
Sistema de monitorización de marcas RFID en estudios de dispersión de áridos en playas

José Antonio Hernández Rodríguez

Tutor: Jorge Cabrera Gámez
Departamento de Informática y Sistemas
Universidad de Las Palmas de Gran Canaria

Cotutor: Ignacio Alonso Bilbao
Departamento de Física
Universidad de Las Palmas de Gran Canaria

Las Palmas de Gran Canaria – 29 de Julio de 2013

Índice

1. Introducción	5
1.1 Estado Actual	5
1.2 Objetivos	6
1.3 Organización	8
2. Competencias	10
3. Aportaciones	23
4. Normativa y Legislación.....	24
5. Plan del Proyecto	28
5.1 Tareas.....	28
5.2 Planificación Temporal	28
6. Desarrollo	30
6.1 Análisis	32
6.1.1 Interfaz gráfica, Análisis.....	32
6.1.2 Aplicaciones Waspote, Análisis	47
6.2 Diseño	53
6.2.1 Diseño de la Interfaz Gráfica	53
6.2.2 Diseño de las aplicaciones Waspote	58
6.3 Construcción	62
6.1.1 Construcción de la Interfaz Gráfica.....	63
6.1.2 Construcción de las aplicaciones Waspote	65
7. Conclusión	71

7.1 Conclusión del trabajo realizado	71
7.2 Líneas futuras	72
8. Referencias.....	74
Apéndice: Manual de usuario del sistema	76
Preparación e instalación	76
Manual de Uso	78

Listado de Tablas y Figuras

■ Tablas

Tabla 1 – Planificación temporal del desarrollo del proyecto.....	29
Tabla 2 – Componentes del equipo móvil o de detección de marcas.....	50

■ Figuras

Fig. 1 – Actores UML.....	41
Fig. 2 – Relación asociación UML.....	41
Fig. 3 – Relación inclusión UML.....	42
Fig. 4 – Relación extensión UML.....	42
Fig. 5 – Relación generalización UML.....	42
Fig. 6 – Casos de uso principales.....	45
Fig. 7 – Desglose del CdU ‘Realizar Informes’.....	46
Fig. 8 – Desglose del CdU ‘Monitorización’.....	47
Fig. 9 – Dispositivo Wasp mote.....	48
Fig. 10 – Lector RFID.....	52
Fig. 11 – Alimentación del lector RFID.....	52
Fig. 12 – Diseño de la base de datos ‘seguimientorfid’.....	54
Fig. 13 – Estructura del sistema.....	56
Fig. 14 – Estructura APIs Wasp mote.....	59
Fig. 15 – Setup del Wasp mote Base.....	60
Fig. 16 – Loop del Wasp mote Base.....	60
Fig. 17 – Setup del Wasp mote Rover.....	61
Fig. 18 – Loop del Wasp mote Rover.....	62
Fig. 19 – Imagen de la interfaz gráfica de usuario.....	65
Fig. 20 – Ejemplo de la precisión de los datos obtenidos por el GPS del Wasp mote.....	68
Fig. 21 – Sistema en funcionamiento 1.....	69
Fig. 22 – Sistema en funcionamiento 2.....	69
Fig. 23 – Sistema en funcionamiento 3.....	70

Fig. M1 – Menú XAMPP.....	77
Fig. M2 – Diálogo de configuración de los parámetros de la ET.....	79
Fig. M3 – Pantalla principal de la IU.....	79
Fig. M4 – Establecer Ángulo α	80
Fig. M5 – Confirmación punto topográfico.....	81
Fig. M6 – Diálogo para editar registro.....	81
Fig. M7 – Diálogo de edición correcta.....	82
Fig. M8 – Entrada de texto del modo siembra.....	82
Fig. M9 – Ventana de configuración.....	83
Fig. M10 – Diálogo para guardas los informes.....	84
Fig. M11 – Diálogo de guardado correcto.....	84
Fig. M12 – Menú de búsqueda de la IU.....	85

1. Introducción

El proyecto “Sistema de monitorización de marcas RFID en estudios de dispersión de áridos en la playa” surge con la intención de apoyar un proyecto de fin de Máster del Departamento de Física de la Universidad de Las Palmas de Gran Canaria, y de fomentar la colaboración multidisciplinar.

Dicho proyecto consiste en marcar una serie de cantos y colocarlos en una playa, concretamente en el norte de Gran Canaria, con el fin realizar un seguimiento de los mismos y analizar el desplazamiento que sufren por el efecto del oleaje. Nuestro papel en el proyecto consiste en servir de soporte informático para facilitar la obtención y la representación de la información para realizar este proyecto, proporcionando así, más comodidad y sencillez a los encargados de desarrollarlo.

Las principales herramientas que se deberán utilizar durante el desarrollo del proyecto para la consecución del mismo son las siguientes:

- Dispositivos Waspnotes
- Estación total Topcon GTS – 300
- API de Google Maps

1.1 Estado Actual

En este apartado se expone el estado en el que se encuentran las distintas herramientas empleadas en el proyecto a día de hoy.

En primer lugar tenemos los dispositivos **Waspnotes** [6] [14] [15], los cuales son plataformas basadas en una placa con microcontrolador y un entorno de desarrollo, diseñada para facilitar el uso de la electrónica en proyectos multidisciplinarios.

Estos dispositivos han tenido una excelente acogida en todo el mundo, un gran número de empresas han hecho uso de estos dispositivos, tales como Telefónica, Sony Ericsson, Vodafone, IBM, etc., así como distintas universidades a nivel internacional. Esta gran aceptación va condicionada por la gran diversidad de utilidades de estos dispositivos, empleados en proyectos como:

- Monitorización de ruido y polución en ciudades.
- Medición de niveles de radiación.
- Monitorización de calidad del aire en interiores.
- Medición del nivel de radiación ultravioleta.
- Prevención de incendios forestales.
- Etc.

Pieza esencial en el proyecto es la **estación total** [7] (en este TFG se utiliza la Topcon GTS-300), aparato electro-óptico cuyo funcionamiento se apoya en la tecnología electrónica. Consiste en la incorporación de un distanciómetro y un microprocesador a un teodolito (instrumento de medición para obtener ángulos verticales y horizontales con una alta precisión) electrónico. Este tipo de aparato es usado principalmente en topografía para el cálculo de distancias horizontales, distancias inclinadas, diferencia de alturas entre dos puntos, etc.

Y por último tenemos la **API de Google Maps** [9], la más popular y conocida de las tres herramientas. Ésta es un servidor de aplicaciones de mapas en la web, ofrece imágenes de mapas desplazables, así como fotografías por satélite de todo el mundo.

Con la introducción de las herramientas de búsqueda y el movimiento en el mapa, ha hecho crecer el interés en el uso de las imágenes satelitales, tanto para la investigación como para fines personales. Además la posibilidad de integrar la API en cualquier página Web ha revolucionado el uso de la misma.

En el siguiente apartado se comenta cual es el papel de cada una de estas herramientas dentro del proyecto.

1.2 Objetivos

El objetivo principal de este trabajo de fin de grado está muy claro: desarrollar una interfaz gráfica capaz de almacenar y representar la información obtenida a partir de los distintos sensores y herramientas empleados, además de los programas necesarios para obtener dicha información mediante los dispositivos Waspnotes.

Para lograr este objetivo principal hay que tener claro cuál es el rol que tendrá cada una de las herramientas mencionadas en el apartado anterior dentro del proyecto, aparte de estas, también se debe hacer mención de la interfaz gráfica que posee un papel primordial en el proyecto.

- **Wasmotes:** la función de los Wasmotes será la detección de los distintos cantos marcados con las marcas RFID a lo largo de la playa donde estén situadas.
- **Estación Total:** la estación total se encargará de medir las distancias entre un punto de referencia y las localizaciones de las detecciones registradas por los dispositivos Wasmotes, con la finalidad de obtener las coordenadas geográficas de las mismas.
- **API Google Maps:** la API de Google Maps servirá para representar las coordenadas calculadas a partir de los datos obtenidos de la estación total sobre el mapa correspondiente.
- **Interfaz Gráfica:** la interfaz gráfica tendrá como objetivos la coordinación de las tres herramientas anteriores para la obtención y representación de los datos obtenidos en tiempo real, servir de soporte a la API de Google Maps, y permitir consultas y extracciones de informes posteriores.

Pero aparte de este objetivo general se plantea otra serie de objetivos secundarios y secuenciales, que son los que se citan a continuación:

1. Establecer la comunicación entre dos dispositivos Wasmotes y obtener la información oportuna de los sensores de los que están provistos estos dispositivos.
2. Conseguir la comunicación entre el ordenador portátil y los dispositivos Wasmote.
3. Conseguir la comunicación entre el ordenador portátil y la estación total Topcon GTS-300.
4. Conocer y elegir una de entre las distintas APIs disponibles para la representación de puntos en mapas.
5. Representar los datos obtenidos sobre la API de mapas seleccionada.

6. Desarrollar una interfaz gráfica capaz de coordinar las distintas herramientas empleadas.

1.3 Organización

Esta memoria consta de ocho capítulos y dos apéndices. El primero, el presente, es de carácter introductorio. Como ya se ha expuesto en parte, en él se explica de forma general en qué consiste el proyecto, cuál es su finalidad y qué características posee. También se destacan cuáles se destaca cuál es el estado en el que actualmente se encuentran las principales herramientas empleadas en el desarrollo de este trabajo de fin de grado. Posteriormente se mencionan los objetivos que se esperan conseguir una vez finalizado el mismo. Y por último, el presente apartado, se resume la estructura y organización general que presenta la memoria.

El segundo capítulo versa sobre cómo se han cubierto las distintas competencias asociadas al título de Grado en Ingeniería Informática con el fin de culminar los estudios y como preparación para el desempeño futuro de actividades profesionales en el ámbito correspondiente a la titulación obtenida.

En el tercero se hace referencia a las distintas aportaciones que han surgido gracias al desarrollo de este proyecto.

El cuarto capítulo se trata la normativa y legislación vigente que afecta a este trabajo de fin de grado, se mencionan las distintas licencias a las que se acogen algunas de las herramientas empleadas y los propietarios de los derechos de otras aplicaciones también usadas.

El quinto expone cuales van a ser las tareas en las que se va a dividir el desarrollo de este trabajo de fin de grado y la temporización de las mismas.

El capítulo número seis contiene la parte más importante del proyecto, aquí se cuenta todo lo relacionado con el qué y el cómo se ha realizado. Este capítulo está compuesto por tres apartados: análisis, diseño y construcción; separando así el desarrollo en las principales etapas que se establecen en la ingeniería del software.

El séptimo sirve de conclusión, donde se exponen cuales son los resultados conseguidos y en qué aspectos se puede seguir trabajando en el futuro.

Además, en esta memoria se incluyen un apéndice donde se encuentra un sencillo manual para instalar y usar el sistema.

2. Competencias

En el presente trabajo de fin de grado se nos exige el cumplimiento de una serie de competencias profesionales. En este capítulo mostraremos las competencias que debemos cumplir, y explicaremos detalladamente de qué manera hemos cubierto cada una de ellas.

- ***CIIO1: capacidad para diseñar, desarrollar, seleccionar y evaluar aplicaciones y sistemas informáticos, asegurando su fiabilidad, seguridad y calidad, conforme a principios éticos y a la legislación y normativa vigente.***

Esta competencia queda cubierta con lo planteado en los capítulos de “Análisis”, “Requisitos de Hardware y Software” y “Desarrollo”. En ellos se muestra como ha sido todo el proceso de desarrollo del proyecto con el fin de crear un producto que satisfaga la demanda, asegurando así la fiabilidad, seguridad y calidad del mismo. Además de poner de manifiesto los conocimientos adquiridos durante la realización de los estudios, aplicándolos correctamente para el buen desarrollo de este trabajo de fin de grado.

- ***CIIO2: capacidad para planificar, concebir, desplegar y dirigir trabajos, servicios y sistemas informáticos en todos los ámbitos, liderando su puesta en marcha y su mejora continua y valorando su impacto económico y social.***

Esta competencia se cumple con lo establecido en los siguientes capítulos:

- **Plan del proyecto:** en este capítulo se muestra cual es la planificación de las tareas a desarrollar, estableciendo su descripción y duración.
- **Aportaciones:** aquí se hace referencia a las aportaciones conseguidas con el desarrollo del presente trabajo de fin de grado, valorando su impacto social y económico.

- ***CI104: capacidad para elaborar el pliego de condiciones técnicas de una instalación informática que cumpla los estándares y normativas vigentes.***

Para cumplir con esta competencia se muestra a continuación el pliego de condiciones técnicas necesarias para realizar la monitorización de marcas RFID en estudios de dispersión de áridos en playas.

Objeto del contrato

El contrato tiene como objeto el desarrollo de un sistema capaz de monitorizar una serie de marcas RFID colocadas en distintos cantos a lo largo de un playa para el posterior estudio de la dispersión de los mismos.

Presupuesto

- a) **Dispositivos Waspnotes:** 2 x 250 €/unidad aprox. = 500 € aprox.
- b) **Lector RFID con sistema de alimentación y antena:** 300 € aprox.
- c) **Ordenador Portátil:** 500 € aprox.
- d) **Modem 3G USB + tarifa durante dos meses:** 60 € + 20 € / cuota x 2 meses de uso aprox. = 100 €
- e) **Coste de desarrollo:** 15 € / hora x 300 horas de trabajo = 4500 €.

El presupuesto para el presente contrato es de 5900.00 €, al que incluyendo el 7% de I.G.I.C. (tras el incremento de este porcentaje en el año 2012) da como resultado una suma de **6313.00 €**. El precio del elemento **d** (Modem 3G USB y la tarifa durante dos meses, que es la duración prevista del usuario final para la realización del estudio) varía con respecto a la compañía que se desee contratar, sin embargo, en el presupuesto se añade una cifra aproximada. Se debe hacer mención que la estación total Topcon GTS – 300 que ha sido empleada para este TFG ya se encontraba disponible para su uso, por lo que no ha sido necesario adquirirla, en caso contrario se habría puesto dentro del presupuesto.

Especificaciones técnicas

Las especificaciones técnicas que debe cumplir el sistema desarrollado son las siguientes:

- Con las herramientas necesarias, el sistema debe ser capaz de monitorizar las marcas RFID depositadas en una serie de cantos a lo largo de una playa, con el fin de estudiar su dispersión por el efecto de la marea.
- El sistema debe almacenar los datos obtenidos (coordenadas geográficas de los cantos) y representarlos en la API de Google Maps en tiempo real. Además debe permitir el postproceso de los datos, es decir, dar la posibilidad a los usuarios la realización de búsquedas e informes mientras no se esté realizando la monitorización.

Asimismo, el adjudicatario debe prepara un manual de usuario sin coste adicional alguno, con la suficiente información como para que las personas interesadas puedan utilizar la aplicación sin problemas.

Condiciones particulares

- **Plazo de ejecución:** el contratista se compromete a elaborar el sistema de conformidad con lo establecido en este documento y con la oferta que hubiese presentado, así como a entregarlo al contratante. Dicha entrega deberá hacerse en el plazo máximo de 5 meses, a contar desde la fecha de firma del contrato.

El producto solicitado se considerará debidamente entregado cuando se haya recibido el sistema al completo, esté disponible para su inmediato funcionamiento y se haya facilitado el manual y la documentación técnica correspondiente.

- **Penalizaciones:** cualquier incumplimiento del servicio técnico postventa y de mantenimiento y plazos de entrega, conllevará una penalización del 0.001% por cada día natural de incumplimiento sobre el precio del contrato. En caso de que el pago por parte del contratante estuviera realizado con anterioridad, la empresa contratada deberá abonar al contratante el importe de la penalización. La penalización máxima por este concepto no excederá del 10% del importe final del contrato.

- **Documentación:** las empresas licitadoras deberán poner claramente de manifiesto las posibles actualizaciones del sistema.
- **CII18: conocimiento de la normativa y la regulación de la informática en los ámbitos nacional, europeo e internacional.**

1. Ámbito Internacional

En el contexto internacional, son pocos los países que cuentan con una legislación apropiada. Entre ellos destacan Estados Unidos, Alemania, Austria, Gran Bretaña, Holanda, Francia, Argentina y Chile.

A continuación se mencionan algunos aspectos relacionados con la ley en diferentes países, así como los delitos informáticos que persigue.

a. Estados Unidos

Este país adoptó en 1994 el Acta Federal de Abuso Computacional que modificó el Acta de Fraude y Abuso Computacional de 1986.

Con la finalidad de eliminar los argumentos hipertécnicos acerca de qué es y que no es un virus, un gusano o un caballo de Troya, la nueva acta proscribía la transmisión de un programa, información, códigos o comandos que causan daños a la computadora, a los sistemas informáticos, a las redes, información, datos o programas. La nueva ley es un adelanto porque está directamente en contra de transmisión de virus.

Asimismo, en materia de estafas electrónicas, defraudaciones y otros actos dolosos relacionados con los dispositivos de acceso a sistemas informáticos, la legislación estadounidense sanciona con pena de prisión y multa, a la persona que defraude a otra mediante la utilización de una computadora o una red informática.

En el mes de Julio del año 2000, el Senado y la Cámara de Representantes de este país establecen el Acta de Firmas Electrónicas en el Comercio Global y Nacional. La ley sobre la firma digital responde a la necesidad de dar validez a documentos informáticos (mensajes electrónicos y contratos

establecidos mediante Internet) entre empresas (para el B2B) y entre empresas y consumidores (para el B2C).

b. Alemania

Este país sancionó en 1986 la Ley contra la Criminalidad Económica, que contempla los siguientes delitos:

- Espionaje de datos.
- Estafa informática.
- Alteración de los datos.
- Sabotaje informático.

c. Austria

La ley de reforma del Código Penal, sancionada el 22 de Diciembre de 1987, sanciona a aquellos que con dolo causen un perjuicio patrimonial a un tercero influyendo en el resultado de una elaboración de datos automática a través de la confección del programa, por la introducción, cancelación o alteración de datos por actuar sobre el curso del procesamiento de datos. Además contempla sanciones para quienes cometan este hecho utilizando su profesión de especialistas en sistemas.

d. Gran Bretaña

Debido a un caso de hacking en 1991, comenzó a regir en este país la Computer Misuse Act (Ley de Abusos Informáticos). Mediante esta ley el intento, exitoso o no, de alterar datos informáticos es penado con hasta cinco años de prisión o multas. Esta ley tiene un apartado que especifica la modificación de datos sin autorización.

e. Holanda

El 1 de Marzo de 1993 entró en vigencia la Ley de Delitos Informáticos, en la cual se penalizan los siguientes delitos:

- El hacking

- El preacking (utilización de servicios de telecomunicaciones evitando el pago total o parcial de dicho servicio).
- La ingeniería social (arte de convencer a la gente de entregar información que en circunstancias normales no entregaría).
- La distribución de virus.

f. Francia

En Enero de 1988, este país dictó la Ley relativa al fraude informático, en la que se consideran aspectos como:

- Intromisión fraudulenta que suprima o modifique datos.
- Conducta intencional en la violación de derechos a terceros que haya impedido o alterado el funcionamiento de un sistema de procesamiento automatizado de datos.
- Conducta intencional en la violación de derechos a terceros, en forma directa o indirecta, en la introducción de datos en un sistema de procesamiento automatizado o la supresión o modificación de los datos que éste contiene, o sus modos de procesamiento o de transmisión.
- Supresión o modificación de datos contenidos en el sistema, o bien en la alteración del funcionamiento del sistema (sabotaje).

g. Chile

Chile fue el primer país latinoamericano en sancionar una Ley contra delitos informáticos, la cual entró en vigencia el 7 de junio de 1993. Esta ley se refiere a los siguientes delitos:

- La destrucción o inutilización de los datos contenidos dentro de una computadora es castigada con penas de prisión. Asimismo, dentro de esas consideraciones se encuentran los virus.
- Conducta maliciosa tendiente a la destrucción o inutilización de un sistema de tratamiento de información o de sus partes componentes, o que dicha conducta impida, obstaculice o modifique su funcionamiento.

- Conducta maliciosa que altere, dañe o destruya los datos contenidos en un sistema de tratamiento de información.

2. **Ámbito europeo**

Hasta ahora, el principal esfuerzo europeo por regular el tema de los delitos informáticos dio como resultado el “Convenio sobre la Ciberdelincuencia”, del 21 de noviembre de 2011. Este documento fue firmado por los representantes de cada país miembro del Consejo de Europa, aunque su eficacia depende de su posterior refrendo por los órganos nacionales de cada país firmante.

El “Convenio sobre la Ciberdelincuencia” permitió la definición de los delitos informáticos y algunos elementos relacionados con éstos, tales como “sistemas informáticos”, “datos informáticos”, o “proveedor de servicios”. Estos delitos informáticos fueron clasificados en cuatro grupos:

- a. Delitos contra la confidencialidad, la integridad y la disponibilidad de los datos y sistemas informáticos.
 - Acceso ilícito a sistemas informáticos.
 - Interceptación ilícita de datos informáticos.
 - Interferencia en el sistema mediante la introducción, transmisión, provocación de daños, borrado, alteración o supresión de éstos.
 - Abuso de dispositivos que faciliten la comisión con delitos.
- b. Delitos informáticos.
 - Falsificación informática que produzca alteración, borrado o supresión de datos informáticos que ocasionen datos no auténticos.
 - Fraudes informáticos.
- c. Delitos relacionados con el contenido.
 - Delitos relacionados con la pornografía infantil.

d. Delitos relacionados con infracciones de la propiedad intelectual y derechos afines.

Conviene destacar que en el “Convenio sobre la Ciberdelincuencia” se encomienda a cada parte que tome las medidas necesarias para tipificar como delito en su derecho interno de cada uno de los apartados descritos en cada categoría.

En la Disposición 1441 del BOE num. 226 de 2010, encontramos el Instrumento de Ratificación del “Convenio sobre la Ciberdelincuencia”, hecho en Budapest el 23 de Noviembre de 2001

3. Ámbito nacional

Ley Orgánica de Protección de datos de carácter personal: régimen sancionador aplicable (BOE nº 298 de 14/XII/99 que publicó la Ley Org. 15/1999 de Dic.).

Objeto: proteger y garantizar las libertades públicas y derechos fundamentales de las personas, especialmente su HONOR e INTIMIDAD personal y familiar.

Aspectos de interés: serán responsables “los responsables de los ficheros o de los tratamientos” y “los encargados de los tratamientos”.

Tipo de infracciones:

- *Leves* (art.44.2): multas de 100.000 a 10 millones de las antiguas pesetas. Ej: rectificar datos o no comunicarlos a la Agencia de Protección Datos.
- *Graves* (art.43): multas de 10 a 50 millones de las antiguas pesetas. Ej: no mantener sistemas de seguridad, obstrucción a inspección, uso en provecho propio.
- *Muy Graves* (art.45): multas de 50 a 100 millones de las antiguas pesetas (“Conductas reprochables”). Ej: vulnerar a propósito el secretismo, etc.

Ley 7/1998 de 13 de Abril que regula las condiciones generales de contratación.

R.D. 1906/1999 de 17/XII que regula la contratación telefónica.

R.D. Ley 14/1999 de 17/XII sobre Firma Electrónica (BOE Nº 224 de 18/XII).

Firma electrónica: dispositivo electrónico que permite la identificación del signatario de las operaciones realizadas por Internet.

Identifica: el firmante (autenticación) y evita el retracto (no repudio).

Código Penal

Ley Orgánica 10/1995 de 23/XI: Tipifica delitos y faltas por el uso de la informática, concretamente contra la Intimididad, Patrimonio, Socioeconómicos y Propiedad Intelectual.

Título X: “Delitos contra la intimidad, derecho a la propia imagen y la inviolabilidad del domicilio”.

- Apoderarse de papeles, e-mails, mensajes, otros.
- Cracks: delitos.
- Obtener datos de terceros.

Recomendaciones de la APD

a. Información en la recogida de datos

- Cuando suministre datos personales a cualquier organización (proveedores de acceso, proveedores de contenido, vendedores a través de comercio electrónico, etc.) sea consciente de a quién se los facilita y con qué finalidad.
- Procure averiguar la política de sus proveedores y administradores de listas y directorios en lo que se refiere a venta, intercambio o alquiler de los datos que les suministra. Solicite que sus datos personales no vayan unidos a su identificación de acceso a Internet.

b. Finalidad para la que se recogen los datos

- Desconfíe si los datos que le solicitan son excesivos para la finalidad con la que se recogen o innecesarios para el servicio que se le presta.
- Tenga en cuenta que cuando introduce su dirección de correo electrónico en un directorio, lista de distribución o grupo de noticias, dicha dirección puede ser recogida por terceros para ser utilizada con una finalidad diferente, como por ejemplo remitirle publicidad no deseada.
- Cuando navegue por Internet, sea consciente de que los servidores Web que visita pueden registrar tanto las páginas a las que accede como la frecuencia y los temas o materias por las que busca, aunque no le informen de ello. Asimismo, su pertenencia a determinados grupos de noticias y listas de distribución puede contribuir a la elaboración de perfiles más o menos detallados sobre su persona.

c. Seguridad en el intercambio de los datos

- Utilice, siempre que sea posible, las últimas versiones de los programas navegadores, ya que cada vez suelen incorporar mejores medidas de seguridad. Considere la posibilidad de activar en dichos programas las opciones que alerten sobre los intercambios de datos no deseados y no rellene aquellos datos que no desee hacer públicos (por ejemplo, dirección de correo electrónico, nombre, apellidos, etc.).
- No realice transacciones comerciales electrónicas a través de proveedores con sistemas inseguros o no fiables. Consulte el manual de su navegador para averiguar cómo informa de que se ha establecido una conexión con un servidor seguro.
- Recuerde que existen sistemas de dinero electrónico que preservan el anonimato de sus compras en Internet.

- Utilice los mecanismos de seguridad que tenga a su alcance para proteger sus datos de accesos no deseados. El medio más fiable para conseguirlo es el cifrado de los mismos.
- Salvo que se utilicen mecanismos de integridad, autenticación y certificación (firma digital, notarios electrónicos, etc.) no confíe ciegamente en que la persona u organización que le remite un mensaje es quien dice ser y en que el contenido del mismo no se ha modificado, aunque esto sea así en la inmensa mayoría de las ocasiones.
- Siempre que se le soliciten datos personales que no está obligado legalmente a suministrar, sopesese los beneficios que va a recibir de la organización que los recoge frente a los posibles riesgos de utilización irregular de los mismos.
- Ante cualquier duda sobre la legalidad de la utilización de sus datos de carácter personal, póngase en contacto con la Agencia de Protección de Datos.

La protección jurídica de programas de ordenador. Piratería informática

- El Real Decreto Legislativo 1/1996, por el que se aprueba el Texto Refundido sobre Propiedad Intelectual, la protección jurídica de los programas de ordenador, antes regulada por la Ley de Protección Jurídica de Programas de Ordenador y por la Ley de Propiedad Intelectual, crea un marco jurídico en contra de la piratería informática.
- El Texto Refundido desarrolla una serie de medidas para combatir la piratería informática, como la posibilidad de que los fabricantes de programas de ordenador soliciten a la justicia española la realización de un registro sorpresa en empresas en las que existan sospechas fundadas o evidencias de delito. España es uno de los países en los que se puede acudir a esta medida cautelar. De esta manera se erradica la posibilidad de que los presuntos infractores puedan destruir las pruebas existentes, lo cual, indudablemente ocurrirá si se les notifica por adelantado la realización de un registro.

¿En qué casos se infringe la Ley?

- Al copiar distribuir un programa de ordenador o la documentación que se le acompaña, incluidas aplicaciones, datos, códigos y manuales, sin permiso expreso o licencia del propietario o licencia del propietario de los derechos de explotación.
- Al utilizar un programa sin la correspondiente licencia o autorización del fabricante, con independencia de que se utilice en un solo ordenador o en varios de forma simultánea.
- Al utilizar programas de ordenador en un número de copias superior al autorizado por el fabricante en sus contratos o licencias de uso.
- En empresas y demás organizaciones, al fomentar, consciente o inconscientemente, permitir, obligar o presionar a los empleados a realizar o distribuir copias no autorizadas del programa.
- Al efectuar copias no autorizadas porque alguien lo requiere u obliga a ello. Al ceder o prestar el programa de forma que pueda ser copiado o al copiarlo mientras está en su posesión en calidad de cedido o prestado.
- Al crear, importar, poseer o negociar con artículos destinados a burlar o neutralizar cualquier medio técnico aplicado para proteger el programa de ordenador.

Medidas Judiciales

Si finalmente existe evidencia de delito, las medidas judiciales que pueden adoptarse son:

- Solicitar al Juez un registro sorpresa de las instalaciones del presunto infractor, tanto por la vía civil como por la penal.
- Solicitar al Juez la adopción urgente de medidas cautelares de protección.
- Exigir indemnizaciones acordes con los daños materiales y morales causados.
- El cierre del centro de actividad del infractor.

- El secuestro de todos aquellos medios destinados a suprimir los dispositivos técnicos que protegen un programa desarrollado y comercializado por un fabricante de programas.
- ***TFG01: Ejercicio original a realizar individualmente y presentar y defender ante un tribunal universitario, consistente en un proyecto en el ámbito de la Ingeniería Informática de naturaleza profesional en el que sinteticen e integren las competencias adquiridas en las enseñanzas.***

Esta competencia quedará cubierta una vez realizada la presentación de este TFG antes los correspondientes miembros del tribunal.

3. Aportaciones

La realización de este trabajo de fin de grado deja al descubierto distintas aportaciones, que son las siguientes:

- Capacidad de obtener, monitorizar y almacenar las posiciones geográficas en tiempo real de marcas RFID, siendo capaz de representar dichas coordenadas en la API de Google Maps, y posibilitando también la consulta de las mismas en un proceso posterior.
- Carácter interdisciplinar del proyecto, donde los elementos de tecnología informática que se aportan permiten impulsar tecnológicamente la faceta científica (geológica, en este caso).
- Dar a conocer a otras disciplinas la utilidad que pueden aprovechar de utilizar la informática para el desarrollo de sus propios proyectos.
- La última y no menos importante, es que se fomente el trabajo multidisciplinar tanto dentro de la ULPGC como fuera de ella, que tan buenos resultados ha dado durante la realización de este proyecto.

4. Normativa y Legislación

En este cuarto capítulo se trata la legislación vigente que afecta a este trabajo de fin de grado, se mencionan las distintas licencias a las que se acogen algunas de las herramientas empleadas y los propietarios de los derechos de otras aplicaciones también usadas.

Antes de entrar en estos aspectos, hay que hacer mención del **software libre**, que tienes un gran protagonismo en este trabajo de fin de grado. El software libre es la denominación del software que respeta la libertad de los usuarios sobre su producto adquirido y , por tanto, una vez obtenido puede ser usado, copiado, estudiado, modificado y redistribuido libremente. Según la Free Software Foundation, el software libre se refiere a la libertad de los usuarios para ejecutar, distribuir, estudiar, modificar el software y distribuirlo modificado.

EL software libre suele estar disponible gratuitamente, o al precio de costo de la distribución a través de otros medios; sin embargo no es obligatorio que sea así, por lo que no hay que asociar software libre a “software gratuito” (denominado usualmente freeware) ya que, conservando su carácter de libre, puede ser distribuido comercialmente (“software comercial”). Análogamente, el software gratuito incluye en ocasiones el código fuente; no obstante, este tipo de software no es libre en el mismo sentido que el software libre, a menos que se garanticen los derechos de modificación y redistribución de dichas versiones modificadas del programa.

Tampoco debe confundirse software libre con “software de dominio público”. Este último es aquel software que no requiere de licencia, pues sus derechos son de explotación para toda la humanidad, porque pertenece a todos por igual. Cualquiera puede hacer uso de él, siempre con fines legales y consignando su autoría general. Este software sería aquel cuyo autor lo dona a la humanidad o cuyos derechos de autor han expirado, tras un plazo contado desde la muerte éste, habitualmente 70 años. Si un autor condicione su uso bajo una licencia, por muy débil que sea, ya no es de dominio público.

GNU General Public Licence (GPL)

Los dispositivos *Waspnotes*, el servidor multiplataforma *XAMPP* y el entorno de desarrollo *Netbeans* ejecutan código liberado bajo esta licencia.

La Licencia Pública General de GNU, o más conocida por su nombre en inglés GNU General Public License, es una licencia creada por la Free Software Foundation en 1989 (su primera versión), y está orientada principalmente a proteger la libre distribución, modificación y uso de software. Su propósito es declarar que el software cubierto por esta licencia es software libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios.

Existen varias licencias “hermanas” de la GPL, como la licencia de documentación libre de GNU (GFDL), la Open Audio License, para trabajos musicales, etc., y otras menos restrictivas, como la LGPL, o la LGPL (Lesser General Public License, antes Library General Public License), que permiten el enlace dinámico de aplicaciones libres a aplicaciones no libres.

La licencia GPL, al ser un documento que cede ciertos derechos al usuario, asume la forma de un contrato, por lo que usualmente se le denomina contrato de licencia o acuerdo de licencia.

Common Development and Distribution License (CDDL)

El entorno de programación NetBeans además de estar bajo la licencia GPL, también se encuentra condicionada por esta otra.

Common Development and Distribution License (Licencia Común de Desarrollo y Distribución) o CDDL es una licencia de código abierto (OSI) y libre, producida por Sun Microsystems, basada en la Mozilla Public License o MPL, versión 1.1. La licencia CDDL fue enviada para su aprobación al Open Source Initiative el 1 de diciembre de 2004, y fue aprobada como una licencia de código abierto a mediados de enero de 2005. EN el primer borrador hecho por el comité de divulgación de licencias OSI, la CDDL es una de las nueve licencias más populares, mundialmente usadas o con fuertes comunidades. Además la Free

Software Foundation afirma que se trata de una licencia libre pero es incompatible con GNU GPL debido principalmente a algunos detalles.

Google

La API de Google Maps es el servidor de aplicaciones de mapas en la web propiedad de Google Inc.

En este trabajo de fin de grado, el usuario final del sistema desarrollado accederá al mismo de forma pública y gratuita, por lo tanto, los servicios que proporciona la API de Google Maps, en este caso, son gratuitos.

Google Inc. es la empresa propietaria de la marca Google, cuyo principal producto es el motor de búsqueda de contenido en Internet del mismo nombre, creado originalmente por Larry Page y Sergey Brin.

Aunque su principal producto sea éste, la empresa ofrece también otros servicios: un servicio de correo electrónico (Gmail, su mapamundi en 3D Google Earth), un servicio de mensajería instantánea basado en XMPP llamado Google Talk, el sitio web de vídeos YouTube, Google Libros o Google Noticias, el navegador web Google Chrome, el servicio de red social Google+, al igual lidera el desarrollo del sistema operativo basado en Linux, Android, usado actualmente en teléfonos móviles y tablets, entre otros muchos.

Microsoft

Para la realización de este trabajo también se ha hecho uso de la aplicación Microsoft Office Word y Microsoft Office Power Point para la confección de la presente memoria y la presentación de la misma respectivamente. Además estas dos junto al desarrollo de la interfaz gráfica y los programas de los dispositivos Waspote se han hecho sobre el sistema operativo Windows, en concreto, sobre el XP y el 7. Todo este software es propiedad de Microsoft Corporation teniendo las licencias correspondientes para el uso de las mismas.

Microsoft Corporation es una empresa multinacional de origen estadounidense, fundada el 4 de abril de 1975 por Bill Gates y Paul Allen. Dedicada al sector del software, tiene su sede en Redmond, Washington, Estados Unidos. Microsoft desarrolla, fabrica, licencia y produce software y

equipos electrónicos, siendo sus productos más usados el sistema operativo Microsoft Windows y la suite Microsoft Office, los cuales tienen una importante posición entre los ordenadores personales.

Durante su historia, ha sido objeto de críticas, como acusaciones de realizar prácticas monopolísticas que la han llevado ante la Comisión Europea y el Departamento de Justicia de los Estados Unidos. En concreto, esta controversia en la industria de la informática surge en 1980, debido principalmente a las tácticas de negocios empleadas por esta empresa, pues inicialmente adopta un estándar o producto, para después producir versiones de los mismos incompatibles con el estándar, que termina extinguiéndose e impide a la competencia utilizar la nueva versión del producto. Estas tácticas han dado lugar a que diversas empresas y gobiernos interpongan demandas contra la empresa.

Los partidarios del software libre se muestran preocupados por la formación de la Trusted Computing Platform Alliance (TCPA), formada por un grupo de empresas entre las que destacan Microsoft e Intel que ha impulsado, alegando que de esta forma se producirá un aumento de la seguridad y privacidad en el ordenador de los usuarios, una estrategia denominada Trusted Computing. La TCPA ha sido vista por sus detractores como un medio para permitir a los desarrolladores de software para imponer algún tipo de restricción en su software.

Además los detractores de Microsoft discrepan con la promoción de la Gestión de derechos digitales (DRM) y el coste total de propiedad (CTP). La gestión de los derechos digitales es una tecnología que permite a los proveedores de contenido a imponer restricciones a los métodos y a la forma en que sus productos son usados. Los detractores del DRM sostienen que esta tecnología puede infringir el uso legítimo y de otros derechos, dado que restringe actividades legales o la reproducción de material para su uso en representaciones de diapositivas.

5. Plan del Proyecto

En este capítulo se muestran cuáles son las tareas realizadas y cuál es la planificación temporal que se ha seguido para realizarlas.

5.1 Tareas

El Trabajo de Fin de Grado “Sistema de monitorización de marcas RFID en estudios de dispersión de áridos en playas” tiene una división de tareas como la que se muestra a continuación:

- **Tarea 1:** Establecer la comunicación entre dos dispositivos Waspnotes y obtener la información oportuna de los sensores de los que están provistos estos dispositivos.
- **Tarea 2:** Conseguir la comunicación entre el ordenador portátil y los dispositivos Waspnote. Realización de las pruebas oportunas.
- **Tarea 3:** Desarrollo de la interfaz de control de la estación total a través de su interfaz RS232. Realización de las pruebas oportunas.
- **Tarea 4:** Representar datos obtenidos sobre la API de Google Maps.
- **Tarea 5:** Desarrollar una interfaz gráfica capaz de coordinar las distintas herramientas empleadas.
- **Tarea 6:** Prueba conjunta de las herramientas y arreglo de errores.

5.2 Planificación Temporal

Para llevar a cabo las tareas indicadas en el apartado anterior, la planificación que se ha previsto para cubrir las 300 horas previstas es la que se puede ver a continuación. Consta de una tabla (Tabla 1), que temporiza las distintas tareas en horas dedicadas, las cuáles se desarrollan prácticamente de manera secuencial.

Tareas	Duración (horas)
Tarea 1	40 horas
Tarea 2	40 horas
Tarea 3	20 horas
Tarea 4	15 horas
Tarea 5	135 horas
Tarea 6	50 horas

Tabla 1 – Planificación temporal del desarrollo del proyecto

6. Desarrollo

La orientación de este capítulo es la relativa al desarrollo de cualquier software, tal y como define la **Ingeniería del Software**. Hay que mencionar previamente que aquí se añaden los apartados de “Análisis” y “Requisitos de hardware y software”, de obligatoria presencia en la memoria, por motivos de estructuración, claridad y condicionamiento del desarrollo del mismo.

El sistema desarrollado en este Trabajo Fin de Grado sigue un **ciclo de vida clásico o en cascada** ([3] y [4]), enfoque metodológico que ordena rigurosamente las etapas del proceso para el desarrollo del software de forma que el inicio de cada etapa debe esperar a la finalización de la etapa anterior. Este modelo consta normalmente de las siguientes etapas principales de creación:

1. **Análisis de requisitos**
2. **Diseño**
3. **Codificación**
4. **Prueba**
5. **Mantenimiento**

En este capítulo se contemplan principalmente las cuatro primeras etapas. El mantenimiento se quedará al margen, aunque si surgiera cualquier contratiempo con el sistema después de las cuatro primeras se trataría de solventarlo lo antes posible. Las cuatro etapas que se tratan aquí se dividen en tres bloques: “Análisis” y “Diseño”, que cubren la primera y segunda etapas respectivamente; y el último bloque, “Construcción”, que abarca las etapas tres y cuatro.

A pesar de haber empleado el ciclo de vida clásico o en cascada, durante el desarrollo del sistema se ha tenido que realizar dos iteraciones de este método, debido a que originariamente el sistema estaba contemplado para que sólo se utilizase los dispositivos Waspnote para la obtención de las coordenadas geográficas. Sin embargo, la orografía del lugar donde se iba a realizar el estudio, exactamente en la playa de San Felipe bajo el Risco de Silva,

impide la utilización efectiva de los receptores GPS con el grado de precisión requerido en este proyecto. Así que por petición de los usuarios finales después de haberle comunicado la falta de precisión en los datos que proporcionaban los Wasmotes, se decantó por el uso de una estación total, en concreto, la Topcon GTS-300.

Este contratiempo provocó la modificación del rumbo del desarrollo del sistema, por lo que prácticamente se hizo una nueva iteración incremental parcial, aplicando también el ciclo de vida clásico o en cascada, pues aparte de añadir nuevas características a la funcionalidad del sistema se tuvo que suprimir otras cuantas.

Sin embargo, en este capítulo se muestra el desarrollo del sistema en una sola iteración, como si desde el principio se hubiera especificado como requisito el uso de la estación total. Simplificando y dando más claridad a la lectura de la presente memoria.

El sistema desarrollado en este Trabajo de Fin de Grado se puede separar en dos partes claramente diferenciables:

- **Aplicaciones Wasmote:** esta parte del sistema es la encargada de detectar las marcas RFID alojadas en los cantos a lo largo de la playa.
- **Interfaz gráfica de usuario:** la interfaz será la encargada de sincronizar todas las herramientas empleadas en la obtención de las coordenadas geográficas de los cantos, siendo capaz de manipular de distintas formas estos datos. Uno de los aspectos más importantes de la IU es su labor de control sobre la estación total y la manipulación de los datos recibidos por la misma.

A continuación se abordan las distintas etapas del desarrollo del sistema en los tres bloques citados anteriormente (Análisis, Diseño y Construcción), dividiéndose cada uno en dos, uno para la parte del sistema encargada de la aplicación de los dispositivos Wasmotes, y otro para la interfaz de usuario.

6.1 *Análisis*

El primer bloque hace referencia a la **Ingeniería de Requisitos**. La Ingeniería de Requisitos comprende todas las tareas relacionadas con la determinación de las necesidades o de las condiciones a satisfacer por un software nuevo o modificado. Además, el propósito de ésta es hacer que los requerimientos alcance un estado óptimo antes de llegar a la fase de diseño en el proyecto. Los buenos requisitos deben ser medibles, comprobables, sin ambigüedades o contradicciones, etc. Estamos aquí ante una etapa crucial en el desarrollo de la aplicación, pues su ejecución tiene repercusiones directas en el posible logro de los objetivos finales.

Es muy importante tener claro desde el principio qué es lo que se va a desarrollar; cualquier ambigüedad puede ser desastrosa si no se soluciona antes de cambiar de etapa. Así que este va a ser el primer paso a seguir; una vez queda claro se pasa a la especificación de los requisitos del software.

6.1.1 *Interfaz gráfica, Análisis*

Como ya se ha visto, este proyecto se divide en dos partes principales, en el caso de la interfaz gráfica, su misión consiste en la de sincronizar todos los dispositivos participantes en el sistema de monitorización en tiempo real, es decir, coordinar los dispositivos Waspmote, la estación total y la API de Google Maps (visible desde la propia interfaz) de manera de que el usuario pueda ver el progreso de las detecciones en tiempo real. Además de esta funcionalidad, el usuario puede realizar búsqueda e informes en el postproceso de los datos obtenidos durante la monitorización.

Una vez se tiene claro cuál es la idea general de lo que se dispone a desarrollar, se entra ya en la descripción y especificación de los requisitos. En ella se realiza una descripción completa del comportamiento del sistema. Para elaborar la especificación debemos diferencia entre **requisitos funcionales** y **requisitos no funcionales**.

- **Requisitos No Funcionales**

Un **requisito no funcional** es aquel que especifica criterios que pueden usarse para juzgar la operación de un sistema en lugar de sus comportamientos específicos, ya que éstos corresponden a los requisitos funcionales. Por tanto, se refieren a todos los requisitos que ni describen información a guardar ni funciones a realizar. Este tipo de requisitos afecta a los servicios o funciones del sistema, tales como restricciones de tiempo, sobre el proceso de desarrollo, estándares, etc.

Los requisitos no funcionales de este Trabajo de Fin de Grado toman un papel bastante importante en el desarrollo de este proyecto y debido a que la lista es bastante amplia, se clasificarán en cuatro grupos (la información expuesta en este apartado ha sido extraída en su mayoría de [5]):

- **Hardware:** el hardware necesario para el desarrollo de este proyecto es el siguiente:
 - ❖ **Ordenador Portátil:** en él se desarrollará la interfaz gráfica y las aplicaciones Waspnote, además será el alojamiento de la propia interfaz gráfica donde se llevará a cabo la correspondiente monitorización una vez preparado el sistema. Hay que destacar que para la realización del mismo se han hecho uso de dos ordenadores portátiles distintos, el primero del ordenador personal del autor de este trabajo y otro proporcionado para uso dedicado a este fin.
 - ❖ **Dispositivos Waspnotes:** como ya se ha visto es necesario el uso de dos dispositivos Waspnotes para la detección de las marcas RFID. Uno de ellos dispondrá de sistema para la detección de las marcas RFID.
 - ❖ **Estación Total Topcon GTS-300:** dispositivo necesario para calcular las coordenadas geográficas de las detecciones de las marcas RFID.
 - ❖ **Módem 3G USB:** también es necesario un pincho para realizar la conexión a internet durante la monitorización, ya que es necesario de

usar la API de Google Maps, por lo que el acceso a Internet es fundamental.

- **Software:** el software que se ha utilizado para el desarrollo y el funcionamiento del sistema es el que aparece a continuación:

- ❖ **Windows:** la realización del sistema se ha hecho en el sistema operativo Windows, tanto en Windows XP como en Windows 7.

- ❖ **Netbeans:** entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java. Existe además un número importante de módulos para extenderlo, además es un producto libre y gratuito sin restricciones de uso.

Netbeans ha sido usado para desarrollar la interfaz gráfica en Java.

- ❖ **Notepad++:** editor de texto y de código fuente libre con soporte para varios lenguajes de programación. De soporte nativo a Microsoft Windows. Se distribuye bajo los términos de la GNU General Public License (GPL).

Notepad ha sido empleado para confeccionar lo relacionado con las distintas páginas HTML, además de realizar el script para la ejecución de la interfaz gráfica.

- ❖ **Microsoft Word:** software destinado al procesamiento de textos. Fue creado por la empresa Microsoft, y actualmente viene integrado en la suite ofimática Microsoft Office. Originalmente fue desarrollado por Richard Brodie para el computador IBM bajo sistema operativo DOS en 1983. Ha llegado a ser el procesador de texto más popular del mundo.

Microsoft Word ha sido utilizada para la redacción de la presente memoria.

- ❖ **Microsoft PowerPoint:** programa de presentación desarrollado por la empresa Microsoft para sistemas operativos Microsoft Windows y Mac OS, ampliamente usado en distintos campos como la enseñanza,

negocios, etc. Con PowerPoint y los dispositivos de impresión adecuados se puede realizar muchos tipos de resultado relacionados con las presentaciones: transparencias, documentos impresos para los asistentes a la presentación, notas y esquemas para el presentador, o diapositivas estándar de 35mm.

Microsoft PowerPoint ha sido necesaria para la elaboración de la presentación de este trabajo de fin de grado.

- ❖ **XAMPP:** servidor independiente de plataforma, software libre, que consiste principalmente en la base de datos MySQL, el servidor web Apache y los intérpretes de lenguajes de script: PHP y Perl. EL nombre proviene del acrónimo de **X** (para cualquiera de los diferentes sistemas operativos), **A**pache, **M**ySQL, **P**HP, **P**erl. El programa está liberado bajo la licencia GNU y actúa como un servidor web libre, fácil de usar y capaz de interpretar páginas dinámicas.

XAMPP es empleado como servidor de base de datos (MySQL) y como servidor web interpretando el lenguaje de script PHP. Específicamente, MySQL es usado tanto por la interfaz como por la API de Google Maps, y el servidor web es necesario el correcto funcionamiento de la API de Google Maps.

- ❖ **API Google Maps:** Google ofrece una API para su famosa aplicación Maps, que nos permite incorporar un mapa a nuestro dominio web. EL API proporciona diversas utilidades para manipular mapas y añadir contenido al mapa mediante diversos servicios, permitiendo crear sólidas aplicaciones de mapa en los sitios web.

En nuestro caso la API de Google Maps se utiliza en la interfaz gráfica para mostrar las coordenadas geográficas de las detecciones de las marcas RFID. Para poder mostrar la API de Google Maps se embebe un navegador dentro de la propia interfaz gráfica, que en definitiva es una aplicación de escritorio.

- ❖ **AJAX:** técnica de desarrollo web para crear aplicaciones interactivas. Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador

de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, la interactividad, velocidad y usabilidad en las aplicaciones. JavaScript es el lenguaje interpretado en el que normalmente se efectúan las funciones de llamada de AJAX mientras que el acceso a los datos se realiza mediante XMLHttpRequest, objeto disponible en los navegadores actuales. En cualquier caso, no es necesario que el contenido asíncrono esté formateado en XML.

AJAX es utilizado para actualizar las detecciones de las marcas RFID en tiempo real la API de Google Maps durante la monitorización.

▪ **Librerías**

- ❖ **RXTXcomm:** esta librería es necesaria para establecer la conexión con los dispositivos Waspote y la estación total vía USB (puerto serie).
- ❖ **poi-3.9-20121203:** las librerías POI son de un proyecto dirigido por Apache Software Foundation, que surgen previamente de un subproyecto de Jakarta Project. Estas librerías proporcionan funcionalidades que permiten leer y escribir fichero en formatos de Microsoft Office como Word, PowerPoint o Excel.

Para poder realizar los informes de las coordenadas geográficas de las detecciones de las marcas RFID en Excel se ha hecho uso de esta librería.

- ❖ **jdom-2.0.4:** JDOM es una biblioteca de código abierto para manipulaciones de datos XML optimizados para Java. Al ser creado exclusivamente para este lenguaje, se beneficia de las características del mismo, incluyendo sobrecarga de métodos, colecciones, etc.

Ha sido utilizada esta librería para el manejo de ficheros XML.

- ❖ **jQuery y jQuery UI:** **jQuery** es una biblioteca de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM,

manejar eventos, desarrollar animaciones FLV y agregar interacción con la técnica de AJAX a páginas web. jQuery es software libre y de código abierto, posee un doble licenciamiento bajo la Licencia MIT y la GNU GPL, permitiendo su uso en proyectos libres y privativos. jQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funcionalidades propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio.

jQuery UI es una biblioteca de componentes para el framework jQuery que le añaden un conjunto de plug-ins, widgets y efectos visuales para la creación de aplicaciones web. Cada componente o módulo se desarrolla de acuerdo a la filosofía jQuery (find something, manipulate it).

Se ha hecho uso de estas herramientas para hacer uso de AJAX y dar estilo a las páginas HTML que contienen la API de Google Maps.

▪ Lenguajes

- ❖ **Java:** el lenguaje de programación Java fue originalmente desarrollada por James Gosling de Sun Microsystems y publicado en 1995 como un complemento fundamental de la plataforma Java de la misma compañía. Su sintaxis deriva mucho de C y C++, pero tiene menos facilidades de bajo nivel que cualquiera de ellos. Las aplicaciones de Java son generalmente compiladas a bytecode que pueden ejecutarse en cualquier máquina virtual Java sin importar la arquitectura de la computadora subyacente. Java es un lenguaje de programación de propósito general, concurrente, orientado a objetos y basado en clases que fue diseñado específicamente para tener tan pocas dependencias como fuera posible. Su intención es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo, lo que quiere decir que el código que es ejecutado en una plataforma no tiene que ser recompilado para ejecutarse en otra.

Java ha sido el lenguaje elegido para realizar la interfaz gráfica.

- ❖ **XML:** lenguaje de marcas desarrollado por el World Wide Web Consortium (W3C). A pesar de ello, XML no ha nacido sólo para su aplicación para Internet, sino que se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas. Se puede usar en bases de datos, editores de texto, hojas de cálculo y casi cualquier cosa imaginable. XML es una tecnología sencilla que tiene a su alrededor otras que la complementan y la hacen muchos más grande y con unas posibilidades mucho mayores. Tienen un papel muy importante en la actualidad ya que permiten la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil.

Se dispondrá de un fichero XML donde se almacenan distintos parámetro de configuración de la interfaz gráfica.

Todos los lenguajes que se citan a continuación son necesarios para el control de la API de Google Maps que se encuentra embebida en la interfaz gráfica.

- ❖ **PHP:** lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas, y es usado principalmente para la interpretación el lado del servidor. Se encuentra instalado en más de 20 millones de sitios web y en un millón de servidores, y es también el módulo Apache más popular entre los equipos informáticos que usan Apache como servidor web.

PHP se usará para acceder a la base de datos para mostrar las coordenadas geográficas que se encuentran guardados en ella.

- ❖ **HTML:** son las siglas de HyperText Markup Language, hace referencia al lenguaje de marcado predominante para la elaboración de páginas web que se utiliza para describir y traducir la estructura y la información en forma de texto, así como para complementar el texto con objetos tales como imágenes. EL HTML se escribe en forma de etiquetas, rodeadas por corchetes angulares (<, >). HTML también

puede describir, hasta un cierto punto, la apariencia de un documento, y puede incluir un script (por ejemplo, JavaScript), el cual puede afectar el comportamiento de navegadores web y otros procesadores de HTML.

HTML se empleará para estructurar la página donde se mostrará la API de Google Maps.

- ❖ **CSS:** las hojas de estilo en cascada (CSS) hacen referencia a un lenguaje de hojas de estilo usado para describir la presentación semántica (el aspecto y formato) de un documento escrito en lenguaje de marcas. Su aplicación más común es dar estilo a páginas web escritas en lenguaje HTML, pero también puede ser aplicado a cualquier tipo de documentos XML, incluyendo SVG y XUL. La información de estilo puede ser adjuntada como un documento separado o en el mismo documento HTML. EN este último caso podrían definirse estilos generales en la cabecera del documento o en cada etiqueta particular mediante el atributo “<style>”.

CSS se usa para dar estilo a los componentes que forman las páginas HTML donde se encuentra la API de Google Maps.

- ❖ **JavaScript:** lenguaje de programación interpretado que se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. Al contrario que PHP, se usa principalmente del lado del cliente. Además su uso en aplicaciones externas a la web, como por ejemplo en documentos PDF o aplicaciones de escritorio, es también significativo.

Aparte de usarse en la aplicación de jQuery y jQuery UI, su principal uso es el del manejo de la API de Google Maps.

- **Requisitos Funcionales**

Un **requisito funcional** define el comportamiento interno del software: cálculo, detalles técnicos, manipulación de datos y otras funcionalidades específicas que muestran cómo serán llevado a la práctica los casos de uso. La

especificación de los requisitos funcionales del software es la fase de análisis del problema, donde se debe aportar un método de análisis y en la que el desarrollador es el máximo responsable. Básicamente el objetivo es obtener una definición del software que se va a construir.

Estos requisitos deben describir los servicios o actividades que hay que proporcionar con todo detalle. Sin embargo, éstos por sí solos no están detallados. Los escenarios son descripciones de cómo se utilizará el sistema en la práctica (completan o sustituyen los requisitos funcionales). Estos escenarios son representados mediante los **casos de uso**, incorporados en UML para describir la interacción entre los actores y el sistema. Un conjunto de casos de uso describe todas las posibles interacciones con el sistema, lo que puede ir mal y cómo manejar en su caso el problema.

Para poder crear diagramas de casos de uso hay que entender la semántica para su construcción. UML (Unified Modeling Language) 2.0 definió una notación para cada elemento del diagrama ([21]).

Actor: el caso de uso es una descripción funcional de la interacción entre uno o varios actores y el sistema que se está modelando específicamente sobre la funcionalidad desarrollada en el caso de uso. El actor está asociado a un rol específico que cumple en la interacción con el sistema (Ej: Usuario administrador, empleado, etc.). Un actor del sistema generalmente corresponde con un ser humano que tiene un rol de interacción con el caso de uso aunque no siempre es así; un actor puede ser un hardware, otro sistema, un servicio que consume las funcionalidades modeladas en el caso de uso, etc. En UML 2.0 un actor se expresa simplemente con una figura en forma de ser humano. Esta figura va acompañada del nombre del actor que representa.

Caso de Uso: un caso de uso define un comportamiento del sistema capaz de cumplir un objetivo particular para un actor particular. Básicamente se podría decir que un requerimiento funcional del sistema puede dar origen a uno o muchos casos de uso. Es por eso que definir correctamente todos los requerimientos funcionales y no funcionales previo al diseño de diagramas de casos de uso es tan importante. EL diagrama de casos de uso no muestra el detalle de estas operaciones pero sí presenta la relación entre el actor y los

casos de uso. La notación UML para un caso de uso se denota por una elipse y el nombre del caso de uso en su interior (Fig. 1).

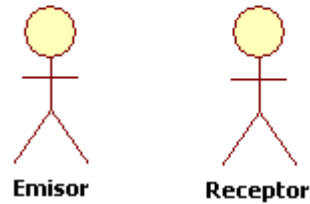


Fig. 1 – Actores UML

Relaciones: para la construcción de los diagramas de casos de uso se han identificado un conjunto de relaciones que se pueden dar entre ellos. Estas relaciones significativas, de interacción o de dependencia entre comportamientos del sistema a través de los casos de uso. A continuación se detalla estas relaciones:

- a. **Asociación:** una asociación es la relación de comunicación más simple entre un actor y el sistema. Las asociaciones implican que un actor interactúa con el comportamiento modelado en el caso de uso. La notación para las asociaciones es una línea simple entre el actor y el caso de uso en el que participa (Fig. 2).



Fig. 2 – Relación asociación UML

- b. **Inclusión (INCLUDE):** la inclusión en los diagramas de caso de uso es un tipo de relación que describe explícitamente el uso obligatorio de otro caso de uso. Es decir, que un caso de uso incluido en otro indica que su ejecución es obligatoria en el comportamiento modelado en el caso de uso que lo incluye. La notación UML para una inclusión se ve como una flecha de líneas discontinuas con la palabra reservada “<<include>>” (Fig. 3).

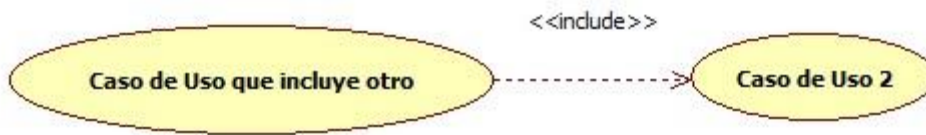


Fig. 3 – Relación inclusión UML

- c. **Extensión (EXTEND)**: la extensión en los diagramas de casos de uso es un tipo de relación que indica que un caso de uso puede extender su comportamiento a otro caso de uso. La extensión no es obligatoria como sí ocurre en la inclusión, es decir, que el caso de uso al que se extiende puede o no ejecutar el comportamiento del caso de uso. La notación UML para una extensión se ve como una flecha de líneas discontinuas con la palabra reservada “<<extend>>” (Fig. 4).

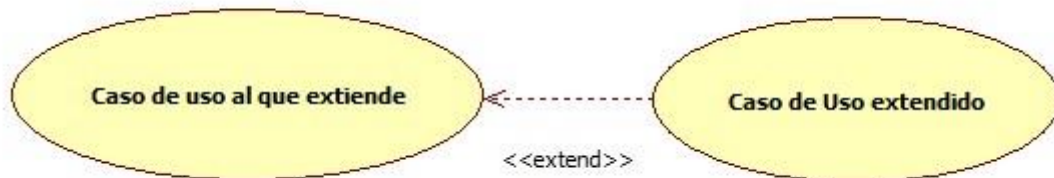


Fig. 4 – Relación extensión UML

- d. **Generalización**: en algunos momentos surge la necesidad, al igual que en la orientación a objetos, de que un caso de uso se pueda especializar en otros más concretos. A esto se le llama Generalización. En un caso de uso especializado, siempre se esperará que el caso de uso padre se ejecute. La notación UML para la generalización se representa con una punta de flecha triangular que va desde el caso de uso hijo hasta el caso de uso padre (Fig. 5).



Fig. 5 – Relación generalización UML

Antes de mostrar los diagramas de casos de uso de la interfaz gráfica se comenta como debería funcionar ésta para comprender mejor la finalidad de los casos de uso necesarios para el desarrollo de esta parte del sistema.

Como ya se ha visto el objetivo fundamental de este trabajo de fin de grado es monitorizar las marcas RFID con la finalidad de estudiar la dispersión de los cantos a lo largo de una playa. Para conseguirlo es necesario el uso de los dispositivos Waspnote, una estación total y un ordenador portátil que contendrá la interfaz gráfica de usuario; esta última será la encargada de coordinar todo el proceso, el cual está basado en la detección de las marcas RFID. Cada detección compuesta por una serie de etapas que son las que se enumeran a continuación:

1. En un primer momento la detección es realizada por los dispositivos Waspnote, que proporcionan la información correspondiente a la interfaz gráfica.
2. Tras la detección de la marca RFID, hay que realizar una medición por parte de la estación total en el lugar donde se produjo la detección. Las medidas obtenidas serán enviadas de nuevo a la interfaz gráfica.
3. Y finalmente la interfaz gráfica manipulará los datos obtenidos tanto de los dispositivos Waspnote como de la estación total, para obtener los resultados correspondientes.

Para que este proceso funcione correctamente es indispensable situar la estación total en un punto cuyas coordenadas geográficas sean conocidas, puesto que con este punto y los datos obtenidos tras realizar las correspondientes mediciones, nos dará las coordenadas de las detecciones de las marcas RFID.

Además es necesario realizar una primera medición en otro punto conocido, que hará de punto de referencia, ya que no es posible la orientación exacta de la estación total al norte geográfico, con la finalidad de hallar el ángulo que debe aplicarse a los puntos obtenidos para representarlos correctamente. Así que con los tres puntos obtenidos: posición de la estación total, punto conocido que hace de referencia, y el punto obtenido de la estación calculado

sobre el punto de referencia, podemos hacer un sistema de ecuaciones para calcular dicho ángulo, entonces a cada punto obtenido, fruto de las detecciones de las marcas RFID, se le aplicará este ángulo representándolo de forma correcta sobre la API de Google Maps.

Para el cálculo del ángulo citado anteriormente, al cual llamaremos α , se consigue aplicando las ecuaciones de rotación en dos dimensiones y los puntos mencionados anteriormente, quedando resuelto de la siguiente manera.

$$\begin{pmatrix} X' \\ Y' \end{pmatrix} = \begin{pmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

Multiplicamos y resulta:

$$\begin{cases} X' = x * \cos \alpha - y * \sin \alpha \\ Y' = x * \sin \alpha + y * \cos \alpha \end{cases}$$

Siendo P_{RR} el punto de referencia con coordenadas reales, P_{RM} el punto de referencia medido por la estación total, y P_B el punto donde está situado la estación total. Consideraremos también la X como el este geográfico y la Y como el norte geográfico, quedando el sistema de ecuaciones de la siguiente manera:

$$\begin{cases} E_{P_{RR}} = [E_{P_{RM}} - E_{P_B}] * \cos \alpha - [N_{P_{RM}} - N_{P_B}] * \sin \alpha \\ N_{P_{RR}} = [E_{P_{RM}} - E_{P_B}] * \sin \alpha + [N_{P_{RM}} - N_{P_B}] * \cos \alpha \end{cases}$$

Resolvemos el sistema de ecuaciones utilizando el método de sustitución despejando el $\sin \alpha$ en la primera ecuación:

$$\sin \alpha = \frac{([E_{P_{RM}} - E_{P_B}] * \cos \alpha) - E_{P_{RR}}}{[N_{P_{RM}} - N_{P_B}]}$$

Sustituimos en la segunda ecuación y nos resulta que el ángulo α es igual a lo siguiente:

$$\alpha = \cos^{-1} \frac{([N_{PRM} - N_{PB}] * N_{PRR}) + ([E_{PRM} - E_{PB}] * E_{PRR})}{[N_{PRM} - N_{PB}]^2 + [E_{PRM} - E_{PB}]^2}$$

Este proceso nos servirá para hallar el ángulo que hay que rotar las coordenadas geográficas obtenidas por la estación total, por tanto, una vez hallado éste, bastará con sustituir los valores de las coordenadas de la posición de la estación total (y) y las coordenadas de la detección de la marca RFID obtenidas por la estación total (y) para calcular la posición exacta de la propia detección, es decir, calcular los valores de y de las ecuaciones anteriores.

Cabe destacar que el tipo de coordenadas geográficas con las que se va a trabajar son tanto coordenadas de latitud y longitud en grados sexagesimales como UTM, estas últimas a diferencia del sistema de coordenadas geográficas, expresadas en longitud y latitud, las magnitudes en el sistema UTM se expresan en metros únicamente al nivel del mar, que es la base de la proyección del elipsoide de referencia. Para este proyecto se ha trabajado teniendo como elipsoide de referencia el elipsoide WGS84, que es el mismo en el que se basa el sistema GPS.

Teniendo claro estos aspectos, se procede a continuación a ilustrar los diferentes casos de uso de la interfaz gráfica de usuario. En la siguiente imagen se muestran los casos de uso principales:

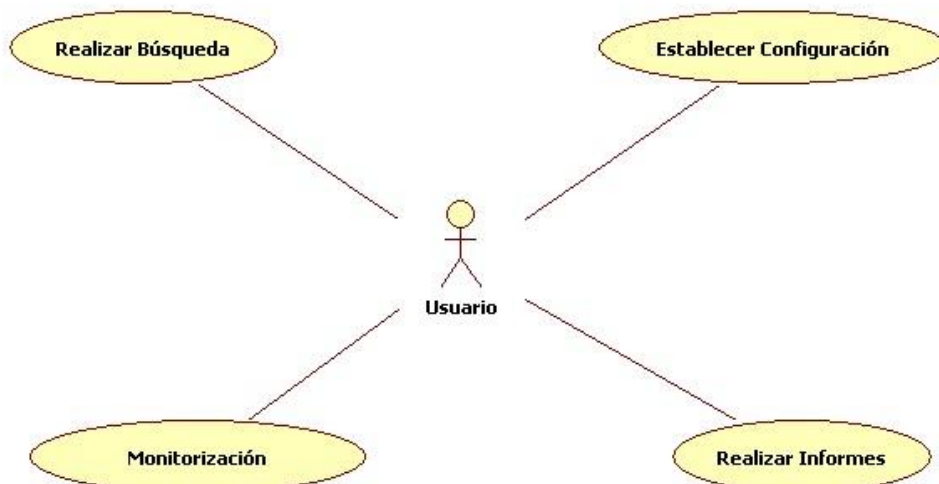


Fig. 6 – Casos de Uso principales

En el diagrama anterior se muestra los casos de uso principales para que el usuario trabaje con la interfaz gráfica. A continuación se enumera cada una de ellas con sus correspondientes características:

- **Realizar Búsqueda:** los estudios para la dispersión de los cantos se harán por campañas, es decir, cada sesión de monitorización será una campaña, por tanto, la interfaz se regirá de esta manera; entonces el usuario podrá realizar búsquedas por campañas, mostrándole la información oportuna de cada una de ellas.
- **Establecer Configuración:** el usuario tiene la posibilidad de establecer los parámetro de configuración entre los que destacan: parámetros de la base de datos, coordenadas de los puntos de referencia y de la estación total, etc.
- **Realizar Informe:** como se muestra en la Fig. 7, este caso de uso se divide en otros dos, ya que se pueden realizar dos tipos de informe: uno es obtener la imagen de la API de Google Maps con las correspondientes coordenadas, y el otro es obtener un informe con todos los datos obtenidos en formato Excel, por ser el formato usado por el cliente para transferir los datos a otras aplicaciones.

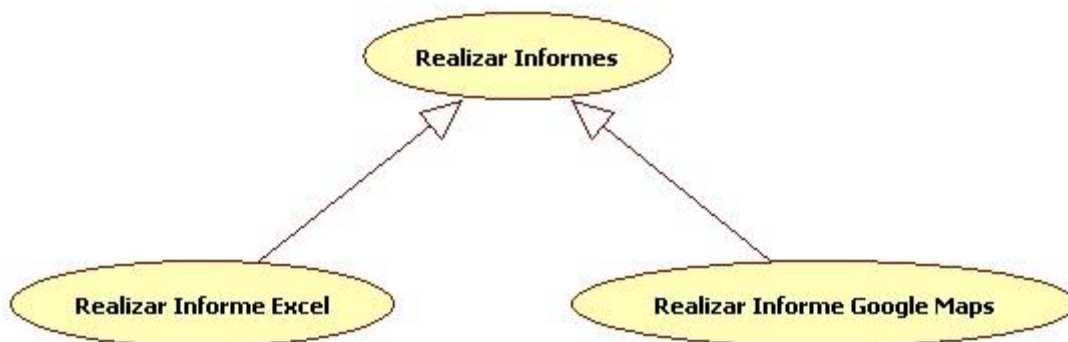


Fig. 7 – Desglose del CdU 'Realizar Informes'

- **Monitorización:** como sucede con el caso de uso anterior, se desglosa en otra serie de casos de uso más específicos. Mientras está transcurriendo la monitorización el usuario puede realizar varias acciones, por ejemplo, modificar o eliminar algún dato obtenido, añadir una medición topográfica (aparte de las coordenadas de las detecciones, también se obtendrán

puntos topográficos a lo largo de la monitorización), y establecer el modo “siembra”, este modo será necesario solo para la primera campaña, con el fin de ir colocando los cantos a lo largo de la playa de una forma rápida y eficiente sin hacer uso de los dispositivos Waspote. Este proceso puede llevarse a cabo debido a que todos los cantos se encuentran marcados visiblemente con un ID numérico cada uno, entonces a medida que se van sembrando la IU debe dar la posibilidad de relacionar el ID correspondiente con la coordenada obtenida a través de la estación total. En la siguiente figura se muestra el desglose comentado de este caso de uso:

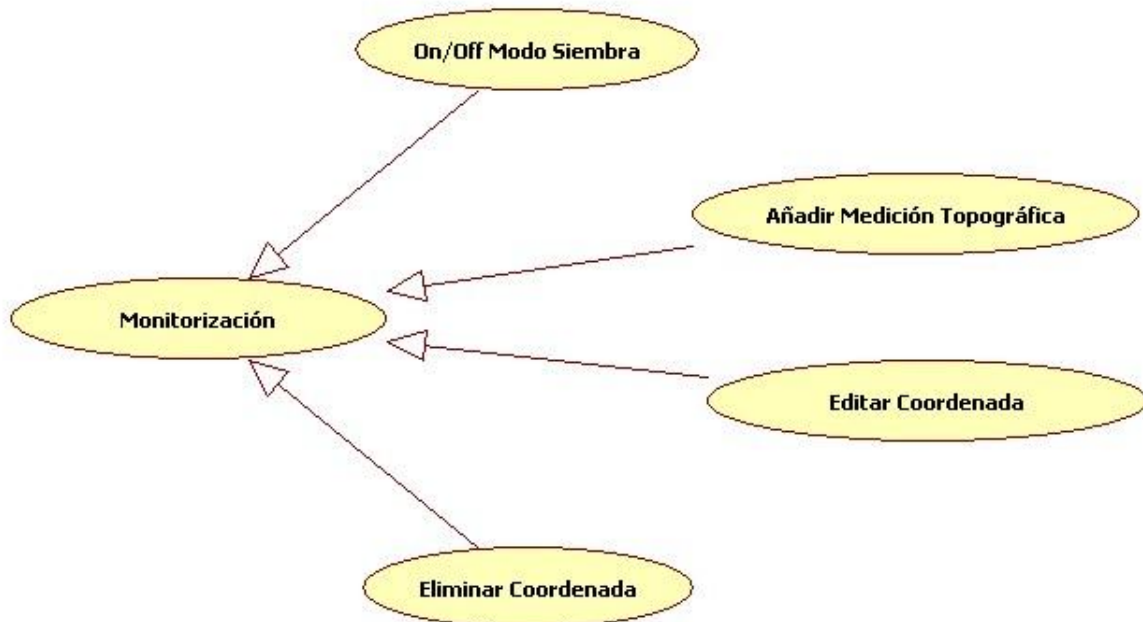


Fig. 8 – Desglose del CdU ‘Monitorización’

6.1.2 Aplicaciones Waspote, Análisis

Para el caso de las aplicaciones Waspote, el objetivo es la detección de las marcas RFID alojadas en una serie de cantos que se distribuirán a lo largo de la playa con la finalidad de estudiar su dispersión.

Los dispositivos Waspotes (Fig. 9) son el resultado de dos años de investigación por parte de la empresa aragonesa Libelium, empresa de diseño y fabricación de hardware para la implementación de redes sensoriales

inalámbricas, redes malladas y protocolos de comunicación para todo tipo de redes inalámbricas distribuidas.

Wasmote es una plataforma modular para construir redes de sensores inalámbricas de muy bajo consumo. La plataforma comprende:

- La placa Wasmote con microcontrolador, memoria, batería, acelerómetro y sockets para añadir módulos.
- API y compilador OpenSource.
- Diferentes radios de comunicación inalámbrica con alcances de hasta 40 Km.
- Diversos módulos opcionales para añadir comunicación Bluetooth, GPRS y GPS.
- Una gran variedad de placas de sensores para medir gases, eventos físicos y parámetros necesarios en el smart metering.



Fig. 9 – Dispositivo Wasmote

Para este proyecto se hará uso de dos dispositivos Wasmotes, a los cuales identificaremos de la siguiente manera:

- **Wasmote Rover:** este Wasmote se encontrará conectado al lector RFID y será el equipo que uno de los usuarios habrá de desplazar por el área objeto de estudio para detectar los cantos marcados. Por cada

detección enviará por radio una serie de parámetros que se indicarán posteriormente, además del identificador RFID del canto en cuestión.

- **Wasmote Base:** este dispositivo tiene como objetivo recibir la información enviada por el *Wasmote Rover* y pasarla a la interfaz gráfica de usuario que se encuentra en un ordenador portátil al que está conectado, para que este último haga las acciones oportunas.

Por tanto, habrá que realizar dos aplicaciones distintas, una para cada *Wasmote*, para ello será necesaria la consulta de la información de [1] y [13]. A continuación, tal y como se hizo para la interfaz gráfica se hace una distinción entre los requisitos funcionales y no funcionales.

- **Requisitos No Funcionales**

Los requisitos no funcionales del TFG con respecto a las aplicaciones de los *Wasmote* se enumeran de la siguiente manera:

- ❖ Se necesitan dos dispositivos *Wasmote* para el correcto funcionamiento del sistemas
- ❖ Necesidad de un ordenador personal con el IDE específico instalado en él para el desarrollo de las aplicaciones correspondientes.
- ❖ Ambos dispositivos *Wasmotes* deben disponer del módulo de radio para establecer enlaces en banda ISM (869 MHz), una tarjeta de memoria microSD y del módulo GPS, que a pesar de que no proporciona la precisión deseada, se usarán como un medio secundario de localización y para contrastar los errores de GPS frente las coordenadas proporcionadas por la estación total.
- ❖ El *Wasmote Rover* debe contar con una antena para la detección de las marcas RFID

Para cubrir estas necesidades, se dispone de dos dispositivos *Wasmote* con las siguientes características:

Componentes del Wasmote Base: equipado con un receptor GPS SiRF III de 12 canales y un modem radio XBee 868 PRO, conectada por USB a un portátil/netbook.

Componentes del Wasmote Rover: equipado con un receptor GPS SiRF III de 12 canales y un modem radio XBee 868 PRO. El sistema de detección RFID se conecta al mote mediante uno de sus puertos serie y se alimenta mediante una batería portátil. En la tabla siguiente se muestran las características específicas de este último dispositivo.

Subsistema	Componente	Modelo
Alimentación	Batería de 24V@7Ah (2 x 12V@7Ah)	PowerSonic 12V
	Convertor DC/DC	Convertor DC/DC
Detección RFID	Lector RFID	RI-CTL-MB2B
	Módulo RFM	RI-RFM-008B
	Módulo de autosintonización	RI-ACC-008B
Microprocesador	Antena	Antena de diseño propio
	Sistema empotrado	Wasmote V1.1
	Receptor GPS	Receptor SirFIII, 12 canales
	Modem radio	Módulo XBee 868 PRO
	Adaptador RS232 - 3V3	

Tabla 2 – Componentes del equipo móvil o de detección de marcas

A continuación se cita la información sobre el **lector RFID**, su alimentación y el resto de componentes necesarios para su funcionamiento:

- **Alimentación**

El lector y los módulos RF pueden alimentarse de una única fuente de tensión o de fuentes separadas. Los RI-RFM-007B, RI-RFM-008B y RI-ACC-008B requieren de una tensión de alimentación regulada en el rango 7-24VDC. En esta implementación se ha diseñado un sistema de alimentación mediante dos baterías de gel-plomo de 12V y 7Ah conectadas en serie. La salida de la batería está conectada a un convertor DC-DC que permite convertir y estabilizar la

tensión de salida de las baterías a cualquier tensión en el rango 7-24V y entregar hasta 10A.

El rango de detección del lector RFID aumenta con la tensión de alimentación del circuito de radio por lo que en principio la alimentación a 24V, máximo admisible, podría ser la mejor opción. Sin embargo, a la máxima tensión se genera un exceso de calor que es necesario extraer del interior de la caja que aloja la electrónica. Tras experimentar con varias tensiones de alimentación se comprobó que a 18V se obtenía el mejor resultado global; esto es, la temperatura del detector se mantenía dentro de unos límites razonables, mientras que la distancia máxima de detección no disminuía de manera apreciable (alrededor de un metro para una alineación detector - antena óptima).

Para evitar dañar las baterías de plomo debido a una descarga excesiva, el convertor DC-DC se ha programado para desconectar las baterías si la tensión de las mismas cae por debajo de 21V. No obstante, dicha tensión se mide simultáneamente mediante uno de los convertidores A/D de la mota y se transmite a la estación base para su monitorización por los operadores. De esta manera, es posible estimar el tiempo restante de autonomía del equipo de detección.

- **Lector RFID**

El sistema de detección de marcas RFID comprende cuatro elementos: el lector RFID (modelo RI-CTL-MB2B), el módulo RFM (modelo RI-RFM-008B) y el módulo sintonización (modelo RC-ACC-008B), todos ellos de Texas Instruments, y una antena de diseño propio.

La antena es una estructura circular de 80 cm de diámetro que contiene una bobina de tres vueltas de hilo de cobre macizo. El módulo de sintonización se usó para ajustar el circuito de radiofrecuencia a la frecuencia de resonancia de la antena.

El lector está conectado a la mota desde la que se controla a través de una interfaz serie de tres hilos. Dado que el lector opera a los niveles del estándar

RS232 y el puerto serie de la mota lo hace a niveles TTL 3.3V se ha usado un circuito conversor de niveles basado en el IC MAX3223.

Desde la mota se monitoriza constantemente el estado del lector, emitiéndose alertas sonoras tanto cuando se produce una detección como cuando la comunicación entre la mota y el lector falla.

A continuación se muestra dos imágenes: una del lector RFID (Fig. 10) y otra del sistema completo del lector RFID (Fig. 11), en la parte izquierda de esta imagen se puede ver la antena junto con el Wasp mote Rover, en el centro el lector RFID y a su derecha la alimentación de este último.



Fig. 10 – Lector RFID



Fig. 11 – Sistema RFID completo

- **Requisitos Funcionales**

En cuanto a los requisitos funcionales, no se establecen casos de uso como se hizo para la interfaz gráfica, ya que se trata de una aplicación autónoma y no demasiado compleja, así que se enumerarán los requisitos funcionales de la misma forma que los requisitos no funcionales de esta parte del sistema:

- ❖ El Wasmote Rover debe detectar las marcas RFID, enviándola la información de la misma al Wasmote Base. El Base también debe pasar dicha información al portátil/netbook al que esté conectado. Además ambos Wasmotes deben informar periódicamente para hacer un seguimiento del estado de los mismos.
- ❖ Todos los mensajes intercambiados deben ser almacenado en las tarjetas microSD de las que disponen los Wasmote.
- ❖ La aplicación de este sistema en un entorno como el que se puede encontrar en el litoral norte de Gran Canaria requiere el uso de señales visuales y auditivas para informar sobre el estado de la monitorización o de los dispositivos.
- ❖ El intercambio de mensajes será a través de radio cuando la comunicación sea entre los Wasmote, y a través de USB cuando sea Wasmote Base y el portátil/netbook al que esté conectado.
- ❖ Los mensajes deberá contener como mínimo información sobre: el Wasmote emisor, temperatura y batería del Wasmote correspondiente, fecha y hora del mensaje, identificador de la marca RFID cuando se trate de una detección, y a pesar de no hacer uso de los datos del GPS deberán incorporarse igualmente en el mensaje por posible necesidad futura.

6.2 Diseño

Una vez realizado el correspondiente análisis del desarrollo del sistema, nos encontramos ante una nueva etapa: el **Diseño**.

El diseño hace posible la transformación de los requisitos en un producto software acabado mediante la obtención de modelos físicos del software y tiene como objetivo definir la estructura del sistema y a la facilidad del mantenimiento, ayudando a construir sistemas estables.

6.2.1 Diseño de la Interfaz Gráfica

El diseño de la interfaz gráfica de usuario engloba el diseño de tres aspectos principales:

▪ **Diseño de la base de datos**

Como ya se ha visto, se hace uso de MySQL como gestor de base de datos. Con este gestor se ha creado una nueva base de datos con el nombre de ‘*seguimientorfid*’, la cual cuenta con una serie de tablas quedando como se muestra en la figura siguiente:

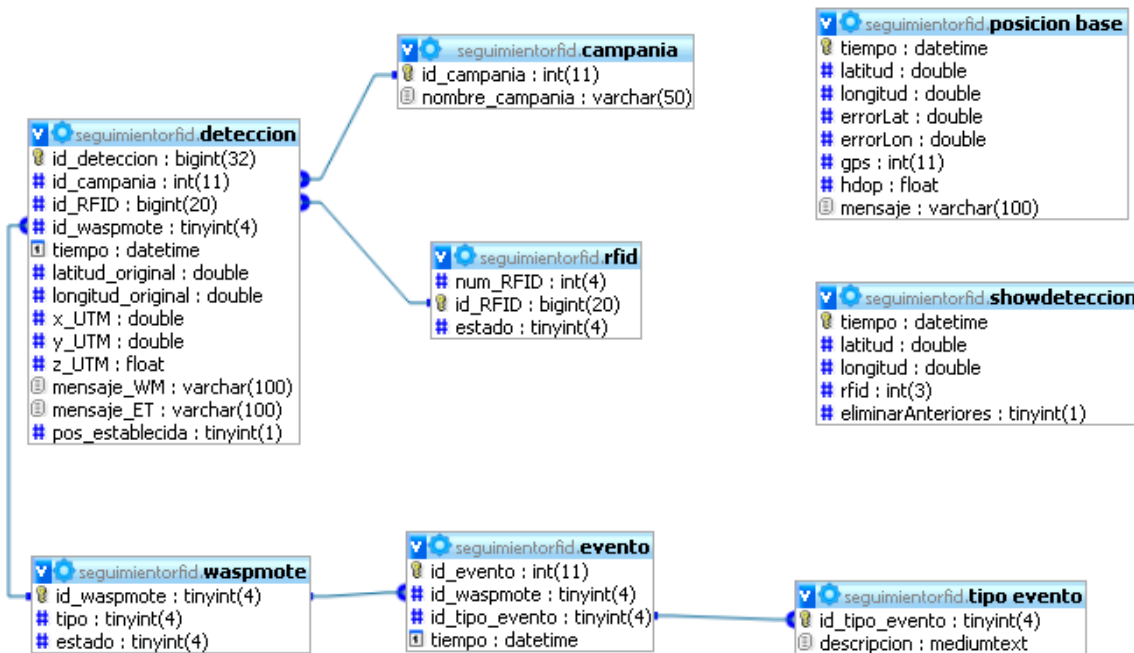


Fig. 12 – Diseño de la base de datos ‘seguimientorfid’

En esta base de datos se cuenta con las tablas que se enumeran a continuación:

- ❖ **posicion_base**: esta tabla tiene como finalidad guardar la información de los mensajes generados por el Waspote Base.
- ❖ **showdeteccion**: el contenido que almacena esta tabla es temporal, se utiliza para la actualización de la API de Google Maps en tiempo real. Se insertan las coordenadas que se desean mostrar, junto a otros parámetros como el ID de la marca RFID detectada. A continuación, la página que contiene la API de Google Maps ejecuta un script que accede a esta tabla temporalmente para actualizar el mapa, para finalmente eliminar de la tabla los puntos ya dibujados.

- ❖ **campania:** aquí se van almacenando las campañas que se van generando a lo largo del estudio de dispersión, es decir, habrá tantos registros como campañas realizadas.
- ❖ **rfid:** esta tabla contiene toda la información de las marcas RFID utilizadas en el estudio. El campo *'num_rfid'* tendrá valores de 1 a n, donde n indica el total de marcas RFID, y el campo *'id_rfid'* representa el id de cada marca.
- ❖ **waspmote:** el contenido que guarda esta tabla hace referencia a los distintos dispositivos Waspote empleados.
- ❖ **tipo_evento:** esta tabla contiene los eventos que pueden producirse en las aplicaciones de los Waspote, como por ejemplo: encendido del Waspote, detección de marca RFID, conexión del GPS, etc.
- ❖ **evento:** aquí se almacenan los eventos que se han producido durante la monitorización (la funcionalidad de esta tabla no se aplica al 100%).
- ❖ **detección:** esta es la tabla más importante, contiene la información de todas las detecciones de las marcas RFID, utilizando distintas *'foreign key'* para referenciar a otras tablas (waspmote, campania y rfid).

▪ **Diseño de la estructura del sistema**

El siguiente paso es ilustrar como se estructura todo el sistema en su conjunta, a pesar de que este apartado se refiera solo a la interfaz gráfica, ya que éste requiere de todos los elementos para su correcto funcionamiento.

Como se puede ver en la Fig. 13, los rectángulos con el borde redondeado hacen referencia al hardware empleado en el sistema, en este caso tenemos cuatro componentes hardware distintos: la estación total, los dos dispositivos Waspote (Base y Rover) y el portátil o netbook donde estará alojado el resto del sistema. El resto (rectángulos normales) representan el software utilizado, donde los que no se encuentran dentro de otro rectángulo son los principales y los que sí lo están son los “secundarios”. Los principales son la interfaz gráfica, la API de Google Maps y el XAMPP, este último contendrá tanto el servidor Apache como MySQL.

Los dispositivos que se comunican con la interfaz gráfica son el Waspote Base, que a su vez está conectado al Rover, y la estación total. Con respecto a

la interfaz gráfica y la API de Google Maps necesitarán de los servicios proporcionados por el gestor de base de datos MySQL, además los Mapas necesitarán también de Apache.

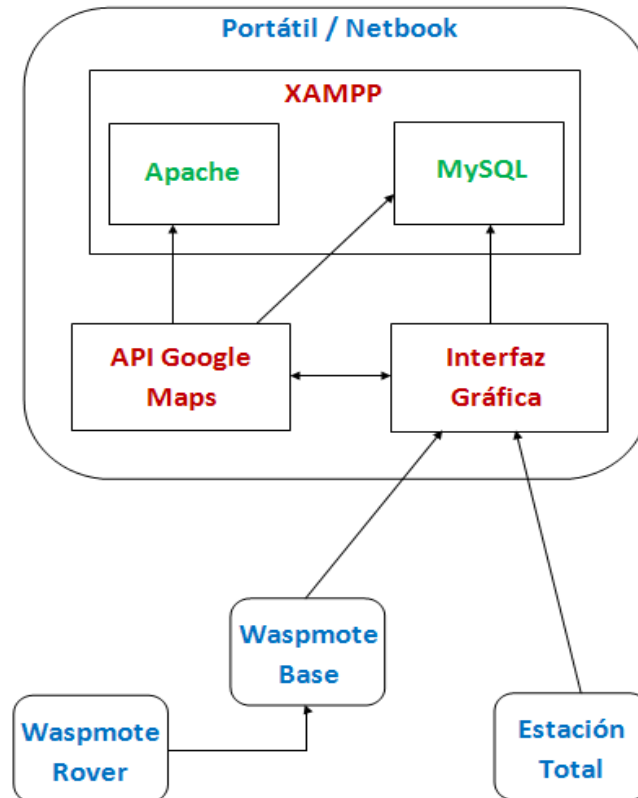


Fig. 13 – Estructura del sistema

▪ Diseño de la Interfaz Gráfica

Atendiendo al análisis obtenido en la etapa anterior, el diseño de la interfaz gráfica de usuario, resumida con aspectos generales, es el que se describe a continuación:

El componente principal de la interfaz gráfica es el **JFrame**, clase propia de la biblioteca gráfica ‘Swing’ de Java, en concreto se creará una clase que herede de esta primera. Desde esta clase se podrá realizar tres de los cuatro principales casos de uso de la aplicación: ‘Realizar Informes’, ‘Realizar Búsquedas y ‘Manejar Monitorización’.

Para el caso de uso ‘Establecer Configuración’ se aplicará una clase que herede de otra clase propia de la biblioteca Swing, **JDialog**.

Otro de las clases importantes de Java que se utilizará como clase padre de la herencia, es **JApplet**, que se utilizará como navegador Web, con la finalidad de mostrar el fichero HTML que contiene la API de Google Maps.

El control de la estación total juega un papel crucial en la funcionalidad del sistema. La interfaz gráfica intercambia con la estación total una serie de mensajes con formatos propios del protocolo de comunicación de esta última, entre ellos, los más importantes son aquellos que contienen la información correspondiente a las medidas, las cuales contienen la información de la localización de las detecciones de las marcas RFID. Dicha información puede obtenerse de distintos modos según se establezca en la estación total, para este TFG el modo empleado es el NEZ (N or E or Z coordinates mode), que tiene la estructura del siguiente ejemplo (esta información ha sido extraída de la referencia [2]):

```

U-00596337+01011930+00095802m+1203040d
|   |           |           |   |   |   |
ID  N           E           Z   m/f  H   d/g/m

N : N coordinate           -596.337m  (+/-, 8 digits)
E : E coordinate           +1011.930m (+/-, 8 digits)
Z : Z coordinate           +95.802m   (+/-, 8 digits)
    
```

De este mensaje la información realmente importante son los tres parámetros que vienen a continuación del ID del mensaje, los cuales hacen referencia a las coordenadas en UTM en dirección norte (primer parámetro), en dirección este (segundo parámetro) y el último la altura en metros de la medida.

Además de utilizar estas clases, la interfaz a nivel global, se compone de otras muchas que además de complementar a las citadas anteriormente para satisfacer los distintos casos de uso, serán necesarias para cubrir otras necesidades: como acceder a la base de datos, configurar la página con las detecciones que se mostrarán el mapa, etc.

6.2.2 Diseño de las aplicaciones Wasmote

Para la parte de diseño de las aplicaciones Wasmote nos centraremos en dos aspectos: el formato de los mensajes enviados por los dispositivos Wasmote y los diagramas de flujo de las aplicaciones de cada uno de ellos.

Con respecto a los **mensajes**, tanto para el Wasmote Rover como para el Wasmote Base el formato es coincidente, para simplificar el desarrollo y la comunicación final con la interfaz gráfica. La estructura de un mensaje sería el siguiente:

w#b#t#x#.#y#.#i#g#p#.#fDDMMYYhHHMMSS.Se#d#s#

En este mensaje cada vez que aparece una letra significa que es un nuevo parámetro el que aparece y que los dígitos que aparecen a su derecha representan su valor. En concreto, cada letra tiene el siguiente significado:

- **w**: este parámetro indica cual es el Wasmote del que proviene el mensaje (0 → Wasmote Rover, 1 → Wasmote Base).
- **b**: este parámetro representa la cantidad de batería restante del Wasmote correspondiente.
- **t**: con esta letra se hace referencia a la temperatura a la que se encuentra el Wasmote correspondiente.
- **x**: a pesar de no hacer uso de los datos obtenidos por el GPS de los Wasmote, se almacenan para comparar datos y por la futura necesidad de estos datos. La **x** representa la latitud en grados decimales con una precisión de hasta seis decimales.
- **y**: este parámetro indica la longitud del Wasmote correspondiente. Al igual que con la latitud se encuentra en grados decimales con una precisión de hasta seis decimales.
- **i**: en caso de que se detecte una marca RFID, este parámetro tendrá el valor en notación hexadecimal del valor del ID de dicha marca RFID, en caso contrario siempre tendrá el valor cero.
- **g**: la **g** indica el número de satélites a los que se encuentra enganchado el módulo GPS del Wasmote correspondiente.
- **p**: este parámetro representa el valor HDOP (Horizontal Dilution of Precision) del GPS, es decir, la incertidumbre 2D de las coordenadas

obtenidas por éste. Este valor viene determinado por la geometría de los satélites, a diferencia de otros errores que vienen dados por la calidad de la señal [17]. Mientras más bajo sea este valor, el resultado obtenido será de mejor calidad.

- **f**: aquí se indica la fecha del mensaje con formato DDMMYY.
- **h**: aquí se indica la hora del mensaje con formato HHMMSS, además de proporcionar las milésimas de segundo separadas por un punto. Este parámetro completa el campo anterior de fecha.
- **e**: este parámetro representa el evento del que está informando el Wasmote correspondiente, por ejemplo, encendido del Wasmote, conexión establecida del GPS, detección de una marca RFID, etc.
- **d**: este parámetro está destinado para ir enumerando cada uno de los mensajes.
- **s**: el último campo está reservado para obtener la información que indica el estado de la señal de radio entre el Wasmote Rover y el Wasmote Base.

Y para finalizar con el diseño de las aplicaciones Wasmote, se hace uso de unos sencillos diagramas de flujo que explican el transcurso de la ejecución de cada una de ellas. Comentar que dichas aplicaciones, sea cual sea su finalidad, sigue una estructura de inicialización (**Setup**) y de bucle (**Loop**), es decir, en un primer momento se ejecuta una parte del código que tiene como objetivo preparar la ejecución con las inicializaciones que se crean oportunas, y finalmente un trozo de código que se ejecutará cíclicamente hasta que se considere necesario o indefinidamente. Podemos ver esta estructura en la siguiente imagen (Fig. 14):

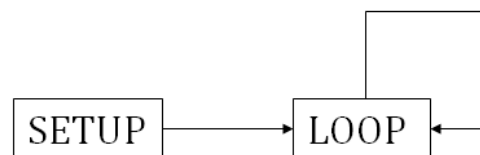


Fig. 14 – Estructura APIs Wasmote

A continuación se muestran las figuras que representan los **diagramas de flujo** del Wasmote Rover y del Wasmote Base, en primer lugar (Fig. 15) aparece el diagrama del 'Setup' del Base. Aquí se empieza con una señal acústica

(Zumbador) para avisar al usuario del inicio del proceso, para continuar con la inicialización de los distintos módulos que van a intervenir (radio, tarjeta microSD y GPS), en caso de no haber ningún problema se vuelve a realizar otra señal acústica para indicar al usuario que finalmente el GPS ha conectado con los satélites y se procederá a entrar en el 'Loop'.

En la Fig. 16 se puede ver la ejecución del bloque Loop de la aplicación del Waspote Base. En este bucle se empieza por comprobar si se ha recibido un mensaje del Waspote Rover, para tratarlo y enviarlo a la interfaz gráfica. Independientemente de que se haya recibido o no un mensaje, el propio Waspote Base construye uno con información de sí mismo para pasarlo también a la interfaz gráfica, y finalmente empezar un nuevo ciclo del bucle.

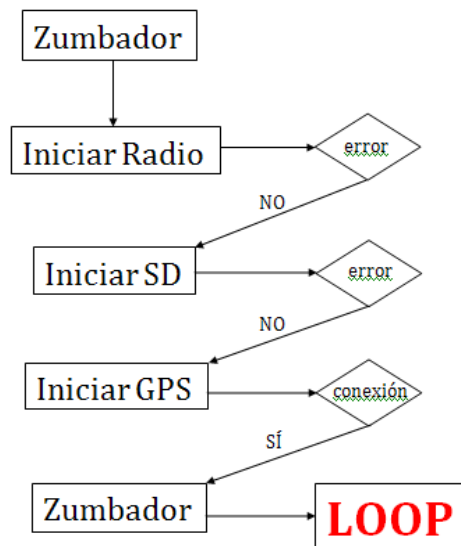


Fig. 15 – Setup de Waspote Base

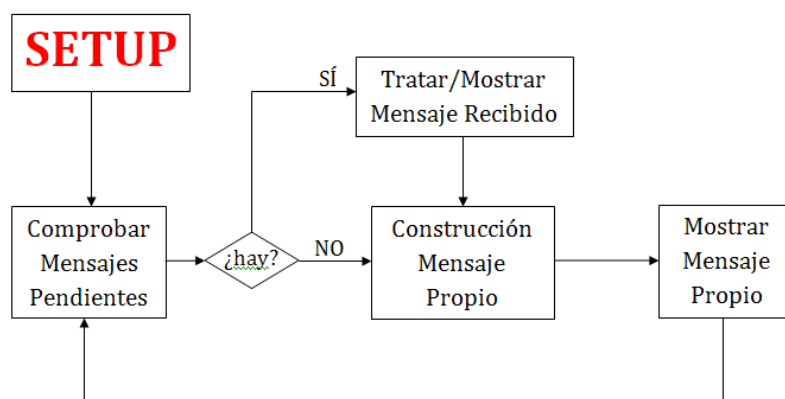


Fig. 16 – Loop del Waspote Base

Por parte del Waspnote Rover, el 'Setup' (Fig. 17) en rasgos generales es igual que el del Waspnote Base, salvo que hay que realizar la correspondiente inicialización del módulo de detección RFID.

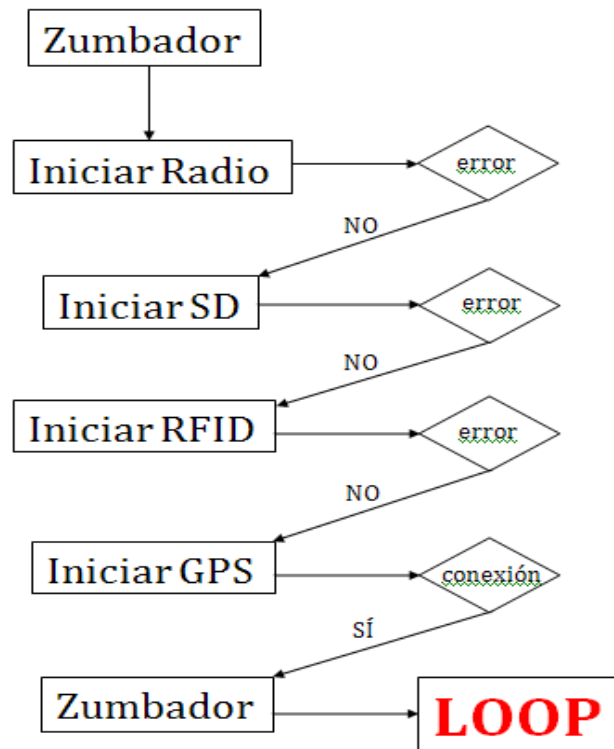


Fig. 17 – Setup del Waspnote Rover

Atendiendo al bloque Loop (Fig. 18), se comienza por comprobar si se ha realizado una detección de alguna de las marcas RFID. Si se ha producido se envía el mensaje con los parámetros oportunos al Waspnote Base. En caso negativo se comprueba si ha vencido el tiempo del período establecido, si lo ha hecho se envía un mensaje al Base, si no se comienza un nuevo ciclo del bucle. Este período es utilizado para realizar un seguimiento del estado del Waspnote a lo largo de la monitorización, y no restringir el envío de paquetes a las detecciones exclusivamente. Además, cuando se realiza una detección se produce una señal acústica para informar al usuario que esté llevando el Waspnote y la antena de la misma, y realizar la correspondiente medida con la estación total.

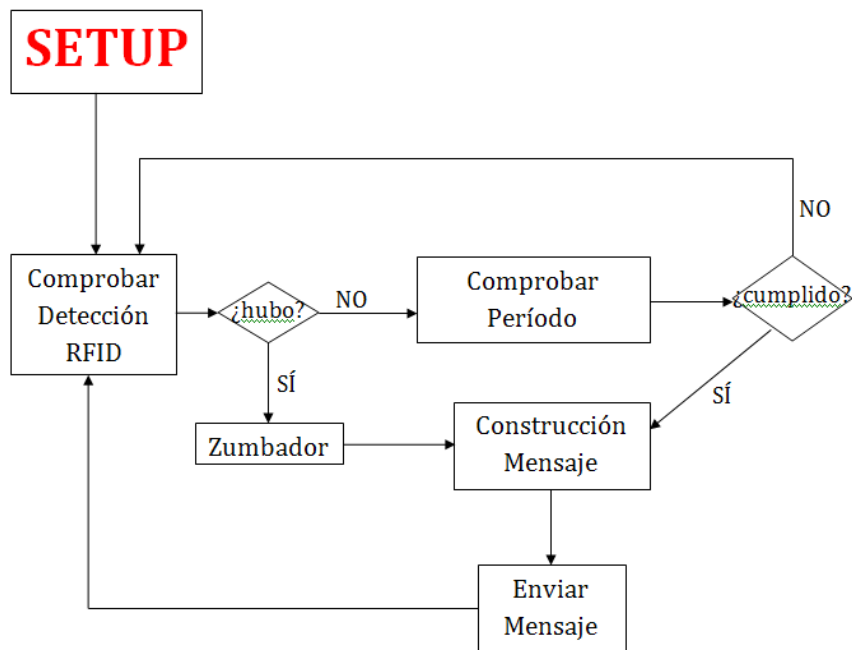


Fig. 18 – Loop del Waspote Rover

6.3 Construcción

En este último bloque del capítulo se hace referencia a las siguientes dos etapas del desarrollo del sistema:

- **Implementación:** la implementación es la etapa en la que se reduce el diseño elaborado a código, es decir, la codificación. En otras palabras, la programación es el proceso de diseñar, codificar, depurar y mantener el código fuente de un software. El código fuente normalmente debe estar escrito en un lenguaje de programación determinado. El propósito de la programación es crear programas que exhiban un comportamiento deseado. El proceso de escribir código requiere frecuentemente conocimientos en varias áreas distintas, además del dominio del lenguaje a utilizar, algoritmos especializados y lógica formal.
- **Prueba:** la prueba es un proceso de ejecución de un programa con la intención de descubrir un error. Consiste en comprobar que el software realice correctamente las tareas indicadas en la especificación del

problema. Es muy difícil realizar pruebas ideales para el testeo de un software, así que es muy importante la planificación de las mismas para obtener mejores resultados posibles. En general hay dos grandes formas de organizar un área de pruebas. La primera es que esté compuesta por personal inexperto y que desconozca el tema de pruebas, de esta forma se evalúa que la documentación entregada sea de calidad, que los procesos descritos sean tan claros que cualquiera pueda entenderlos y el software haga las cosas tal y como están descritas. El segundo enfoque es tener un área de pruebas conformada por programadores con experiencia, personas que saben sin mayores indicaciones en qué condiciones puede fallar una aplicación y que pueden poner atención en detalles que personal inexperto no consideraría.

Cabe destacar el hecho de que el sistema, complejo por el número de componentes hardware que integra, va a ser usada en el campo por usuarios que no son expertos en informática. Por ello es importante que la validación del sistema así como su operación sean robustas. En este apartado se contarán los aspectos en rasgos generales, haciendo hincapié solo en aquellos aspectos más importantes durante el desarrollo del TFG. Básicamente, durante esta etapa las referencias utilizadas para el correcto desarrollo de las mismas han sido [1], [2], [9], [11], [12], [13], [16], [18], [19] y [20]. A continuación se muestra como han sido estas dos etapas en las dos partes en las que se ha dividido el sistema.

6.1.1 Construcción de la Interfaz Gráfica

La **implementación** de la interfaz gráfica se puede separar en dos bloques, uno orientado a la interfaz gráfica pura, escrita en Java, y otro relacionado con la API de Google Maps, donde se encuentran tipos de ficheros que son usados en aplicaciones web: HTML, CSS, PHP y JavaScript.

El código de la interfaz gráfica escrita en Java está dividida en varias partes (directorios) principales, que son las siguientes:

- ❖ **estacion_total**: este directorio tiene las clases que son usadas para manejar los mensajes recibidos de la estación total, además de realizar

los cálculos para obtener las coordenadas exactas, el ángulo α y otros parámetros que pueden verse en el apartado de '*Análisis*' de este mismo capítulo.

- ❖ **images:** contiene las imágenes que son usadas en la interfaz gráfica.
- ❖ **seguimientorfid:** las clases que implementan la propia interfaz gráfica están alojadas en este directorio. Estas clases son las encargadas de cubrir las necesidades expuestas en los casos de uso que aparecen en el apartado de '*Análisis*' de este mismo capítulo.
- ❖ **serialport:** aquí están las clases necesarias para la comunicación por USB, de la interfaz gráfica con la estación total y el Waspote Base.
- ❖ **util:** en esta carpeta se encuentran clases complementarias para el desarrollo de la interfaz gráfica, como el manejo de ficheros XML o HTML, paso de coordenadas UTM a decimales y viceversa (utilizando el algoritmo de la referencia [16]), etc.

Los ficheros que componen el manejo de la API de Google Maps se encuentran en el directorio específico que es usado por el servidor XAMPP para el alojamiento de las distintas aplicaciones Web desarrolladas. Este directorio está compuesto por:

- ❖ **Dos directorios**, '*css*' y '*js*', que contienen los ficheros de estilos y de scripts respectivamente, necesarios para el uso de jQuery UI. Obtenido principalmente de [19] y [20].
- ❖ Un fichero de estilo, **estilo.css**, encargado de dar formato a los componentes de la página que contiene la API de Google Maps.
- ❖ La versión de **jQuery, 1.8.2** para el uso de JavaScript y AJAX, obtenida de [19].
- ❖ Un fichero PHP, **refresh.php**, cuya función es actualizar la API de Google Maps con las detecciones de las marcas RFID en tiempo real, a través del uso de AJAX.
- ❖ Y finalmente un fichero HTML, **index.html**, que es la página principal que contiene la API de Google Maps, con el contenido necesario para el manejo de la misma. Este fichero lo iré actualizando la interfaz gráfica durante la ejecución de esta, para mostrar el contenido correspondiente en cada momento.

Las **pruebas** más significativas de la interfaz gráfica son las realizadas para el correcto funcionamiento de la API de Google Maps, haciéndolas directamente sobre un navegador, en concreto, Google Chrome. Para ello debía tenerse activados el servidor Apache y MySQL del XAMPP previamente, luego en la barra de direcciones del navegador bastaba con escribir “localhost/NombreDirectorioEspecificoEnHTDOCS”.

En la siguiente imagen (Fig. 19) puede verse como ha quedado la interfaz gráfica una vez finalizada:

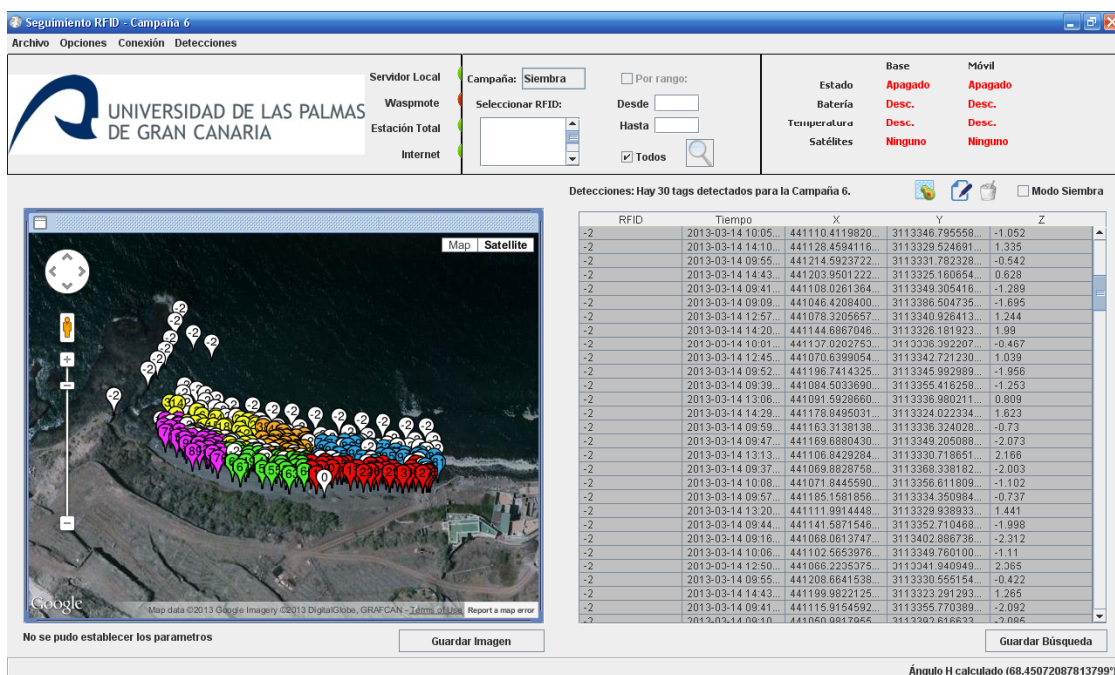


Fig. 19 – Imagen de la interfaz gráfica de usuario

6.1.2 Construcción de las aplicaciones Waspnote

Para la parte de **implementación** de las aplicaciones Waspnote, mencionar que para esta parte del sistema cuenta con dos aplicaciones, una para el Waspnote Rover y otra para el Waspnote Base.

Cada una de ellas está compuesta por una serie de ficheros, uno para las declaraciones con extensión “.h” y el resto con extensión “.pde” (Processing Source Code). Ahora se verá cuales son los ficheros que forman cada una de las aplicaciones por separado.

▪ **Aplicación Wasmote Base**

La aplicación del Wasmote Rover, que se encarga de recibir las detecciones del Rover y enviárselas a la interfaz gráfica, es la siguiente:

- ❖ **rfid_fijo.h** → En este fichero se establecen todas las declaraciones a nivel global de la aplicación.
- ❖ **rfid_fijo.pde** → En este fichero es el principal de la aplicación, aquí es donde se inicia la aplicación, se puede decir que es el método 'main' de la misma.
- ❖ **rfid_fijo_comm.pde** → Este módulo se encarga de la comunicación por radio con el Wasmote Rover.
- ❖ **rfid_fijo_gps.pde** → Aquí se maneja todo lo relacionado con el GPS del Wasmote en cuestión.
- ❖ **rfid_fijo_message.pde** → Aquí se encuentra el código necesario para formar los distintos mensajes que se van intercambiando a lo largo de la ejecución del sistema de monitorización durante el estudio.
- ❖ **rfid_fijo_sd.pde** → Este módulo se encarga de manejar la tarjeta SD del dispositivo Wasmote para almacenar los mensajes intercambiados durante la monitorización.
- ❖ **rfid_fijo_status.pde** → Código necesario para comprobar distintos parámetro del Wasmote, como puede ser la temperatura o la batería.
- ❖ **rfid_fijo_zumbador.pde** → Este fichero contiene todo lo relacionado con el zumbador del Wasmote, utilizado para avisar al usuario de los distintos acontecimientos a lo largo de la monitorización.

▪ **Aplicación Wasmote Rover**

La aplicación del Wasmote Rover, que se encarga de detectar las distintas marcas RFID que se encuentran a lo largo de la playa, es la que aparece a continuación:

- ❖ **rfid_movil.h** → En este fichero se establecen todas las declaraciones a nivel global de la aplicación.

- ❖ **rfid_movil.pde** → En este fichero es el principal de la aplicación, aquí es donde se inicia la aplicación, se puede decir que es el método 'main' de la misma.
- ❖ **rfid_movil_comm.pde** → Este módulo se encarga de la comunicación por radio con el Wasmote Base.
- ❖ **rfid_movil_gps.pde** → Aquí se maneja todo lo relacionado con el GPS del Wasmote en cuestión.
- ❖ **rfid_movil_message.pde** → Aquí se encuentra el código necesario para formar los distintos mensajes que se van intercambiando a lo largo de la ejecución del sistema de monitorización durante el estudio.
- ❖ **rfid_movil_sd.pde** → Este módulo se encarga de manejar la tarjeta SD del dispositivo Wasmote para almacenar los mensajes intercambiados durante la monitorización.
- ❖ **rfid_movil_status.pde** → Código necesario para comprobar distintos parámetro del Wasmote, como puede ser la temperatura o la batería.
- ❖ **rfid_movil_zumbador.pde** → Este fichero contiene todo lo relacionado con el zumbador del Wasmote, utilizado para avisar al usuario de los distintos acontecimientos a lo largo de la monitorización.

Con respecto a las **pruebas**, las aplicaciones Wasmote han sido las primeras que han pasado por esta etapa. Probándose cada módulo por separado antes de probar el conjunto entero, es decir, probar el código destinado al manejo del GPS, de la tarjeta de memoria SD, del módulo de radio, etc. Las pruebas de la interfaz gráfica de usuario serían posteriores a esta, debido a que se consideraba primordial tener unas aplicaciones Wasmote robustas, dejándolas listas para centrarse de lleno en la interfaz gráfica.

Mencionar que durante estas pruebas fue cuando se decidió optar por la estación total para la obtención de las coordenadas geográficas de las marcas RFID. Se probó usando la API de Google Maps por separado, evaluando la precisión de los datos obtenidos. En la siguiente figura se muestra un ejemplo de la baja precisión de los mismos, aquí las coordenadas deberían estar sobre la acera que rodea el aparcamiento que aparece en la Fig. 20.

Ya haciendo uso de la estación total los resultados fueron muy buenos, a pesar de no haber un recorrido como en la imagen anterior, se hicieron las pruebas en sitios estratégicos, fácilmente detectables en el mapa para comprobar la exactitud, comprobando que efectivamente los resultados eran bastante precisos.



Fig. 20 – Ejemplo de la precisión de los datos obtenidos por el GPS de los Waspnote

Por último, se añaden una serie de imágenes en las que se puede ver el sistema en funcionamiento durante la realización de las campañas.

En la primera imagen (Fig. 21) se puede apreciar la estación total, que aparece a la derecha de la imagen situada encima de un trípode, y a la izquierda de ésta están situados el ordenador portátil y el Waspnote Base sobre una mesa.

En la segunda imagen (Fig. 22), la persona que está a la izquierda está sujetando un prisma, que es la herramienta necesaria para hacer las mediciones con la estación total. Y la persona de la derecha transporta la parte del sistema encargada de realizar las detecciones de las marcas RFID, en concreto, el Waspnote Rover y el lector RFID con su correspondiente alimentación y su antena.

La última (Fig. 23) muestra el ordenador portátil en la que se está ejecutando la IU, y la estación total, donde la persona que se ve en la imagen está realizando una medición con ella.



Fig. 21 – Sistema en funcionamiento 1



Fig. 22 – Sistema en funcionamiento 2



Fig. 23 – Sistema en funcionamiento 3

7. Conclusión

En este capítulo se concreta, a modo de resumen, los objetivos conseguidos con el desarrollo del presente proyecto. Asimismo, se indican sucintamente algunas líneas de actuación que permitirán mejorar algunos de los aspectos desarrollados.

7.1 Conclusión del trabajo realizado

El trabajo de fin de grado tiene por objetivo la elaboración de un proyecto informático aplicando los conocimientos adquiridos a lo largo de las asignaturas del Grado en Ingeniería Informática.

Cuando se inició este trabajo de fin de grado se plantearon una serie de objetivos ligados al prototipo que se quería desarrollar, siendo el objetivo fundamental el dar soporte a las necesidades de los usuarios finales del sistema que plantearon unos requerimientos claros en cuanto a la funcionalidad y robustez del sistema.

Para la consecución de este objetivo era necesario la creación de dos subsistemas que trabajaran en conjunto, por un lado estaba la aplicación de los dispositivos Waspnote, cuyo objetivo es la detección de las marcas RFID, y por el otro la interfaz gráfica de usuario, que debe sincronizar y controlar todos los dispositivos participantes en el sistema, esto incluye los dispositivos Waspnote y la estación total, y además de tratar toda la información obtenida por estos, con el fin de monitorizar, almacenar y representarla.

Dichos subsistemas han sido desarrollados, tal y como se refleja a lo largo del capítulo de desarrollo y con el material entregado junto con esta memoria. El resultado finalmente obtenido ha sido evaluado por los usuarios como satisfactorio, quedando reflejado en el uso intenso del sistema como herramienta de campo en no menos de seis campañas.

Los conocimientos técnicos adquiridos, mejorados y/o empleados en este proyecto comprenden varias áreas de conocimiento: la programación de sistemas empujables (Waspnote), la programación en el lenguaje Java, la

programación web, el análisis y diseño de base de datos y la ingeniería del software.

En definitiva, el desarrollo de este proyecto ha significado una muy buena experiencia a nivel personal y profesional, donde he aprendido muchísimas cosas, tanto por mi cuenta, como por parte de mi tutor, Jorge Cabrera Gámez, y los destinatarios del sistema desarrollado, Mariona Casamayor Font e Ignacio Alonso Bilbao (cotutor), agradeciéndoles enormemente el trato y la ayuda recibido por ellos.

7.2 Líneas futuras

En este apartado se exponen las posibles funcionalidades o mejoras que se le pueden añadir al presente trabajo, pero que por falta de tiempo no ha sido posible implementar o que por la experiencia del mismo se ha decidido aplicar en un futuro.

- **Mejorar la monitorización de los estados de los Waspote**

En la interfaz gráfica hay un sector, donde se debe mostrar el estado de los Waspote durante la monitorización, situado en la parte superior derecha (puede verse en la Fig. 17). La funcionalidad de este bloque informativo no está implementada completamente, por lo que una de las tareas a realizar en un futuro es la de terminar dicha implementación, que no ha podido ser completada debido a la falta de tiempo disponible para ello, ya habiéndose empleadas las 300 horas previstas.

- **Extensión de aplicación del sistema**

En consecuencia de los buenos resultados obtenidos, se pretende extender el uso del sistema en otras zonas de estudio, así como incluir más de un Waspote de tipo Rover en el sistema para abordar zonas más amplias.

- **Representar el recorrido de los cantos en la API de Google Maps**

En este momento, la interfaz gráfica de usuario es solo capaz de mostrar las coordenadas geográficas por campañas, es decir, que no puede mostrar al

usuario cual ha sido el movimiento de los cantos cronológicamente en una misma pantalla.

Evidentemente una mejora para el sistema sería poder representar las distintas posiciones de un solo canto distinguiendo cuales han sido sus posiciones a lo largo del estudio de la dispersión, enumerándolas desde la más antigua a la más reciente, por ejemplo, la primera detección del canto se marcaría con el número 1 y la más reciente con el número n , correspondiente a la cantidad de detecciones que ha habido de dicho canto, además de pintar sobre el mapa el recorrido de dicho éste.

8. Referencias

Finalmente se citan las fuentes que han sido de ayuda para poder realizar este trabajo de fin de grado. Sin ellas el tiempo necesario para llevarlo a cabo habría sido notablemente superior.

[1] Manuales Waspote

[2] Electronic Total Station GTS -750, Interface Manual (GTS75080401).
Topcon Corporation 2007

[3] Roger S. Pressman, “Ingeniería del Software: Un enfoque práctico”, 6ª Edición, Editorial McGraw Hill, 2005

[4] Apuntes de Ingeniería del Software 1 y 2, de la ULPGC

[5] <http://www.wikipedia.es> (Última consulta: 24/06/2013)

[6] <http://www.libelium.com> (Última consulta: 20/06/2013)

[7] <http://www.topcon-positioning.eu/75/76/products.html> (Última consulta: 11/07/2013)

[8] <http://www.topotienda.com> (Última consulta: 17/06/2013)

[9] <http://developers.google.com/maps> (Última consulta: 11/07/2013)

[10] <http://www.dorcus.co.uk/carabus//jscalculators.html> (Última consulta: 22/02/2013)

[11] <http://www.stackoverflow.com> (Última consulta: 05/03/2013)

[12] <http://www.w3schools.com/> (Última consulta: 20/02/2013)

[13] <http://www.arduino.cc/es> (Última consulta: 13/01/2013)

[14] <http://unpocodejava.wordpress.com/2012/08/21/que-es-waspote/>
(Última consulta: 20/06/2013)

[15] <http://www.aragoninvestiga.org/Presentacion-de-Waspmote-nueva-plataforma-para-Redes-Sensoriales-Inalambricas> (Última consulta: 20/06/2013)

[16] <http://www.gabrielortiz.com/index.asp?Info=058a> (Última consulta: 10/03/2013)

[17] <http://www.pdaexpertos.com/foros/viewtopic.php?t=48484> (Última consulta: 27/06/2013)

[18] <http://docs.oracle.com/javase/7/docs/api> (Última consulta: 15/02/2013)

[19] <http://jquery.com> (Última consulta: 20/02/2013)

[20] <http://jqueryui.com> (Última consulta: 20/02/2013)

[21] <http://casosdeuso.com/2134/diagramas-de-casos-de-uso-concepto-y-ejemplos/> (Última consulta: 21/08/2012, link no disponible a 15/07/2013)

Apéndice: Manual de usuario del sistema

En este apéndice se expone un resumido y sencillo manual de la aplicación, donde se muestran ejemplos con capturas de pantalla para facilitar la comprensión del funcionamiento del mismo.

El sistema descrito en este Trabajo de Fin de Grado tiene como finalidad monitorizar las marcas RFID que se encuentran depositadas en una serie de cantos que se colocarán a lo largo de una playa, con la finalidad de estudiar la dispersión de los mismos.

Este manual explica tanto el proceso de preparación e instalación como el manejo del sistema, a continuación se verá detalladamente cada uno de estos aspectos, además se debe tener presente que se cuenta con el material proporcionado con todo el software necesario para el funcionamiento del sistema.

Preparación e instalación

Para el correcto funcionamiento del sistema, los distintos componentes hardware deben estar configurados como se cita a continuación:

- **Estación Total**

La estación total debe estar en su modo NEZ para realizar las campañas, este modo obtiene las medidas dando las coordenadas Norte (N), Este (E) y Altura (Z), con respecto a unos valores iniciales que deben establecerse con anterioridad. Estos valores iniciales deben ser exactamente las coordenadas en las que se posicionará la estación durante la monitorización.

- **Dispositivos Waspote**

Los dispositivos deben tener acoplados los programas correspondientes, para ello se hará uso del IDE propio del Waspote. Además al Waspote Rover se le debe conectar la antena de detección RFID.

- **Portátil / Netbook**

En el caso del ordenador, donde estará alojada la interfaz gráfica, es mucho más complejo. En primer lugar, se debe instalar el servidor XAMPP, para ello se ha descargado el archivo necesario desde la página oficial de XAMPP.

Tras la descarga procedemos a la instalación ejecutando el archivo que hemos descargado. Seguiremos los pasos de instalación normales y tendremos el XAMPP listo para usar.

Una vez finalizada la instalación del XAMPP se tiene que poner en marcha los dos servicios necesarios para que el sistema se ejecute correctamente: Apache y MySQL. Para arrancarlos tenemos que irnos al menú principal del XAMPP (Fig. M1) y hacer click en 'Start' en cada uno de ellos. Si se arrancan de forma correcta ambos servicios deben mostrar a la derecha de su nombre el texto 'Running' en un fondo verde, tal y como se ve en la Fig. M1. Mientras no esté de esta forma no se podrá hacer uso del sistema correctamente.

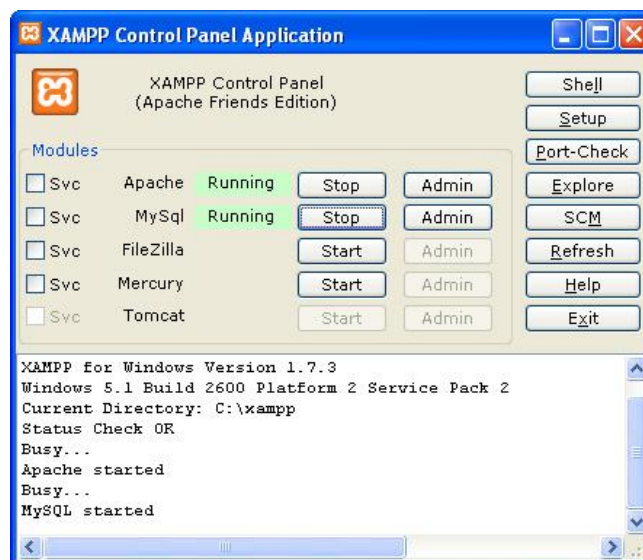


Fig. M1 – Menú XAMPP

Tras haber realizado estos pasos debemos colocar dentro del directorio, que tienes configurado el XAMPP para almacenar las distintas aplicaciones Web, para copiar y pegar la carpeta '*rfid*' que se adjunta con el material proporcionado. También se debe crear desde el 'phpmyadmin' una base de datos con el nombre 'seguimientorfid' e importar el fichero SQL que se proporciona con el material entregado que tiene por nombre el mismo que el de la base de datos.

El siguiente paso es alojar la carpeta con el nombre '**Seguimiento RFID**' en la ubicación que se prefiera del ordenador que se vaya a emplear para la monitorización. Para ejecutar la interfaz gráfica es imprescindible tener

instalado también el JRE (Java Runtime Environment), que es un conjunto de utilidades que permite la ejecución de programas Java, en concreto debe tenerse instalado la versión 7, que fue la utilizada para el desarrollo de la interfaz (este software puede obtenerse de la página oficial de Oracle). Además debemos seleccionar dos archivos: 'rxtxSerial.dll' y 'seguimientorfidNuevo.xml'; el primero de ellos es necesario para poder establecer la comunicación con los Waspote y la estación total, y el segundo es el fichero de configuración de la interfaz. Una vez seleccionados debemos copiar y pegarlos en la carpeta 'bin' del JRE que se haya instalado.

Finalmente debemos crear un fichero '.bat' con las siguientes instrucciones para permitir la ejecución del programa.

```
1 cd C:\Program Files\Java\jre7\bin
2 java -jar C:\SeguimientoRFID\SeguimientoRFID.jar
```

El código que aparece aquí es meramente orientativo, la primera línea tiene como objetivo cambiar de directorio para situarse en la carpeta bin del JRE que se tiene instalado en el equipo, así que esta ruta dependerá de donde se tenga instalado este software. Y la segunda realiza la ejecución de la interfaz haciendo uso del propio JRE, así que la ruta dependerá de donde este alojado el ejecutable de la interfaz gráfica en el equipo.

Habiendo seguido todos los pasos anteriores la interfaz gráfica de usuario está lista para usarse.

Manual de Uso

Cuando se ejecuta la interfaz gráfica la primera pantalla que nos aparece es un diálogo como el que se muestra en la Fig. M2, aquí se establecen los parámetros necesarios para configurar la estación total: las coordenadas geográficas y la altura de la posición de la estación total y del punto de referencia (necesario para el cálculo del ángulo α), la altura que tiene la plataforma que hace de soporte de la estación total y del prisma, que es un jalón con el que se hace las medidas oportunas, y además el ángulo α .

Establecer Parámetros Estación Total

Se va a iniciar el programa con los parámetros siguientes:

Posición de la base y de la referencia:

X: 441146.75 Xref: 441020.0
 Y: 3113314.69 Yref: 3113364.81
 Z: 5.0 Zref: 3.0

Ángulo H: 1.194690454687132 radianes

Altura de la estación y del prisma:

Prisma: 2.0 Estación: 1.425

Modificar No Modificar

Fig. M2 – Diálogo de configuración de los parámetros de la estación total

Si se quieren conservar los parámetros, que aparecen en el diálogo, durante la monitorización se debe hacer click en ‘No Modificar’; en caso contrario se establecen los parámetros oportunos correctamente y se hace click en ‘Modificar’. Tras elegir la opción deseada se muestra la pantalla principal de la interfaz gráfica, esta pantalla se muestra en la Fig. M3.

Seguimiento RFID - Campaña 6

Archivo Opciones Conexión Detecciones

UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA

Servidor Local Waspnote Estación Total Internet

Campaña: Siembra

Por rango: Desde Hasta

Selecciónar RFID: [dropdown]

Desde Hasta

Estados: Base (Apagado, Desc.), Móvil (Apagado, Desc.), Temperatura (Desc., Desc.), Satélites (Ninguno, Ninguno)

Detecciones: Hay 30 tags detectados para la Campaña 6.

RFID	Tiempo	X	Y	Z
-2	2013-03-14 10:05...	441110.4119820...	3113346.795558...	-1.052
-2	2013-03-14 14:10...	441128.4594116...	3113329.524691...	1.335
-2	2013-03-14 09:55...	441214.5923722...	3113331.782328...	-0.542
-2	2013-03-14 14:43...	441203.9501222...	3113325.160654...	0.628
-2	2013-03-14 09:41...	441108.0261364...	3113349.305416...	-1.289
-2	2013-03-14 09:09...	441046.4208400...	3113386.504735...	-1.695
-2	2013-03-14 12:57...	441078.3205657...	3113340.926413...	1.244
-2	2013-03-14 14:20...	441144.6867046...	3113326.181923...	1.99
-2	2013-03-14 10:01...	441137.0202753...	3113336.392207...	-0.467
-2	2013-03-14 12:45...	441070.6399054...	3113342.721230...	1.039
-2	2013-03-14 09:52...	441196.7414325...	3113345.992989...	-1.956
-2	2013-03-14 09:39...	441084.5033690...	3113355.416258...	-1.253
-2	2013-03-14 13:06...	441091.5928660...	3113336.980211...	0.809
-2	2013-03-14 14:29...	441178.8495031...	3113324.022334...	1.623
-2	2013-03-14 09:59...	441163.3138138...	3113336.324028...	-0.73
-2	2013-03-14 09:47...	441169.6880430...	3113349.205088...	-2.073
-2	2013-03-14 13:13...	441106.8429284...	3113330.718651...	2.166
-2	2013-03-14 09:37...	441069.8828758...	3113368.338182...	-2.003
-2	2013-03-14 10:08...	441071.8445590...	3113356.811809...	-1.102
-2	2013-03-14 09:57...	441185.1581856...	3113334.350984...	-0.737
-2	2013-03-14 13:20...	441111.9914448...	3113329.938933...	1.441
-2	2013-03-14 09:44...	441141.5871546...	3113352.710468...	-1.998
-2	2013-03-14 09:16...	441068.0613747...	3113402.886736...	-2.312
-2	2013-03-14 10:06...	441102.5653976...	3113349.760100...	-1.11
-2	2013-03-14 12:50...	441068.2235375...	3113341.940949...	2.365
-2	2013-03-14 09:55...	441208.6641538...	3113330.555154...	-0.422
-2	2013-03-14 14:43...	441199.9822125...	3113323.291293...	1.265
-2	2013-03-14 09:41...	441115.9154592...	3113355.770389...	-2.092
-2	2013-03-14 09:10...	441060.9817056...	3113303.646632...	-2.066

Map | Satellite

No se pudo establecer los parametros Guardar Imagen

Guardar Búsqueda

Ángulo H calculado (68.45072087813799°)

Fig. M3 – Pantalla principal de la IU

Las acciones que se pueden realizar con la interfaz gráfica se dividirán en dos partes: una con las acciones que se pueden realizar durante el proceso de monitorización y otra con las que se puede realizar durante el postproceso.

■ Monitorización

Durante la monitorización se puede realizar diferentes acciones. Por ejemplo, previo a la monitorización se puede **establecer** nuevamente el **ángulo α** , para ello se debe ir a ‘Detecciones → Calcular Ángulo H’ de el menú superior de la pantalla principal de la IU (Fig. M3), así se deja preparado para que la siguiente medición por parte de la estación total sirva para el cálculo de dicho ángulo. En caso de que el usuario se haya equivocado y no quiere realizar este cálculo de ir a la misma ruta (‘Detecciones → Calcular Ángulo H’) cancelando así el proceso. En la Fig. M4 se puede ver exactamente donde se debe hacer click para realizar esta acción.

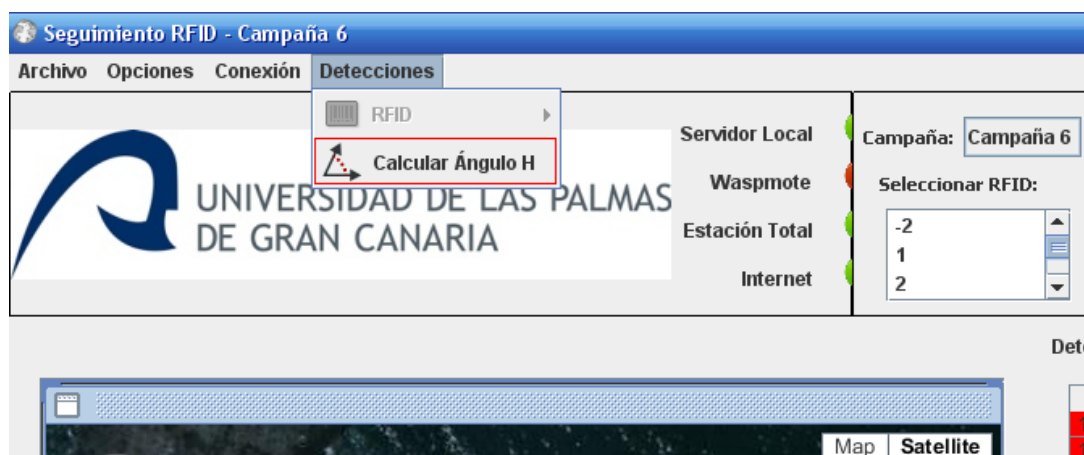


Fig. M4 – Establecer Ángulo α

Además si se selecciona la opción de establecer el ángulo α , deberá aparecer el texto “A la espera de calcular el ángulo H, haga la medida con la estación total” en la barra inferior de la pantalla principal de IU.

Otra de las acciones que puede realizarse es **realizar una medida topográfica**, aquí es necesario apretar el botón con la imagen siguiente:



Antes de que se añada el registro en la tabla para proceder a la medición topográfica, se le pide al usuario que confirme la misma, con un diálogo como el de la Fig. M5.

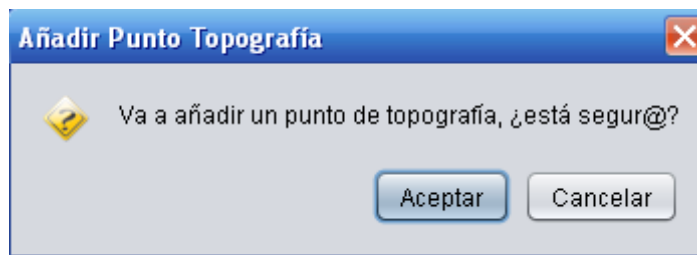


Fig. M5 – Confirmación punto topográfico

Si el usuario finaliza por confirmar la acción se debe realizar la medición con la estación total para completar el proceso.

El usuario también tiene posibilidad de **editar** los distintos **registros** que aparecen en la tabla de detecciones, restringiendo la edición al rfid del registro; si se desea realizar esta acción el usuario debe seleccionar previamente el registro que quiere modificar y posteriormente hacer click en el botón que tiene como imagen:



Una vez que el usuario selecciona el botón le aparece el diálogo de la Fig. M6, aquí establece el número de RFID correspondiente y se hace click en 'Aceptar'.



Fig. M6 – Diálogo para editar registro

Si la edición se produce satisfactoriamente, la IU se encarga de mostrar un mensaje informado al usuario que la edición del registro con el número de RFID que ha establecido se ha realizado correctamente. En la Fig. M7 se muestra dicho diálogo informativo.

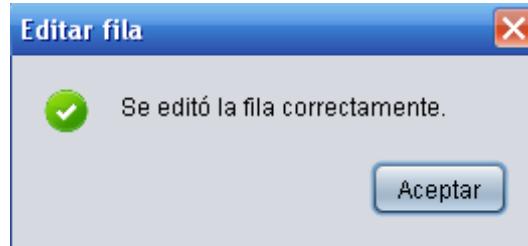


Fig. M7 – Diálogo de edición correcta

Además de editar los registros, éstos también pueden ser eliminado, el proceso es exactamente igual, se selecciona la fila correspondiente, se elige la opción de eliminar, habiendo un dialogo de eliminado posterior y el informe de que se ha realizado correctamente; la única diferencia es que el botón destinado para la eliminación de los registros es el siguiente:



Otro de los aspectos que se puede tratar durante la monitorización, es **realizar la siembra**, esta es la etapa del estudio de dispersión en que se coloca los cantos a lo largo de la playa. Dicha etapa se tratará como una campaña independiente. Para establecer este modo hay que seleccionar el *CheckBox* titulado 'Modo Siembra' que se encuentra a la derecha de los botones de edición y eliminación de registros vistos anteriormente. Cuando se activa éste se muestra al usuario una entrada de texto donde deberá escribir el número de RFID que se va a sembrar en ese momento, esta entrada de texto puede verse en la Fig. M8.

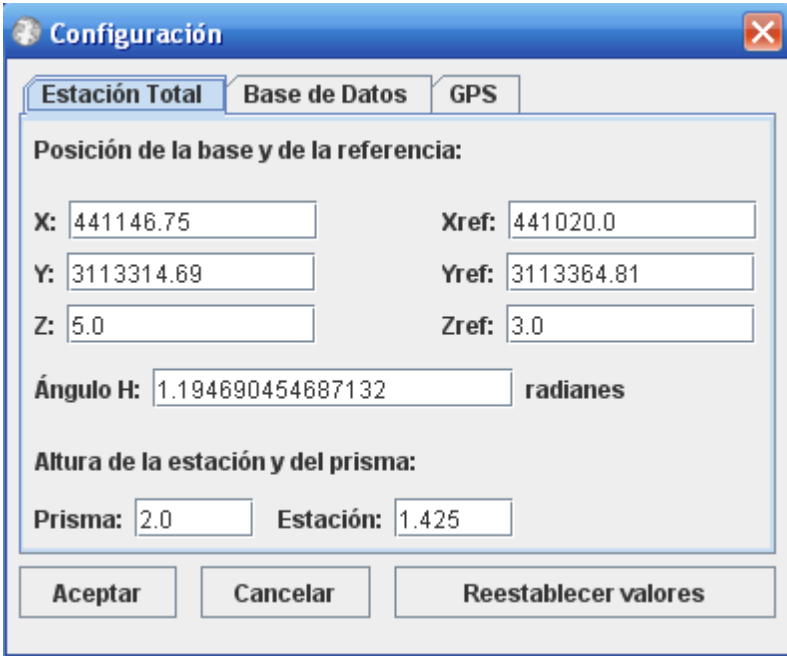


Fig. M8 – Entrada de texto del modo siembra

Si se quiere realizar una medida topográfica debe introducirse el número '-2'. La interfaz gráfica trabaja de tal manera que cuando recibe una medida de la estación total crea un registro con las coordenadas obtenidas y como rfid el número que se haya puesto en la entrada de texto. En caso de que no se quiere

sembrar una piedra, y lo que quiera hacerse sea una medida topográfica debe introducirse el número '-2'.

Por último el usuario puede realizar la configuración de los parámetros de la estación total, de la base de datos y del GPS (estos últimos tienen que ver con las opciones de los Wasmotes, pero que han quedado obsoletas debido al desuso provocado por los malos resultados obtenidos). El usuario deberá ir a 'Opciones → Configuración' para proceder a la configuración, entonces se le mostrará una pantalla como la de la Fig. M9, aquí el usuario pondrá los parámetros como considere oportuno y finalizará haciendo click en 'Aceptar'.



The screenshot shows a window titled 'Configuración' with three tabs: 'Estación Total', 'Base de Datos', and 'GPS'. The 'Estación Total' tab is active. The window contains the following fields and labels:

- Posición de la base y de la referencia:**
 - X: 441146.75
 - Y: 3113314.69
 - Z: 5.0
 - Xref: 441020.0
 - Yref: 3113364.81
 - Zref: 3.0
- Ángulo H:** 1.194690454687132 radianes
- Altura de la estación y del prisma:**
 - Prisma: 2.0
 - Estación: 1.425

At the bottom of the window are three buttons: 'Aceptar', 'Cancelar', and 'Reestablecer valores'.

Fig. M9 – Ventana de configuración

■ Postproceso

En el postproceso las operaciones principales que se pueden realizar son: realizar informes y realizar búsquedas. Para **realizar informes** hay dos opciones posibles: obtener una imagen del mapa, en formato JPEG, que se está visualizando en la interfaz u obtener un fichero en formato EXCEL con la tabla que aparece a la derecha del mapa (Fig. M3) con la información de las detecciones y de los puntos topográficos resultantes de la realización de las campañas.

En caso de que se quiere **obtener la imagen del mapa** en cuestión, se debe hacer click sobre el botón que se encuentra debajo del mapa (Fig. M10) con el texto *'Guardar Imagen'*. Tras haber hecho click en este botón aparecerá un diálogo para guardar la imagen como el que sale en la Fig. M10, aquí se selecciona el directorio deseado y se establece un nombre para la imagen, una vez elegido se hace click en *'Guardar'*.

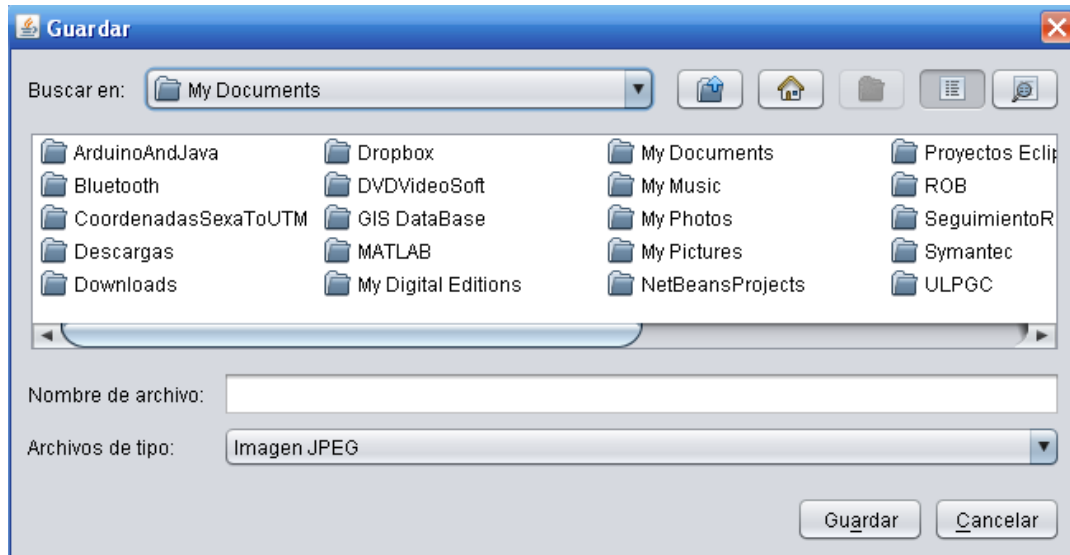


Fig. M10 – Diálogo para guardar los informes

Si el proceso de guardado se ha realizado correctamente, se nos informa con un dialogo como el de la imagen siguiente (Fig. M11).

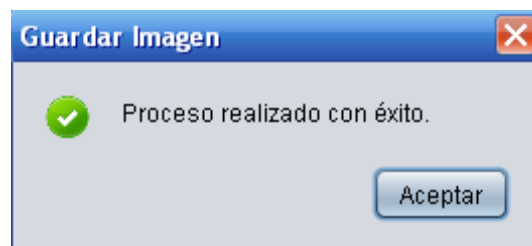


Fig. M11 – Diálogo de guardado correcto

El proceso para **obtener el extracto de la tabla de detecciones** en formato EXCEL es igual que el anterior, la única diferencia consiste en que para empezar el proceso se debe hacer click en el botón que se encuentra debajo de la tabla de detecciones (Fig. M3) que tiene por nombre *'Guardar Búsqueda'*.

Si lo que se desea es **realizar una búsqueda**, debe atenderse al menú que aparece en la Fig. M12, que es una porción de la pantalla principal de la interfaz, que se encuentra en la parte superior central de la misma.

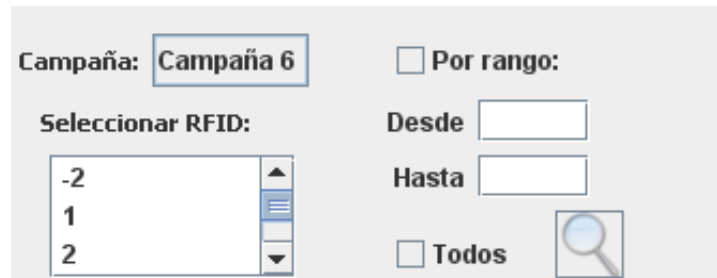


Fig. M12 – Menú de búsqueda de la IU

Las búsquedas están orientadas a las campañas y a las marcas RFID. Para las campañas hay un *SelectBox* donde se seleccionará una de las campañas que se hayan realizado, dando la posibilidad también de que se busque en todas las campañas existentes. En cuanto a las marcas RFID hay tres opciones:

- Todas las marcas RFID. Se debe seleccionar el *CheckBox* titulado '*Todos*'.
- Por rango, las marcas están numeradas de 1 a n, entonces el rango debe ser un conjunto dentro de este intervalo. Se selecciona el *CheckBox* titulado '*Por rango*' y se establece el rango con los números deseados.
- Por la selección de una serie de marcas concretas, se hace click en el RFID del que se quiere realizar la búsqueda, y en caso de que sea más de una marca, deberemos ir seleccionándolas mientras se mantiene pulsado el botón 'Ctrl' del teclado.

Una vez se hayan establecido los parámetros se debe apretar el botón sobre el que aparece una imagen de una lupa.