostrando el proceso de configuración y lecturas en la pantalla Oled.

Una vez realizado el paso anterior, el siguiente paso es registrar el microcontrolador en el servidor, la forma de diferenciar un sistema Arduino de otro es mediante el envío de su dirección Mac, que es única. En el código PHP que figura en la siguiente imagen, se puede observar el proceso, primeramente, se comprueba si la Mac existe en la base de datos, obteniendo el id que le corresponde para después insertar una un registro nuevo con los datos correspondientes, pero con la misma id, eso sí, con una fecha diferente.

```

132     $id_existe_mac=$this->existe_mac($mac);
133
134     if($id_existe_mac>0)
135         $id=$id_existe_mac;
136     else
137         $id=$this->mira_ultima_id($mac);
138
139     $res = $this->consulta('INSERT INTO `mcu` (`id`,`fecha`,`mac`,`ip`,`atributos`,`estado`,`intentos`) VALUES
140 ('.$id.','.$fecha.','.$mac.','.$ip.','.$atributos.','.$estado.','.$intentos.')
141 ON DUPLICATE KEY UPDATE
142 `fecha` = ".$fecha."',
143 `mac` = ".$mac."',
144 `ip` = ".$ip."',
145 `estado` = ".$estado."',
146 `atributos` = ".$atributos."',
147 `intentos` = ".$intentos.'");

```

Figura 52. Bloque sql inserción microcontrolador



En el caso de no existir en el sistema ningún microcontrolador con esta misma Mac, pues se procede de igual forma, pero obteniendo la última id+1.

Una vez echado a andar el sistema microcontrolador, se dedicará a enviar lecturas cada determinado intervalo de tiempo y siempre ordenado por la aplicación Android ejecutada en un dispositivo móvil, con la aplicación a cada instante se pueden modificar los sensores que puede leer el microcontrolador, así como modificar su estado, al mismo tiempo que se envían las lecturas, cada Arduino lee sus órdenes en el servidor para actuar en consecuencia.

Hasta aquí la ejecución del lado de Arduino, ahora se describe la ejecución del lado de la aplicación móvil diseñada para Android. Una vez instalada la aplicación en el dispositivo móvil, se accede al sistema a través de la pantalla de login, como la que figura en la imagen.

Una vez efectuado el login con éxito se obtiene un listado con el número de sistemas Arduino conectado, se muestra el estado indicando si está activo o no el microcontrolador en cuestión y las lecturas, en esta última nos indica si hay error o no, si hay error es debido a que microcontrolador(mcu) no está enviando lecturas o está desconectado.



Figura 54. Listado de Mcus

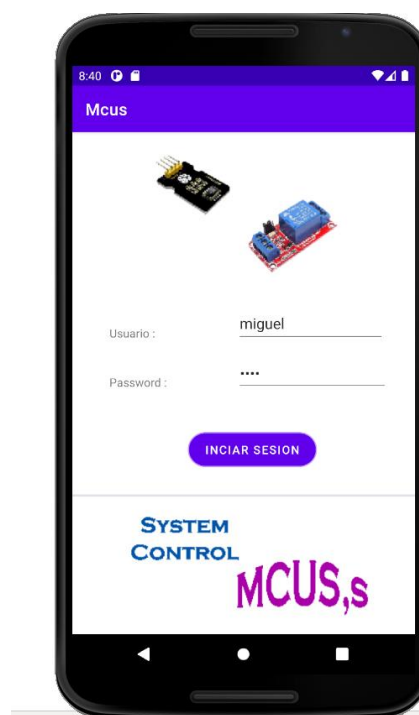


Figura 53. Pantalla login

Si se pulsa en cualquier microcontrolador del listado pasamos a la siguiente pantalla, en la cual se muestra información de las lecturas, estado de los sensores y estado del mcu.

Pulsando un simple botón se pueden activar/desactivar tanto sensores como el microcontrolador.

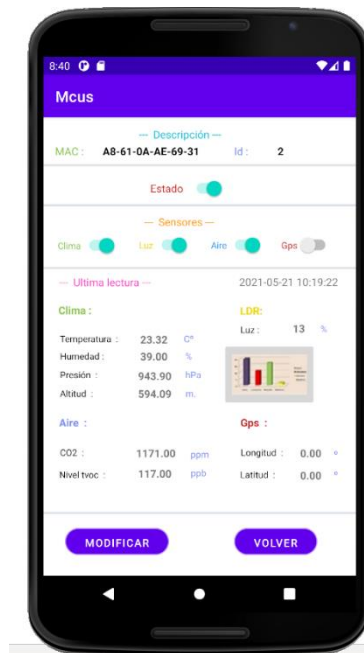


Figura 55. Pantalla lecturas y estado

Por último se encuentra la pantalla de información gráfica, se pueden observar las gráficas de las lecturas de luz, aire(CO₂) y Clima(temperatura).



Figura 56. Gráficas



6 PRESUPUESTO

El presupuesto del hardware corresponde a la siguiente tabla, denotar que este presupuesto puede variar en gran medida dependiendo de la fecha y lugar de la compra:

Componente	Unidades	Precio/uni.	Total
Fuente de alimentación MB102	1	2,50	2,50
Arduino Mega 2560	1	42,00	42,00
Módulo Ethernet Shield	1	25,80	25,80
Módulo LDR	1	1,00	1,00
Sensor ambiental GY-BME280	1	11,69	11,69
Conector Pila 9v	1	1,00	1,00
Sensor de aire CCS811	1	14,00	14,00
Módulo GPS GY-NEO6MV2 NEO-6M	1	16,29	16,29
Pantalla OLED ARCELI 0.96 Inch 128x64	1	6,15	6,15
Placa protoboard	4	4,20	12,60
Cables conexión	1	7,99	7,99
Pack leds 600 unidades	1	6,66	6,66
Módulos de Relés de 4 Canales	1	7,99	7,99
Resistencias	1	10,99	10,99
Total base imponible..... =			166,66 €
IVA 21 %... =			35,00 €
TOTAL... =			201,66 €

Tabla 3. Tabla de presupuesto

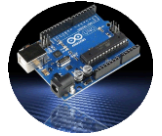


7 LÍNEAS DE TRABAJO FUTURAS

Como futuras mejoras en el proyecto se consideran las siguientes para futuras versiones:

- Sistema de Backup a través de la aplicación Android que contengan una opción para realizar copias de seguridad de la base de datos.
- La interfaz de la App está diseñada para poder ser utilizada en dispositivos móviles, no estaría de más un diseño versión para Tablet.
- Otra mejora sería disponer de una web para realizar las mismas gestiones que se realizan a través de la App.
- En método actual para conectarse a la red es través de Ethernet, no estaría de más tener la posibilidad de utilizar conexión Wifi.
- Al hilo de la mejora anterior, una opción válida proporcionar claves de acceso a la red wifi, sería a través de una conexión bluetooth.
- Avisos desde la aplicación móvil indicando los sistemas nuevos conectados, también de algún error con los sensores o microcontroladoras, por ejemplo, si se encuentran un tiempo estimado sin enviar datos estando en estado activo y alguno de sus sensores también.
- Mejorar el sistema de gestión de usuarios del sistema.

Este proyecto en caso seguir avanzando en su diseño en vistas a poder ser comercializado, se debería de cambiar el diseño para implementarlo en una placa de circuito impreso, con el microcontrolador Atmega2560 y todos los sensores, todo ello encapsulado e integrado en un soporte o caja en la cual este todo el sistema incorporado con sensores exteriores.



8 REFERENCIAS

- [1] «Características Arduino Mega 2560,» [En línea]. Available: <https://proyectoarduino.com/arduino-mega-2560/>.
- [2] Aprendiendoarduino, «Ethernet shield Aprendiendo Arduino,» [En línea]. Available: <https://aprendiendoarduino.wordpress.com/category/ethernet-shield/>. [Último acceso: 2021].
- [3] «Sensor ambiental,» [En línea]. Available: <https://naylampmechatronics.com/sensores-posicion-inerciales-gps/357-sensor-de-presion-temperatura-y-humedad-bme280.html>.
- [4] S. Aire, «Sensor Aire CCS811,» Keyestudio, [En línea]. Available: <https://shop.innovadidactic.com/ca/standard-sensores/983-keyestudio-ccs811-sensor-de-calidad-del-aire-dioxido-de-carbono.html>.
- [5] G. GY-NEO6MV2, «LEANTEC,» GPS GY-NEO6MV2, [En línea]. Available: <https://leantec.es/tienda/modulo-ublox-gps-gy-neo6mv2-arduino-uno-r3-nano-mega-original-raspberry-pi/>.
- [6] az-delivery, «Pantalla OLED az-delivery,» OLED az-delivery, [En línea]. Available: <https://www.az-delivery.de/es/products/1-3zoll-i2c-oled-display>.
- [7] Arduino, «Software Arduino,» Software Arduino, 8 Abril 2021. [En línea]. Available: <https://www.arduino.cc/en/software>.
- [8] Notepad, «Notepad plus plus,» [En línea]. Available: <https://notepad-plus-plus.org/downloads/v8/>.
- [9] Google, «Android studio,» [En línea]. Available: <https://developer.android.com/studio/archive>.
- [10] JetBrains, «JetBrains,» JetBrains, [En línea]. Available: <https://es.wikipedia.org/wiki/JetBrains>.
- [11] JetPack, «JetPack,» Google, [En línea]. Available: <https://developer.android.com/jetpack?hl=es>.
- [12] Filezilla.org, «Filezilla.org,» [En línea]. Available: <https://filezilla-project.org/>.
- [13] Domiteca, «Domiteca,» [En línea]. Available: www.domiteca.com.
- [14] Interdominios, «interdominios.com,» [En línea]. Available: <https://www.interdominios.com/hosting-sin-limitas.aspx>.
- [15] hardzone.es, «Raspberry Pi y Arduino,» Raspberry Pi y Arduino, 2021. [En línea]. Available: <https://hardzone.es/reportajes/comparativas/raspberry-pi-vs-arduino/>.



- [16] Processing, «Programacion facil arduino,» [En línea]. Available: <https://programarfacil.com/blog/arduino-blog/processing-wiring-arduino/>.
- [17] E. Hub, «Arduino Mega Pinout,» Arduino Mega Pinout, 20 Enero 2021. [En línea]. Available: <https://www.electronicshub.org/arduino-mega-pinout/>.
- [18] «Xamarin,» Xamarin, [En línea]. Available: <https://inmediatum.com/blog/ingenieria/ventajas-y-desventajas-de-apps-desarrolladas-en-xamarin/>.
- [19] «Bus de comunicación I2C,» Bus de comunicación I2C, [En línea]. Available: <http://robots-argentina.com.ar/didactica/descripcion-y-funcionamiento-del-bus-i2c/>.
- [20] «Primeros pasos con la placa Arduino Mega 2560,» [En línea]. Available: <https://descubrearduino.com/arduino-mega/>.
- [21] «SPI interface de periféricos en serie,» SPI interface de periféricos en serie, [En línea]. Available: <https://www.luisllamas.es/arduino-spi/>.

Fecha en la cual fueron revisadas las referencias por última vez, 30 junio de 2021