



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Informática

Desarrollo de un equipo de diagnóstico de baterías

Trabajo Fin de Grado

Grado en Ingeniería Informática

AUTOR/A: Iborra Barber, Pau

Tutor/a: Rubio Moreno, Alicia

Cotutor/a externo: PANERO TAPIA, JUAN

CURSO ACADÉMICO: 2021/2022



## Agradecimientos

---

Agradecer principalmente a mis padres por todo el esfuerzo que han hecho para que pueda llegar hasta aquí, a los compañeros de la Ford que me hicieron la estancia muy cómoda, a mi tutora Alicia por solucionarme cualquier problema que haya podido tener y ser mi tutora de prácticas y, por último, al resto de becarios que conocí con los que estuve pasando una bonita estancia en la Ford.



## Resumen

---

La empresa Ford España de Almussafes propuso la realización de este proyecto con el que ayudar al taller eCats, que se encuentra en la planta de producción dentro de la factoría.

El Proyecto consiste en desarrollar un equipo de medición de estado de baterías “atlas copco” con una aplicación móvil. Esta aplicación permite realizar y controlar el proceso, además se añade un circuito para medir el voltaje y la intensidad.

A través de un hardware y software, primero se cargarán y posteriormente se descargarán durante 10 segundos aplicando una resistencia, para así medir la caída de tensión y diagnosticar si es funcional o no. El equipo de diagnóstico se conectará a través de bluetooth al móvil con un programa diseñado para Android donde se mostrarán todos los parámetros medidos.

La aplicación cuenta con una interfaz intuitiva y fácil de usar para que sea dominada por cualquier trabajador. Además, también cuenta con una pequeña guía de cómo usar para el usuario.

Las tecnologías empleadas para el desarrollo de la aplicación serán HTML, JavaScript, CSS, Angular, Ionic y Arduino entre otras.

**Palabras clave:** baterías, diagnóstico, aplicación móvil, circuito, estado.

L'empresa Ford Espanya d'Almussafes va proposar la realització d'aquest projecte per ajudar el taller eCats, que es troba a la planta de producció dins de la factoria.

El Projecte consisteix a desenvolupar un equip de mesurament d'estat de bateries "atlas copco" amb una aplicació mòbil. Aquesta aplicació permet fer i controlar el procés, a més s'afegeix un circuit per mesurar el voltatge i la intensitat.

Mitjançant un maquinari i programari, primer es carregaran i posteriorment es descarregaran durant 10 segons aplicant una resistència, per així mesurar la caiguda de tensió i diagnosticar si és funcional o no. L'equip de diagnòstic es connectarà a través de bluetooth al mòbil amb un programa dissenyat per a Android on es mostraran tots els paràmetres mesurats.

L'aplicació compta amb una interfície intuïtiva i fàcil de fer servir perquè sigui dominada per qualsevol treballador. A més, també compta amb una petita guia de com utilitzar per a lusuari.

Les tecnologies emprades per al desenvolupament de l'aplicació seran HTML, JavaScript, CSS, Angular, Ionic i Arduino entre d'altres.

**Paraules clau:** bateries, diagnòstic, aplicació mòbil, circuit, estat.

## Abstract

---

Ford España in Almussafes proposed this project to help the eCats workshop, which is located in the production plant inside the factory.

The project consists of developing a battery status measurement equipment "atlas copco" with a mobile application. This application allows the process to be carried out and controlled, in addition to a circuit to measure voltage and current.

Through hardware and software, the batteries will first be charged and then discharged for 10 seconds by applying a resistor, in order to measure the voltage drop and diagnose whether it is functional or not. The diagnostic equipment will be connected via bluetooth to the mobile phone with a program designed for Android where all the measured parameters will be displayed.

The application has an intuitive and easy-to-use interface so that it can be mastered by any worker. In addition, it also has a small user's guide on how to use it.

The technologies used for the development of the application will be HTML, JavaScript, CSS, Angular, Ionic and Arduino among others.

**Keywords:** batteries, diagnosis, mobile application, circuit, status.

## Tabla de contenidos

---

1.	Introducción	12
1.1	Motivación	12
1.2	Objetivos	13
1.3	Impacto esperado	13
1.4	Metodología para el desarrollo del trabajo	14
1.5	Estructura de la memoria	15
2.	Situación actual	16
2.1	Crítica al estado del arte	16
2.2	Propuesta	17
3.	Análisis y modelado	18
3.1	Análisis de seguridad	20
3.2	Análisis energético o de eficiencia algorítmica	20
3.3	Análisis del marco legal y ético	20
3.4	Análisis de riesgos	21
3.5	Identificación y análisis de soluciones posibles	21
3.5.1	Soluciones alternativas	21
3.6	Solución propuesta	21
3.7	Plan de trabajo	22
3.8	Presupuesto	23
4.	Diseño de la solución	24
4.1	Arquitectura del sistema	24
4.2	Diseño detallado	29
4.2.1	Mapa de navegación	30
4.2.2	Selección empezar comprobación del estado de la batería	33
4.3	Tecnología utilizada	35
4.3.1	Herramientas para el desarrollo	35
4.3.2	Lenguajes para su programación	36
4.3.3	Frameworks utilizados	37
5.	Desarrollo de la solución propuesta	39
5.1	Propuesta de la solución final	39
5.2	Problemas durante el desarrollo	40
6.	Implantación	43



7.	Pruebas	44
7.1	Pruebas en el circuito	44
7.2	Pruebas en la implementación de código	44
7.3	Pruebas de la aplicación	44
8.	Conclusiones	45
8.2	Relación del trabajo desarrollado con los estudios cursados	45
8.3	Competencias transversales	46
9.	Futuros trabajos	47
10.	Referencias	48
	Anexo 1. Pruebas funcionamiento	50
	Anexo 2. Funcionamiento ngOnInIt y constructor	51
	Anexo 3. Conexión BLE	53
	Anexo 4. Objetivos de desarrollo sostenible	55
	Glosario	58



Figura 1. Fases modelo en cascada .....	14
Figura 2. Diagrama de flujo comprobación estado batería antiguo. ....	18
Figura 3. Diagrama de flujo comprobación estado batería con aplicación. ....	19
Figura 4. Planificación fases del trabajo. ....	22
Figura 5. Esquema funciones entradas y salidas Arduino. ....	24
Figura 6. Arduino Nano 33 IoT. ....	25
Figura 7. Circuito completo. ....	26
Figura 8. Sensor de consumo. ....	26
Figura 9. Fuente de alimentación. ....	27
Figura 10. Resistencia de descarga con ventilador. ....	27
Figura 11. Circuito de mando. ....	28
Figura 12. Muestra de mediciones. ....	28
Figura 13. Ultracaps. ....	29
Figura 14. Mapa de navegación de la aplicación. ....	30
Figura 15. Pantalla de inicio de la aplicación. ....	31
Figura 16. Interfaz página bluetooth. ....	32
Figura 17. Página de guía sobre la aplicación. ....	33
Figura 18. Diagrama de flujo test comprobación estado batería. ....	34
Figura 19. Código Arduino comprobación voltaje. ....	34
Figura 20. Control carga/descarga Arduino .....	35
Figura 21. Directorio 'src' de la App .....	40
Figura 22. Estado conexión .....	41
Figura 23. Mal contacto entre resistencias .....	42
Figura 24. Importaciones /home.page .....	43
Figura 25. Plugins utilizados. ....	43
Figura 26. Importación OnInit.....	51



Figura 27. Método ngOnI nit .....	51
Figura 28. Escáner de dispositivos bluetooth cercanos.....	53
Figura 29. Código dispositivos encontrados .....	53
Figura 30. Selección de dispositivo.....	53
Figura 31. Dispositivo conectado .....	54

## Índice de tablas

---

Tabla 1. Funciones Arduino .....	24
Tabla 2. Programación según herramienta.....	40
Tabla 3. Competencias transversales .....	46



# 1. Introducción

---

En todas las empresas, ya sean grandes o pequeñas, siempre hay problemas por solucionar y funciones a optimizar para conseguir un mayor desarrollo de estas mismas.

El empleo de tecnología y, concretamente, de la informática pueden suponer una gran ayuda a la mayoría de las empresas. Estas deben evolucionar y crecer para mantenerse en el mercado, y actualmente, la informática es la que está revolucionando todos los mercados.

En este trabajo se ha ayudado a la empresa agilizando el trabajo. Gracias a los conocimientos y a la aceptación que se obtuvo por parte del equipo, consiguiendo así mejorar el funcionamiento y cumplir los objetivos establecidos por la empresa.

También sirvió para mejorar con la formación personal como trabajador y con la experiencia en nuevos campos durante el tiempo que duró el proyecto.

Al final de este trabajo se encuentra un pequeño glosario con algunas definiciones que pueden ayudar a entender ciertos conceptos.

## 1.1 Motivación

La Ford Motor Company, la empresa que ofrecía las prácticas, específicamente en el taller de eCats, tenía constantemente problemas con la cantidad de baterías atlas copco que fallaban. Era un proceso largo y costoso, en cuanto a tiempo, el tener que estar reparando cada una y posteriormente comprobando su funcionamiento. Se ofreció la idea de realizar una aplicación que ayudara a comprobar el estado de las baterías en cuestión de segundos, cosa que les ahorraría mucho tiempo y les dejaría libres para desempeñar otras tareas.

Anteriormente se propuso un proyecto sobre inteligencia artificial, el cual también parecía interesante y despertaba la curiosidad. Después de unos días, valorando las partes positivas y negativas de cada proyecto, se decidió trabajar en el primero, el cual era nuevo para la empresa y aun no se había investigado.

Era un proyecto con muchas visiones de futuro, compuesto por una gran diversidad de lenguajes de programación y herramientas tecnológicas, cosa que motivaba más aun a trabajar en él.

En la empresa hubo un gran interés constante por este proyecto y fue apoyado por muchos componentes del equipo.

## 1.2 Objetivos

El principal objetivo de este proyecto era ayudar a los trabajadores de la empresa, desarrollando un software y hardware, con una aplicación Android enlazada a un Arduino, para que compruebe el estado de las baterías a través de un circuito.

Este proyecto consta, por una parte, del diseño de la interfaz de usuario, la cual es muy intuitiva; y, por otra parte, de la implementación de una aplicación móvil.

Se pretende crear una interfaz sencilla y ágil para que cualquier empleado sin requerir de ninguna experiencia ni conocimiento pueda manejarla. El código quedará detallado para cualquier necesidad de comprensión de algún trabajador.

La aplicación contará con varios botones, los cuales te permiten medir el voltaje y la intensidad, empezar el proceso de comprobación del estado de batería o parar el proceso, y otro por si queremos descargarlas. Además, se añadió un pequeño botón de consulta donde se detalla el funcionamiento.

Por tanto, toda esta información puede ser expuesta en subobjetivos:

- Reducir el tiempo de comprobación del estado de las baterías.
- Facilidad de utilización y comprensión de la aplicación móvil.
- Funcionamiento a través de conexión bluetooth.
- Diseño con circuito reducido.

## 1.3 Impacto esperado

Debido a la cantidad de baterías que posee la empresa, se pretende poder comprobar el estado de las baterías que generan fallos de la forma más rápida y sencilla posible.

Toda esta información que genera la aplicación es suficiente para saber si la batería debe de ser reparada o esta correcta para su utilización.

Las baterías están compuestas en su interior por “ultracaps”, que es donde se hace la comprobación del estado. En la empresa hay una cantidad inimaginable de estos por lo que el tiempo de comprobación es inmenso. Los trabajadores empleaban una pequeña máquina para hacer la comprobación que tardaba horas en dar un resultado, el cual se basaba en cargarlas y comprobar una vez cargadas su funcionamiento. Con la aplicación se pueden comprobar todos los “ultracaps” que componen una batería en menos de 1 minuto, cosa que agilizaba drásticamente el trabajo.

Esta mejora va a generar más tiempo para los trabajadores para realizar otras actividades necesarias, y por lo tanto una mejora en el rendimiento general de la empresa.



## 1.4 Metodología para el desarrollo del trabajo

Una vez claros los conceptos básicos y la intención del proyecto, se tiene que adoptar una metodología que marcará los pasos del proyecto.

Se valoraron distintas metodologías y finalmente se decidió utilizar la metodología tradicional de modelo en cascada [1].

Este método es un proceso de desarrollo secuencial de proyectos que se utiliza a menudo en el desarrollo de software. Este enfoque concibe el trabajo como un conjunto de etapas que deben realizarse una tras otra. Toma su nombre de las distintas fases que componen un proyecto, ya que deben superponerse en un orden concreto y estricto de arriba a abajo. Por ejemplo, no podemos comenzar la fase de diseño sin completar la fase de requisitos.

Estas son las fases que componen la metodología de modelo en cascada:

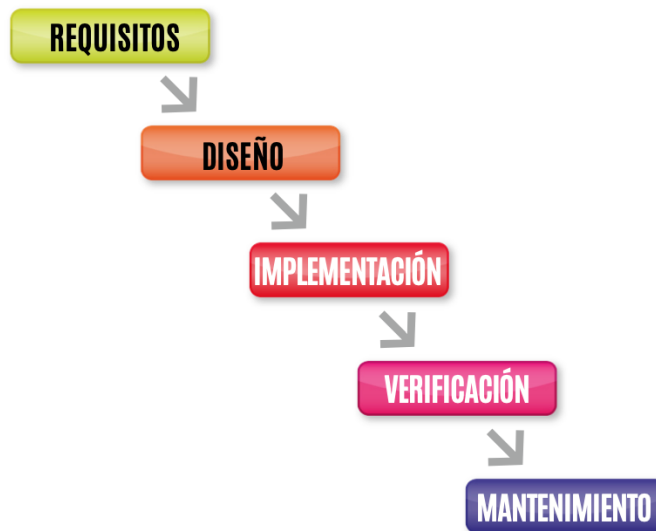


Figura 1. Fases modelo en cascada

- La primera fase es la de requisitos, consiste en planificar y analizar las necesidades.
- En segundo lugar, el diseño, donde se piensa una solución para las necesidades vistas en la fase anterior.
- La fase de implementación, en la cual se realiza la programación y se buscan posibles errores, realizamos pruebas y comprobamos el funcionamiento.
- En este momento, el cliente realiza la verificación, y a la vez, se explica el manejo de la aplicación.
- Finalmente, se entrega con su correspondiente mantenimiento y la opción de realizar mejoras.

## 1.5 Estructura de la memoria

El trabajo se presenta estructurado por los siguientes capítulos:

1. **Introducción.** Capítulo que ya ha sido explicado, en él se introduce al proyecto mostrando la motivación que ha llevado a realizarlo, los objetivos y el impacto esperado que generó sobre la gente, tanto para los trabajadores como para la empresa misma.
2. **Situación Actual.** En esta sección se habla sobre el problema y la necesidad de la empresa. Además, de una pequeña comparación con la actualidad.
3. **Análisis y Modelado.** Aquí se explica el problema y se analiza en un diagrama para que se vea el proceso. Junto con distintos análisis y la propuesta de solución. Además de un plan de trabajo con las fases que lo componen.
4. **Diseño de la solución.** Se detalla y se muestra con imágenes las partes que componen el proyecto, explicando la función de cada una de estas. También se enseñan las principales tecnologías empleadas. Además, de un diagrama de flujo para entender el proceso de funcionamiento.
5. **Desarrollo de la solución propuesta.** Este capítulo enseña la propuesta de solución final y todos los problemas con los que nos hemos encontrado durante el desarrollo del proyecto.
6. **Implantación.** Pequeña muestra de componentes de la programación.
7. **Pruebas.** Aquí se explican todas las pruebas realizadas a lo largo del trabajo en las distintas partes que lo componen.
8. **Conclusión.** En esta sección se encuentran los errores y problemas que ha habido y como se han solucionado. También se explica cómo se han cumplido los objetivos marcados al principio, las competencias transversales más trabajadas y la relación con los estudios cursados.
9. **Futuros trabajos.** Se explican las mejoras que podrían realizarse en un futuro.
10. **Referencias.** Aquí se listan todas las referencias empleadas a lo largo del trabajo.



## 2. Situación actual

---

En este segundo capítulo se trata sobre las tecnologías actuales relacionadas con este mismo campo. La mayoría de las empresas requieren de tecnologías que les faciliten las tareas, con el fin de poder mejorar sus funciones.

La Ford Motor Company es una empresa muy grande que cuenta con una gran cantidad de maquinaria necesitada de una tecnología para funcionar. Las principales tecnologías en las que se centran para la innovación son las siguientes [2]:

- Robots colaborativos.
- Impresoras 3D
- Exoesqueletos
- Realidad aumentada
- Código QR

Estas 5 están mayoritariamente centradas en el desarrollo del vehículo, pero detrás existe una gran cantidad de otros trabajos que también requieren de tecnologías para su correcto funcionamiento. Siendo una empresa de estas dimensiones, hay miles de departamentos y subprocesos que no se centran en la creación de vehículos, también existe un control informático de toda la empresa, una seguridad que mantenga todos los datos privados, unos trabajadores que se encarguen de la reparación y el mantenimiento de las maquinas, ...

Para la realización de la aplicación se emplearon diversos lenguajes de programación, entre los que encontramos JavaScript, HTML, CSS, Arduino, ... Anteriormente no se constaba con ninguna aplicación para realizar la comprobación del estado de las baterías, por lo que tenían que cargarlas al máximo y probarlas, dependiendo de su funcionamiento sabrían si están bien o por lo contrario necesitaban ser reparadas.

### 2.1 Crítica al estado del arte

Hay varios proyectos relacionados realizados en la Ford, incluso en el mismo departamento, pero ninguno que se haya centrado en el proyecto realizado en este TFG. Por eso mismo, es más interesante un problema que anteriormente no se haya tratado y pueda ser solucionado gracias a este proyecto.



## 2.2 Propuesta

Comparando con lo existente actualmente, en lo que refiere a estas funciones, no existía nada que pudiese afrontar estos problemas de forma rápida y eficaz. Aportando una ampliación de capacidades de programación en los distintos lenguajes, y a la empresa, una herramienta ágil de trabajo.

Desde un primer momento esta propuesta ha tenido grandes expectativas y ha sido muy valorada por todo el equipo.



### 3. Análisis y modelado

---

El problema era siempre el mismo y siempre llevaba a cabo las mismas acciones:

1. Alerta fallo en batería atlas copco.
2. El trabajador se encargaba de abrir la batería y coger los ultracaps.
3. Los dejaba en la base de carga que tenían en el taller hasta que estuviesen cargados por completo.
4. Volvían a montar las baterías.
5. Probaban su funcionamiento.

A continuación, se ve en un diagrama el desarrollo del proceso para su comprobación.

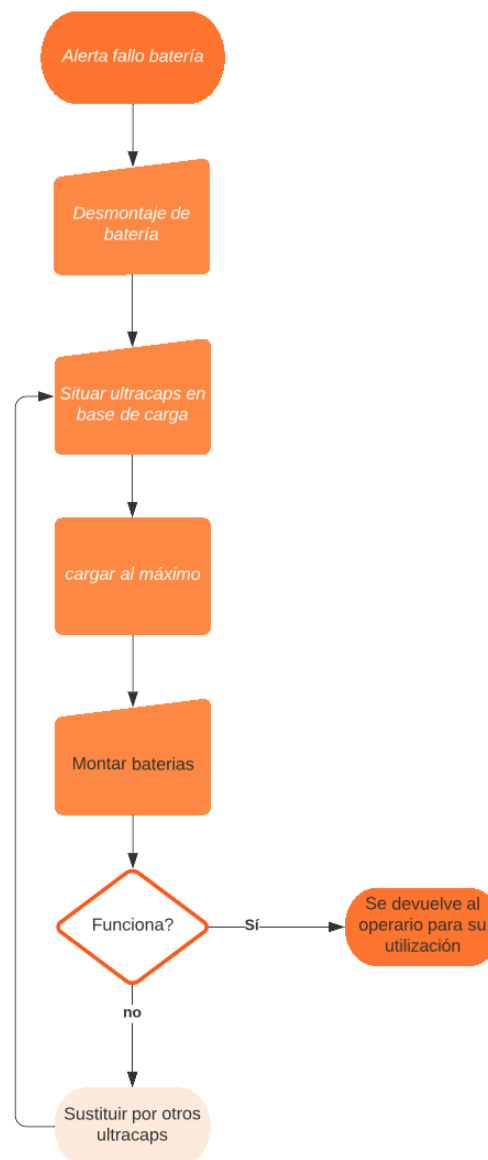


Figura 2. Diagrama de flujo comprobación estado batería antiguo.

Posteriormente con la aplicación se ahorra el paso de cargar los ultracaps al máximo y comprobar manualmente si funciona, ya que, eso lo haría el mismo programa, por lo tanto, representado con otro diagrama de flujo quedaría de la siguiente forma:

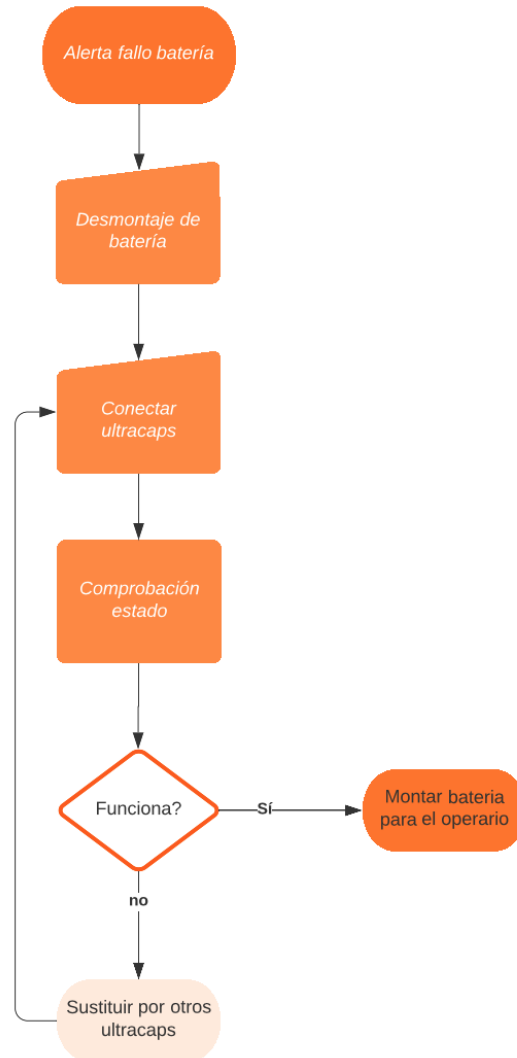


Figura 3. Diagrama de flujo comprobación estado batería con aplicación.

El paso 1 y 2 se mantendrían y a partir del 3, en vez de conectar a la base de carga, se conectan por cable al circuito de comprobación. En 4 lugar se chequea el estado con la aplicación y dependiendo del estado se vuelve a montar o se sustituyen los ultracaps por otros.

### 3.1 Análisis de seguridad

La empresa Ford Motor Company, cuenta con un alto nivel de seguridad por toda la factoría. Físicamente cuentan con cuerpos de seguridad, los cuales verifican tu identificación para poder entrar dentro de la fábrica, y para salir tienen el poder de revisarte por si has cogido algo de la empresa, está totalmente prohibido extraer cualquier cosa de dentro de la factoría, en caso de requerir llevarte algún utensilio debes realizar una solicitud de permiso que será revisada y valorada por superiores.

A nivel informático cuentan con un departamento TI encargado de todo el mantenimiento y control de cada ordenador o utensilio electrónico. Los permisos de descarga de softwares son bastantes limitados, teniendo un pequeño catálogo que ofrece la empresa donde detalla que programas se pueden descargar. Además, cada trabajador cuenta con un móvil privado que solo puede ser usado dentro de la empresa.

### 3.2 Análisis energético o de eficiencia algorítmica

Respecto al consumo, la aplicación ocupa muy poco espacio en tu dispositivo móvil y no requiere una gran potencia de procesador, por lo que consume poca batería y puede ser ejecutada prácticamente en cualquier dispositivo móvil Android.

También hay que añadir que con el diseño del circuito se ha reducido el coste de los elementos al mínimo, haciendo así un mínimo gasto energético para la carga y descarga de los ultracaps con unas resistencias.

### 3.3 Análisis del marco legal y ético

Hay dos apartados que debería remarcar por el marco legal.

**Análisis de la protección de datos:** La aplicación requiere tanto de permisos de ubicación como detección de dispositivos bluetooth para poder buscar los dispositivos cercanos y conectarse al Arduino.

**Propiedad intelectual:** La empresa, Ford España, no permite la extracción de material fuera de la factoría, por lo cual, tanto el circuito como las baterías necesarias para el funcionamiento correcto de la aplicación solo puede ser utilizado dentro de las instalaciones.

### **3.4 Análisis de riesgos**

La aplicación está debidamente diseñada para que pueda ser utilizada por cualquier trabajador, y consta con un apartado de guía de funcionamiento. Pero aun así puede haber gente que no esté nada relacionada con las tecnologías móviles y pueda requerir de alguien que le ayude. Por lo tanto, no está de más impartir un pequeño curso para explicar a todos los trabajadores que van a utilizar la aplicación como deben hacerlo de forma que no haya ningún riesgo de error a la hora de comprobar el estado de las baterías.

### **3.5 Identificación y análisis de soluciones posibles**

El problema ya explicado obtiene como soluciones factibles una única opción que es la mencionada en este proyecto. De todas formas, se podría optar por otras soluciones que para algún usuario les puede parecer interesante.

#### **3.5.1 Soluciones alternativas**

##### **Renovación de las baterías.**

Solicitar que todas las baterías que fallen se eliminen y sean sustituidas por unas nuevas. Esta solución es muy eficaz pero poco factible, ya que conllevaría un gasto inmenso para la empresa.

Aunque esta solución fuese muy rápida, hay que ser objetivos y mirar por el bien de la empresa, no se puede gastar una cantidad de dinero tan grande en algo que no es realmente necesario, además de todo el material que habría que tirar.

##### **Actual**

La solución que se explicó, la cual empleaban actualmente, es eficaz y eficiente, pero tiene un costo de tiempo elevado para el trabajador. En este caso se tiene que desmontar la batería y dejar cargar los ultracaps en una base de carga. Una vez cargados, volver a montarlos y comprobar su funcionamiento. Efectivamente se consigue el resultado esperado, pero ese tiempo podría haber sido empleado para otro trabajo que requiera del mismo trabajador.

### **3.6 Solución propuesta**

Como se ha comentado en capítulos anteriores, la solución elegida por su eficiencia y bajo costo de realización y mantenimiento, ha sido una aplicación que facilite la comprobación del estado de las baterías midiendo el voltaje y la intensidad de los ultracaps que las componen. Con esto, el trabajador sabe si la batería puede volver a



ser utilizada por el operario o va a necesitar una reparación. En caso de requerirla, se abriría la batería comprobando que ultracaps fallan, una vez detectados se repararían, o si por lo contrario no tienen solución, como último recurso serían remplazados por otros.

### 3.7 Plan de trabajo

La elaboración del trabajo se divide en varios procesos. A lo largo de la realización del proyecto ha habido unas fases más simples y otras más costosas, pero todas ellas han repercutido de forma positiva en el resultado. El tiempo definido para su elaboración era aproximadamente de 3 meses, siendo este el tiempo de permanencia en la empresa de prácticas Ford España.

Las fases fueron estructuradas adaptándolas al tiempo según su avance:

Núm. Fase	Nombre Fase	Fecha inicio	Fecha fin
1	Estructurar Arduino y conexiones	01/03/2022	15/03/2022
2	Programación Arduino	16/03/2022	07/04/2022
3	Implementación apk y conexión bluetooth	08/04/2022	26/05/2022
4	Realización de mejoras	27/05/2022	

Figura 4. Planificación fases del trabajo.

**Estructurar Arduino y conexiones.** En esta primera fase, se conoce el funcionamiento a fondo del Arduino, concretamente el Arduino Nano 33 IoT, con librerías extraídas de la página oficial<sup>[2]</sup>, para así poder desarrollar unos conocimientos mayores sobre el Arduino, tanto a nivel de programación como a nivel de conexiones.

**Programación Arduino.** La segunda fase ya requería de programación. Arduino es un hardware libre con el que grabar instrucciones escritas en un lenguaje de programación poco utilizado a lo largo de la carrera, por lo que se requería una gran iniciativa de aprendizaje. Estas instrucciones programadas a su vez interactuaban con un circuito electrónico. Se debía saber el funcionamiento de cada instrucción y como reaccionaría el circuito. Una gran cantidad de pruebas fueron realizadas a lo largo de esta fase.

**Implementación apk y conexión bluetooth.** Esta fase es la más larga y probablemente con más dificultades encontradas a lo largo de su realización. Empleando el lenguaje JavaScript + Framework de Ionic con Angular<sup>[3]</sup> y HTML<sup>[4]</sup> se consiguió el diseño de la interfaz de toda la aplicación, más el funcionamiento de cada uno de los botones que componen la App.

Además, de conseguir la conexión bluetooth entre la aplicación del dispositivo móvil con el del Arduino. Esto se realizó gracias a una de las características más importantes que componen el Arduino Nano 33 IoT, que es la función BLE (Enables

Bluetooth® Low Energy)<sup>[14]</sup>, la cual está optimizado para un uso de baja potencia a velocidades de datos bajas, cosa que nos genera un ahorro de energía y es suficiente para su correcto funcionamiento.

**Realización de mejoras.** Una vez finalizada la aplicación, tanto de software como de hardware solo quedaba hacer mejoras, ya fuesen estéticas de interfaz como internas con la recopilación de datos. No se llegó a finalizar esta fase porque se mantuvo en constante actualización.

**Otras fases.** En principio, se dio la opción de realizar una 5 fase, en la cual diseñar unas piezas 3D con la App Inventor, una vez finalizada la aplicación principal, para así cubrir y dejar de una forma más organizada todo el circuito electrónico. Esta fase por falta de tiempo en la empresa no se pudo realizar como principalmente se planeó, pero un compañero y trabajador de la empresa consiguió realizarla explicando de una forma muy superficial el proceso de fabricación.

### 3.8 Presupuesto

Este trabajo ha sido requerido de bastante material para poder ser llevado a cabo. En este caso, todo ello fue proporcionado por la propia empresa, Ford España.

El material empleado para su realización fue el siguiente:

- Ordenador
- Teléfono móvil
- Arduino Nano 33 IoT
- Cable USB
- Cables circuito
- Ventiladores
- Resistencias
- Sensor de consumo
- Condensadores
- Fuente de alimentación
- Mosfet<sup>[15]</sup>
- Transistores
- Relés

Como se observa la mayoría de los materiales son los requeridos para el circuito, pero como ha sido comentado, todo esto ha sido proporcionado por la empresa. Ya que para la programación se han empleado softwares gratuitos, por lo que el costo ha sido cero.

## 4. Diseño de la solución

Sabiendo la solución tomada al problema, y teniendo todo el material requerido para conseguirla, se procede a su realización. La solución está documentada en dos niveles, la arquitectura del sistema y el diseño detallado. Además de la tecnología empleada para la realización del proyecto.

### 4.1 Arquitectura del sistema

En primer lugar, se muestra el esquema del Arduino Nano 33 IoT para así comprender su funcionamiento. Este dispositivo, cuenta con entradas analógicas y digitales, y salidas digitales, desde las cuales se pueden leer los valores, aplicar cargas,

...

Aquí también se puede observar la función de cada entrada/salida:

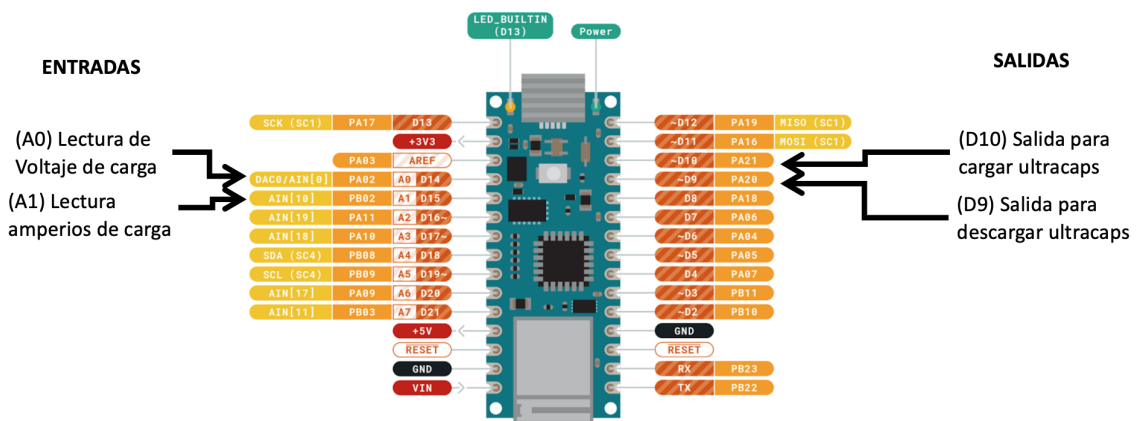


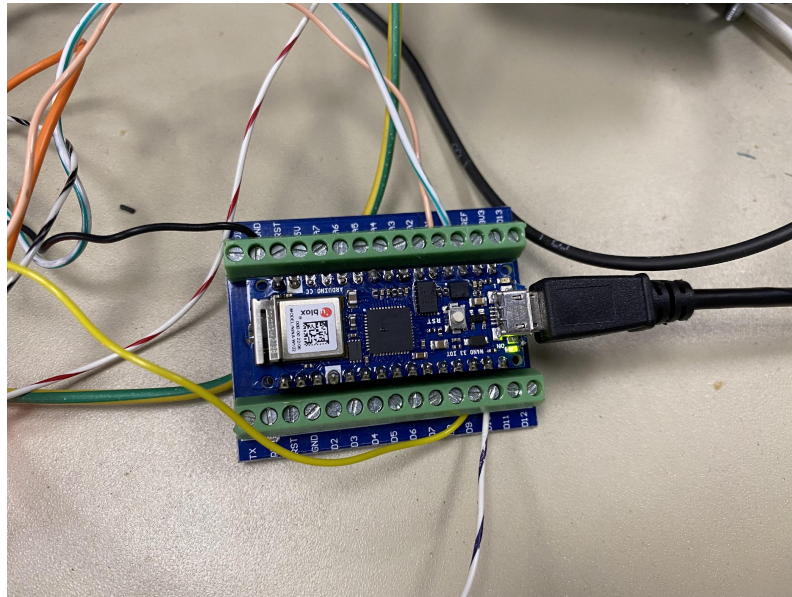
Figura 5. Esquema funciones entradas y salidas Arduino.

Se muestra también en una tabla para su mejor comprensión:

	Entrada	Salida	Función
A0 (analógica)	X		Lee el voltaje
A1 (analógica)	X		Lee los amperios
D10 (digital)		X	Activa la carga
D9 (digital)		X	Activa la descarga

Tabla 1. Funciones Arduino





*Figura 6. Arduino Nano 33 IoT.*

A la izquierda de la Figura 5 se observan todas las entradas (Analógicas y digitales) y a la derecha las salidas digitales. Aún quedarían entradas y salidas inutilizadas con las cuales se podrían hacer más funciones, pero no han sido requeridas para este proyecto.

En la figura 6 aparece el Arduino con sus conexiones mencionadas en la figura anterior, y conectado al PC, a través del cual se mandan los comandos programados.

Una vez vista la función clara del Arduino, el proceso prosigue por el circuito. Este circuito consta de diversas partes, cada una con su función.

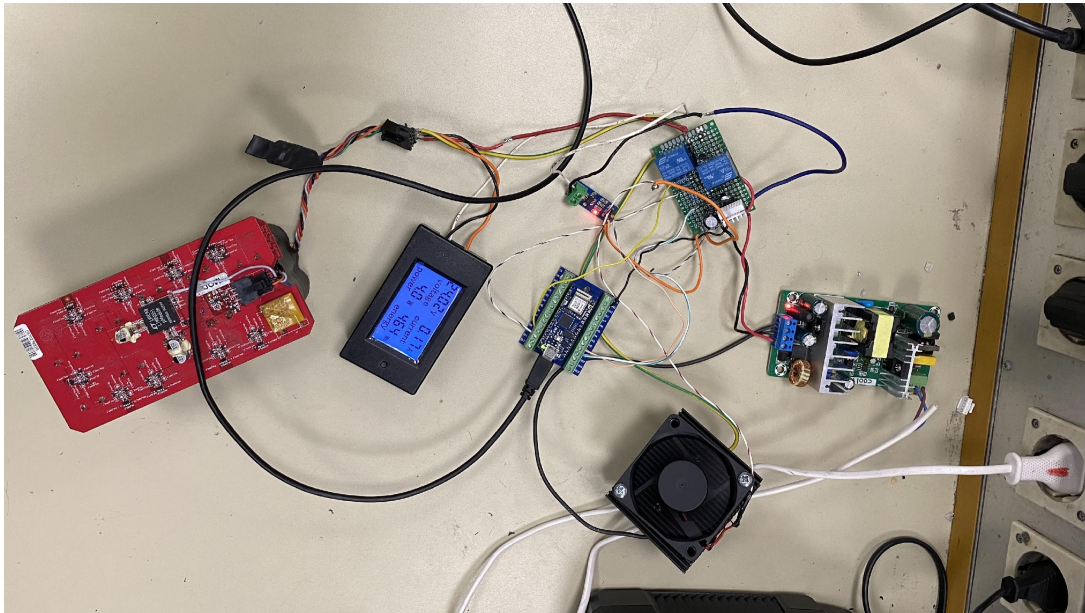


Figura 7. Circuito completo.

En la figura 7 se puede ver el circuito completo, todas estas conexiones son las que hacen capaz que se cumplan las funciones programadas para la aplicación. A continuación, se muestra dividido por partes para su mejor comprensión.

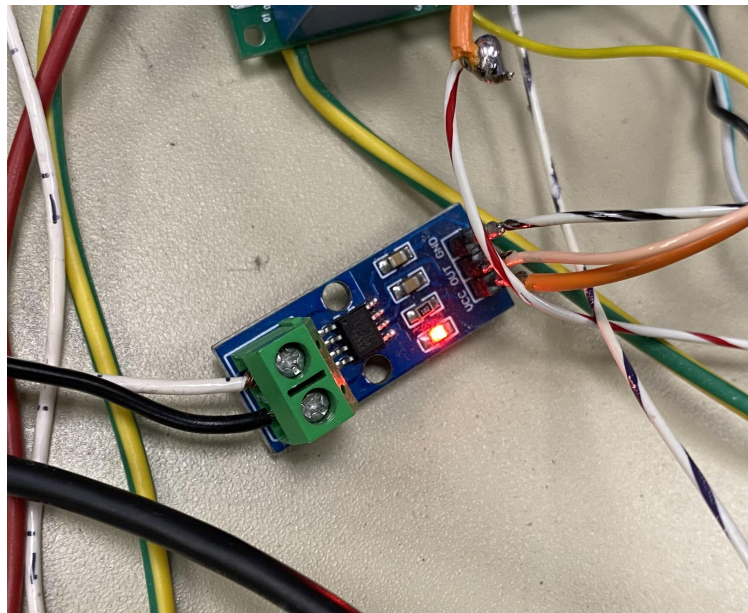
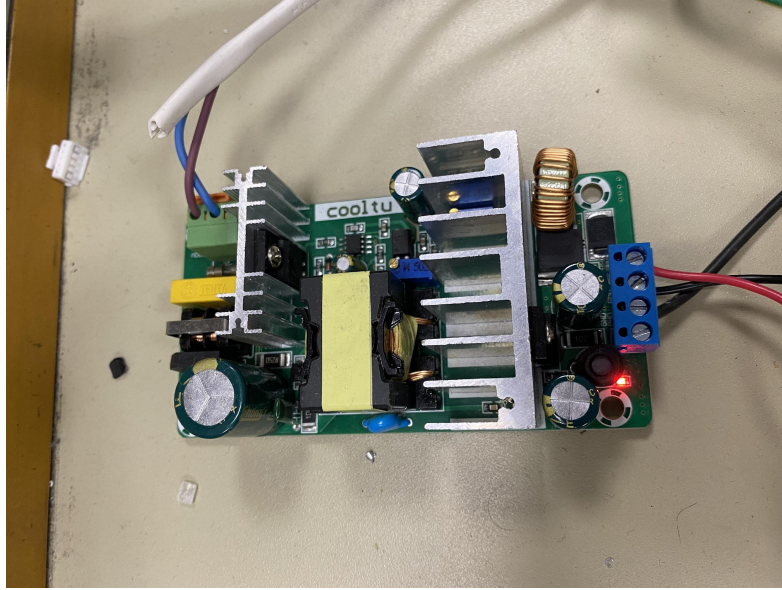
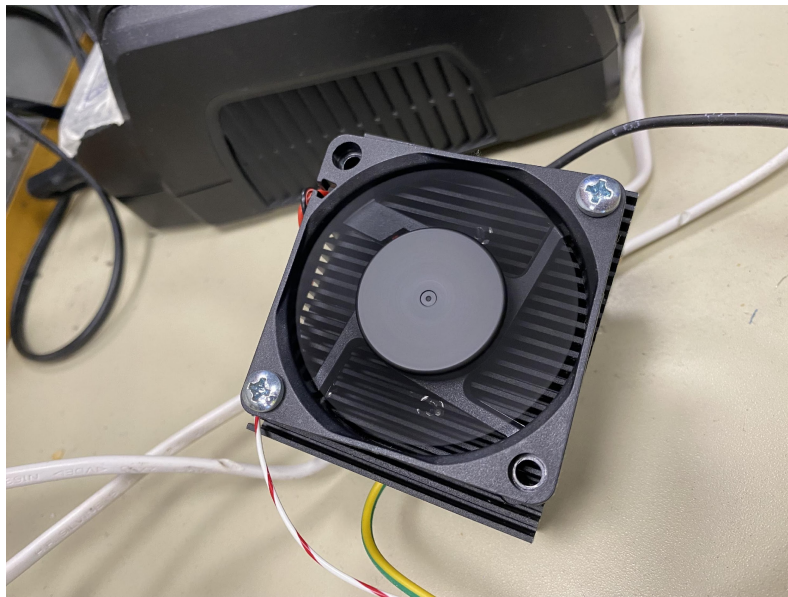


Figura 8. Sensor de consumo.



*Figura 9. Fuente de alimentación.*



*Figura 10. Resistencia de descarga con ventilador.*

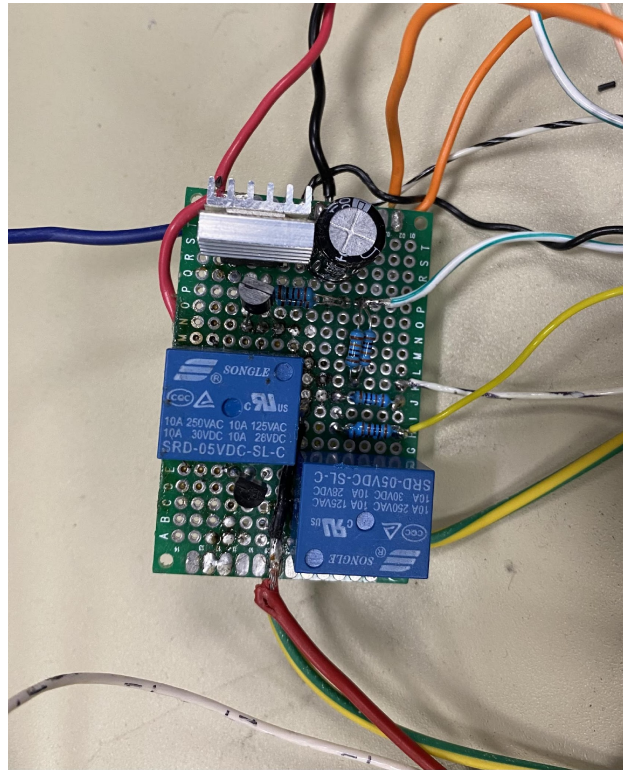


Figura 11. Circuito de mando.



Figura 12. Muestra de mediciones.

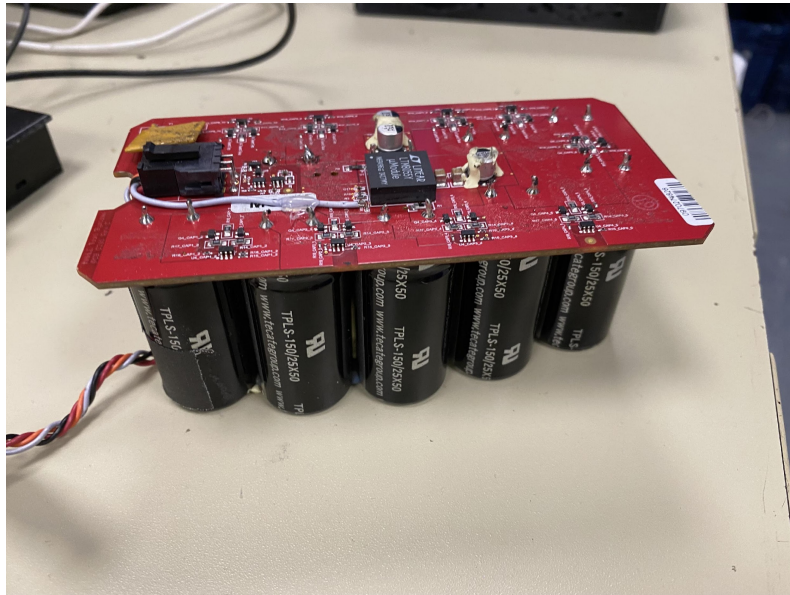


Figura 13. Ultracaps.

Estas serían todas las partes que componen el circuito que hacen que funcione correctamente. La figura 11 es una de las partes más importantes, este circuito de mando cuenta con un MOSFET, un condensador, 2 transistores y 2 relés para activar y desactivar la carga y la descarga.

A través de la pantalla de la figura 12 se puede ver las mediciones que se realizan, pero estas mismas también pueden ser vistas a través de la aplicación móvil. Y en la figura 13 se observan los ultracaps sobre los cuales se hacen las comprobaciones de su correcto funcionamiento.

Todo este conjunto forma la parte física del proyecto, a todo esto, hay que sumarle la programación realiza en los distintos lenguajes (Arduino, JavaScript, HTML) necesaria para el correcto funcionamiento. En la cual el usuario principalmente a través de la aplicación deberá buscar el Arduino en las conexiones bluetooth. Una vez conectado solicitará que se le muestre la intensidad y el voltaje por pantalla, y seguidamente elegirá la acción que quiera realizar, esto accionará los relés para empezar la carga o la descarga de los ultracaps o si por lo contrario quiere detener el proceso contara con una opción de detención, el cual parará tanto la carga como la descarga.

## 4.2 Diseño detallado

A continuación, se detalla el funcionamiento de la aplicación partiendo del mapa de navegación. Después, se explica el proceso de comprobación del estado de una batería.

### 4.2.1 Mapa de navegación

En la siguiente figura se observa el mapa de navegación de la aplicación. Los distintos apartados con los que cuenta y por los que navegara el usuario.

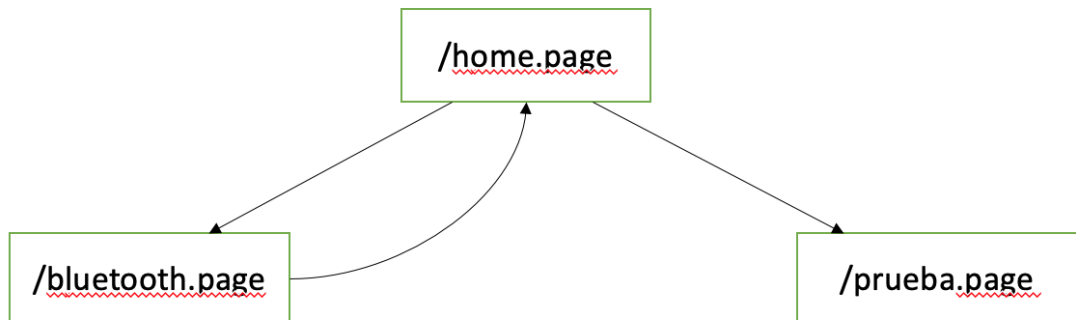


Figura 14. Mapa de navegación de la aplicación.

El nombre de la aplicación es BatteryTest y a continuación se explicará el funcionamiento de cada apartado.

- Página principal (/home):** Aquí se muestran los distintos botones, en la parte superior derecha aparece una cruz por si se quiere cerrar la aplicación. En azul se encuentra el primer botón para empezar un escaneo completo de dispositivos, este botón llevará a la página ' /bluetooth.page '. Abajo se cuenta con dos campos de texto (Voltaje y Amperios), donde, más adelante enseñará los datos de la batería conectada. Más abajo está el botón medir, el cual hace que una vez la conexión bluetooth esta activada y conectada muestre los valores por pantalla. El botón verde inicia el proceso de comprobación del estado, pero para esto se tiene que estar previamente conectado. El amarillo aplica una descarga por si se necesita reducir el voltaje de una forma rápida. Y el rojo de stop, finaliza cualquier proceso por si es requerido en algún momento, ya sea por sobrecalentamiento de la batería, problemas en el circuito o cualquier cosa. Un último campo de texto imprimirá el resultado final de la batería. Y, además, en la parte inferior izquierda se cuenta con un botón en forma de bombilla, el cual te lleva a la página ' /prueba.page ', aquí se explica el funcionamiento de cada botón, por si es requerido por algún usuario, y el funcionamiento del proceso de medición, para saber lo que hace el circuito en todo momento.



Figura 15. Pantalla de inicio de la aplicación.

- **Página bluetooth (/bluetooth):** Una vez se entre a esta página, cargarán todos los dispositivos bluetooth cercanos al móvil. Desde aquí se buscará el Arduino, en este caso llamado 'BateriaTest' y una vez seleccionado se emparejará. Seguidamente, mandará al usuario de vuelta a la página principal.

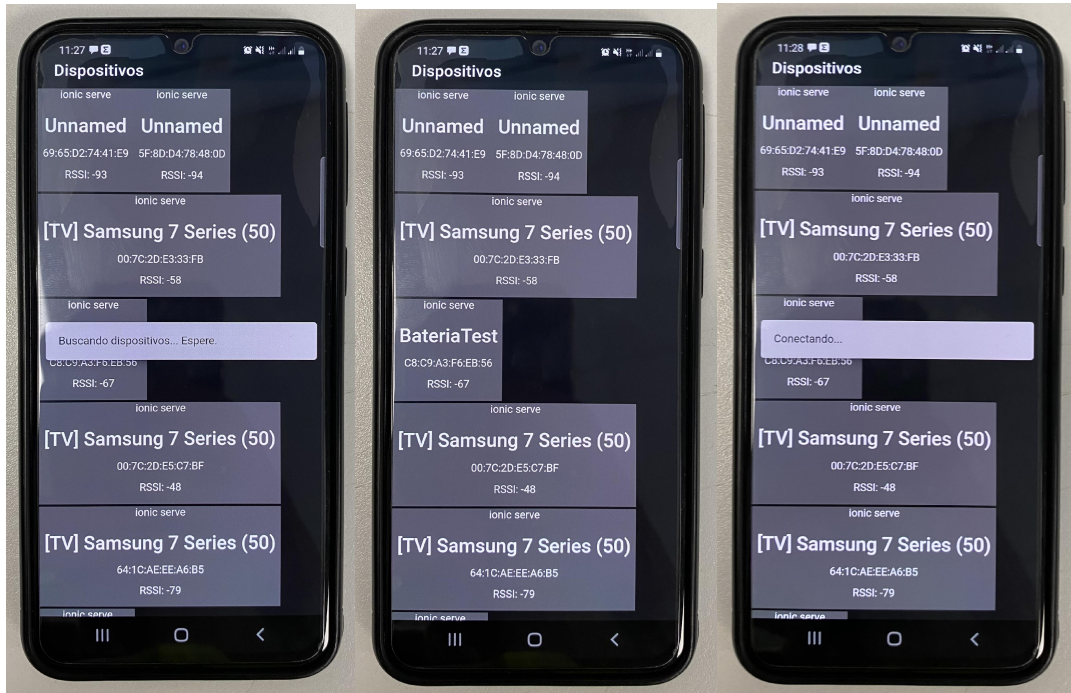


Figura 16. Interfaz página bluetooth.



- **Página prueba (/prueba):** Se accede aquí cuando se pulsa el icono de la bombilla en la pantalla de inicio. En esta pestaña se encuentra una guía con las instrucciones sobre la función de cada botón y, además, explica el proceso de realización de la medición de baterías.

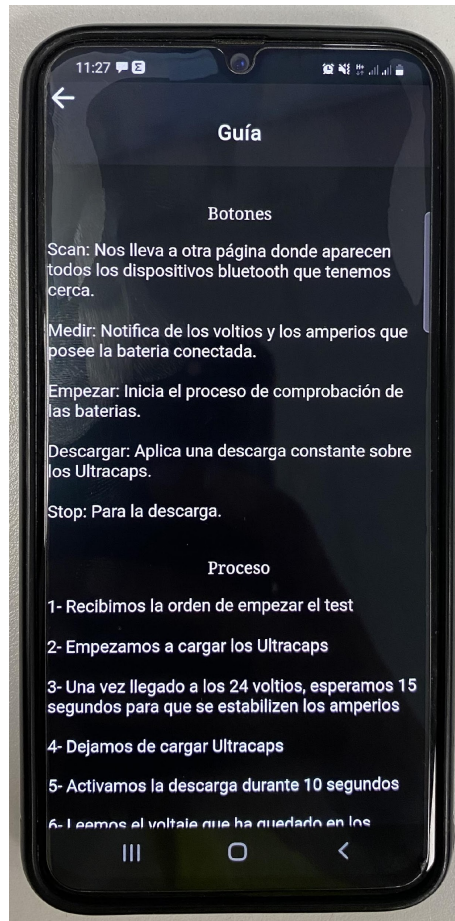


Figura 17. Página de guía sobre la aplicación.

#### 4.2.2 Selección empezar comprobación del estado de la batería

Una vez pulsado el botón 'Empezar' se realizan todos los pasos necesarios para determinar el estado de la batería. En la figura 18 vemos un diagrama del proceso.

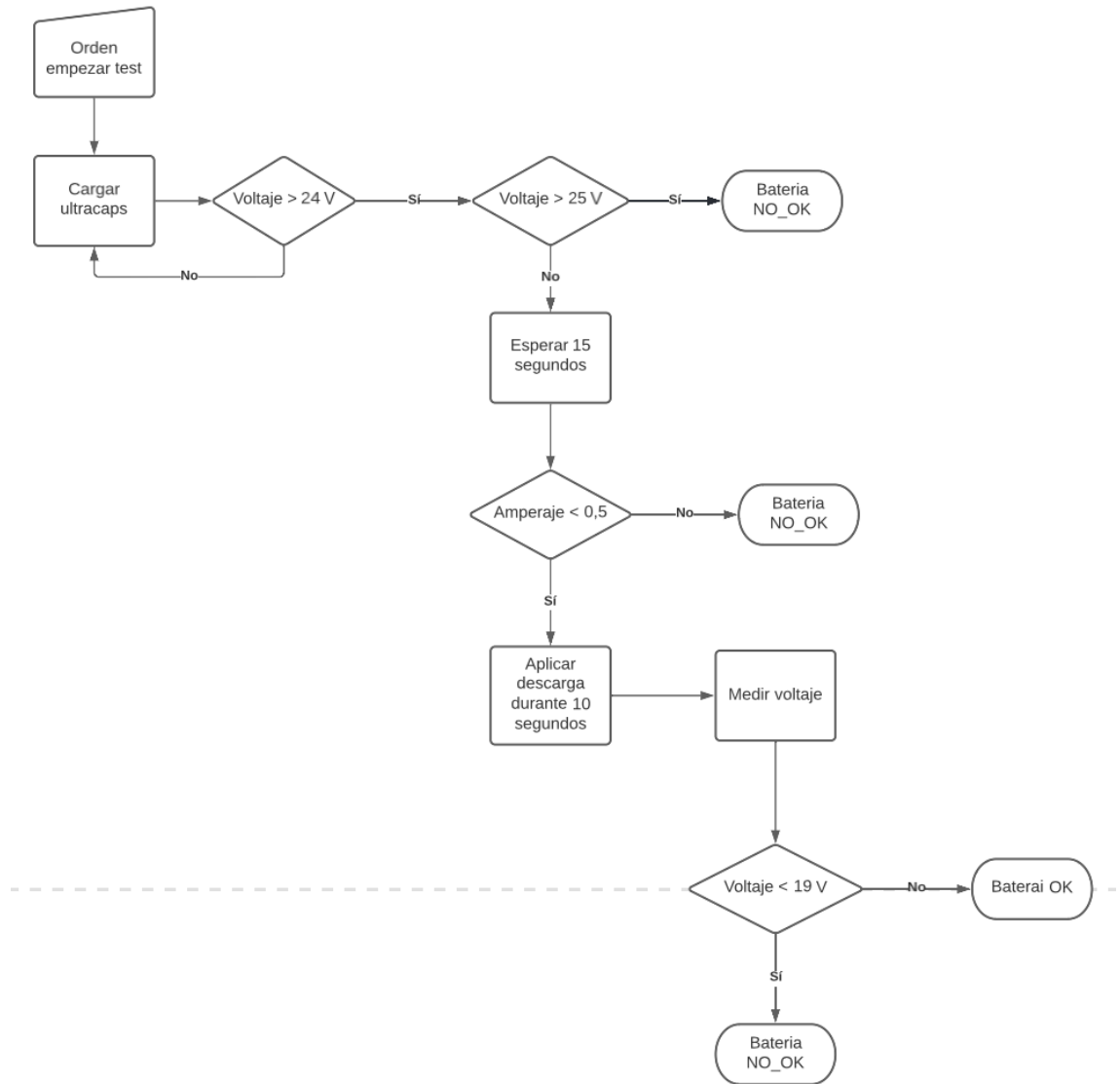


Figura 18. Diagrama de flujo test comprobación estado batería.

Las comprobaciones de las decisiones se realizarán en el lenguaje Arduino, quedando, por ejemplo, de la siguiente manera:

```

if(valor_voltaje2 < 19){
  Serial.println("BATERIA NO_OK");
  resultado2.writeValue(String("NO_OK, voltaje bajo"));
  digitalWrite(Ultracaps_nok, HIGH);
}else
{
  Serial.println("BATERIA OK");
  resultado2.writeValue(String("BATERIA OK"));
  digitalWrite(Ultracaps_ok, HIGH);
}
  
```

Figura 19. Código Arduino comprobación voltaje.

Esta figura 19, es la última comprobación en la cual se llega a la conclusión de si el estado de la batería es correcto o incorrecto. En caso de tener una carga superior a 19 Voltios, se considera que la batería está en buen estado (OK), en el caso contrario, si el voltaje es inferior, la batería estará en mal estado (NO OK).

Al igual que el tiempo de espera y activar/desactivar carga/descarga también se realiza desde Arduino.

```
digitalWrite(Cargar_ultracaps, LOW); //desactivar carga de ultracaps  
digitalWrite(Descargar_ultracaps, HIGH); //activar descarga de ultracaps
```

Figura 20. Control carga/descarga Arduino

En esta última figura se observa como deja de cargar y empieza a descargar. Es muy importante que nunca este la carga y la descarga activas a la vez.

El resto de las partes, como es la interfaz y su diseño, la conexión entre dispositivo móvil y Arduino, todo es programado a través de JavaScript y sus frameworks requeridos.

## 4.3 Tecnología utilizada

A continuación, se encuentran las herramientas, lenguajes, tecnologías y frameworks empleados para el desarrollo de este proyecto.

### 4.3.1 Herramientas para el desarrollo

#### Visual Studio Code<sup>[4]</sup>

Es un editor de código fuente desarrollado por Microsoft para Windows, Linux, macOS y Web. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código. También es personalizable, por lo que los usuarios pueden cambiar el tema del editor, los atajos de teclado y las preferencias. Es gratuito y de código abierto,<sup>12</sup> aunque la descarga oficial está bajo software privativo e incluye características personalizadas por Microsoft.



### Arduino<sup>[5]</sup>

Arduino es una plataforma electrónica de código abierto basada en hardware y software fáciles de usar. Las placas Arduino pueden leer entradas y convertirlas en una salida: activar un motor, encender un LED, publicar algo en línea. Puede decirle a su placa qué hacer enviando un conjunto de instrucciones al microcontrolador en la placa. Para ello se utiliza el lenguaje de programación Arduino (basado en Wiring), y el Software Arduino (IDE), basado en Processing.



### Ultimaker Cura<sup>[6]</sup>

Es una aplicación diseñada para impresoras 3D, en la que se pueden modificar los parámetros de impresión y después transformarlos a código G. Fue creada por David Braan, que después de un tiempo trabajaría para Ultimaker, una empresa dedicada al diseño y fabricación de impresoras 3D.



## 4.3.2 Lenguajes para su programación

### JavaScript<sup>[7]</sup>

Es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. Se utiliza principalmente del lado del cliente, implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas.



### Arduino<sup>[8]</sup>

El lenguaje estándar es C++, aunque es posible programarlo en otros lenguajes. No es un C++ puro, sino que es una adaptación proveniente de avr-libc que provee de una librería de C de alta calidad para usar con GCC en los microcontroladores AVR de Atmel y muchas funciones específicas para los MCU AVR de Atmel.

## HTML<sup>[9]</sup>

Siglas en inglés de HyperText Markup Language (lenguaje de marcado de hipertexto), hace referencia al lenguaje de marcado para la elaboración de páginas web. Es un estándar que sirve de referencia del software que conecta con la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código para la definición de contenido de una página web, como texto, imágenes, videos, juegos, entre otros.



## Node.JS<sup>[10]</sup>

Es un entorno de ejecución para JavaScript, del cual requerimos para poder realizar instalaciones de paquetes, con el comando 'npm'.



## CSS<sup>[11]</sup>

Es un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado. Utilizado para describir la presentación de documentos HTML o XML.



### 4.3.3 Frameworks utilizados

## Angular<sup>[12]</sup>

Es un framework para aplicaciones web desarrollado en TypeScript, de código abierto, mantenido por Google, que se utiliza para crear y mantener aplicaciones web de una sola página. Su objetivo es aumentar las aplicaciones basadas en navegador con capacidad de Modelo Vista Controlador (MVC), en un esfuerzo para hacer que el desarrollo y las pruebas sean más fáciles.



### **Ionic**<sup>[13]</sup>

Es un SDK (Kit de desarrollo de software) de frontend de código abierto para desarrollar aplicaciones híbridas basado en tecnologías web (HTML, CSS y JS). Es decir, un framework que nos permite desarrollar aplicaciones para iOS nativo, Android y la web, desde una única base de código.



## 5. Desarrollo de la solución propuesta

---

En este apartado vamos a ver como se ha llegado a la solución, el proceso con el cual se consigue el objetivo establecido y algunos de los problemas y dificultades que más impedimentos han supuesto para cumplir con el trabajo.

### 5.1 Propuesta de la solución final

Principalmente, se estableció la propuesta diseñando el proceso para que posteriormente fuese implementado con el código necesario.

Lo que fue requerido convertir a código para el correcto funcionamiento fue el siguiente proceso:

- 1- Conectamos los dispositivos por bluetooth.
- 2- Recibimos la orden de empezar el test a través del móvil.
- 3- Damos la orden de empezar a cargar los Ultracaps (D10).
  - a. Monitorizamos tanto el Voltaje como el Amperaje de carga
    - i. Si el voltaje pasa por encima de 25v, ultracaps NOK Voltaje alto (información a mostrar en móvil)
    - ii. El amperaje, cuando pasen los ultracaps de 24v debe de ser menor a 0,5Amperios
      1. Esperar 15 segundos después de pasar de 24V para ver si se estabiliza por debajo de este amperaje, si no los damos como NOK
    - iii. Si los ultracaps pasan de 24V y el amperaje se estabiliza después de los 15s procedemos al paso 3
- 4- Dejamos de cargar ultracaps (D10 a 0).
- 5- Activamos la salida de descarga de ultracaps (D9) durante 10s.
- 6- Leemos el voltaje que ha quedado en los ultracaps, si está por debajo del límite establecido, los damos como NOK, si está por encima, los damos como OK.

\*OK: correcto; NOK: no funciona correctamente

\*La información del tipo de fallo se debe de mostrar en el móvil

\*La entrada de STOP test, detendrá el test en cualquier momento.

\*D10 y D9, nunca deben de poder estar activas a la vez, hay que programar un bloqueo en el software para que esto nunca ocurra. Ej. Si (D10 ==1) No se ejecuta nunca la rutina de D9

Las iniciales que aparecen como 'Dx' son las salidas del Arduino que se vieron anteriormente.



A continuación, se detallan las herramientas con las que se ha programado cada parte de la aplicación para entender su complejidad de conexión.

Programación correspondiente según herramienta empleada:

	Arduino	Visual Studio Code
Interfaz		√
Conexión bluetooth	√	√
Cambio entre páginas		√
Carga/Descarga	√	

Tabla 2. Programación según herramienta

La mayor parte de la aplicación de Visual Studio va a ser diseñada en el directorio /src/app:

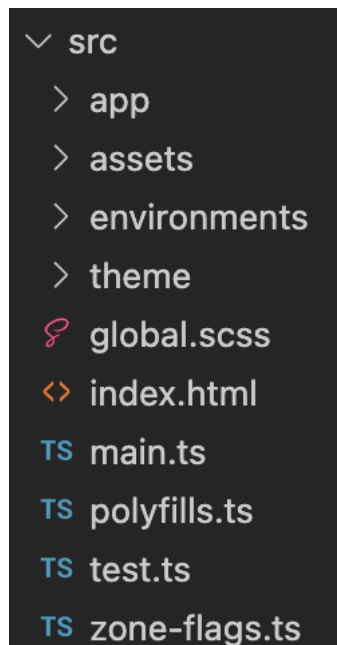


Figura 21. Directorio 'src' de la App

## 5.2 Problemas durante el desarrollo

En relación con la escritura del código, donde más problemas se encontraron fue en conseguir la conexión bluetooth entre el Arduino y el dispositivo móvil. Primero, se debía conseguir que mostrase todos los dispositivos, una vez los enseñara por pantalla tener la opción de seleccionar el que fuese necesario y finalmente, que estableciese la conexión con este. Se necesitaron unas cuantas clases enlazadas entre sí para



conseguirlo. También se empleó una clase para el caso en el que el bluetooth se desconecte o se pierda la conexión.

Además, para poder comprobar el correcto funcionamiento, se requería hacer las pruebas directamente desde el móvil, ya que el ordenador no contaba con bluetooth.

Todo este proceso, al igual que las acciones que se realizaban cuando se comprobaba el estado de la batería, se mostraba un mensaje en la parte inferior de la pantalla para que el usuario pudiera ver en todo momento en qué situación se encontraba, ya que sin esto muchas veces no se sabía si la conexión había sido efectiva o, por lo contrario, había algún fallo.

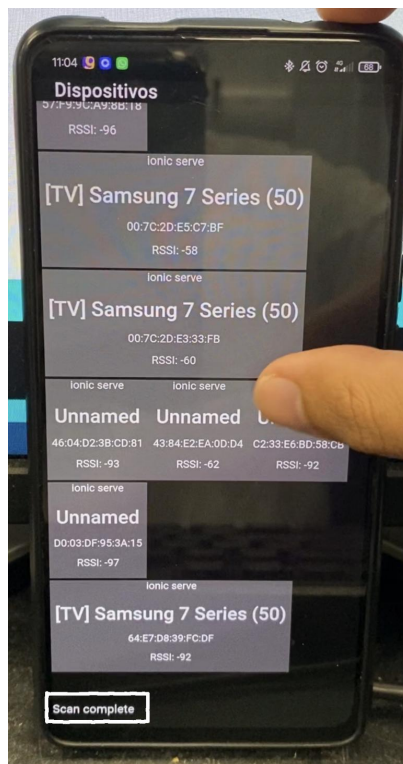
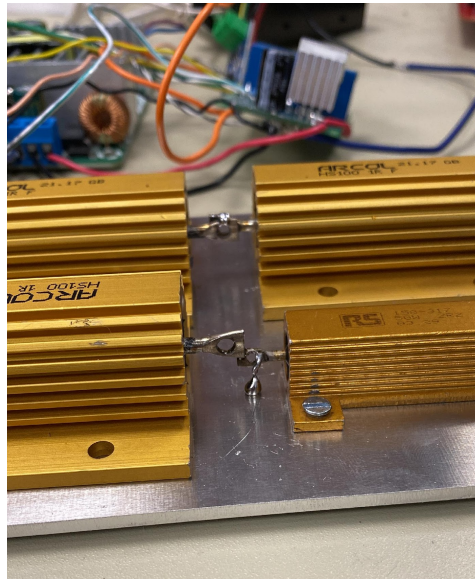


Figura 22. Estado conexión

Otro problema que se presentó fue con el circuito, en primer lugar, por las resistencias ya que no tenían la potencia suficiente para aplicar la carga necesaria a corto plazo de tiempo, por lo que se tuvieron que cambiar por otras más potentes.

Y en segundo lugar hubo fallos en las conexiones del circuito. Al principio no se encontraba el problema, y después de verlo todo con detalle se detectó un mal contacto entre las resistencias del circuito, debido al movimiento inadecuado del estaño. Esto hacía que se cortara la corriente y por tanto no se accionaban las tareas que debía realizar el circuito.



*Figura 23. Mal contacto entre resistencias*

Este problema se solucionó con una nueva soldadura, una vez se mantuvo el contacto entre las resistencias funcionó sin ningún problema.

## 6. Implantación

---

El proyecto se desarrolló e implemento como estaba planeado, construyendo todo el código desde cero y posteriormente exportándolo a '.apk' para las pruebas en el dispositivo Android.

Se empleó el uso de una gran cantidad de librerías para tener de sus servicios, tanto para conexiones bluetooth, como el paso de parámetros entre páginas y para la navegación de estas.

Por ejemplo, estas son todas las importaciones realizadas para la página principal:

```
import { Component, NgZone } from '@angular/core';
import { NavController, NavParams } from '@ionic/angular';
import { Router } from '@angular/router';
import { enableProdMode } from '@angular/core';
import { BLE } from '@awesome-cordova-plugins/ble/ngx';
import { MenuController } from '@ionic/angular';
import { ActivatedRoute } from '@angular/router';
import { ToastController } from '@ionic/angular';
```

Figura 24. Importaciones /home.page

Y estos son los plugins empleados:

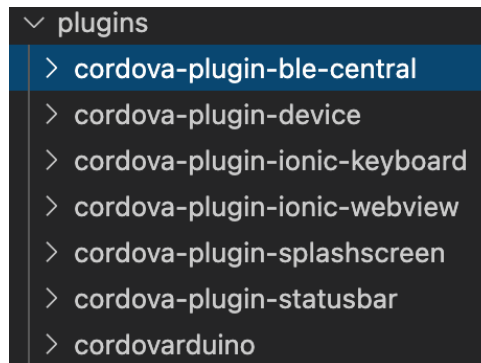


Figura 25. Plugins utilizados.

## 7. Pruebas

---

La aplicación requería de pruebas constantes para verificar sus modificaciones y la eficacia de su funcionamiento. Cada vez que una modificación fuese realizada y se necesitara hacer una prueba, se debía exportar la '.apk' y pasarla al dispositivo móvil. A la misma vez, el circuito debía estar conectado a la luz para tener corriente.

### 7.1 Pruebas en el circuito

En un primer momento se montó un pequeño circuito con el que poder ir probando la aplicación, pero aparecieron los primeros problemas.

El principal impedimento fueron las resistencias, ya que, no tenían la potencia suficiente para aplicar la descarga en un tiempo adecuado, por lo que probamos con distintas resistencias hasta dar con la más adecuada.

Y, en segundo lugar, el circuito alcanzaba temperaturas elevadas y por seguridad, se incluyeron unos ventiladores para mantener una temperatura más baja y estable.

Una vez, todo el circuito montado y funcionando correctamente, se tenía que probar y solucionar los problemas que pudieran aparecer en la aplicación.

### 7.2 Pruebas en la implementación de código

Los primeros pasos para poder utilizar el programa era la conexión bluetooth. Esta parte es donde más problemas aparecieron por la dificultad de conseguir una conexión satisfactoria.

Se realizaron una gran cantidad de pruebas hasta conseguir el código eficiente, además para comprobar si la conexión funcionaba correctamente se tenía que exportar cada vez la '.apk' e instalarla en el móvil.

### 7.3 Pruebas de la aplicación

Una vez, la aplicación completa y con una interfaz bien estructurada, se realizaron las pruebas para verificar su correcto funcionamiento.

Se tuvieron que realizar varias pruebas y en diversas situaciones para comprobar que funcionaba en cualquier estado. Además, de comprobar que no se diese ningún fallo, y que el dispositivo móvil soportaba bien la aplicación.

## 8. Conclusiones

---

Terminando casi con el recorrido de este proyecto, queda concluir enseñando el cumplimiento de los objetivos principalmente expuestos. Con un gran tiempo de aprendizaje, de pruebas y de errores, finalmente la aplicación cumple con su trabajo.

El primer objetivo, el cual era ayudar a los trabajadores con el tiempo empleado para estas comprobaciones, fue cumplido con altas expectativas, ya que a los trabajadores les ayudó el programa y obviamente tenían mucho más tiempo para realizar otras tareas.

Con la pestaña de ayuda que aparece en la aplicación se consigue que esté al alcance de cualquier usuario y sepa cómo usarla. Ya que, no todos están familiarizados con estos dispositivos y tecnologías.

Una parte difícil de conseguir, propuesta como objetivo fue el funcionamiento a través de bluetooth, ya que es un punto muy importante para que pueda funcionar el trabajo.

Y, por último, lograr que se accione sin fallos el circuito, aprendiendo nuevas tecnologías. Al principio no estaba muy claro por la poca experiencia en este trabajo, pero con ayuda se consiguió un circuito que cumpliera todo lo requerido.

Por lo tanto, decir lo satisfecho que se estuvo con esta aplicación y la experiencia que aporta personalmente, además de los conocimientos adquiridos para futuros trabajos. Gracias a esto, se ha estado desarrollando en distintos ámbitos y ha ayudado a saber manejar una gran variedad de softwares, incluso alguno empezando desde cero. También esto ha hecho crecer como trabajador, donde aprendes que es muy importante la dedicación y la constancia.

### 8.2 Relación del trabajo desarrollado con los estudios cursados

En este proyecto se han empleado una gran cantidad de conocimientos adquiridos a lo largo de los 4 años de la carrera de ingeniería informática, y a la vez se han adquirido otros a lo largo de todo el proceso. Gran parte de haber conseguido este trabajo ha sido gracias a la programación, ya que se ha requerido en todo momento saber desarrollarse en estos entornos de programación y tener la capacidad de solventar los problemas.

Muchas tecnologías de las que se han utilizado, vistas anteriormente, no suelen ser usadas en una gran cantidad de asignaturas durante la carrera, pero las relacionadas te ofrecen una base de conocimiento que con la práctica se desarrolla y así consiguiendo una gran experiencia en el ámbito.

Obviamente todo lo aprendido de JavaScript fue fundamental para este proyecto, junto la programación de HTML. Se necesitó juntar estos conocimientos con aprender Angular e Ionic<sub>[16]</sub>, los cuales no se han visto con el detalle necesario para poder



completar este trabajo. Pero con dedicación e investigado se aprende, y esta es la forma en la que un estudiante recién graduado que entra al mundo laboral aprende y obtiene conocimientos. Lo mismo pasa con utilizar Arduino, y diseñar el circuito, con lo cual se requirió de ayuda. Además, de ver desde cero la utilización e impresión de 3D con distintos softwares.

### 8.3 Competencias transversales

Respecto a las competencias transversales, las más trabajadas serían las siguientes:

<b>Competencias transversales</b>	
Análisis y resolución de problemas	Por la cantidad de problemas que han aparecido, y todas las búsquedas que se han tenido que realizar para conseguir solucionar estos problemas.
Comprensión e integración	Se ha visto una gran variedad de campos nuevos y de conceptos no conocidos previamente.
Instrumental específica	Porque como se ha dicho, se han tenido que aprender nuevas tecnologías.

*Tabla 3. Competencias transversales*

## 9. Futuros trabajos

---

En principio los objetivos expuestos son los mencionados en la memoria y los que finalmente han sido conseguidos, pero estaba la opción de un futuro mejorar esta misma aplicación.

Las mejoras que se podrían haber implementado son centradas en un registro de baterías, es decir, cada vez que se hiciese la comprobación de una batería, poder almacenar los datos y su resultado, por si en un futuro vuelve a fallar la misma poder consultar un historial.

Se pensó aplicar esta mejora añadiendo otra pestaña en la aplicación que mostrara el ID de la batería, y a continuación, la fecha del proceso en que se realizó la comprobación y su solución.

Finalmente, no se pudo llevar a cabo por falta de tiempo en la empresa.



## 10. Referencias

---

[1]: "MODELO WATERFALL O EN CASCADA",

<[https://www.dtagency.tech/cursos/metodologias\\_gestion\\_proyectos/tema\\_1-ModeloWaterfall.pdf](https://www.dtagency.tech/cursos/metodologias_gestion_proyectos/tema_1-ModeloWaterfall.pdf)>

[2]: "El impacto de cinco tecnologías en la fabricación de vehículos de Ford",

<<https://thelogisticsworld.com/manufactura/cinco-tecnologias-que-han-potenciado-la-fabricacion-de-vehiculos-de-ford/>>

[3]: Servicios en Angular,

<<https://desarrolloweb.com/articulos/servicios-angular.html> >

[4] Juan Diego Gauchat. El gran libro de HTML5 CSS3 y JavaScript.

MARCOMBO. (2012)

[5]: Visual Studio Code,

<<https://code.visualstudio.com>>

[5]: Arduino plataforma,

<<https://www.arduino.cc>>

[6]: Ultimaker cura,

<<https://ultimaker.com/es/software/ultimaker-cura>>

[7]: JavaScript, documentation

<<https://developer.mozilla.org/es/docs/Web/JavaScript>>

[8]: Arduino lenguaje,

<<https://www.arduino.cc/reference/es/>>

[9]: HTML,

< <https://developer.mozilla.org/es/docs/Web/HTML>>

[10]: Node.JS, documentation

< <https://nodejs.org/en/docs/> >

[11]: CSS, documentation

<<https://developer.mozilla.org/es/docs/Web/CSS>>



[12]: ANGULAR. Angular.io.

<<https://angular.io/docs>>

[13]: Ionic,

<<https://ionicframework.com>>

[14] Bluetooth Low Energy (BLE) Central Plugin for Apache Cordova,  
<https://github.com/don/cordova-plugin-ble-central>

[15] ¿Qué es un MOSFET y para qué sirve?,  
<<https://www.geeknetic.es/MOSFET/que-es-y-para-que-sirve>>

[16] Ionic : Hybrid Mobile App Development : Khanna, Rahat, Yusuf, Sani, Phan,  
Hoc (2017)



## Anexo 1. Pruebas funcionamiento

---

En este apartado se muestran las pruebas realizadas con el trabajo finalizado, se ve y se explica el funcionamiento de la aplicación como estaba planeado y cómo reacciona en cada situación.

En primer lugar, se muestra que pasaría si la batería estuviera en mal estado:

<https://drive.google.com/file/d/1SebXIQ4rDp2xniHIGUiPQeXdityEKHj2/view>

En este Segundo caso la batería está en buen estado y así es como debería reaccionar la aplicación:

<https://drive.google.com/file/d/1J0TDUQFivtCd27xWuFfEVmO-HwfdFDjw/view>

Además, se adjuntan unos videos para observar cómo se ve y funciona la página de guía para el usuario y la conexión de bluetooth.

En el siguiente video se ve la información que almacena la página de Guía:

[https://drive.google.com/file/d/1IW\\_1RaTTzaTrNC-DUTW4e7gmUtIFNSDp/view](https://drive.google.com/file/d/1IW_1RaTTzaTrNC-DUTW4e7gmUtIFNSDp/view)

Por último, se grabó el proceso con el cual se consigue la conexión a través de bluetooth:

<https://drive.google.com/file/d/16PEKfb1lobXanTCkpz48hIgb1N8QTNsj/view>

## Anexo 2. Funcionamiento ngOnInit y constructor

---

El constructor siempre se ejecuta en primer lugar al cambiar de página, pero la función ngOnInit hace que ejecute el código por primera vez cuando carga la página, este método pertenece a Angular.

Para que un componente de Angular pueda ejecutar el ngOnInit se tiene que importar OnInit de Angular:

```
import { Component, OnInit } from '@angular/core';
```

Figura 26. Importación OnInit

El componente tiene que implementar en OnInit y meter el método ngOnInit dentro del componente:

```
import { Component, OnInit } from '@angular/core';

@Component({
  selector: 'app-home',
  templateUrl: './home.component.html',
  styleUrls: ['./home.component.css']
})

export class HomeComponent implements OnInit {
  constructor() { }
  ngOnInit() {
    console.log("En este instante el componente ha cargado")
  }
}
```

Figura 27. Método ngOnInit

Este es solo un ejemplo en el que se lanza un console.log para que sea más fácil de comprender.

Hay que entender la diferencia entre el método `ngOnInit` y el constructor. Este segundo se ejecuta antes que el `ngOnInit`. Por lo general, los constructores se usan para inicializar variables, mientras que `ngOnInit` se usa para inicializar o realizar tareas relacionadas con Angular. Podríamos poner todo esto directamente en el constructor y funcionaría de la misma manera, pero vale la pena separar más el código para que sea más fácil de mantener.

## Anexo 3. Conexión BLE

---

Una de las partes más complicadas ha sido conseguir la conexión bluetooth entre el dispositivo móvil y el Arduino. A continuación, se muestra como funcionaría una conexión exitosa, explicada a través del código.

Se empieza con un escaneo de dispositivos cercanos nada más se cambia a la pestaña de 'bluetooth'. Esto pasa, ya que, el código de escanear está dentro del constructor.

```
this.ble.scan([], 5).subscribe(  
  device => this.onDeviceDiscovered(device), // busca los dispositivos disponibles  
  error => this.scanError(error) // en caso de error  
);
```

Figura 28. Escáner de dispositivos bluetooth cercanos

El método 'onDeviceDiscovered' recoge la información de los dispositivos:

```
onDeviceDiscovered(device) { //coge los dispositivos disponibles  
  console.log('Discovered ' + JSON.stringify(device, null, 2));  
  this.ngZone.run(() => {  
    this.devices.push(device);  
  });  
}
```

Figura 29. Código dispositivos encontrados

Una vez todos los dispositivos aparecen, se selecciona el requerido por el usuario y se procede el siguiente código:

```
deviceSelected(device) { // selecciona dispositivo  
  console.log(JSON.stringify(device) + ' selected');  
  
  this.setStatus('Connecting to ' + device.name || device.id);  
  
  this.ble.connect(device.id).subscribe(  
    peripheral => this.onConnected(peripheral),  
    peripheral => this.onDeviceDisconnected(peripheral))  
}
```

Figura 30. Selección de dispositivo

En caso de conectarse:

```
onConnected(peripheral) { //cuando esta conectado
  this.ngZone.run(() => {
    this.setStatus('conectado');
    this.peripheral = peripheral;
    this.router.navigate(['/home'])
  });
}
```

Figura 31. Dispositivo conectado

Una vez conectado se volvería a la página principal 'home' y ya se podría seguir con el proceso.

## Anexo 4. Objetivos de desarrollo sostenible

---

Grado de relación del trabajo con los Objetivos de Desarrollo Sostenible (ODS).

<b>Objetivos de Desarrollo Sostenibles</b>	<b>Alto</b>	<b>Medio</b>	<b>Bajo</b>	<b>No Procede</b>
ODS 1. <b>Fin de la pobreza.</b>				<b>X</b>
ODS 2. <b>Hambre cero.</b>				<b>X</b>
ODS 3. <b>Salud y bienestar.</b>				<b>X</b>
ODS 4. <b>Educación de calidad.</b>				<b>X</b>
ODS 5. <b>Igualdad de género.</b>				<b>X</b>
ODS 6. <b>Agua limpia y saneamiento.</b>				<b>X</b>
ODS 7. <b>Energía asequible y no contaminante.</b>				<b>X</b>
ODS 8. <b>Trabajo decente y crecimiento económico.</b>	<b>X</b>			
ODS 9. <b>Industria, innovación e infraestructuras.</b>			<b>X</b>	
ODS 10. <b>Reducción de las desigualdades.</b>				<b>X</b>
ODS 11. <b>Ciudades y comunidades sostenibles.</b>				<b>X</b>
ODS 12. <b>Producción y consumo responsables.</b>			<b>X</b>	
ODS 13. <b>Acción por el clima.</b>				<b>X</b>
ODS 14. <b>Vida submarina.</b>				<b>X</b>
ODS 15. <b>Vida de ecosistemas terrestres.</b>				<b>X</b>
ODS 16. <b>Paz, justicia e instituciones sólidas.</b>				<b>X</b>
ODS 17. <b>Alianzas para lograr objetivos.</b>				<b>X</b>



## **Reflexión sobre la relación del TFG/TFM con los ODS y con el/los ODS más relacionados.**

El trabajo realizado ofrece una importante repercusión en el ámbito de crecimiento de la empresa. No un crecimiento físico, pero sí de tecnologías y de producción.

Por lo tanto el punto más destacado es el **trabajo decente y crecimiento económico**, el cual se puede ver fácilmente reflejado, ya que se han aportado unas mejoras que reducen el tiempo de realización de tareas y por lo tanto más tiempo para otros procesos, produciendo más beneficios en menos tiempo.

La meta de este objetivo que más lo representa sería el 8.2 ' Lograr niveles más elevados de productividad económica mediante la diversificación, la modernización tecnológica y la innovación, entre otras cosas centrándose en los sectores con gran valor añadido y un uso intensivo de la mano de obra'. Esto es justo lo que se ha conseguido con este proyecto, a partir de una idea de mejora e innovación se realizó este trabajo cumpliendo las expectativas y los objetivos propuestos inicialmente.

Otro objetivo de desarrollo sostenible sería la **industria, innovación e infraestructuras**, esta relación se puede atribuir principalmente a que el trabajo se realiza en un departamento en el cual se crean proyectos nuevos para innovar con la tecnología.

Este entorno facilita el avance de la industria de la informática, ya que el departamento en el que se ha desarrollado el proyecto utiliza técnicas de programación avanzadas.

También se ha tenido que hacer un proceso de investigación con el cual conseguir datos e información sobre los campos empleados, los softwares con los que se ha desarrollado todo el trabajo, y comprender el funcionamiento y estado de los componentes que forman el proyecto.

Un último objetivo con el cuál se relaciona, es la **producción y consumo responsables**. Este apartado es bastante importante porque en una empresa grande por pequeño que sea el consumo si es constante puede generar grandes devastaciones.

El punto más importante es saber que las baterías son reparadas, es decir, las baterías no son desechadas cuando aparece algún fallo. Esto reduce bastante la contaminación, ya que con la cantidad de baterías que cuenta cada factoría de la Ford, si se deshicieran de todas ellas por sus fallos aumentaría fácilmente la contaminación.

Respecto al consumo se puede decir que hay una reducción empleada, ya que anteriormente se requería conectar las baterías a la corriente y cargarlas completamente. Actualmente se aplica una pequeña carga y descarga en cuestión de segundos.

Además, empleamos un BLE (Bluetooth Low Energy), emite una señal de 2.4Ghz con un alcance de hasta 100 metros, con la gran ventaja de que consume muy poca energía y permite estar emitiendo la señal durante meses e incluso años sin tener que recargar la batería.



Como se mencionó anteriormente, a pesar de ser un trabajo con un impacto empresarial importante, es difícil vincularlo con los 17 puntos propuestos por los ODS. Esto es así porque las metas o puntos más sociales relacionados con la sostenibilidad no son particularmente relevantes en este trabajo.



## Glosario

---

**Atlas copco:** Baterías para funcionar y durar con pocos requisitos de mantenimiento bajo condiciones extremas. Constan con una mayor duración frente a ciclos de carga/descarga.

**Ultracaps:** Son ultra capacitores electrónicos capaces de almacenar y descargar energía de manera muy rápida y eficaz.

**APK:** Es un archivo propio de Android que se utiliza para instalar aplicaciones.

**Framework:** Es un conjunto estandarizado de herramientas, módulos de código y procedimientos establecidos para enfocar el desarrollo de alguna determinada tarea. En este proyecto se ha utilizado framework de Ionic y Angular.

**MOSFET:** Es un transistor, un componente eléctrico que se encarga de regular la salida de voltaje a partir de una tensión de entrada dada.

**Back-end:** parte trasera de una aplicación invisible para el usuario, encargada de gestionar los datos y la lógica necesarios para el intercambio de datos con un servidor o una base de datos.

**Front-end:** consiste en la transformación de los datos de una aplicación de modo que sean utilizables en una interfaz gráfica para facilitar la interacción del usuario con ellos. Referido a una web, es la parte visual con la que interactúa el cliente.

**Método:** es una subrutina cuyo código es definido en una clase y puede pertenecer tanto a una clase, como es el caso de los métodos de clase o estáticos, como a un objeto, como es el caso de los métodos de instancia.

**Diagrama de flujo:** representa la esquematización gráfica de un algoritmo, el cual muestra gráficamente los pasos o procesos a seguir para alcanzar la solución de un problema.

**BLE:** Bluetooth de baja energía es una tecnología wireless basada en el estándar Bluetooth 4.0, que gracias al uso de poca energía tiene transpondedores que siguen emitiendo un ping rate ajustable, emitiendo una señal de 2.4 GHz con un alcance de hasta 100 metros.

