



Sistema Apuntador Autónomo

Manuel Illescas Gómez

Grado de Ingeniería Informática

Arduino

Jose Lopez Vicario

Pere Tuset Peiró

7 Junio 2020



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>Sistema Apuntador Autónomo</i>
Nombre del autor:	<i>Manuel Illescas Gómez</i>
Nombre del consultor/a:	<i>Jose Lopez Vicario</i>
Nombre del PRA:	<i>Pere Tuset Peiró</i>
Fecha de entrega (mm/aaaa):	06/2020
Titulación:	<i>Grado Ingeniería Informática</i>
Área del Trabajo Final:	<i>Arduino</i>
Idioma del trabajo:	<i>Castellano</i>
Palabras clave	<i>Arduino, Posicionamiento, Localización</i>
Resumen del Trabajo:	
<p>Este proyecto presenta el diseño y desarrollo de un nuevo prototipo de dispositivo de sistema apuntador autónomo cuya funcionalidad principal será la de servir como método de guía para los participantes en actividades al aire libre como el geocaching.</p> <p>El sistema basa su desarrollo en la plataforma de código abierto Arduino, que permite la integración de los distintos módulos necesarios, un sensor de campo magnético y un sensor GNSS, que recolectan los datos de orientación y posición del dispositivo, una pantalla LCD para mostrar los datos de orientación al usuario y una batería que proporcionará la alimentación necesaria todo el conjunto.</p> <p>El dispositivo utiliza la información de los sensores para geolocalizar la ubicación y la orientación del sistema con respecto al norte geográfico e informa al usuario del rumbo a seguir para llegar al destino prefijado en la pantalla LCD.</p> <p>Asimismo, el prototipo provee, mediante una comunicación inalámbrica establecida con la placa Arduino, de un servicio web integrado para la configuración y monitorización del sistema y que es accesible desde cualquier navegador web.</p> <p>Finalmente, la realización de las pruebas finales del sistema ha permitido verificar los requisitos de diseño establecidos y confirmar que los objetivos del proyecto definidos en la fase inicial han sido cumplidos.</p>	

Abstract:

This project shows the design and development of a new prototype of a self-contained pointing system device whose main functionality will be to serve as a guidance method for participants in outdoor activities such as geocaching.

The system is based on the open source platform Arduino, which allows the integration of the different modules needed, a magnetic field sensor and a GNSS sensor, which collect the orientation and position data of the device, a LCD screen to display the orientation data to the user and a battery that will provide the required power to the whole system.

The device uses the information from the sensors to geolocate the location and orientation of the system with respect to true geographic north and informs the user of the course to follow to reach the preset destination on the LCD screen.

Likewise, the prototype provides, through a wireless communication established with the Arduino board, an integrated web service for the configuration and monitoring of the system and which is accessible from any web browser.

Finally, the final testing of the system has allowed to verify the established design requirements and to confirm that the project objectives defined in the initial phase have been met.

Índice

1. Introducción.....	1
1.1 Contexto y justificación del Trabajo.....	1
1.2 Objetivos del Trabajo.....	3
1.3 Enfoque y método seguido.....	4
1.4 Planificación del Trabajo.....	4
1.4.1 Seguimiento de la planificación.....	5
1.5 Viabilidad.....	6
1.5.1 Viabilidad Técnica.....	6
1.5.2 Viabilidad Económica.....	7
1.5.3 Viabilidad Legal.....	8
1.6 Breve resumen de productos obtenidos.....	8
1.7 Breve descripción de los otros capítulos de la memoria.....	9
2. Estado del Arte.....	10
2.1 Dispositivos específicos.....	10
2.2 Dispositivos genéricos.....	11
2.3 Contribución del Trabajo.....	11
3. Diseño de la solución.....	12
3.1 Arquitectura.....	13
3.2 Componentes de la solución.....	15
3.2.1 Componentes Hardware.....	15
3.2.1.1 Arduino.....	15
3.2.1.2 Magnetómetro.....	16
3.2.1.3 Sensor GNSS.....	17
3.2.1.4 Pantalla LCD.....	17
3.2.1.5 Batería.....	18
3.2.1.6 Otros componentes hardware.....	18
3.2.1.7 Matriz de cumplimiento de los requisitos.....	19
3.2.2 Herramientas Software.....	19
3.2.2.1 Software de desarrollo.....	19
3.2.2.2 Software de monitorización.....	20
3.2.2.3 Software ofimático.....	20
3.3 Diagrama de componentes.....	21
4. Implementación.....	22
4.1 Salvaguarda de datos en memoria flash.....	23
4.2 Integración del magnetómetro.....	23
4.3 Integración del sensor GNSS.....	26
4.3.1 Cálculo del rumbo hasta el destino.....	27
4.4 Integración pantalla LCD.....	28
4.5 Desarrollo del software de control.....	30
5. Pruebas y resultados obtenidos.....	37
5.1 Pruebas de conexión.....	37
5.1.1 Conexión WiFi Arduino.....	37
5.1.2 Magnetómetro.....	39
5.1.3 Sensor GNSS.....	40
5.1.4 Pantalla LCD.....	41

5.2 Pruebas funcionales	42
5.2.1 Orientación con respecto al norte geográfico.....	42
5.2.2 Cálculo de la dirección de destino	43
5.2.3 Servicio de Administración Web	45
5.2.4 Presentación de la información en la pantalla LCD	48
5.2.5 Pruebas de alimentación.....	49
6. Conclusiones.....	50
6.1 Futuras líneas de acción	51
7. Bibliografía	52

Lista de figuras

Ilustración 1. Arquitectura del Sistema	2
Ilustración 2. Calendario Laboral del Proyecto	4
Ilustración 3. Planificación del Proyecto	5
Ilustración 4. Dispositivos GNSS específicos. Fuente: [4] y [5]	10
Ilustración 5. Esquema de conexiones del prototipo realizado con Fritzing [6]	14
Ilustración 6. Arduino MKR WiFi 1010. Fuente: [7]	16
Ilustración 7. Magnetómetro QMC5883L. Fuente: [8]	16
Ilustración 8. Sensor GPS NEO-6M. Fuente: [9]	17
Ilustración 9. Pantalla LCD Nokia 5110. Fuente: [11]	17
Ilustración 10. Diagrama de componentes	21
Ilustración 11. Prototipo	22
Ilustración 12. Instalación Magnetómetro	24
Ilustración 13. Corrección Declinación Magnética. Fuente: [15]	25
Ilustración 14. Ejemplo de localización GPS. Fuente [16]	27
Ilustración 15. Cálculo dirección de destino	28
Ilustración 16. Integración pantalla LCD	28
Ilustración 17. Diagrama servicio web	31
Ilustración 18. Contenido dinámico actualizable	34
Ilustración 19. Formulario declinación magnética	34
Ilustración 20. Formulario posición de destino	35
Ilustración 21. Página web de administración	36
Ilustración 22. Prueba de conexión WiFi	38
Ilustración 23. Prueba de conexión con el magnetómetro	40
Ilustración 24. Prueba de conexión con sensor GNSS	41
Ilustración 25. Corrección declinación magnética	42
Ilustración 26. Medida de orientación promedio	43
Ilustración 27. Mapamundi ofrecido por Google Maps.	44
Ilustración 28. Envío datos XML	46
Ilustración 29. Actualización automática declinación magnética.	47
Ilustración 30. Fichero XML, declinación automática.	47
Ilustración 31. Direccionamiento, incremento entre medidas mostradas.	49

1. Introducción

1.1 Contexto y justificación del Trabajo

En los últimos años es cada vez más frecuente la inclusión de las tecnologías de la información y la comunicación en todos los ámbitos de la vida cotidiana, especialmente desde la proliferación de los dispositivos móviles conectados, como los smartphones, tablets, smartwatch, etc.

Las actividades lúdicas no son una excepción a esta norma y se han creado multitud de dispositivos para ser usados al hacer deporte (podómetros, pulsómetros, monitores de actividad, localización GPS, etc.) o como juguetes o complementos de estos (Lego Boost, Lego Mindstorms Sphero, Disney Infinity, Starlink, etc.).

Una de las actividades lúdicas al aire libre que más ha aumentado el número de participantes es el Geocaching [1], que consiste en esconder objetos en cualquier lugar del mundo, bien en un entorno urbano o bien en el campo, anotando las coordenadas geográficas del mismo y que posteriormente se harán públicas para que otras personas puedan buscar y encontrar los ‘tesoros’ escondidos cerca de su posición. Los jugadores necesitan, por tanto, la situación del objeto escondido (en forma de georreferencia) y un sistema global de navegación por satélite (Global Navigation Satellite System, GNSS [2]), usualmente un smartphone, para conocer la posición actual y la distancia al destino, para poder encontrar el ‘tesoro’.

Este proyecto se basa en la creación de un prototipo de dispositivo para ser usado por los jugadores de geocaching pero introduciendo una diferencia, el usuario no debe conocer las coordenadas exactas del tesoro escondido.

Este dispositivo debe funcionar de forma autónoma y constará de una placa Arduino que procesará las señales de los sensores integrados: un sensor GNSS para averiguar la localización del sistema en todo momento, un magnetómetro para calcular la orientación del prototipo hacia el norte geográfico y un dispositivo señalador que indicará al usuario el camino a seguir.

Para que el sistema funcione correctamente de forma autónoma y sin conexión, tanto las coordenadas de la posición de destino como el dato de la declinación magnética deben ser introducidas con anterioridad al uso del dispositivo.

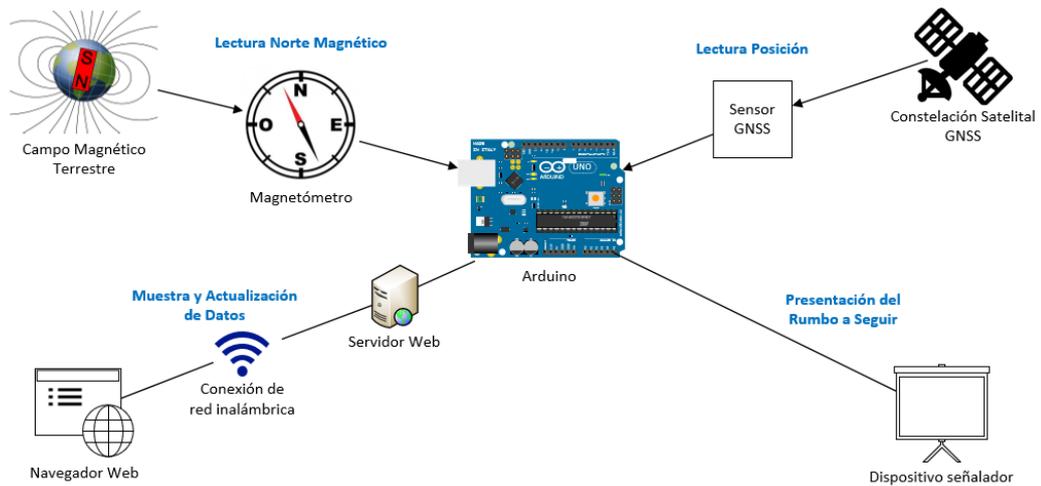


Ilustración 1. Arquitectura del Sistema

Tal y como se describe en la Ilustración 1, el funcionamiento de este dispositivo apuntador consiste en la toma de datos de la unidad GNSS, que nos proporcionará la ubicación del sistema sobre el globo terráqueo, y del magnetómetro, lo que nos indicará la diferencia entre dirección hacia donde apunta el sistema y el norte magnético. La placa Arduino se encargará de calcular la orientación real del sistema con respecto al norte geográfico ya que es necesario corregir la lectura del norte magnético, obtenido con el magnetómetro, con la declinación magnética de la zona (por ejemplo, la declinación magnética actual para Madrid es de $-0^{\circ} 17'$, mientras que para Barcelona es $+1^{\circ} 15'$).

Con ambas medidas podremos conocer la localización y orientación exacta del prototipo y así poder calcular la dirección a seguir para llegar al lugar donde se encuentra escondido el tesoro, una vez realizado el cálculo la placa Arduino notificará al usuario la dirección a seguir por medio del dispositivo señalador. Este cálculo y ajuste del rumbo correcto debe realizarse de forma continua según el jugador se vaya moviendo por el entorno.

El sistema proporcionará así mismo, a través de la conexión inalámbrica, una página web por la que se podrá actualizar la información de la posición del tesoro escondido (para reutilizar el sistema cuantas veces como se desee) así como de la declinación magnética de la zona, puesto que ésta varía levemente con el paso del tiempo y entre distintas áreas geográficas.

1.2 Objetivos del Trabajo

Este proyecto tiene como finalidad la realización de un prototipo de sistema apuntador autónomo y portable que permita al usuario orientarse hacia una posición (coordenadas geográficas) predeterminada.

Este sistema apuntador podría utilizarse en los siguientes casos de uso.

- Recreativo: como complemento al deporte de ‘geocaching’, por ejemplo, en una competición el patrocinador reparte uno de estos sistemas a cada participante con las coordenadas del tesoro escondido prefijadas para que todos ellos vayan a buscar el objeto sin saber las coordenadas exactas de éste y sin la posibilidad de usar un dispositivo externo como smartphone.
- Religioso: para determinadas religiones, como la musulmana, los rezos diarios deben ser realizados, varias veces al día, orientándose hacia un punto determinado, en este caso La Meca, con este sistema se puede disponer de la orientación correcta incluso en lugares sin cobertura de datos en el teléfono móvil.
- Souvenir: como “Recuerdo de Cuenca” o “Recuerdo de Roma” para saber exactamente la dirección hacia la cual se encuentra la ciudad.

Se definen los siguientes objetivos específicos a alcanzar durante el trabajo de fin de grado. En función de la criticidad de cada objetivo se ha asignado un nivel de prioridad:

- Realizar el montaje de un prototipo autónomo y funcional del sistema apuntador. Prioridad alta.
- Calcular con ayuda de los sensores el rumbo a seguir para llegar al destino. Prioridad alta.
- Presentar al usuario de forma visual la información del rumbo a seguir para llegar a destino de forma continua e independiente de los movimientos del sistema. Prioridad media.
- Permitir que el administrador/operador del sistema pueda revisar los datos recibidos por los sensores. Prioridad baja
- Actualizar los datos almacenados en el sistema como, por ejemplo, los valores de declinación magnética y de destino. Prioridad media.

1.3 Enfoque y método seguido

Para la realización de este proyecto se ha decidido seguir una estrategia de creación de un nuevo producto debido a que no existe un dispositivo similar en el mercado. El prototipo resultante incorporará las funcionalidades establecidas en los objetivos del proyecto (véase el apartado anterior). Este sistema se basará en la integración hardware de componentes COTS disponibles en el mercado y el desarrollo ad-hoc del software de control asociado.

1.4 Planificación del Trabajo

La planificación de las tareas a realizar en este proyecto ha sido estimada en base a un calendario que incluye todos los días naturales como hábiles exceptuando los festivos siguientes:

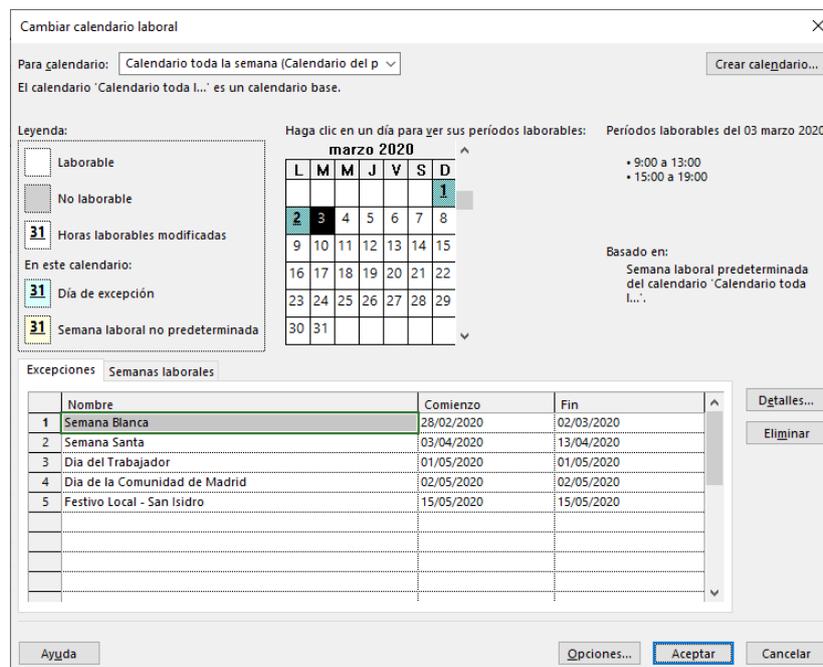


Ilustración 2. Calendario Laboral del Proyecto

Todas y cada una de las tareas serán realizadas por un único responsable que podrá dedicar un tiempo medio máximo al proyecto de 3 horas al día.

A continuación, se muestra el diagrama de Gantt correspondiente a las tareas básicas del proyecto junto con la estimación de tiempo y su interdependencia. Esta estimación se ha realizado teniendo en cuenta los conocimientos y experiencia previa tanto en integración hardware, desarrollo de software y documentación del responsable de este proyecto.

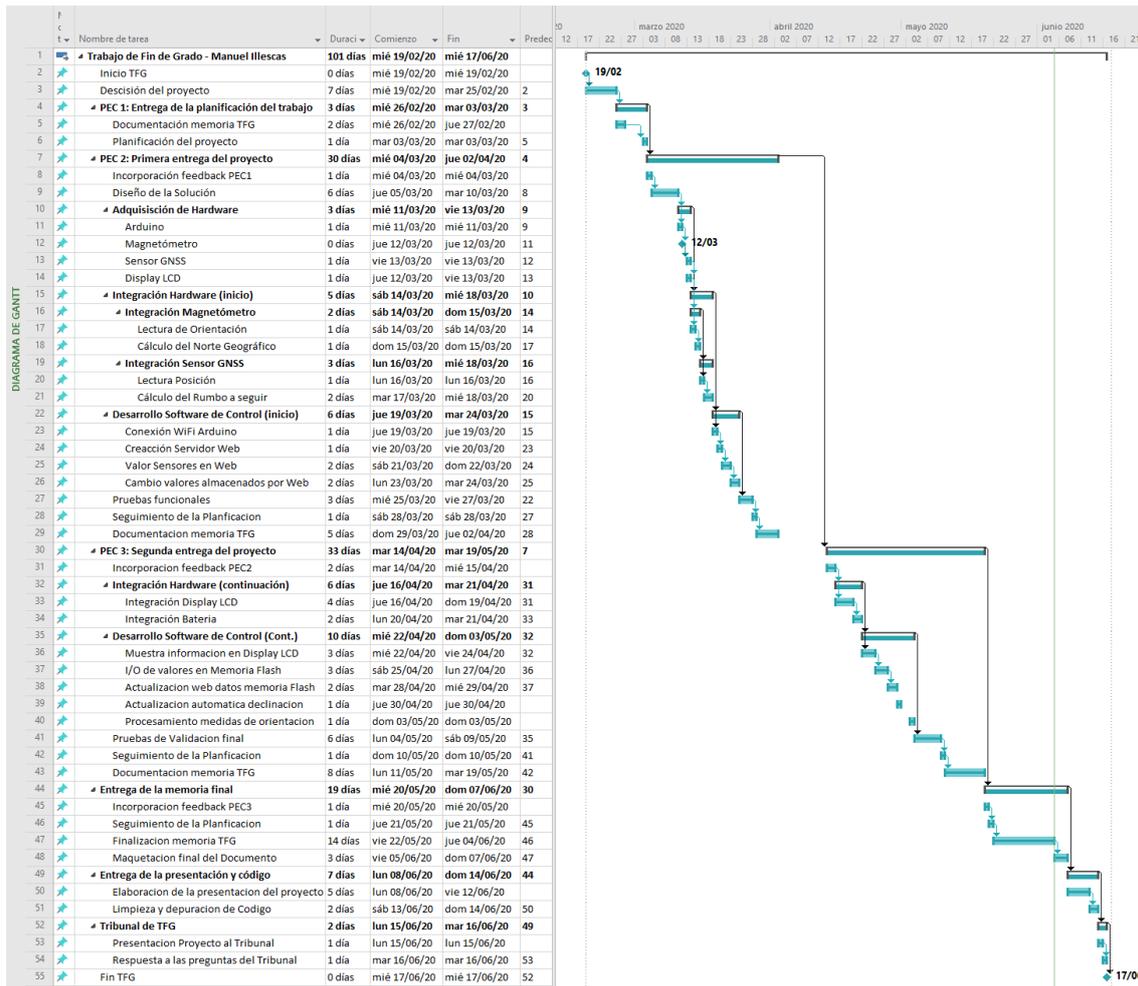


Ilustración 3. Planificación del Proyecto

Como resultado de la planificación inicial se estima que el responsable único del proyecto trabajará 101 jornadas, teniendo en cuenta el horario laboral de 3 horas, por tanto, dedicará un máximo de 303 horas de trabajo al proyecto completo, lo que está en consonancia con el marco temporal teórico previsto para un Trabajo de Fin de Grado.

1.4.1 Seguimiento de la planificación

A causa de la pandemia global de coronavirus COVID-19 se detectó una posible desviación en la segunda entrega parcial del proyecto debido a la sobrecarga de trabajo y a las circunstancias anómalas que conlleva el confinamiento, por ello se pidió una extensión del plazo de entrega de una semana.

Este aplazamiento no generó una desviación global en la planificación inicial del trabajo de fin de grado, puesto que fue compensado realizando las actividades durante los días, inicialmente inhábiles, de Semana Santa.

Durante las conversaciones con el tutor de TFG se le propuso el cambio de sistema señalizador. En la propuesta inicial estaba previsto que fuese una flecha apuntadora accionada por un motor, la propuesta final se basa en que la presentación e indicación del rumbo a seguir sea realizada usando una pantalla LCD. Este cambio fue aceptado e implantado durante el proyecto sin impacto sobre la planificación acordada.

Se añadieron posteriormente dos nuevas tareas (tarea 39 y 40) que incluyen la integración de un sistema de actualización de la declinación magnética automática y el post procesamiento de las medidas del magnetómetro para minimizar el error mostrado al usuario. Estas tareas se han realizado acortando el tiempo de integración de la pantalla LCD, que inicialmente estaba sobreestimado dado que se desconocía su complejidad.

1.5 Viabilidad

El producto para el que se está desarrollando este prototipo puede tener cabida en el mercado actual de actividades lúdicas unido al 'culto al cuerpo' o la consciencia de la necesidad de realizar actividades físicas. A continuación, se realiza un estudio inicial de la viabilidad técnica, económica y legal de este sistema.

1.5.1 Viabilidad Técnica

La complejidad técnica de realizar este proyecto no radica tanto en los componentes hardware que lo componen si no, más bien, en la integración software del conjunto de ellos.

El hardware que se utilizará en este proyecto se basa en soluciones comerciales disponibles en el mercado (COTS) que han sido probadas con anterioridad y forman parte de multitud de productos en el mercado. Este hardware se basa principalmente en 4 componentes, un sistema de localización satelital GNSS que podrá ubicar el prototipo de forma precisa, una brújula magnética que detectará la orientación del campo magnético de la tierra, un sistema de interconexión de los sensores con acceso inalámbrico basado en Arduino y un subsistema de apuntamiento (accionado por medio de un motor) que mostrará al usuario la dirección a tomar para llegar a su destino.

La integración software de cada uno de estos componentes hardware y el proceso de conexión inalámbrica y método de actualización de los datos internos del sistema es, como se indicaba anteriormente, la parte crítica del sistema, pero dedicando el tiempo necesario para la comprensión e integración de cada uno de ellos es realizable.

Es por tanto un **producto técnicamente viable**.

1.5.2 Viabilidad Económica

La viabilidad económica del sistema se estimará en función del coste total de realización del prototipo y el cálculo de unidades vendidas necesarias para retornar la inversión inicial.

El coste estimado inicial para la adquisición de cada uno de los componentes del prototipo es el siguiente:

Producto	Cantidad	Precio (€)
Placa Arduino WiFi	1	30
Módulo GNSS	1	10
Magnetómetro	1	10
Motor	1	3
Compartimento de Batería	1	2
Protoboard	1	2
Cableado	30	2
SW IDE Arduino	1	Gratuito
Total		59

A estos 59 euros hay que añadir el coste/hora de la ejecución completa del proyecto que como vimos en apartados anteriores es de 303 horas. Suponiendo un coste hora de 35 euros, el coste de ejecución se eleva a 10.605 euros, resultando un coste total de producción del prototipo de 10664 euros.

Producir cada unidad nos cuesta 59 euros y, sin incluir costes de producción, envío, etc., como se espera obtener un 15% de beneficio por unidad, se debería poner el precio de venta por cada sistema a 67,85 euros.

Realizando un cálculo simple obtenemos la tabla siguiente:

	Unidades	Coste unitario (€)	Coste (€)
Desarrollo prototipo	1	10664	-10664
Coste sistema	1205	59	-71095
Precio venta	1205	67,85	81759,25
		Resultado	0,25

Donde se puede observar que a partir de la unidad 1205 empezamos a recuperar la inversión inicial y obtendremos beneficios. Por tanto, económicamente hablando **si vendemos más de 1205 unidades el proyecto sería rentable.**

La estimación del coste de los materiales puede verse reducida de forma considerable, se debe tener en cuenta que en la adquisición de los componentes usados para la realización de este prototipo se ha priorizado la rapidez del plazo de entrega frente al coste.

Si este proyecto se intentase convertir en un negocio real se debería buscar nuevos suministradores que provean de elementos de similares características y prestaciones, pero a menor coste, y/o alcanzar acuerdos de descuento en compras por volumen.

En todo caso, revisando esta página web de geocaching a nivel global, se pueden encontrar más de 300 grupos y organizaciones [3] dedicadas a esta actividad. Existen además otro tipo de asociaciones no registradas en esta web, bien porque sean de ámbito más local o bien porque su actividad lúdica no esté centrada exclusivamente en el geocaching, por ejemplo, agrupaciones orientadas a la actividad al aire libre y especialmente con niños, que podrían adquirir el sistema.

Teniendo en cuenta que el sistema sobre el que versa este proyecto está orientado al juego en grupo, en el que los patrocinadores son los únicos que conocen donde se esconden los tesoros y que el principal sistema competidor es el Garmin eTrex 10 [4] que cuesta 120 euros pero no permite ocultar la ubicación del tesoro, se estima factible que el proyecto pueda ser viable si se consiguiera promocionar la venta de lotes de este sistema a las diferentes asociaciones mencionadas.

En este estudio de viabilidad no se han explorado las demás opciones de venta del sistema, en concreto el sistema para orientar a un usuario con un destino prefijado y que no vaya a modificarse, como por ejemplo los casos de usos religioso (orientación hacia la Meca) o souvenir (sistema embebido dentro de un recuerdo de Cuenca). El sistema, en estos casos, sería más sencillo y barato por carecer de conexión inalámbrica, puesto que no se requeriría su uso.

1.5.3 Viabilidad Legal

En cumplimiento de la ley vigente, en ningún caso se perturba el espacio radioeléctrico restringido puesto que los sensores incluidos en este dispositivo son receptores de señales (GNSS o magnéticas) excepto los utilizados para la interconexión (Wifi 2.4/5.8 GHz o Bluetooth) que actúan en unas bandas de frecuencia abiertas al público.

De igual manera, el uso del sistema y sus componentes internos no entrañan riesgos para las personas, animales o el medio ambiente.

Por tanto, **desde el punto de vista legal el sistema es viable.**

1.6 Breve sumario de productos obtenidos

El producto resultante de este Trabajo de Fin de Grado es un prototipo de un Sistema Apuntador Automático y en concreto se obtendrán los siguientes entregables:

- Prototipo Hardware del Sistema Apuntador Automático
- Código software de integración de los componentes
- Memoria Técnica descriptiva del proyecto
- Presentación de los principales elementos del proyecto

1.7 Breve descripción de los otros capítulos de la memoria

A continuación se detallan el resto de apartados que compondrán la memoria del proyecto.

Capítulo 2 – Estado del Arte

En este capítulo se realizará un breve repaso a los dispositivos utilizados en la actualidad para la actividad del Geochacing.

Capítulo 3 – Diseño de la solución.

En este apartado se analizará la arquitectura de la solución propuesta para el prototipo y se llevará a cabo una descripción de cada uno de los componentes usados.

Capítulo 4 – Implementación

Se detalla la implementación y código utilizado para dotar a la solución de las funcionalidades deseadas basándonos en los objetivos y requisitos definidos.

Capítulo 5 - Pruebas y resultados obtenidos.

En este capítulo se expondrán los casos de pruebas utilizados y los resultados obtenidos que conllevan a la validación del sistema.

Capítulo 6 – Conclusiones.

Capítulo 7 – Bibliografía.

2. Estado del Arte

La actividad de Geocaching surgió después de que el gobierno de Estados Unidos eliminase la degradación intencional de la precisión de la señal del Sistema de Posicionamiento Global (GPS, Global Positioning System) en mayo del año 2000. Ese error de precisión en la señal recibida podía alcanzar hasta 100 metros, por lo que se hacía inviable encontrar un objeto pequeño utilizando esta tecnología.

A raíz de la eliminación del error inducido, Dave Ulmer, usuario de la red Usenet publicó en el grupo de noticias sci.geo.satellite-nav la localización del primer 'tesoro' escondido, a partir de ese momento la comunidad creó este juego y hoy miles de usuarios alrededor del mundo buscan los 'geocaches' escondidos.

En la actualidad los usuarios de esta actividad cuentan con diversos dispositivos para localizar los 'geocaches' a través de las coordenadas publicadas:

2.1 Dispositivos específicos

Estos dispositivos han sido creados para su uso por aficionados al senderismo u otras actividades al aire libre. Sus principales ventajas son la resistencia a las inclemencias atmosféricas (agua, polvo, etc.) y la autonomía, dado que suelen utilizar baterías estándar AA o AAA y en algunos casos alcanzan una autonomía superior a las 24h.



Ilustración 4. Dispositivos GNSS específicos. Fuente: [4] y [5]

La Ilustración 4 muestra los modelos eTrex 10 de Garmin [4] y eXplorist GC de Magellan [5], ambos pertenecen a gama básica de dos de las marcas más reconocidas de este tipo de dispositivos. Su principal inconveniente, en ambos casos, es el precio que es superior a los 120 euros.

2.2 Dispositivos genéricos

Los dispositivos genéricos son todos aquellos sistemas que al incluir un módulo de recepción de señales GNSS pueden ser usados para esta actividad. Son dispositivos que la mayoría de nosotros posee en la actualidad: smartphones, tablets o incluso algunos modelos de portátiles.

Se pueden utilizar aplicaciones creadas ad hoc, como la aplicación Geocaching creada por Groundspeak Inc. para dispositivos Android o Apple, que accede a la base de datos de 'geocaches' registrados y muestra los que están escondidos en los alrededores. O también se pueden usar aplicaciones genéricas de cartografía georreferenciada como OpenStreetMap o Google Maps incorporando los detalles de las coordenadas de los 'geocaches' que se quieran encontrar, estas coordenadas están disponibles en páginas web y foros especializados en formato de coordenadas (longitud y latitud) o como fichero con formato GPX.

2.3 Contribución del Trabajo

Los dispositivos existentes, específicos o genéricos, trabajan de igual modo, el usuario conoce y puede ver representado sobre un mapa la posición del 'geocache' escondido en todo momento. Esto facilita la búsqueda y la aproximación a la ubicación estableciendo la ruta más cómoda para llegar a él, por ejemplo, evitando colinas, ríos, bosques cerrados, etc.

Como se ha mencionado anteriormente, el objetivo de este proyecto es la creación de un prototipo de dispositivo que pueda ser utilizado en eventos de Geocaching en las que el promotor del evento entregue a cada participante un sistema preconfigurado con las coordenadas del tesoro escondido.

La contribución del sistema propuesto incluye un factor de indeterminación que propicia y promueve la competición en grupo de la búsqueda del tesoro. Los usuarios no dispondrán de ningún mapa y tampoco conocerán la ubicación o la distancia exacta hasta el 'geocache', solo sabrán el rumbo que deben seguir (en función de la orientación del dispositivo), por lo tanto, se hace imposible seleccionar una ruta óptima que evite accidentes geográficos y complicará en gran medida llegar al destino deseado.

3. Diseño de la solución

Tras el análisis de los objetivos descritos en el apartado 1.2 se deriva la siguiente lista de requisitos, a los que la solución deberá dar respuesta:

- Autonomía: es necesario que el sistema sea autocontenido, no debe requerir ningún elemento o conexión externa para realizar su función. Se requiere una autonomía para una jornada de trabajo de entre 6 y 8 horas.
- Portabilidad: Los componentes y el conjunto total deben ser de dimensiones y peso reducido para que pueda ser manejado de forma fácil. El tamaño del prototipo deberá ser manejado con una sola mano.
- Reutilización: El dispositivo debe ser capaz de permitir una conexión inalámbrica (Wireless o bluetooth) y ofrecer un método (servicio web) para poder actualizar los datos internos de localización de destino y declinación magnética.
- Posicionamiento: El prototipo debe ser capaz de obtener su posición utilizando un sistema de localización satelital, el error de precisión del posicionamiento debe ser inferior a 10 metros.
- Orientación: El sistema debe poder realizar medidas del campo magnético de la tierra y ser capaz de conocer la dirección del norte geográfico realizando la corrección correspondiente del norte magnético, para saber la dirección hacia donde está orientado. Las medidas del norte magnético tomadas con el dispositivo no deben tener una inexactitud mayor a 3°.
- Direccionamiento: El dispositivo debe mostrar al usuario de forma clara la dirección a seguir para alcanzar la ubicación del destino. El paso o salto intermedio entre medidas de dirección contiguas que se muestren en el dispositivo no debe ser mayor a 1°, para no mostrar al jugador a un error muy elevado que le lleva a una posición de destino incorrecta.

3.1 Arquitectura

Según las especificaciones mencionadas en el apartado anterior se diseña la siguiente solución cuyos componentes principales son los siguientes:

Arduino: Esta unidad programable será el centro de cálculo y control de los demás componentes de la solución, además ofrecerá acceso inalámbrico por medio de una conexión wifi y realizará las labores de servidor web.

Magnetómetro: mediante las mediciones realizadas con este sensor se podrá saber la orientación del prototipo con respecto al norte geográfico.

Sensor GNSS: Con este sensor se podrá ubicar de forma precisa la posición del prototipo sobre el globo terráqueo.

Pantalla LCD: Mediante esta pantalla LCD se representará el rumbo que el usuario del sistema debe seguir para alcanzar el destino.

El sistema funcionará mediante la recolección de los datos de orientación y posición del magnetómetro y el sensor GNSS. La placa Arduino realizará las correcciones necesarias entre el norte magnético y el geográfico, teniendo en cuenta el valor de la declinación magnética de la zona. Con este valor de la orientación real del dispositivo junto con la posición actual y el dato almacenado de las coordenadas del 'geocache' harán posible que se pueda realizar el cálculo del rumbo.

En la pantalla LCD se representará de forma visual el valor y/o dirección del rumbo a seguir por el usuario del sistema.

El módulo Arduino realizará las funciones de servidor web, accesible mediante la conexión Wireless, para proveer al administrador del sistema de la información recabada por los sensores y le ofrecerá un medio para modificar los valores guardados en la memoria interna sin tener que acceder al código implementado.

La conexión inalámbrica se utilizará exclusivamente en la fase de configuración del dispositivo por el administrador, previamente al uso en campo por los jugadores. El tipo de conexión inalámbrica escogida es la conexión Wireless (Wi-Fi) puesto que nos permitirá establecer de forma sencilla el acceso al dispositivo desde cualquier red casera, empresarial o incluso desde un dispositivo móvil.

Durante la configuración, el administrador establecerá la conexión inalámbrica bien con la red fija de acceso a internet (ADSL/FTTH) o bien, si está en el campo u otro lugar sin acceso a un dispositivo de tipo router o switch inalámbrico, deberá crear un punto de acceso Wi-Fi con su propio teléfono móvil. En cualquiera de los casos, será necesaria una conexión

a internet para hacer uso de la función de actualización automática del valor de declinación magnética.

Una vez se establezca la conexión con el dispositivo, se podrá conectar con el servidor web integrado y configurar o monitorizar los parámetros necesarios.

Mientras el sistema este en uso por los jugadores no será necesaria ninguna conexión externa, ni a servidores locales ni a servidores de internet. La única conexión necesaria para su correcto funcionamiento será la requerida para la recepción de la señal de los satélites GPS.

En la siguiente ilustración se puede observar el esquema de conexiones entre los distintos módulos del sistema:

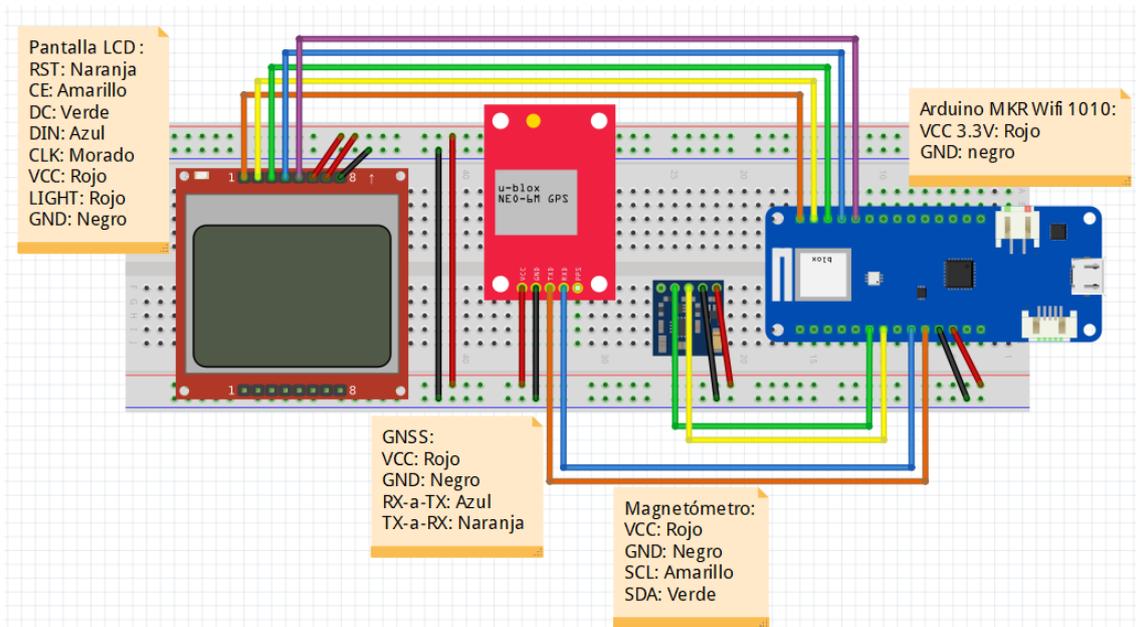


Ilustración 5. Esquema de conexiones del prototipo realizado con Fritzing [6]

3.2 Componentes de la solución

Seguidamente se realizará una descripción en detalle de cada uno de los componentes hardware y herramientas software que se han utilizado en este trabajo de fin de grado.

Como se especificó en el capítulo de viabilidad, la prioridad para la adquisición de los componentes hardware ha sido primordialmente la rapidez de entrega frente al coste del producto, siempre y cuando se cumpliesen con los requisitos funcionales y de tamaño de cada componente.

3.2.1 Componentes Hardware

En los siguientes apartados se realizará una descripción de los elementos hardware utilizados en el desarrollo del trabajo de fin de grado.

3.2.1.1 Arduino

Para este proyecto se utilizará la plataforma de hardware abierto Arduino que ofrece multitud de placas diferentes con un gran abanico interfaces y módulos de ampliación disponibles, se adquirirá una placa Arduino original por la confiabilidad de estas y como apoyo al proyecto Arduino. El criterio básico de selección entre los distintos módulos de Arduino pasa por un primer filtro de tamaño, como se pretende hacer un prototipo pequeño y portable se descartan las familias Uno, Mega, Leonardo, Due, Yun o Zero en sus distintas versiones por sus dimensiones elevadas.

Entre las placas Arduino restantes el siguiente filtro de selección corresponde con la conectividad, se eligen únicamente los que permiten una conexión de red inalámbrica, en concreto y como se especificó anteriormente, la disponibilidad de conexión Wi-Fi. Esto nos permite acotar las distintas opciones hasta seleccionar las familias Nano y MKR.

La diferencia fundamental entre estas familias es el número y tipo de conexiones disponibles, siendo la elegida la familia MKR por tener un puerto de carga para las baterías de polímero de litio (Li-Po) y un conector para placas I2C externas (que en el proyecto actual no va a ser usado)

Dentro de la familia MKR se opta por el último modelo disponible con las prestaciones deseadas, el modelo **MKR WiFi 1010** [7].



Ilustración 6. Arduino MKR WiFi 1010. Fuente: [7]

Con la elección de este modelo nos aseguramos, además, que disponemos de los pines digitales y analógicos necesarios para el montaje del prototipo.

Pese a no contar con memoria EEPROM, esta placa dispone de memoria Flash integrada donde podremos almacenar los datos necesarios para el funcionamiento del sistema.

3.2.1.2 Magnetómetro

No existe una gran variedad de magnetómetros en el mercado y para la construcción de este prototipo se ha adquirido la versión QMC5883L [8] que deriva del módulo original HMC5883L fabricado por Honeywell. Este magnetómetro de 3 ejes, reducidas dimensiones y bajo consumo ($75\mu\text{A}$) nos proporcionará una precisión de las medidas del campo magnético de la tierra con un error inferior a 2° lo que nos permitirá averiguar la orientación del prototipo en todo momento.

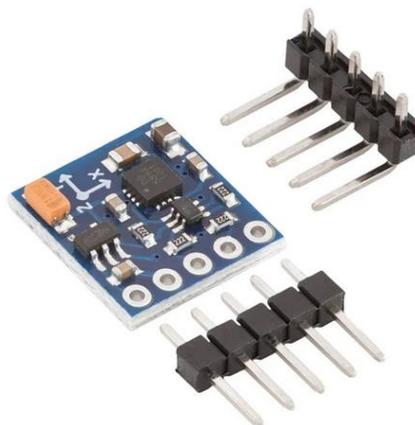


Ilustración 7. Magnetómetro QMC5883L. Fuente: [8]

3.2.1.3 Sensor GNSS

Para la selección del sensor GNSS se ha optado por una versión que viniese acompañada de una antena activa como la de este módulo, cuyo chip ha sido fabricado por U-Blox. El dispositivo Neo-6M [9] es un receptor de 50 canales que trabaja con la banda de frecuencia L1 de la señal GPS.



Ilustración 8. Sensor GPS NEO-6M. Fuente: [9]

Esta unidad permite la conexión por distintos tipos de interfaces (UART, SPI, I2C y USB), soporta el protocolo de NMEA [10] como método de transmisión de los mensajes del GPS y nos ofrece una tasa de refresco de las mediciones de 5Hz junto con una desviación máxima del posicionamiento de 2,5m. Estas características lo hacen más que suficiente para las necesidades de posicionamiento del prototipo.

3.2.1.4 Pantalla LCD

Esta pantalla gráfica LCD similar a las montadas en los teléfonos Nokia 5110 [11] nos ofrece una matriz de 84x48 píxeles sobre los que podremos representar la orientación del rumbo deseado. El criterio de selección para este componente es la facilidad de integración y su reducido consumo, precio y tamaño.

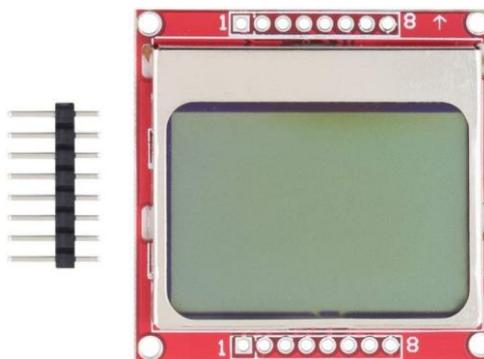


Ilustración 9. Pantalla LCD Nokia 5110. Fuente: [11]

3.2.1.5 Batería

Se ha escogido una batería de polímero de litio (Li-Po) de 3.7V y una capacidad de 1400mAh de la marca fullwat [12] que incorpora un conector JST, que es el requerido para la conexión a la placa Arduino seleccionada. Esta batería proporcionará la alimentación necesaria a todos componentes del prototipo y, además, permite ser cargada a través de la conexión USB integrada en el módulo Arduino.

Para la estimación de la autonomía del prototipo se ha tenido en cuenta los siguientes consumos proporcionados por los fabricantes de los componentes:

Elemento	Consumo
Magnetómetro	75 μ A (máx.)
GPS	67 mA (máx.)
Pantalla LCD	7 mA (80mA si se usan los 4 leds)
MKR 1010	30mA (low power) máx. 93mA
TOTAL	104,075 – 167,075 mA

La pantalla LCD consumirá únicamente 7mA puesto que mantendrá apagados los 4 leds de retroiluminación disponibles, por tanto, el prototipo necesitará un consumo eléctrico de entre 104,075 y 167,075 mA.

Puesto que la batería ofrece 1400mAh permitirá que el dispositivo este encendido durante más de 13 horas (1400mAh/104,075mA) en el caso de bajo consumo sin la conexión Wireless encendida o de más 8 horas (1400mAh/167,075mA) con la conexión inalámbrica habilitada. Estas medidas de consumos satisfacen plenamente el tiempo estipulado de autonomía definido en los requisitos (véase 3. Diseño de la solución).

3.2.1.6 Otros componentes hardware

Los siguientes elementos secundarios han sido utilizados para la realización del proyecto:

Protoboard y cables: Placa de prototipado y cableado necesario para proveer de alimentación de energía e interconexión entre los componentes principales.

Soldador y estaño: Utilizados para soldar los pines del magnetómetro, sensor GNSS y pantalla LCD.

Ordenador y cable USB: Usados para la conexión entre el software de desarrollo y la placa Arduino.

3.2.1.7 Matriz de cumplimiento de los requisitos

La siguiente tabla muestra una tabla resumen con la relación entre el hardware seleccionado y el cumplimiento de los requisitos del sistema.

Requisito	Arduino	Magnetómetro	GPS	LCD	Batería
Autonomía (6-8 horas)					✓
Portabilidad	✓	✓	✓	✓	✓
Reutilización	✓				
Posicionamiento (Error < 10m)			✓		
Orientación (Error <3°)		✓			
Direccionamiento (Paso <1°)				✓	

3.2.2 Herramientas Software

En este apartado se describen las herramientas de software que se han usado para la realización del trabajo de fin de grado.

3.2.2.1 Software de desarrollo

Entre los distintos softwares de desarrollo disponibles para realizar proyectos con Arduino (Processing, Eclipse, Visual Studio, etc.) se ha escogido el IDE de Arduino puesto que su uso es muy sencillo, implementa los drivers de la placa Arduino de forma automática y es un entorno de desarrollo gratuito.

Sobre este entorno de desarrollo se ha instalado las librerías necesarias para acceder a las funciones de los distintos componentes:

WiFiNINA: Librería necesaria para la utilización de la conexión Wireless del módulo Arduino.

QMC5883L Compass: Requerida para la interacción con el magnetómetro.

TinyGPS++: Librería para la conexión con el módulo GPS y la decodificación de los mensajes NMEA recibidos.

Librerías Pantalla LCD: Librerías desarrolladas por Adafruit [13] para la conexión con pantallas LCD monocromo de tipo Nokia 5110

(Adafruit_PCD8544) y para la gestión de elementos gráficos (Adafruit_GFX).

FlashStorage: Librería utilizada para la salvaguarda y restauración de los datos de declinación magnética y coordenadas de destino en la memoria flash interna del módulo Arduino.

3.2.2.2 Software de monitorización

Para monitorizar y verificar los datos recibidos por los sensores y realizar labores de depuración de los cálculos realizados en el desarrollo se utiliza el **IDE de Arduino**, en concreto la funcionalidad de monitorización del puerto serie.

Asimismo, usa **Wireshark** para analizar el tráfico de red y realizar una monitorización y depuración de errores entre:

- La conexión WiFi de la placa Arduino y la red de área local.
- La conexión HTTP generada entre el servidor web embebido y los navegadores web probados.

3.2.2.3 Software ofimático

Para la redacción de las entregas parciales, la memoria final y la presentación del trabajo de fin de grado se ha utilizado la suite ofimática Office de Microsoft, cuyas licencias están disponibles para los estudiantes de la UOC. Cabe destacar el uso de las siguientes piezas de software:

- **Microsoft Word.** Software de edición de textos utilizado para la redacción de esta memoria.
- **Microsoft Project.** Usado para la planificación y seguimiento del proyecto y creación del Diagrama de Gantt.
- **Microsoft Visio.** Software utilizado en la edición de diagramas.
- **Microsoft PowerPoint.** Utilidad para la creación de la presentación final del proyecto.

Asimismo, para la realización del esquema de conexionado del prototipo se ha utilizado la aplicación para hardware de código abierto **Fritzing** [6].

De igual modo para la realización del esquema del diagrama de flujo funcional de los componentes del sistema se ha utilizado la herramienta online **drawio** [14]

3.3 Diagrama de componentes

La siguiente ilustración muestra el diagrama de flujo funcional con los componentes hardware seleccionados para la realización del prototipo (elementos con fondo verde), y su interrelación con las principales funciones software a desarrollar (fondo amarillo). Con el fondo azul se encuentran los datos que deben ser almacenados en la memoria interna del módulo Arduino.

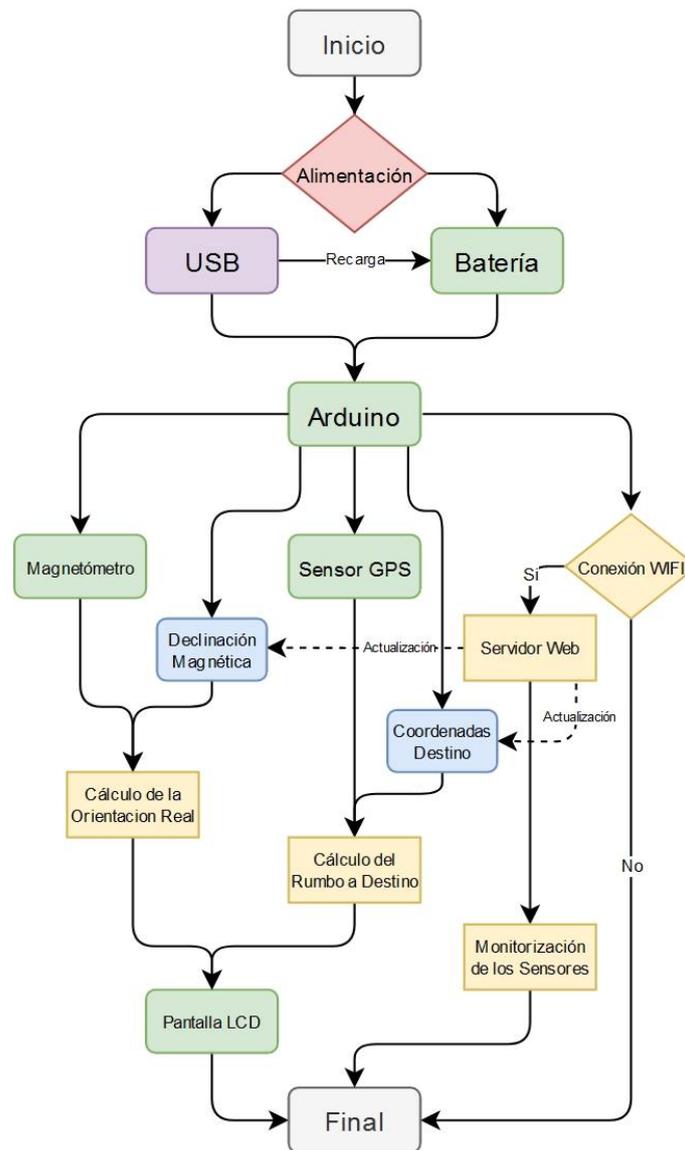


Ilustración 10. Diagrama de componentes

4. Implementación

Tras la adquisición de los materiales necesarios para la construcción del prototipo es necesario realizar el montaje y unas pruebas básicas para asegurar que la instalación ha sido correcta y que cada uno de los componentes funcionan de la forma descrita por los fabricantes.

La primera tarea es la soldadura de los pines del magnetómetro, sensor GNSS y pantalla LCD para lo que se utiliza un soldador casero y estaño. Una vez que dichos componentes están soldados y tras el análisis de las hojas de especificaciones técnicas de cada uno de ellos, se realizan las primeras pruebas de conexión.

Este primer grupo de pruebas (véase el apartado 5.1 Pruebas de conexión) está diseñado para verificar que cada uno de los componentes por separado está correctamente instalado y funcionando.

Después de la comprobación de cada componente se procede a la instalación del prototipo con todos los módulos incorporados, en la siguiente imagen se puede observar el resultado del montaje:

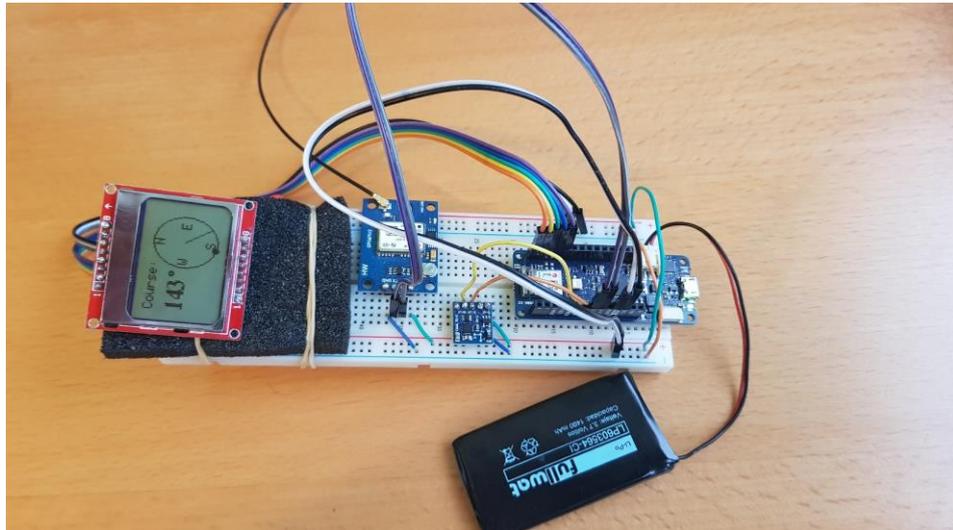


Ilustración 11. Prototipo

En las siguientes secciones se detallará la implementación realizada para cada uno de los subsistemas por separado para poder así explicar de forma precisa las decisiones de desarrollo tomadas.

4.1 Salvaguarda de datos en memoria flash

Existen ciertos datos en el dispositivo que deben ser conservados tras un reinicio o tras el cambio de batería si esta llegase al fin de su vida útil. Se ha creado una nueva estructura de datos para almacenar los valores de declinación magnética y coordenadas de destino configurados por el administrador del sistema:

```
typedef struct {
    double magnetDeclination;
    double target_latitude;
    double target_longitude;
    boolean saved;
} FlashData;
```

Esta estructura permitirá guardar en una única variable todos los valores almacenados para su utilización.

En la fase de inicio del dispositivo, el sistema consultará el estado de la variable booleana 'saved', en caso de que no se haya guardado ningún parámetro (primer uso del sistema) se almacenarán los valores de declinación magnética correspondientes a Madrid (-0° 17') y los datos de coordenadas de destino referentes a la ciudad de Cuenca (40.070393, -2.137416).

```
float declination = -0.28333333; //Magnetic declination
@Madrid -0° 17' converted to decimal degrees

// Target location: Cuenca, Spain
double latitude = 40.070393;
double longitude = -2.137416;
```

Si durante la inicialización el sistema detecta que ya existen unos valores guardados, el sistema cargará, sin modificarlos, dichos valores en una variable para su posterior uso. Como se detallará más adelante, estos valores almacenados podrán ser cambiados desde el servicio web disponible para los administradores.

Todos los valores usados en el sistema se almacenan en formato decimal para su facilidad de transformación en el momento de presentar la información en la pantalla LCD.

4.2 Integración del magnetómetro

La instalación del magnetómetro debe realizarse de forma que uno de sus 3 ejes de medida concuerde con el eje longitudinal del sistema, de este modo se podrá realizar la medida de la orientación de manera correcta. En el montaje de este prototipo se ha orientado el eje X del magnetómetro con el eje longitudinal mayor de la placa de prototipado, por consiguiente, las medidas de orientación se realizarán tomando como referencia el lado opuesto a donde se encuentra situada la placa Arduino.

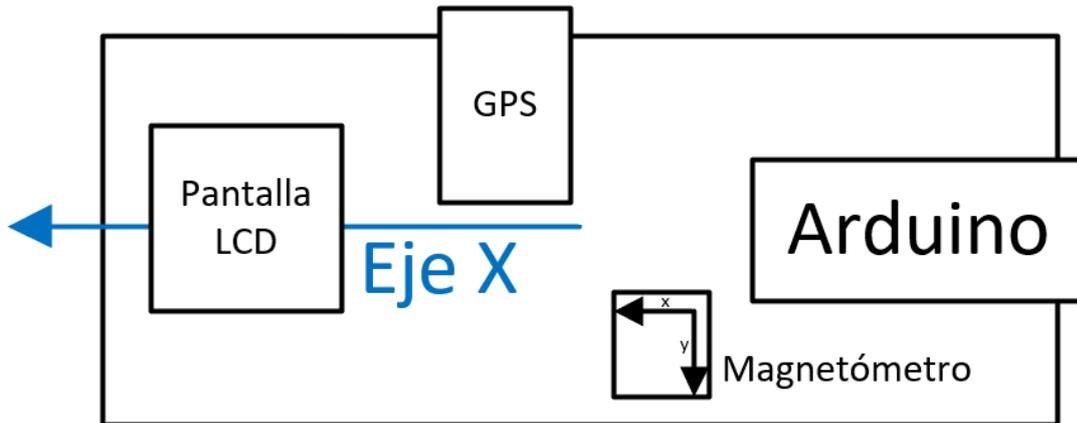


Ilustración 12. Instalación Magnetómetro

El sensor consta de cuatro pines, dos son utilizados para la alimentación eléctrica a 3.3v, uno para el sincronismo con el sistema a través del reloj integrado SCL (Serial Clock Line) y el último pin es el que porta la señal de datos SDL (Serial Data Line) con la placa.

```
Pinout for Arduino MKR 1010
Pin 11 SDA -- Connect to Digital Magnetic Compass SDA pin
Pin 12 SCL -- Connect to DMC SCL pin
```

Se utilizará la librería aportada por el fabricante para el uso de este módulo. Por tanto, usando dicha librería se creará un objeto para operar el magnetómetro.

```
#include <QMC5883LCompass.h> // Digital Magnetic compass library
QMC5883LCompass DMC; // Digital Magnetic compass (DMC)
```

El magnetómetro nos indicará la orientación del prototipo con respecto a la dirección del norte magnético. Será pues necesario conocer la desviación entre el norte magnético y el geográfico y realizar la corrección oportuna.

Ese valor de desviación depende tanto del área geográfica donde se encuentre el sistema como del momento de uso, puesto que el campo magnético de la tierra está en constante movimiento. Se puede encontrar online [15] un modelo planetario de declinación magnética para encontrar el valor preciso del factor de corrección a aplicar. En estos momentos para la ciudad de Madrid, la corrección de declinación magnética es de $-0^{\circ} 17'$.

Es necesario inicializar el módulo con los parámetros de configuración deseados. Se usarán los valores por defecto puesto que permiten la mayor tasa de refresco y de muestra admitidos por el sensor, a la vez que permite utilizar el módulo de forma continua (la opción contraria sería para tomar valores discretos).

```
// Initialize DMC
// Continuous measurement mode, data rate 200Mhz, magnetic
field sensitivity ±8G, over sample rate 512
```

Para el cálculo del azimut (dirección de orientación hacia donde apunta el prototipo), se ha creado la función getAzimuth() que nos dará el ángulo de orientación en base a las medidas del eje X e Y tomadas por el sensor.

Para mejorar la precisión y minimizar la variabilidad de las medidas obtenidas se ha implementado un postprocesado de datos que permite estabilizar la orientación final. Dicha medición final es el resultado de realizar el promedio entre diez mediciones consecutivas, cada una de las cuales ha sido tomada con un espaciado de 25 milisegundos.

Como se detallará en el apartado 5.2.1 Orientación con respecto al norte geográfico, este procesamiento estabiliza la medida al evitar variaciones y saltos en la medida erráticos y es un método suficiente que permite cumplir el requisito de Orientación del sistema.

Esta medida final de azimut se convierte de formato, desde una medición en radianes a otra en grados decimales, por facilidad de uso y posterior tratamiento.

Por último, se corrige el valor de orientación magnética medido con la variación de la declinación magnética de la zona para obtener la orientación real con respecto al norte geográfico.

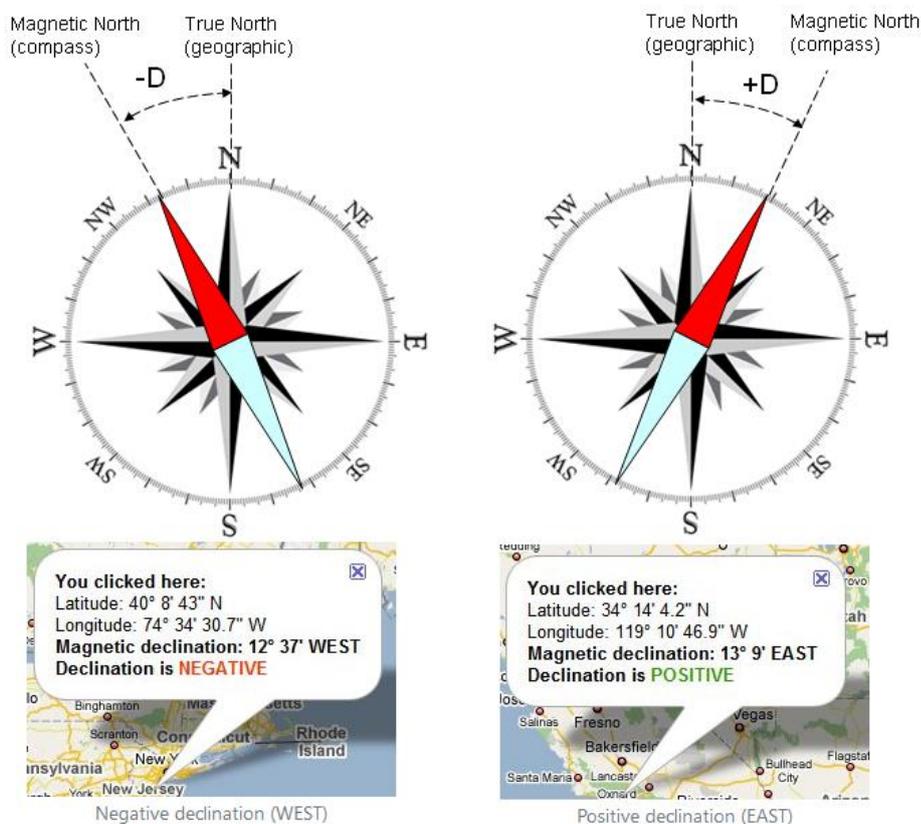


Ilustración 13. Corrección Declinación Magnética. Fuente: [15]

Se hace notar que la medida en elevación correspondiente al eje Z no va a ser utilizada durante este desarrollo puesto que se trabajará con proyección de coordenadas sobre la geosfera de referencia.

4.3 Integración del sensor GNSS

El sensor GNSS es otra unidad pasiva que necesita, en contraposición al magnetómetro, de una antena externa para la recepción de las señales emitidas por la red del sistema global de posicionamiento (GPS). Es por tanto necesario que la unidad, o al menos la antena, se encuentre en el exterior con línea directa (sin obstáculos) al cielo para que funcione de forma correcta.

El módulo GPS consta de cuatro pines de interconexión, dos de ellos son necesarios para la alimentación de energía eléctrica a 3.3V, mientras que los otros dos son necesarios para la transmisión y recepción de datos con la unidad Arduino.

Esta conexión de datos debe realizarse de forma cruzada, el transmisor en un extremo debe comunicar con el receptor en el otro, de igual forma actúa la señal de recepción. Por ello la comunicación con el módulo GPS se realiza a través de los pines de comunicación serie de Arduino.

```
Pinout for Arduino MKR 1010
Pin 13 RX -- Connect to GNSS TX pin
Pin 14 TX -- Connect to GNSS RX pin
```

Se utilizará la librería recomendada por el ensamblador de este módulo para el uso de este sensor y se creará un objeto perteneciente a esta librería para almacenar los datos recolectados por la señal de los satélites GPS.

```
#include <TinyGPS++.h> // GPS sensor library
TinyGPSPlus own_location; // GPS Sensor object
```

Es necesario inicializar el puerto serie hardware de Arduino (Serial1) para permitir la comunicación con el sensor, se usará la máxima velocidad de transmisión admitida, 9600 baudios.

```
Serial1.begin(9600);
```

Una vez que se le suministra energía eléctrica al sensor, éste empieza a recibir la señal de la constelación de satélites. La unidad tardará alrededor de 20 segundos en alcanzar la recepción de al menos 3 satélites, que son los mínimos necesarios para poder realizar la triangulación y el cálculo de la posición donde se encuentra.

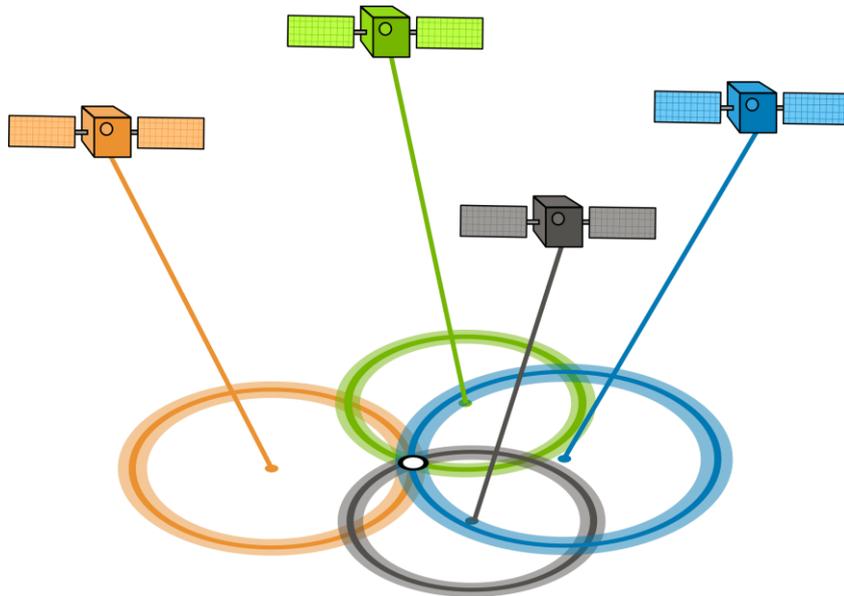


Ilustración 14. Ejemplo de localización GPS. Fuente [16]

La unidad entonces envía todos los datos recolectados (posición, fecha/hora, número de satélites, etc.) en diversas secuencias preformateadas utilizando el protocolo NMEA [10], esta información será recibida en el puerto serie de Arduino y la librería TinyGPS se encargará de codificarla para poder tratarla. La información de posición estará así disponible dentro de un objeto de clase para su visualización o uso.

4.3.1 Cálculo del rumbo hasta el destino

Es necesario conocer la posición del prototipo del sistema para poder realizar el cálculo de la trayectoria a seguir para llegar a destino. Las coordenadas de destino se almacenan en la unidad Arduino y, al igual que la declinación magnética, deben ser prefijadas con anterioridad a su uso. Las coordenadas usadas a modo de ejemplo para realizar la integración corresponden a la ciudad de Cuenca.

Conociendo las coordenadas de origen (posición actual) y las de destino (guardadas en el sistema), un simple calculo trigonométrico permitirá conocer el rumbo a seguir. Tomando la posición actual como origen de coordenadas, se puede representar la localización de destino y calcular el ángulo que forma el norte con respecto a la línea que une ambas localizaciones.



Ilustración 15. Cálculo dirección de destino

El rumbo que el usuario del sistema deberá seguir no corresponde con el ángulo calculado entre las dos coordenadas de referencia con respecto al norte, esa trayectoria deberá ser corregida a posteriori en base a la orientación del dispositivo y será mostrado como rumbo real en la pantalla LCD.

4.4 Integración pantalla LCD

Una vez se ha calculado la orientación con respecto al norte geográfico del sistema y el rumbo a seguir para llegar al destino desde la posición del dispositivo, resta mostrar dicha información al usuario/jugador de forma clara para que pueda tomar la decisión de rotar sobre su propio eje y apuntar de forma correcta hacia el geocache escondido.

Al jugador del sistema se le presentará la siguiente información en la pantalla LCD:

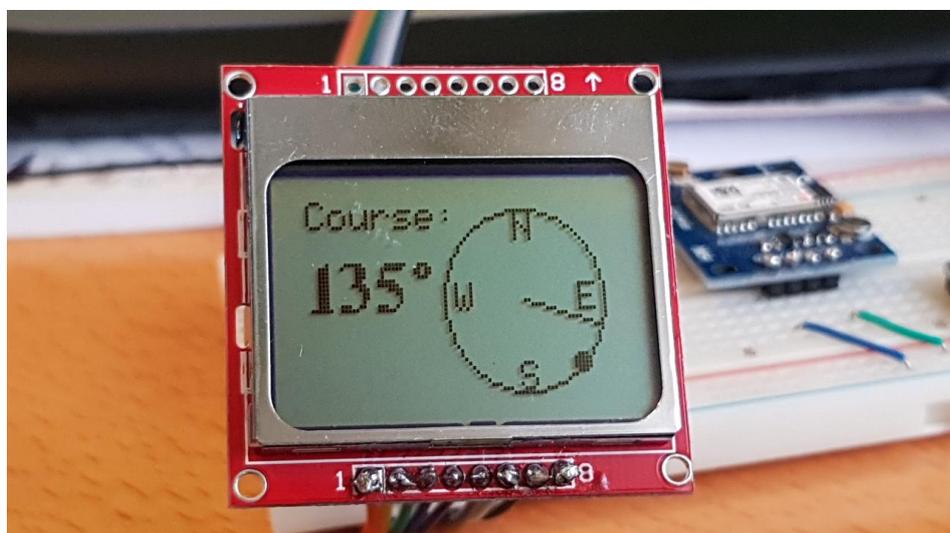


Ilustración 16. Integración pantalla LCD

En la parte izquierda de la pantalla se muestra la dirección de rumbo que debe tomar el jugador, en el ejemplo 135°. Esta información de rumbo se representa dinámicamente tanto en formato numérico como en la esfera de la rosa de los vientos, donde se encuentra representado por un círculo negro en el límite de la esfera. En función del movimiento del jugador por el terreno, dicha información se actualizará según la ubicación del dispositivo.

La orientación del sistema se muestra al usuario en forma de línea (radio) dentro de la esfera. Si se rota el dispositivo sobre sí mismo, la línea dará una (o más) vuelta completa en el mismo sentido de giro de la rotación. Como se explicó anteriormente la orientación toma por referencia el eje mayor del prototipo.

Tanto el rumbo a seguir como la orientación del sistema se muestran referenciadas con el norte geográfico.

Para conseguir que el usuario llegue a su destino se deben alinear la marca de rumbo (círculo negro) y la línea de orientación del sistema, seguidamente el usuario deberá avanzar en la dirección hacia la cual apunta el eje mayor del dispositivo.

La pantalla LCD es una matriz de 6 líneas de 8 píxeles de alto por 84 de largo, haciendo un total de 84x48 píxeles. Para escribir o dibujar en ella se debe especificar las coordenadas de cada elemento representado.

Para la representación de los datos en pantalla se ha utilizado las librerías gráficas proporcionadas por Adafruit [13] para este tipo de pantallas, que permitirán escribir texto y formas geométricas básicas (como el círculo de la esfera o líneas entre 2 puntos).

Se utilizan los datos de orientación y rumbo calculados en los apartados anteriores, para representarlos como:

Orientación: Una línea de orientación, el origen de la misma es el centro de la esfera de la rosa de los vientos y el punto final se calcula en función del ángulo de orientación.

Las coordenadas del final de la línea se obtienen utilizando las funciones trigonométricas que hallan el seno y coseno del ángulo de orientación, esas medidas deben ser posteriormente referenciadas en base al centro del círculo y con una extensión igual la radio de la esfera, el código necesario para el cálculo es el siguiente:

```
int endX, endY;

endX = centerX + (r * sin(toRadians(decimalDegrees)));
endY = centerY - (r * cos(toRadians(decimalDegrees)));
```

Rumbo: De igual modo, el símbolo del rumbo se obtiene dibujando un círculo relleno de radio muy pequeño (2 píxeles), sobre la coordenada del ángulo de rumbo anteriormente calculado y proyectada sobre el borde de la esfera.

```
display.fillCircle(centerX+(r*sin(toRadians(courseToDegrees))),  
centerY-(r*cos(toRadians(courseToDegrees))),2,BLACK);
```

4.5 Desarrollo del software de control

Aunque es necesario proveer al administrador del sistema de la posibilidad de cambiar los valores almacenados en la memoria interna y de monitorizar que los sensores están funcionando de forma correcta, el sistema cumpliendo los requisitos de autonomía y portabilidad, debe funcionar sin que se establezca ninguna conexión entre el dispositivo y un sistema externo.

Esto quiere decir que todo el servicio web de administración estará supeditado a la conexión inalámbrica, pero si ésta no se establece en los primeros 15 segundos tras el arranque del sistema, dicha conexión no intentará restaurarse hasta que no se haga un reinicio y por tanto no será posible la administración del dispositivo.

Esta solución de diseño se ha implementado de este modo para ahorrar el máximo consumo de energía posible puesto que mientras se utilice el dispositivo en el campo no será necesaria, ni posible, ninguna conexión Wireless.

Las funcionalidades de administración estarán disponibles a través de la comunicación entre un navegador web estándar y el servidor web interno, que ofrecerá una página web de monitorización y control del sistema.

La ilustración que se muestra a continuación representa un diagrama del funcionamiento del servicio web implementado. En los siguientes párrafos se explicará cómo trabaja el servicio web en detalle.

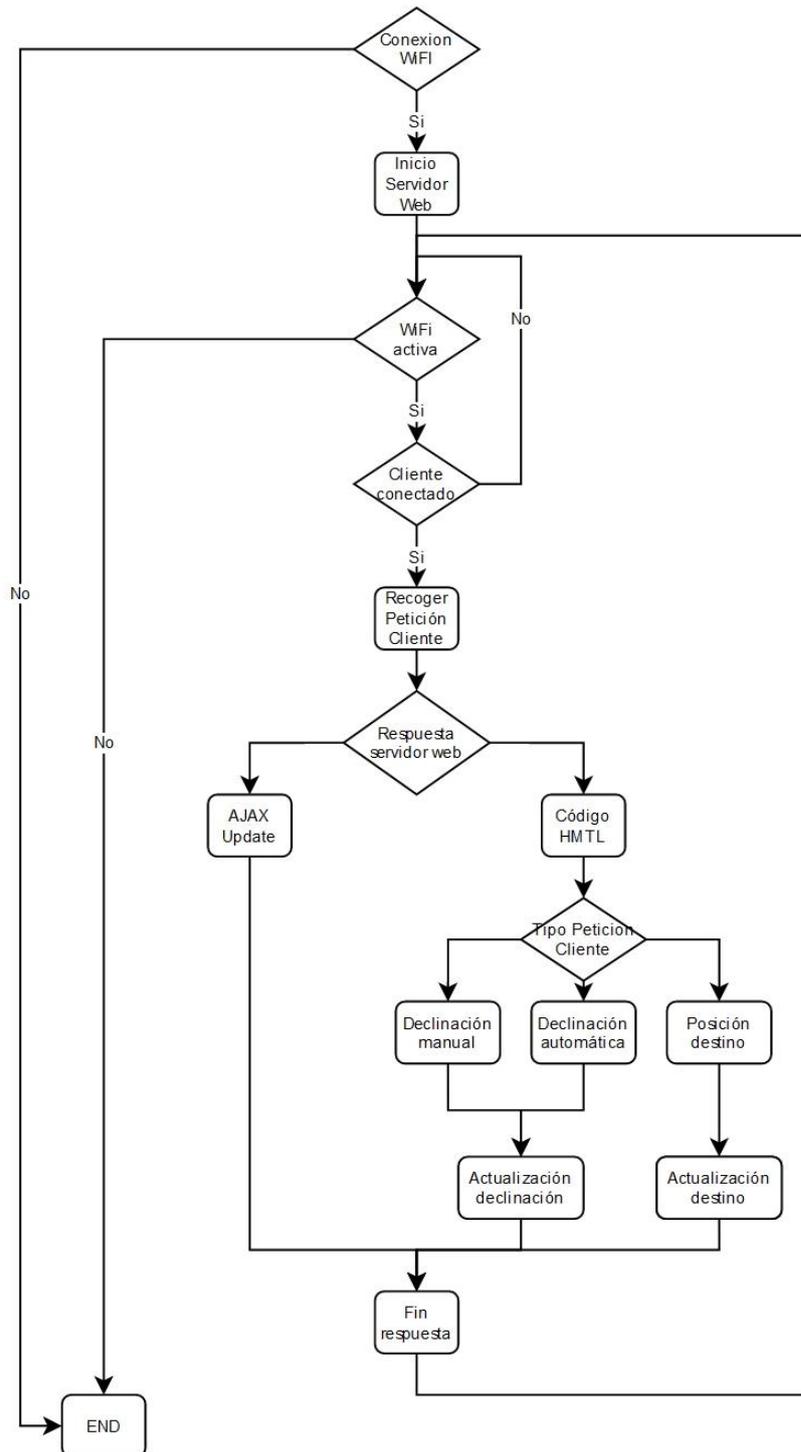


Ilustración 17. Diagrama servicio web

La conexión al servidor web se realizará a través de la conexión WiFi que está disponible a través del módulo Arduino, por consiguiente, la primera tarea necesaria a implementar es la de integrar esa conexión con una red Wireless disponible.

La librería de Arduino WiFinINA proporcionará lo necesario para poder conectar la placa Arduino con la red local. Como no es una red abierta es

necesario indicar el nombre de red (SSID) y la contraseña de acceso, ambos datos se han guardado en un fichero secundario ('arduino_secrets.h') por motivos de seguridad para que no apareciesen en texto claro en el código mostrado.

```
#include <WiFiNINA.h>           // WiFi library
#include "arduino_secrets.h" //Store wireless network info, SSID
and Password

// Wireless variables
char ssid[] = SECRET_SSID;     // WiFi SSID
char pass[] = SECRET_PASS;     // WiFi password
```

La conexión con el servidor web se hará por el puerto http estándar 80. Se realizarán un máximo de cinco intentos para conectar a la red local, con un tiempo de espera entre intentos de 3 segundos, si durante ese tiempo no se ha podido conectar se sobreentenderá que se está fuera del alcance (por ejemplo, en uso en el campo) y no se volverá a intentar la conexión. Si el sistema no ha podido conectarse, el servidor web no será iniciado para no consumir recursos ni batería.

```
int status = WL_IDLE_STATUS; //WiFi status before init is IDLE
WiFiServer server(80);       //WiFi Webserver at port 80

// try to connect to WiFi network (5 times ~15 seconds)
while (status != WL_CONNECTED && counter < 5) {
  Serial.print("Attempting to connect to SSID: ");
  Serial.println(ssid);
  status = WiFi.begin(ssid, pass);
  // wait 3 seconds for connection:
  delay(3000);
  counter++;
}

// Print a comment into serial monitor if connected to WiFi
network
if (status == WL_CONNECTED) {
  Serial.println("Connected to Wireless Network");
  server.begin();
}
```

Si se produce la conexión a la red WiFi y además existe un cliente (navegador web) que está conectado y disponible, el servidor guardará la petición enviada por el navegador para analizarla y, en función del tipo y contenido de la petición, se desencadenará la respuesta apropiada por el servidor web.

```
String HTTP_req;           //variable to store browser request

// check if there is a client connected
if (status == WL_CONNECTED) {
  WiFiClient client = server.available();
  if (client) {
    while (client.connected()) {

      //read and store client request
```

```

while (client.available()) {
    char c = client.read();
    HTTP_req += c;
} //end while client.available

```

Una vez se haya terminado de recibir la petición del navegador la comunicación de protocolo HTTP indica que el servidor debe mandar una confirmación y una respuesta con el contenido que se va a enviar. El tipo de contenido variará dependiendo de la petición, pero el inicio de la respuesta siempre va a ser el mismo, se enviará un aviso de disponibilidad.

```

client.println("HTTP/1.1 200 OK");

```

Si la petición del navegador es el tipo GET estándar o de tipo POST (ver más adelante) se enviará el final de la respuesta HTTP, indicando que ésta será de tipo HTML, y a continuación se mandará la página web completa.

La final respuesta indicando el tipo HTML:

```

//end of server response as HTML
client.println("Content-Type: text/html");
client.println("Connection: keep-alive");
client.println();

```

La página web inicial enviada tiene varias secciones a destacar. Existe contenido dentro del cuerpo (BODY) de dicha página que se encuentra sin datos rellenos entre etiquetas y con un identificador único.

```

client.println("<h1>Sensor measurements</h1>");
client.println("<p>Current Azimuth value: <span
id=\"az\">...</span></p>");
client.println("<p>Current Position:</p><p>Latitude= <span
id=\"lat\">...</span></p>");
client.println("<p>Longitude= <span
id=\"lng\">...</span></p>");

client.println("<p>Current Course: <span
id=\"c\">...</span></p>");

```

Este contenido será rellenado automáticamente por un script que se ejecuta al cargar la página web por primera vez y posteriormente cada segundo.

Este script envía una petición GET al servidor con la cláusula especial 'ajax_update'. Cuando el servidor recibe una petición evalúa si en el contenido aparece la cadena de texto 'ajax_update' y de ser así se envía una respuesta de tipo XML y a continuación un fichero XML en cuyo contenido están los datos que aún no estaban rellenos en la página web principal. Estos datos ya fueron calculados en las secciones de código anteriormente descritas, en concreto azimuth (orientación del dispositivo),

la posición del prototipo (latitud y longitud) y rumbo a destino desde la posición actual.

Sensor measurements

Current Azimuth value: 97° 21' 1.55"

Current Position:

Latitude= 40.396228

Longitude= -3.605228

Ilustración 18. Contenido dinámico actualizable

El script incrustado en la propia página web procesa la respuesta recibida por el servidor e inserta los datos, identificados por una etiqueta única en el fichero XML, dentro de las etiquetas HTML correspondientes. Cada segundo, el script se vuelve a ejecutar por lo que estos datos son actualizados de nuevo con el último valor disponible.

Otra de las secciones disponibles en la página web inicial es el formulario relativo a la declinación magnética.

Magnetic Declination

Current Magnetic Declination value: -0.283333

Manual Update: Insert new Magnetic Declination:

Automatic Magnetic Declination Update

Ilustración 19. Formulario declinación magnética

Este formulario permite cambiar de forma manual el valor de la declinación magnética al administrador rellenando el campo de texto con el valor en grados decimales y pulsando el botón 'Update'. El pulsar el botón se generará una petición de tipo POST al servidor web, que procederá a analizar para extraer el contenido del campo de texto y, tras una conversión a formato numérico, actualizará el valor de la declinación magnética en el sistema.

El valor de la declinación magnética también puede actualizarse de forma automática, para ello se tiene en cuenta la posición actual del dispositivo, adquirida mediante el sensor GPS y a través de la API que provee el Centro Nacional de Datos Geofísicos (National Geophysical Data Center, NGDC) [17] perteneciente al servicio de información ambiental de los Estados Unidos (NOAA), se obtiene el valor de la declinación magnética para las coordenadas indicadas. Posteriormente se procesa la respuesta

devuelva por el NGDC y se extrae el dato de declinación para actualizarlo en la memoria flash interna del sistema.

Ha de tenerse en cuenta que al realizar el cambio de valor de la declinación magnética se modificará automáticamente el valor del azimut real que se muestra por medio de la página web, puesto que el contenido de ese valor es dinámico.

Por último, la página web inicial contiene otro formulario relativo a la posición de destino prefijada en el sistema.

Course to destination

Current destination coordinates:

Latitude= 40.070393

Longitude= -2.137416

New Target Latitude: New Target Longitude:

Ilustración 20. Formulario posición de destino

Este formulario permitirá al administrador cambiar la posición del destino donde se quiere llevar, al ejecutar el cambio, el valor de las variables que almacenan la latitud y longitud de la coordenada de destino será modificado. El valor de rumbo a destino cambiará de forma dinámica, puesto que es un valor calculado cada segundo.

Al rellenar este formulario con los valores, en formato de grados decimales, de la latitud y longitud de la nueva posición de destino, y pulsar el botón 'Update', el navegador enviará una petición POST al servidor web. Éste analizará y procesará la petición y extraerá los nuevos valores de latitud y longitud de dicha petición, posteriormente los convertirá a formato numérico y modificará el valor almacenado en memoria interna del sistema de la coordenada de destino con los nuevos datos.

En la siguiente figura se puede observar el aspecto de la página web una vez los valores han sido actualizados por el script que procesa el fichero XML.

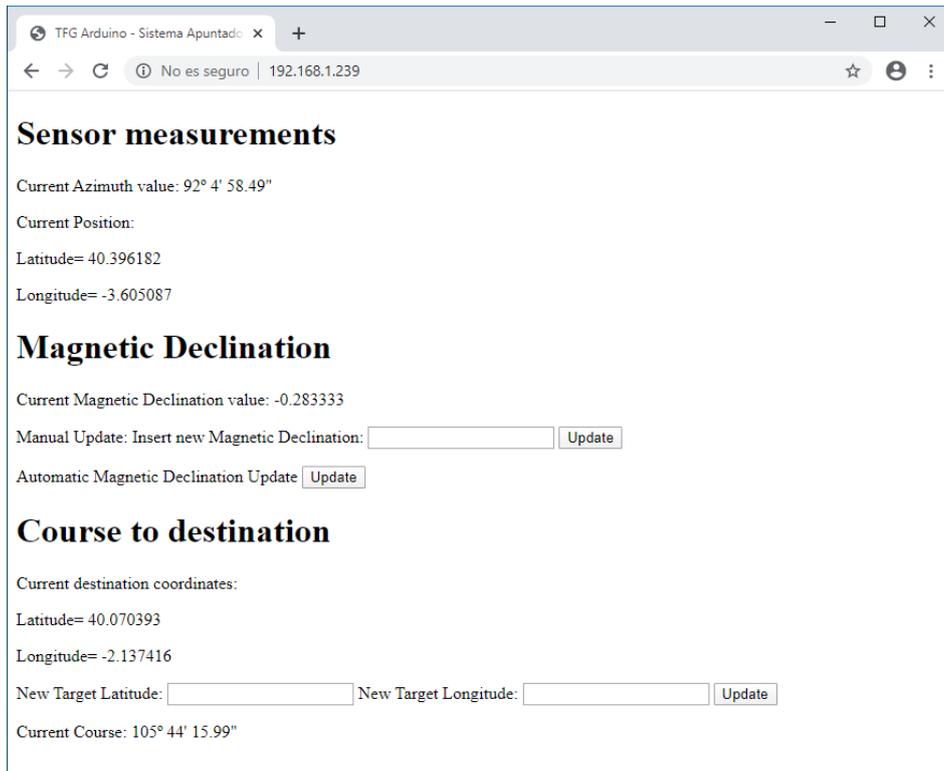


Ilustración 21. Página web de administración

5. Pruebas y resultados obtenidos

En este apartado se describirán las pruebas y resultados obtenidos durante las distintas fases de la implementación del proyecto. La realización de esta batería de pruebas, tanto las pruebas de conexión inicial como las específicas para cada una de las funcionalidades, ha permitido depurar el desarrollo y solventar los errores encontrados durante el mismo.

La siguiente tabla asocia, de forma visual, los grupos de pruebas diseñados para la verificación de los requisitos del sistema

Requisito	Grupo de Pruebas				
	5.2.1	5.2.2	5.2.3	5.2.4	5.2.5
Autonomía (6-8 horas)					✓
Portabilidad	*	*	*	*	*
Reutilización			✓		
Posicionamiento (Error < 10m)		✓			
Orientación (Error <3°)	✓				
Direccionamiento (Paso <1°)				✓	

El requisito de portabilidad se sobreentiende como cumplido al establecer dicho requisito como parte fundamental a la hora de seleccionar el hardware necesario para la construcción del prototipo, cuyo resultado se puede observar en el apartado 4. Implementación.

5.1 Pruebas de conexión

Se realizarán unas pruebas básicas de conexión a los distintos sensores para familiarizarnos con el acceso a los mismo y a sus características.

5.1.1 Conexión WiFi Arduino

Esta prueba se realizará para comprobar que el módulo WiFi de la placa Arduino funciona de forma correcta y se puede conectar la red de área local doméstica. Esta conexión es indispensable para el funcionamiento del servicio web a integrar.

El siguiente código ha sido utilizado para comprobar la conexión WiFi, se hace uno de un fichero adicional 'arduino_secrets.h' donde se almacenan los valores del SSID y contraseña utilizados para establecer dicha conexión.

```

#include <WiFiNINA.h>
#include "arduino_secrets.h" //network stored values

char ssid[] = SECRET_SSID;    // network SSID
char pass[] = SECRET_PASS;    // network password

int status = WL_IDLE_STATUS;

void setup() {
  Serial.begin(9600);

  // attempt to connect to WiFi network:
  while ( status != WL_CONNECTED) {
    status = WiFi.begin(ssid, pass);
    delay(5000); // wait 5 seconds
  }

  // print connectio info
  Serial.print("WiFi SSID: ");
  Serial.println(WiFi.SSID());
  IPAddress ip = WiFi.localIP();
  Serial.print("IP Address: ");
  Serial.println(ip);
  long rssi = WiFi.RSSI();
  Serial.print("Signal strength: ");
  Serial.print(rssi);
  Serial.println(" dBm");
}

void loop() {
}

```

La salida del monitor de puerto serie nos indica que la conexión fue realizada satisfactoriamente y nos provee de información acerca de la dirección IP asignada al módulo, así como de la calidad de la señal inalámbrica. El SSID de la red se ha ocultado por razones de privacidad.

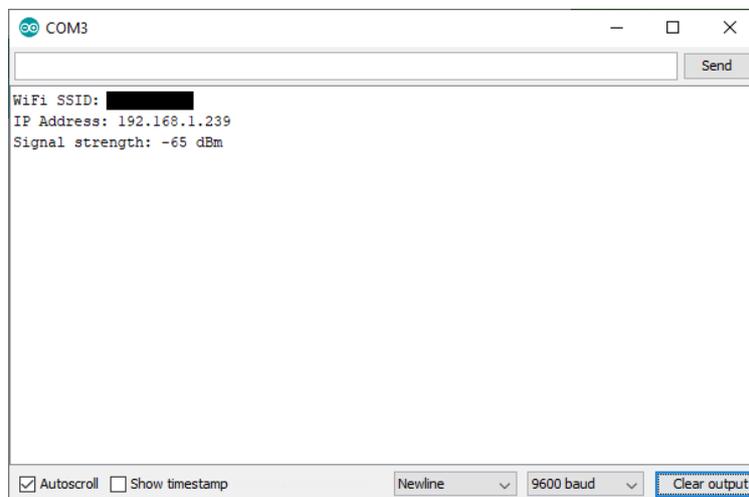


Ilustración 22. Prueba de conexión WiFi

5.1.2 Magnetómetro

Utilizaremos el siguiente código para comprobar tanto que la conexión física con el magnetómetro está funcionando como que el sensor es capaz de realizar lecturas del campo magnético.

Esta rutina simple usa la librería oficial del sensor y lee los valores del campo magnético en cada uno de los 3 ejes.

```
/*
 * Pin 11 SDA -- Connect to Digital Magnetic Compass (DMC) SDA
pin
 * Pin 12 SCL -- Connect to DMC SCL pin
 */

#include <QMC5883LCompass.h>

QMC5883LCompass compass;

void setup() {
  Serial.begin(9600);
  compass.init(); // inti compass with default values
}

void loop() {
  float azimuth; // to store azimuth value

  // Read compass values
  compass.read();

  // Print measured values @ each axis
  Serial.print("X: ");
  Serial.print(compass.getX());
  Serial.print(", Y: ");
  Serial.print(compass.getY());
  Serial.print(", Z: ");
  Serial.println(compass.getZ());

  delay(250);
}
```

El resultado de la prueba es satisfactorio puesto que, como se observa en la salida por el puerto serie, el sensor es capaz de leer el campo magnético de la tierra.

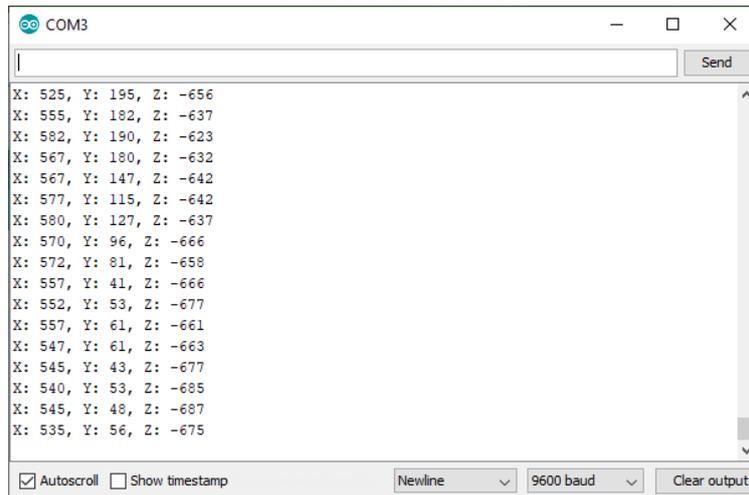


Ilustración 23. Prueba de conexión con el magnetómetro

5.1.3 Sensor GNSS

El código mostrado servirá como prueba para la conexión del sensor GNSS y como verificación de que el módulo puede recibir la señal de los distintos satélites GPS y conocer su localización de forma precisa.

```

/*
 * Pin 13 RX -- Connect to GNSS TX pin
 * Pin 14 TX -- Connect to GNSS RX pin
 */
#include <TinyGPS++.h>

TinyGPSPlus gps;    //GPS object

void setup() {
  Serial.begin(9600);
  Serial1.begin(9600); //GPS Sensor connected to Serial 1
}

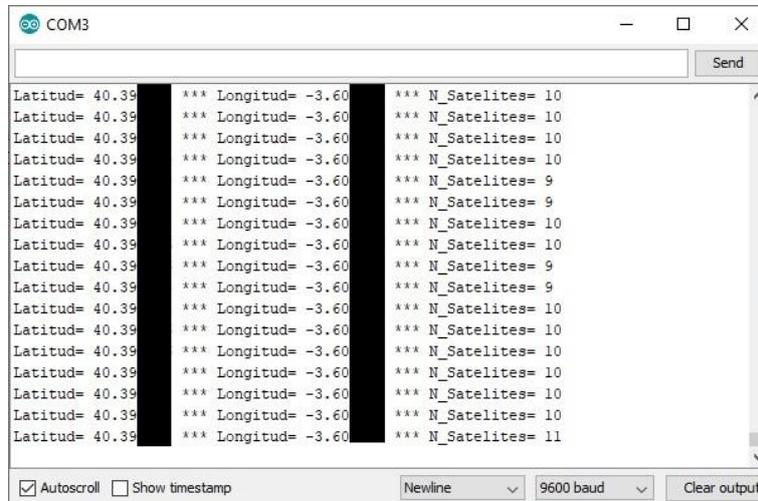
void loop() {
  while (Serial1.available()>0){ //while new NMEA sentence

    gps.encode(Serial1.read()); // encode NMEA sentence

    // print location and number of satellites
    if (gps.location.isUpdated()){
      Serial.print("Latitud= ");
      Serial.print(gps.location.lat(), 6);
      Serial.print(" *** ");
      Serial.print("Longitud= ");
      Serial.print(gps.location.lng(), 6);
      Serial.print(" *** ");
      Serial.print("N_Satellites= ");
      Serial.print(gps.satellites.value());
      Serial.println();
      delay(250);
    }
  }
}

```

Como demuestra la ilustración, la unidad GPS ha podido recibir correctamente la señal de al menos 9 satélites y averiguar las coordenadas de su posición (los últimos dígitos de la posición han sido ocultados por razones de privacidad).



```
COM3
Send
Latitud= 40.39 *** Longitud= -3.60 *** N_Satelites= 10
Latitud= 40.39 *** Longitud= -3.60 *** N_Satelites= 9
Latitud= 40.39 *** Longitud= -3.60 *** N_Satelites= 9
Latitud= 40.39 *** Longitud= -3.60 *** N_Satelites= 10
Latitud= 40.39 *** Longitud= -3.60 *** N_Satelites= 10
Latitud= 40.39 *** Longitud= -3.60 *** N_Satelites= 9
Latitud= 40.39 *** Longitud= -3.60 *** N_Satelites= 9
Latitud= 40.39 *** Longitud= -3.60 *** N_Satelites= 10
Latitud= 40.39 *** Longitud= -3.60 *** N_Satelites= 10
Latitud= 40.39 *** Longitud= -3.60 *** N_Satelites= 10
Latitud= 40.39 *** Longitud= -3.60 *** N_Satelites= 9
Latitud= 40.39 *** Longitud= -3.60 *** N_Satelites= 10
Latitud= 40.39 *** Longitud= -3.60 *** N_Satelites= 10
Latitud= 40.39 *** Longitud= -3.60 *** N_Satelites= 10
Latitud= 40.39 *** Longitud= -3.60 *** N_Satelites= 11
```

Ilustración 24. Prueba de conexión con sensor GNSS

5.1.4 Pantalla LCD

Para comprobar la conexión con la pantalla LCD ejecutaremos el siguiente bloque de código que nos permitirá dibujar un círculo en el centro de la pantalla con el radio máximo permitido

```
#include <Adafruit_GFX.h>
#include <Adafruit_PCD8544.h>

Adafruit_PCD8544 display = Adafruit_PCD8544(1, 2, 3, 4, 5);
uint16_t centerX = 41;
uint16_t centerY = 23;
float radius = 23;

void setup() {
  Serial.begin(9600);

  //Initialize Display
  display.begin();

  display.setContrast(60);

  display.clearDisplay();

  //Draw Circle
  display.drawCircle(centerX, centerY, radius, BLACK);

  display.display();
}

void loop() {}
```

5.2 Pruebas funcionales

Este grupo de pruebas está orientado a verificar que cada una de las funcionalidades deseadas del sistema (requisitos) cumple de forma correcta a su propósito, especialmente en las condiciones de posibles errores.

5.2.1 Orientación con respecto al norte geográfico

Se realizan pruebas para verificar que el cálculo de norte geográfico se corrige de forma adecuada cuando se tomen valores positivos o negativos de la declinación magnética cuando la orientación del prototipo cambie entre los distintos cuadrantes (NE, NW, SW, SE).

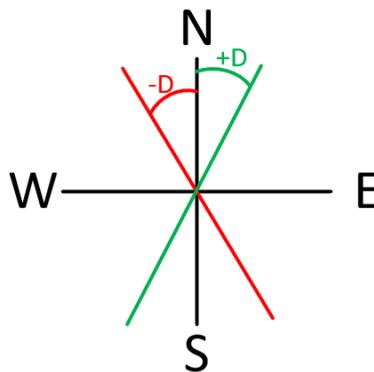


Ilustración 25. Corrección declinación magnética

Las pruebas consisten en establecer valores positivos y negativos de la declinación magnética y observar cómo se comporta el sistema cuando rotando el dispositivo sobre sí mismo se cambia de cuadrante y especialmente cuando la rotación alcanza la orientación cercana al norte geográfico.

El resultado de estas pruebas es satisfactorio puesto que el comportamiento del sistema es el esperado al cambiar entre cuadrantes y cuando la orientación real calculada se acerca progresivamente a 360° o 0° (dependiendo del sentido de giro), el sistema en esas circunstancias nunca ofrece valores negativos o superiores a 360° para la dirección de orientación.

Otra de las pruebas realizadas al sistema de orientación es la estandarización de las medidas obtenidas del magnetómetro. Puesto que las medidas del magnetómetro son lecturas tomadas de forma instantánea de un sensor que de forma continua está midiendo el campo magnético, dichas medidas unitarias pueden llevar un error asociado (el error declarado por el fabricante es menor a 2°). Es por ello por lo que se ha optado por presentar un único dato de orientación que se obtiene de

realizar el promedio de varias de las medidas tomadas de forma consecutiva.

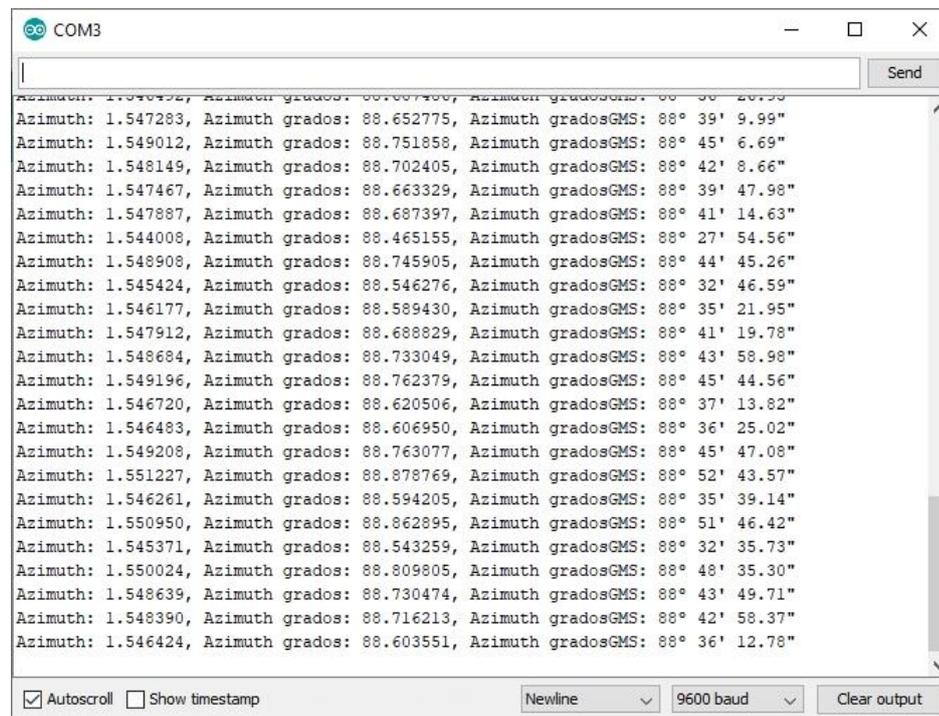


Ilustración 26. Medida de orientación promedio

Como se puede observar en la ilustración anterior las medidas promedio toman una distancia entre ellas de unos 25 minutos de grado sexagesimal. Este dato, junto con la corroboración de la exactitud de la medida tomada por el magnetómetro que se ha verificado tanto con una brújula Silva Ranger S y por la aplicación de Brújula Digital de Axiomatic [18], nos permiten tener plena confianza en que el requisito de Orientación (error $<3^{\circ}$) se ha satisfecho plenamente.

Se ha probado asimismo a desviar las medidas de magnetómetro ubicando un imán en las cercanías del dispositivo. Como era de esperar, las medidas tomadas cuando el imán está muy cerca aparecen distorsionadas, pero si el imán está a una distancia superior a los 50 cm el sistema recupera las lecturas correctas. Por tanto, el jugador de geocaching, mientras lleve el dispositivo, no debería portar ningún imán u objeto que emita un campo magnético o, al menos, separar ambos dispositivos 50 cm como mínimo.

5.2.2 Cálculo de la dirección de destino

Se comprueban las distintas casuísticas de cálculo de la dirección de destino (azimut que une la posición actual y las coordenadas de destino sin tener en cuenta la orientación del sistema) cuando, tomando la

posición del sistema como eje de coordenadas, el destino se encuentre en los distintos cuadrantes (NE, NW, SW, SE).

La primera parte de la prueba consiste en verificar la precisión de la medida de posicionamiento que nos ofrece el sensor GPS. Según el fabricante la precisión de la medida es menor a 2,5 metros, pero las pruebas realizadas en campo ofrecen otro resultado más realista. En concreto se utilizaron 2 smartphones distintos (un Samsung y iPhone) que ofrecieron una medida casi idéntica y que nos arroja un error de medida del prototipo de casi 5 metros.

Este error de medición, aunque se encuentra dentro de los parámetros establecidos dentro de los requisitos de posicionamiento esperados (precisión de posicionamiento menor a 10 metros) en observaciones en campo abierto, aumenta drásticamente si se intenta obtener la posición cuando estamos cerca de un edificio elevado o en una calle relativamente estrecha, dando lugar a mediciones con una desviación de hasta 100 metros.

Por tanto, es preciso notar que los jugadores obtendrán medidas erróneas siempre y cuando se aproximen a lugares angostos como un cañón o un desfiladero o se encuentren junto a una formación rocosa que les haga 'sombra' y la antena del dispositivo no pueda recibir la señal de la constelación GPS.

Una vez que se ha verificado la confiabilidad de la medida de posición obtenida por el sensor GPS procede a verificar si el cálculo de la dirección a destino se realiza correctamente. Para ello se toman los siguientes destinos posibles, Moscú, Nueva York, Buenos Aires y Sídney cada uno de ellos se encuentra en uno de los distintos cuadrantes de un mapamundi.

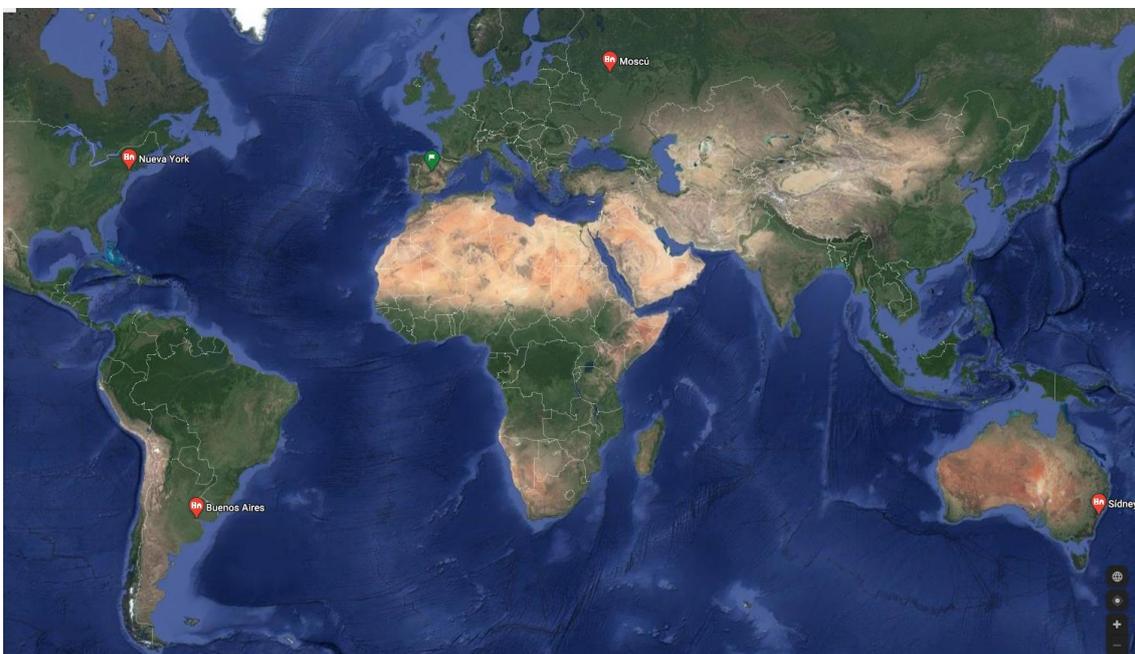


Ilustración 27. Mapamundi ofrecido por Google Maps.

Se han seleccionado dichas ciudades puesto que poseen distintos valores, positivos o negativos, de latitud y longitud en formato decimal.

Para todo estos valores y otros elegidos al azar, se ha realizado el cálculo del rumbo y se ha comprobado matemáticamente y usando una herramienta online de Igismap.com [19] que dichos cálculos eran correctos.

5.2.3 Servicio de Administración Web

Pruebas de conexión inalámbrica

La administración del sistema ha sido diseñada para poder proporcionarse por medio de una conexión inalámbrica. Esta conexión WiFi debe establecerse entre el dispositivo y un punto de acceso de la red local del domicilio/empresa u otro punto de acceso móvil, por ejemplo, un smartphone.

Las pruebas de conexión han sido realizadas con un router casero de la marca Asus y un teléfono móvil Samsung, en ambos casos para una distancia inferiores a 20 metros (distancia máxima probada) entre dispositivos se ha medido el indicador de fuerza de la señal recibida (Received Signal Strength Indication, RSSI) y pese a que la calidad de la señal varía levemente con la distancia, no se ha encontrado ninguna pérdida de señal apreciable en la página de administración debido, posiblemente, al poco flujo de datos necesarios entre el cliente y servidor.

La distancia máxima (20 metros) utilizada en estas pruebas es suficiente para verificar la conexión con el sistema y para realizar las labores de administración y monitorización, puesto que estas tareas son realizadas por el administrador con la presencia física de los dispositivos.

De todos modos, ha de tenerse en cuenta que la placa Arduino integra un módulo WiFi que sigue el estándar 802.11 b/g/n y que la distancia máxima teórica para este tipo de módulo es de 400 metros. Por lo que el uso de cualquier punto de acceso hardware que siga el estándar 802.11 b/g/n debe permitir la conexión con el dispositivo e incluso permitir conexiones a mayor distancia a la utilizada para las pruebas.

Presentación de medidas de sensores en servidor web

Para que el administrador del sistema pueda monitorizar que los sensores están funcionando de manera adecuada se comprobará que los datos mostrados en el servicio web muestran los valores tomados por estos.

Se verifica que los datos de orientación y posición de los sensores se muestran y actualizan de forma dinámica en los datos presentados en el

cliente (navegador web) por el servicio web. Este proceso ha sido comprobado tanto en el entorno de desarrollo como en campo, donde se ha establecido un smartphone como punto de acceso wifi y se ha conectado con el prototipo a través del navegador integrado en el dispositivo.

La siguiente imagen muestra una captura realizada con la aplicación Wireshark donde se observa el envío automático de datos en formato XML a la página web de administración.

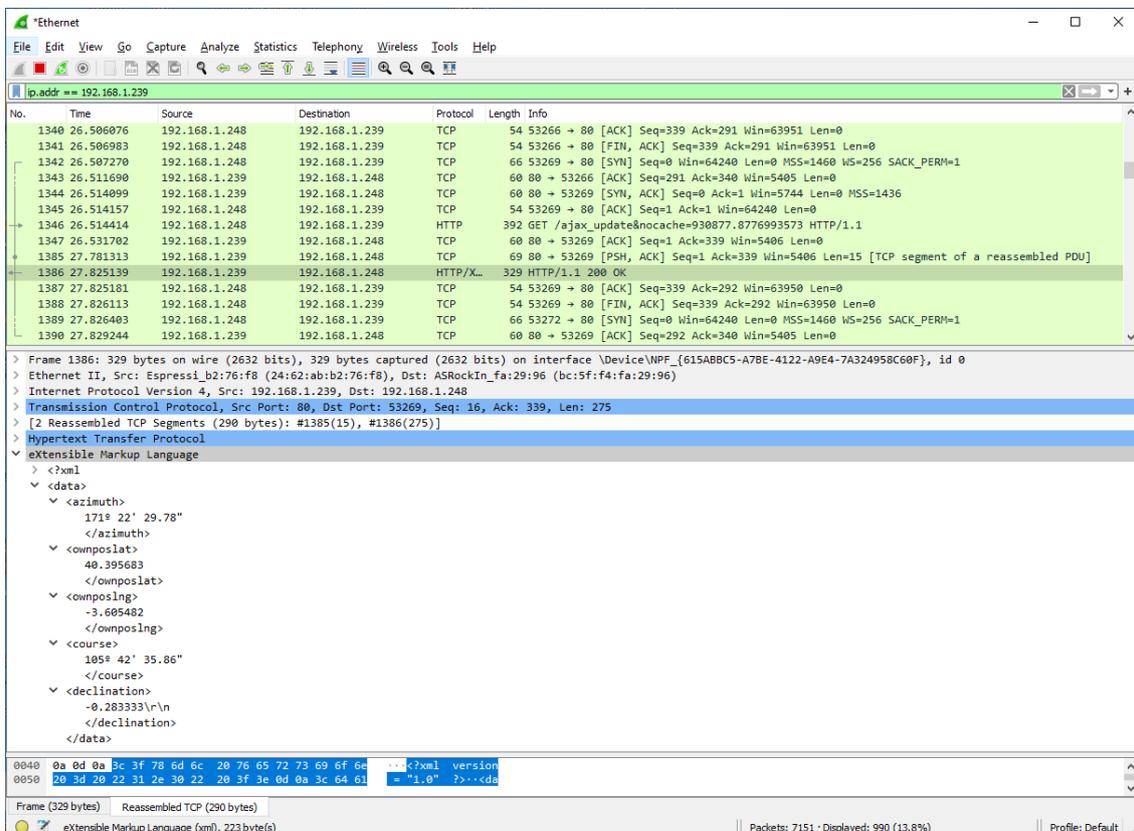


Ilustración 28. Envío datos XML

Actualización de datos mediante servicio web

Se comprobará que el servidor web embebido en el sistema lee y extrae de forma adecuada los datos enviados para actualizar, declinación magnética (manual o automática) y coordenadas de posición del destino, y modifica los valores anteriormente almacenados por los nuevos.

Se procede a introducir en el formulario de declinación magnética distintos valores, tanto positivos como negativos, en formato de grados decimales y se comprueba que el valor de orientación mostrado en el área de 'medidas de sensores' varía de forma acorde al valor del dato introducido.

Posteriormente, y dejando un valor de declinación magnética que no corresponde con el real para la posición actual, se pulsa sobre el botón de actualización automática de la declinación magnética. Se comprueba,

utilizando la herramienta Wireshark, que se construye y envía una petición a los servidores de NGDC [17] y que esta petición es respondida con un nuevo valor de declinación magnética que pasa a ser actualizado e incorporado a la medición de la orientación del dispositivo.

La comunicación con el servidor del NGDC (<https://www.ngdc.noaa.gov>, dirección IP 140.172.190.1) está cifrada al requerir el protocolo HTTPS para la consulta, en la siguiente ilustración se muestra la captura de la conexión con dicho servidor:

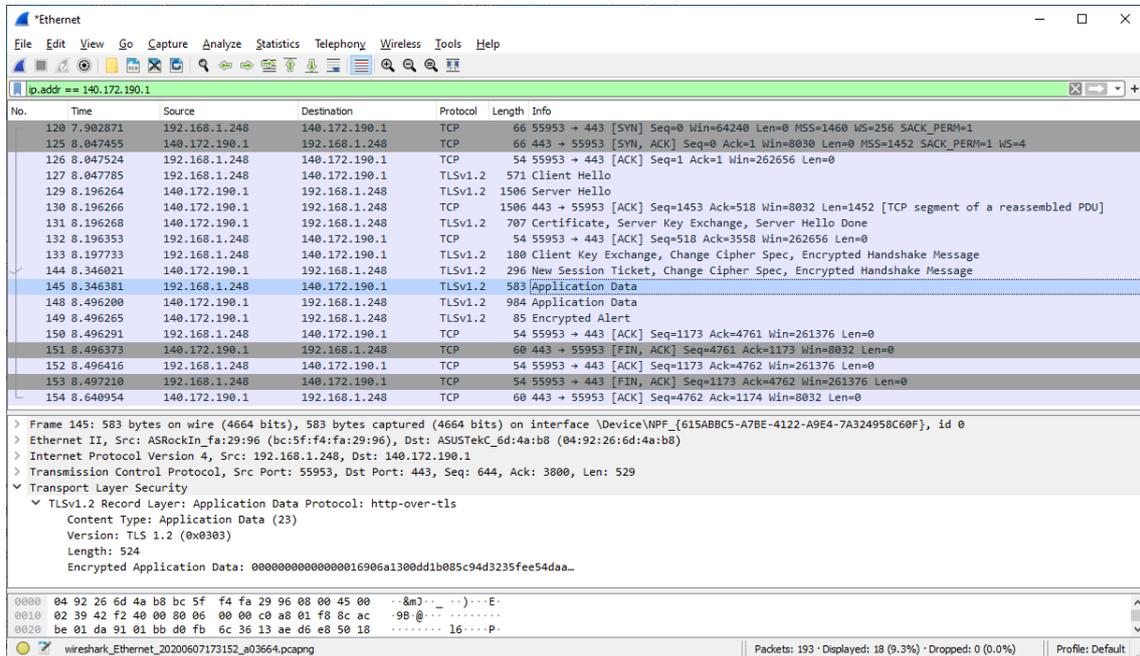


Ilustración 29. Actualización automática declinación magnética.

El fichero XML devuelto por el servidor del NOAA, con la información de la declinación magnética, puede verse utilizando la consola del navegador WEB:

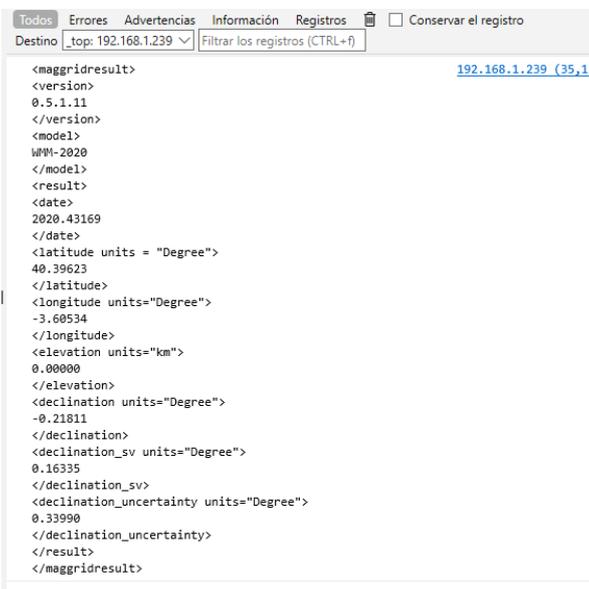


Ilustración 30. Fichero XML, declinación automática.

Por último, se realizan actualizaciones de la posición de destino utilizando el formulario correspondiente. Se introducen varios valores tanto positivos como negativos de los campos de latitud y longitud (en formato de grados decimales) de la posición de destino.

Se comprueba, utilizando la herramienta online lgismap.com [19], que los datos resultantes, calculados en función de la posición actual y los valores de destino introducidos, coinciden plenamente con los mostrados en la página web de administración (además de en la pantalla LCD) y la citada herramienta online.

Persistencia de los datos actualizados

Se realizarán pruebas de acceso de lectura y escritura a la memoria Flash del módulo Arduino y se comprobará que los datos a actualizar desde el servicio web (declinación magnética y coordenadas del destino) se modifican correctamente en la memoria Flash del sistema.

Como continuación a las pruebas anteriores de actualización de datos, se procede, de igual forma, a modificar los valores de la declinación magnética y la coordenada de destino del dispositivo.

Para comprobar que los valores han sido almacenados en la memoria flash interna del sistema se procede a pulsar el botón de 'reset' situado en el módulo Arduino, lo que provoca un reinicio del dispositivo, ejecución del código almacenado y, como parte de este, el acceso a los datos guardados en memoria. Se comprueba que los datos mostrados tras el reinicio concuerdan con los últimos datos actualizados a través de la página de administración del dispositivo.

Alternativamente esta prueba también ha sido realizada desconectando y volviendo a conectar la batería mientras ésta era la única fuente de alimentación del prototipo.

Se concluye, tras todas las pruebas del servicio de administración web que el prototipo está diseñado de forma correcta para poder ser utilizado múltiples veces con distintos puntos de destino y para distintas zonas geográficas (distintas declinaciones magnéticas), dando por cumplido el requisito de reutilización del sistema.

5.2.4 Presentación de la información en la pantalla LCD

Teniendo los valores de orientación con respecto al norte magnético y de la dirección de destino, queda por verificar el requisito de direccionamiento que indica que el sistema debe mostrar las medidas para llegar al destino en incrementos no mayores a 1º sexagesimal para que la indicación al jugador no le lleve a un error muy elevado.

Esta prueba consistirá en la representación de la dirección de rumbo a seguir por el usuario del sistema. Para ello se utilizan distintos ángulos de orientación y destino para comprobar el correcto funcionamiento cuando esas dos direcciones estén en cualquiera de los cuatro cuadrantes (NE, NW, SW, SE).



Ilustración 31. Direccionamiento, incremento entre medidas mostradas.

Como se muestra en la ilustración superior, la representación final al jugador puede llegar a una precisión de un grado (1°) sexagesimal, lo cual resultará muy útil cuando el jugador este cerca del destino escondido para mostrar de forma precisa donde se halla este y cumple con el requisito especificado.

5.2.5 Pruebas de alimentación

Aunque el sistema puede ser alimentado tanto por USB como por batería, el diseño de este prototipo pretende ceñirse al requisito de autonomía que establece que el dispositivo debe funcionar de forma autónoma durante una jornada de trabajo de un 6-8 horas.

Pese a que el cálculo teórico de la duración de la batería seleccionada estima que el requisito se debería cumplir sobradamente (véase 3.2.1.5 Batería), se ha decidido realizar pruebas de duración real de la batería.

Las pruebas consistieron en, tras la carga de la batería a través del módulo Arduino, dejar el sistema funcionando durante toda la noche durante 5 noches distintas. Estas pruebas se llevan a cabo sin la conexión Wireless establecida para emular las condiciones que tendría el sistema en el campo mientras este en uso por un jugador.

Los resultados aportaron una media de duración de la batería de 10 horas y 30 minutos, se hace notar que el dispositivo no arroja información acerca de la carga real de la batería y es posible que no se encontrase totalmente cargada, pero aun siendo así supera el requisito de 8 horas de duración del sistema.

6. Conclusiones

Tras la realización del proyecto de forma satisfactoria y cumpliendo los objetivos definidos para el mismo, se puede decir que ha sido una experiencia muy enriquecedora a la vez que un reto, debido no solo a la situación de confinamiento actual sino también a la integración de distintos componentes y tareas para los que no se tenía experiencia previa.

Ha sido de especial utilidad en todo el proceso de desarrollo del sistema la definición de requisitos, y la integración de métricas en cada uno de ellos, para verificar el cumplimiento de los objetivos del proyecto. Este hecho ha permitido tener un hilo conductor para la realización tanto de las pruebas de evaluación del prototipo como de la redacción de la memoria del proyecto.

Por otro lado, el desafío con mayor complejidad del sistema es el desarrollo de un servidor web desde cero puesto que eso requiere del uso estricto del protocolo HTTP y de no cumplirlo (por ejemplo, por la omisión de algún carácter) puede dejar la conexión entre el servidor y el navegador en un estado de espera eterna y la depuración y solución de estos errores requiere de un conocimiento mayor al estimado inicialmente.

Asimismo, también ha supuesto algún problema, aunque en menor medida, los errores en el cálculo y representación en pantalla LCD del rumbo a seguir cuando la posición del dispositivo y el destino se encontraban en otro cuadrante de latitud y/o longitud

De todos modos y pese a que hubo algún retraso con respecto a la planificación inicial, se ha conseguido mantener los plazos generales para la realización de todas las tareas planificadas, de hecho, se ha podido incluir nuevas funcionalidades no contempladas al inicio del proyecto:

- Actualización automática de la declinación magnética por medio de la integración de una API que conecta con los servidores del NOAA estadounidense, y que provee del valor de declinación magnética en función de una posición dada.
- Procesamiento de las medidas tomadas por el magnetómetro, que minimiza el error inducido al tomar una única medida instantánea, realizando un promedio entre varias medidas consecutivas se consigue estabilizar el dato de orientación del sistema sin que se observen aparentes saltos erráticos.
- La integración de la pantalla LCD ha resultado más fácil de lo esperado al incorporar las librerías de Adafruit que permiten realizar dibujos de formas básicas (círculo, línea, etc.). Una vez solventado el dibujo de esos elementos se pudo dedicar más esfuerzo en la representación de los ángulos de orientación y de

rumbo a destino, realizando los cálculos trigonométricos necesarios.

6.1 Futuras líneas de acción

Una vez finalizado el proyecto se abre la posibilidad de incluir nuevas líneas de acción o mejoras que pueden complementar el sistema desarrollado en este trabajo de fin de grado:

- **Creación de un soporte estanco** para el recubrimiento del dispositivo, lo que permitirá la utilización de este en exteriores independientemente de las condiciones climatológicas.
- **Abaratamiento del dispositivo.** Puesto que la mayor parte de los componentes han sido adquiridos a un revendedor que aseguraba la rapidez de la entrega, se podrían mejorar los márgenes de beneficio al minimizar el coste de adquisición de los elementos del sistema, de forma especial la unidad Arduino que está sobredimensionada para las necesidades del proyecto y que puede ser sustituida por otras de menor coste.
- **Selección automática de destino.** Se podría incluir un módulo en la interfaz para la selección de destino, que utilizando un mapa, el usuario seleccionase un punto sobre el mapa y automáticamente el sistema actualizase el valor de destino almacenado en la memoria del sistema.
- **Mejoras en la interfaz web.** Se podrían implementar cambios visuales en la interfaz web para mejorar la experiencia de usuario.

7. Bibliografía

- [1] Wikipedia, the free encyclopedia, «Geocaching,» [En línea]. Available: <https://en.wikipedia.org/wiki/Geocaching>. [Último acceso: marzo 2020].
- [2] GSA, European Global Navigation Satellite System Agency, «What is GNSS?,» [En línea]. Available: <https://www.gsa.europa.eu/european-gnss/what-gnss>. [Último acceso: Marzo 2020].
- [3] Groundspeak Inc., «Geocaching,» [En línea]. Available: <https://www.geocaching.com/help/index.php?pg=kb.chapter&id=15&pgid=566>. [Último acceso: Marzo 2020].
- [4] Garmin Ltd., «eTrex 10,» Garmin, [En línea]. Available: <https://buy.garmin.com/es-ES/ES/p/87768>. [Último acceso: abril 2020].
- [5] MiTAC Digital Technology Corporation, «eXplorist GC,» Magellan, [En línea]. Available: <https://www.magellangps.com.au/products/devices/explorist-series/explorist-gc>. [Último acceso: abril 2020].
- [6] Friends-of-Fritzing foundation, «Fritzing,» Friends-of-Fritzing foundation, [En línea]. Available: <https://fritzing.org/home/>. [Último acceso: abril 2020].
- [7] Arduino, «Arduino MKR WIFI 1010,» Arduino, [En línea]. Available: <https://store.arduino.cc/arduino-mkr-wifi-1010>. [Último acceso: marzo 2020].
- [8] AZ-Delivery, «Magnetómetro GY-271,» AZ-Delivery, [En línea]. Available: <https://www.az-delivery.de/es/products/gy-271-kompassmodul-kompass-magnet-sensor-fuer-arduino-und-raspberry-pi>. [Último acceso: marzo 2020].
- [9] AZ-Delivery, «Módulo GPS NEO 6-M,» AZ-Delivery, [En línea]. Available: https://www.az-delivery.de/es/products/neo-6m-gps-modul?_pos=2&_sid=d9e7f7acb&_ss=r. [Último acceso: marzo 2020].

- [10] J. Mehaffey, «NMEA Data,» Gpsinformation.org, [En línea]. Available: <https://www.gpsinformation.org/dale/nmea.htm>. [Último acceso: abril 2020].
- [11] AliExpress.com, «Pantalla LCD Nokia 5110,» AliExpress.com, [En línea]. Available: <https://es.aliexpress.com/i/32598718510.html>. [Último acceso: abril 2020].
- [12] Tiendatec.es, «Tiendatec.es,» Tiendatec.es, [En línea]. Available: <https://www.tiendatec.es/electronica/baterias-pilas-y-accesorios/968-bateria-litio-lipo-603564-37v-1400mah-jst-8472496015042.html>. [Último acceso: abril 2020].
- [13] Adafruit Industries, «Adafruit,» Adafruit Industries, [En línea]. Available: <https://www.adafruit.com/>. [Último acceso: abril 2020].
- [14] diagrams.net, «drawio,» diagrams.net, [En línea]. Available: <https://app.diagrams.net/>. [Último acceso: mayo 2020].
- [15] Magnetic-Declination.com, «Magnetic Declination,» Magnetic-Declination.com, [En línea]. Available: <https://www.magnetic-declination.com/>. [Último acceso: marzo 2020].
- [16] <https://www.freepng.es>, «FreePNG,» <https://www.freepng.es>, [En línea]. Available: <https://www.freepng.es>. [Último acceso: mayo 2020].
- [17] NGDC, «Magnetic Field Calculators,» National Geophysical Data Center (NGDC), [En línea]. Available: <https://www.ngdc.noaa.gov/geomag/calculators/magcalc.shtml>. [Último acceso: mayo 2020].
- [18] Axiomatic Inc., «Brújula Digital,» Axiomatic Inc., [En línea]. Available: <https://play.google.com/store/apps/details?id=com.vincentlee.compass>. [Último acceso: 2020 abril].

[19] Igismap.com, «Bearing Angle,» Igismap.com, [En línea]. Available:
<https://www.igismap.com/map-tool/bearing-angle>. [Último acceso: abril 2020].