

▪ Proyecto Fin de Grado ▪

Ingeniería del Software

MapCar: Prototipo de vehículo dirigido con capacidad de captar datos del entorno.

Gonzalo del Palacio Jiménez

Julio 2016

Agradecimientos

Primero y como más importante, me gustaría agradecer sinceramente a mi asesor de PFG, Sr. Alfredo Goñi, su esfuerzo y dedicación.

Por otra parte, y no menos importante, querría hacer notorio y explícito este agradecimiento a las siguientes personas, que si bien en desigual esfuerzo, contribuyeron en todo caso a hacer posible el alumbramiento del proyecto en un inicio, y su progreso a lo largo de todo un cuatrimestre de trabajo.

A mis padres, mi hermana y amigos más cercanos, por haberme dado importantes lecciones y por haberme apoyado en todas las decisiones tomadas a lo largo de mi vida.

A mis tías Blanca y Raquel por el apoyo tanto económico como moral que me han brindado a lo largo de la carrera y de mi vida.

A Erik Fustes, en calidad de asesor en alguna de las partes del coche, sin cuya voluntad nada habría sido de esta forma.

A Luis Marcos Rivera, en calidad de asesor de software para la creación de mapas dinámicos.

Quiero también dar las gracias a todos los profesores de la UPV/EHU de Donostia, por los conocimientos que me han transmitido y las capacidades que me han ayudado a desarrollar, tanto personales como profesionales, durante estos años.

En este proyecto de fin de grado se pretende desarrollar un prototipo de un vehículo que permita captar y manipular cierta información del entorno. Al tratarse de un proyecto de la especialidad ingeniería de software, haremos más énfasis en la parte software del mismo, pero también mostraremos algunos detalles relevantes relacionados con el hardware necesario para que el software pueda desarrollarse y ejecutarse.

El software para controlar el prototipo, se implementará tanto en dispositivos Android como en Ordenadores con Sistema Operativos capaces de ejecutar programas .msi (principalmente Windows).

Dado que no disponemos de un robot captador de datos acorde a las necesidades, se ha modificado un coche radio control mediante el uso de Arduino. Permitiendo así implementar módulos y funcionalidades necesarios para la culminación del proyecto.

La información recopilada se procesa mientras el coche realiza su recorrido formando un mapa dinámico, que, se puede exportar y observar mediante un visualizador de GPS externo a la aplicación.

Para llegar a ello, se estudian los diversos sistemas de comunicación disponibles para el envío de datos y el control del vehículo, y se enumeran varios módulos de captación de datos.

Palabras clave: software de captación de datos, Arduino, Android, robot captador de datos.

1	Introducción	1
1.1	Antecedentes.....	1
1.2	Motivación.....	1
1.3	Tecnologías y herramientas	2
1.3.1	Arduino	2
1.3.2	ArduinoIDE.....	3
1.3.3	Android Studio.....	4
1.3.4	MIT AppInventor.....	5
1.3.5	Visual Studio 2013	6
1.3.6	GMap.Net	7
1.3.7	XInput.NET	7
1.3.8	TinyGPS	8
1.3.9	WIFIESP	8
1.3.10	Sistema de comunicación	9
1.3.11	Sistemas de recolección de datos	13
1.3.12	Elección de Tecnologías de comunicación y de captación de datos.....	17
1.3.13	Sistemas de control.....	17
1.4	Metodología.....	18
1.4.1	Metodología Iterativa.....	18
2	Planificación.....	18
2.1	Alcance.....	19
2.1.1	Descripción de los objetivos concretos del proyecto	19
2.1.2	Restricciones del proyecto	20
2.1.3	WBS/EDT.....	20
2.1.4	Riesgos.....	20
2.2	Períodos de realización de las tareas.....	21
2.2.1	Relacionados con el objeto de proyecto en sí.....	21
2.2.2	Dependencias entre tareas	22
2.2.3	Diagrama de Gantt.....	22
2.2.4	Hitos en el desarrollo del proyecto	23
2.2.5	Estimación de dedicación de cada una de las tareas	24
2.3	Sistema de información.....	26
3	Requisitos	27

3.1 Iteraciones.....	28
3.1.1 Iteración 1.....	28
3.1.2 Iteración 2.....	31
3.1.3 Iteración 3.....	34
3.2 Resultado Final	36
4 Diseño.....	42
4.1 Arquitectura software	42
4.2 Arquitectura hardware	43
4.3 Modelo de datos	45
4.4 Diagramas de secuencia	46
5 Implementación.....	49
5.1 Detalles de implementación.....	50
5.1.1 Router- Cámara.....	50
5.1.2 Móvil.....	50
5.1.3 PC.....	51
5.1.4 Coche	53
5.2 Problemas y soluciones	54
6 Pruebas.....	55
6.1 Prototipo de pruebas	56
6.2 Pruebas en las funcionalidades	56
6.2.1 Prueba P1: Conexión coche (PC)	57
6.2.2 Prueba Explorar directorio:.....	57
6.2.3 Prueba Visualización GPS (PC):.....	58
6.2.4 Prueba Dibujar recorrido:	58
6.2.5 Prueba Exportar Mapa:	58
6.2.6 Prueba Control Cámara:.....	59
6.2.7 Prueba Vídeo Streaming:	59
6.2.8 Prueba Sacar foto:.....	60
6.2.9 Prueba Mostrar foto:	60
6.2.10 Prueba Conexión coche (móvil):	60
6.2.11 Prueba Visualización GPS (móvil):.....	61
6.3 Pruebas en el hardware.....	61
6.3.1 Prueba de autonomía	61
7 Seguimiento y control.....	63
8 LPS	65
8.1 ¿Qué es LPS?	66
8.2 SPL en MapCar	66

9 Conclusiones.....	69
9.1 Reflexiones	70
Bibliografía.....	71
Diccionario.....	76
Anexo A: Manual de Usuario	81
1. Rouer-Cámara	81
2. Aplicación móvil:	82
3. Aplicación ordenador:	84
Anexo B: Actas de reuniones	87
Acta de Reunión 17/02/2016	87
Acta de Reunión 08/03/2016	88
Acta de Reunión 17/03/2016	89
Acta de Reunión 27/04/2016	90
Acta de Reunión 11/05/2016	91

Lista de Figuras

Figura 1 Placa Arduino Uno r3 [A-F-2016]	2
Figura 2 ArduinoIDE	3
Figura 3 Android Studio	5
Figura 4 MIT AppInventor [APPI-2016]	6
Figura 5 GMap.Net	7
Figura 6 XInput.Net.....	8
Figura 7 TinyGPS	8
Figura 8 WIFIESP	8
Figura 9 Funcionamiento Bluetooth [B-2016].....	10
Figura 10 Funcionamiento WI-FI [W-2016].....	11
Figura 11 Funcionamiento GSM [GSM-2016]	12
Figura 12 Funcionamiento Ultrasonidos [SU-2016]	14
Figura 13 Funcionamiento Cámara IP [CIP-2016]	15
Figura 14 Funcionamiento GPS [GPS-F-2016].....	16
Figura 15 Licencia CC BY_NC_SA [L-F-2016]	20
Figura 16 EDT.....	20
Figura 17 Prototipo inicial.....	29
Figura 18 Caso de uso iteración 1.....	30
Figura 19 Prototipo inicio iteración 2.....	32
Figura 20 Caso de uso iteración 2.....	33
Figura 21 Prototipo inicio iteración 3.....	35
Figura 22 Caso de uso iteración 3.....	36
Figura 23 Caso de uso final.....	37
Figura 24 Prototipo final.....	38
Figura 25 Aplicación de PC (Interfaz de control)	38
Figura 26 Aplicación de PC (Interfaz de configuración).....	39
Figura 27 Aplicación de PC (Interfaz de ayuda).....	39
Figura 28 Aplicación móvil (Interfaz de control)	40
Figura 29 Aplicación móvil (Interfaz de conexión)	41
Figura 30 Arquitectura Software.....	43
Figura 31 Arquitectura hardware inicial.....	44
Figura 32 Arquitectura hardware final.....	45
Figura 33 Estructura del esquema GPX.....	46
Figura 34 Diagrama de secuencias Conectar con el coche (Aplicación PC).....	47
Figura 35 Diagrama de secuencias Conectar con el coche (Aplicación Móvil).....	47
Figura 36 Diagrama de secuencia Mover coche y obtener coordenadas (Aplicación PC).....	48
Figura 37 Diagrama de secuencias Mover coche y obtener coordenadas (Aplicación Móvil).....	48
Figura 38 Diagrama de secuencias Vídeo streaming (Aplicación PC).....	49
Figura 39 Acelerómetro	50
Figura 40 Posiciones móvil [A-F-2016].....	51
Figura 41 addMarker	52
Figura 42 addRoute	52

Figura 43 Arduino main	53
Figura 44 Función WI-FI de coche	54
Figura 45 Prototipo de pruebas	56
Figura 46 Batería 800mAh [BA-F-2016]	62
Figura 47 Batería 4800mAh [BB-F-2016]	63
Figura 48 SPL MapCar	68
Figura 49 SPL de ejemplo.....	69
Figura 50 Threads	79
Figura 51 Funcionamiento IPWebcam	81
Figura 52 Aplicación móvil (Ventana de control)	82
Figura 53 Aplicación móvil (Ventana de datos de conexión).....	83
Figura 54 Aplicación Ordenador (Ventana de control).....	84
Figura 55 Aplicación Ordenador (Ventana de configuración)	85
Figura 56 Aplicación Ordenador (Ventana de ayuda)	86
Figura 57 Captura de requisitos Acta_17/02/2016.....	88

Lista de Tablas

Tabla 1 Semanas del proyecto	21
Tabla 2 Diagrama Gantt	23
Tabla 3 Hitos	24
Tabla 4 Estimación de horas/tarea.....	26
Tabla 5 Flujo de eventos iteración 1	31
Tabla 6 Flujo de eventos iteración 2	34
Tabla 7 Flujo de eventos iteración 3	36
Tabla 8 Seguimiento y control.....	65

1 Introducción

1.1 Antecedentes

La información es una herramienta muy valiosa que nos permite predecir situaciones y actuar de la manera más fructífera.

En la actualidad existen robots encargados de recolectar información y procesarla en función de las necesidades para las que esté diseñado (salvamento, reconocimiento, cartografía, etc.) que facilitan enormemente las labores humanas y decremantan considerablemente el riesgo laboral.

1.2 Motivación

Desde el comienzo de la universidad, he desarrollado un peculiar interés por el “mundillo” Arduino. Me resultaba curioso la cantidad de usos que se le pueden dar y la cantidad de módulos que se le pueden incorporar.

Desde que lo vi en un laboratorio que impartió el grupo ITSAS (<http://www.ehu.eus/es/web/itsas>) de la UPV/EHU de informática en Donostia, llevaba buscando algún proyecto funcional para probarlo.

Cerca de las navidades de 2015-2016 encontré un coche radio control al que le faltaba el mando original y cambie su controlador por una placa Arduino para así poder añadirle un mando radio control genérico y poder controlarlo.

A raíz de este “mini proyecto” busqué un profesor que lo viese como el principio de una idea para usar como trabajo de fin de carrera y es ahí donde entró en acción mi director Alfredo Goñi.

1.3 Tecnologías y herramientas

Dado que es un proyecto en el que intervienen varios dispositivos, es necesario el estudio de varias tecnologías para delimitar cuáles se adaptan más a los objetivos del mismo.

1.3.1 Arduino

Arduino es una plataforma de prototipos electrónica de código abierto (open-source) basada en hardware y software flexibles y fáciles de usar. Está pensado para artistas, diseñadores, como hobby y para cualquiera interesado en crear objetos o entornos interactivos.

Arduino puede sentir el entorno mediante la recepción de entradas desde una variedad de sensores y puede afectar a su alrededor mediante el control de luces, motores y otros artefactos. El microcontrolador de la placa se programa usando el *Arduino Programming Language* (basado en Wiring) y el *Arduino Development Environment* (basado en Processing). Los proyectos de Arduino pueden ser autónomos o se pueden comunicar con software en ejecución en un ordenador (por ejemplo con *Flash*, *Processing*, *MaxMSP*, etc.).

Las placas se pueden ensamblar a mano o encargarlas pre-ensambladas; el software se puede descargar gratuitamente. Los diseños de referencia del hardware (archivos CAD) están disponibles bajo licencia open-source, por lo que eres libre de adaptarlas a tus necesidades. [A-2016]

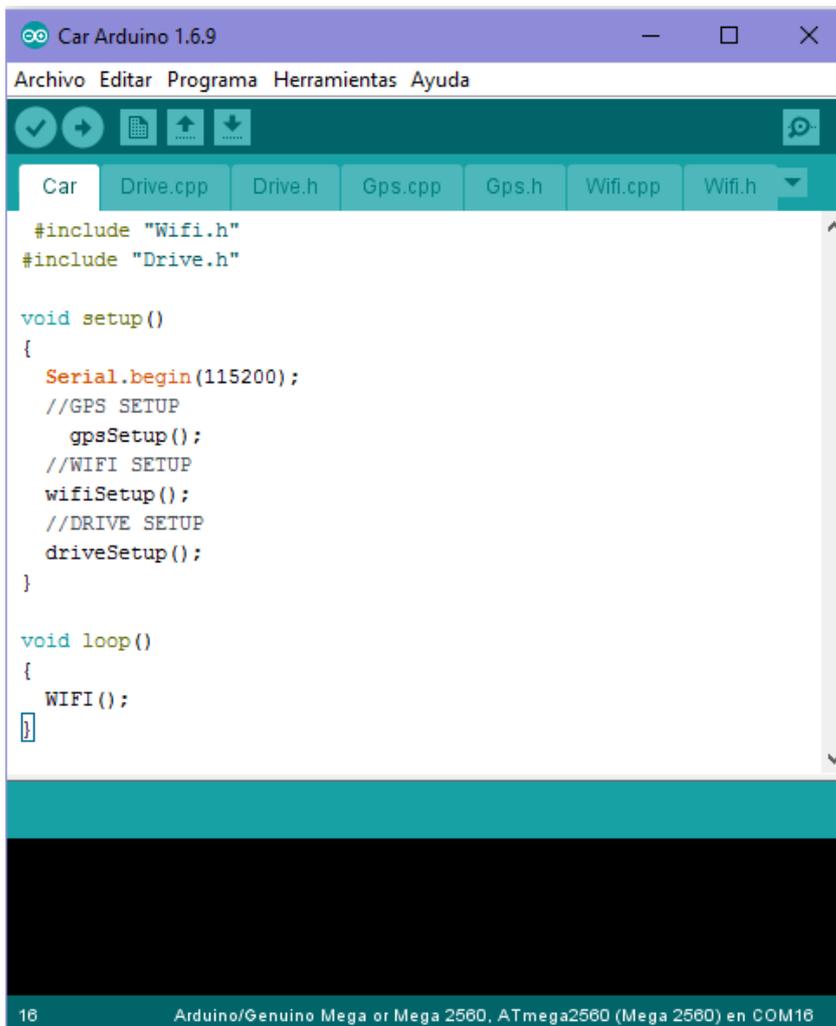


Figura 1 Placa Arduino Uno r3 [A-F-2016]

1.3.2 ArduinoIDE

Es un espacio de desarrollo integrado basado en Processing (lenguaje de programación y entorno de desarrollo integrado de código abierto basado en Java). Processing da soporte a lenguajes de programación como C, C++, C# y Java.

En otras palabras, el IDE de Arduino es un entorno que nos permite comunicar nuestro microcontrolador Arduino con nuestro ordenador y así poder integrarle cualquier programa escrito en C o cualquier lenguaje de programación que derive de C. Dicho programa podrá interactuar con cada salida o entrada de nuestro Arduino. [AIDE-2016]



The screenshot shows the Arduino IDE window titled "Car Arduino 1.6.9". The menu bar includes "Archivo", "Editar", "Programa", "Herramientas", and "Ayuda". The toolbar contains icons for file operations and execution. The file explorer shows a project named "Car" with files: "Drive.cpp", "Drive.h", "Gps.cpp", "Gps.h", "Wifi.cpp", and "Wifi.h". The main editor displays the following code:

```
#include "Wifi.h"
#include "Drive.h"

void setup()
{
  Serial.begin(115200);
  //GPS SETUP
  gpsSetup();
  //WIFI SETUP
  wifiSetup();
  //DRIVE SETUP
  driveSetup();
}

void loop()
{
  WIFI();
}
```

The status bar at the bottom indicates "16 Arduino/Genuino Mega or Mega 2560, ATmega2560 (Mega 2560) en COM16".

Figura 2 ArduinoIDE

1.3.3 Android Studio

Android Studio es un entorno de desarrollo integrado (IDE), basado en IntelliJ IDEA de la compañía JetBrains, que proporciona varias mejoras con respecto al plugin ADT (Android Developer Tools) para Eclipse. Android Studio utiliza una licencia de software libre Apache 2.0, está programado en Java y es multiplataforma.

Fue presentado por Google el 16 de mayo del 2013 en el congreso de desarrolladores Google I/O, con el objetivo de crear un entorno dedicado en exclusiva a la programación de aplicaciones para dispositivos Android, proporcionando a Google un mayor control sobre el proceso de producción. Se trata pues de una alternativa real a Eclipse, el IDE recomendado por Google hasta la fecha, pero que presentaba problemas debido a su lentitud en el desarrollo de versiones que solucionaran las carencias actuales (es indispensable recordar que Eclipse es una plataforma de desarrollo, diseñada para ser extendida a través de plugins).

Android Studio se ha mantenido durante todo este tiempo en versión beta, pero desde el 8 de diciembre de 2014, en que se liberó la versión estable de Android Studio 1.0, Google ha pasado a recomendarlo como el IDE para desarrollar aplicaciones para su sistema operativo, dejando el plugin ADT para Eclipse de estar en desarrollo activo. **[AS-2016]**

Mediante Android Studio se ha desarrollado la aplicación móvil WI-FI, para la comunicación entre aplicación y vehículo.

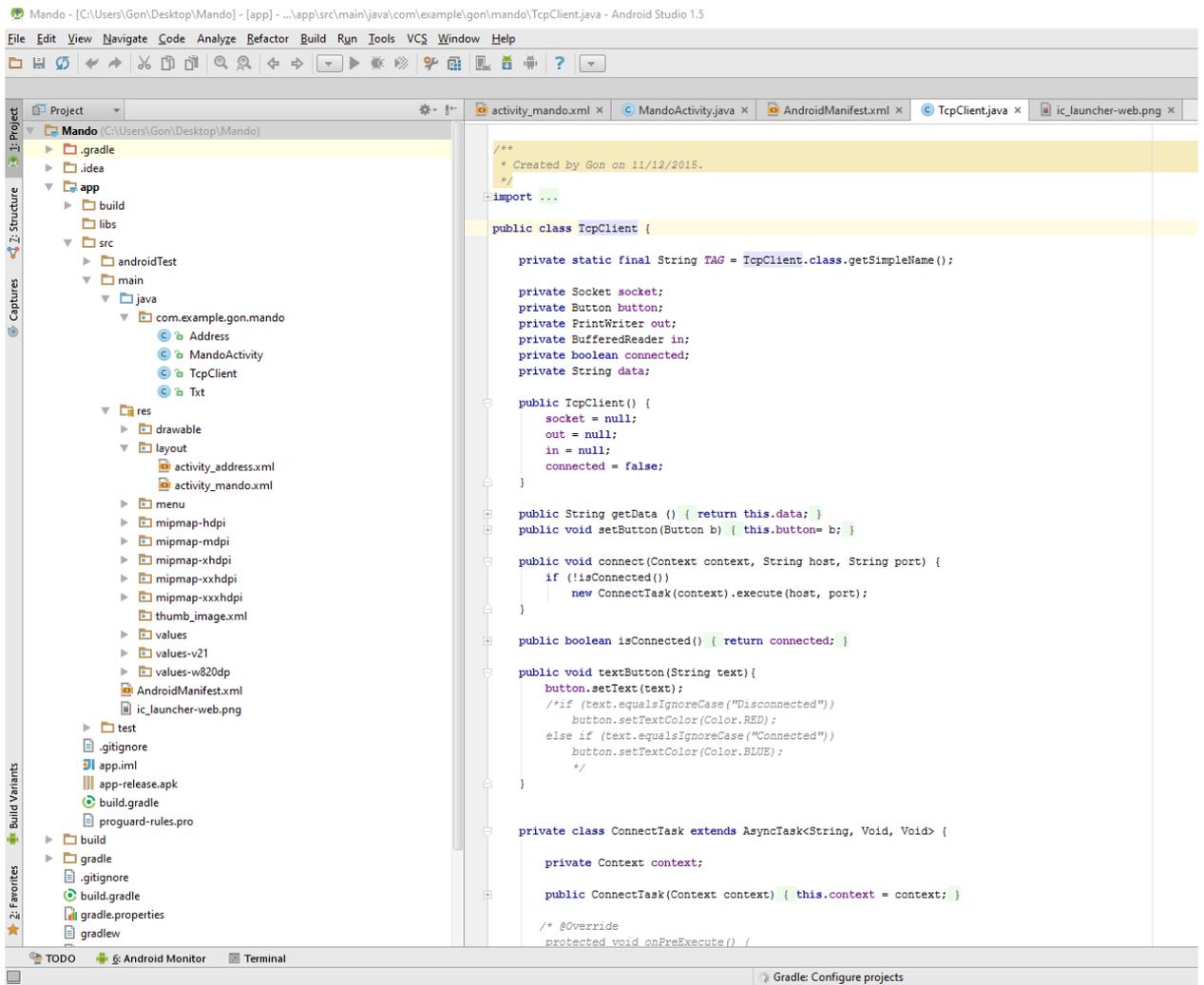


Figura 3 Android Studio

1.3.4 MIT AppInventor

App Inventor es un entorno de desarrollo de aplicaciones para dispositivos Android.

Para desarrollar aplicaciones con App Inventor sólo hace falta un navegador web (ya que se trabaja en la nube) y un móvil o tablet Android (también se puede probar las aplicaciones en un emulador).

App Inventor se basa en un servicio web que permite almacenar el trabajo ayuda a realizar un seguimiento de los proyectos. Es una herramienta de desarrollo visual muy fácil de usar, con la que incluso los no programadores pueden desarrollar sus aplicaciones.

Al construir las aplicaciones para Android se trabaja con dos herramientas: App Inventor Designer y App Inventor Blocks Editor. En Designer se construye el Interfaz de Usuario,

eligiendo y situando los elementos con los que interactuará el usuario y los componentes que utilizará la aplicación. En el Blocks Editor se define el comportamiento de los componentes de tu aplicación. [APPI-2016]

Mediante MIT AppInventor he desarrollado la aplicación móvil Bluetooth para conectar con el vehículo.

A pesar de ser un entorno de desarrollo muy intuitivo y cómodo, aún carece de muchas funcionalidades que impiden que se pueda emplear para todo tipo de proyectos. Por ejemplo y en mi caso, para realizar la aplicación que controla el coche con el móvil, he tenido que emplear Android Studio.

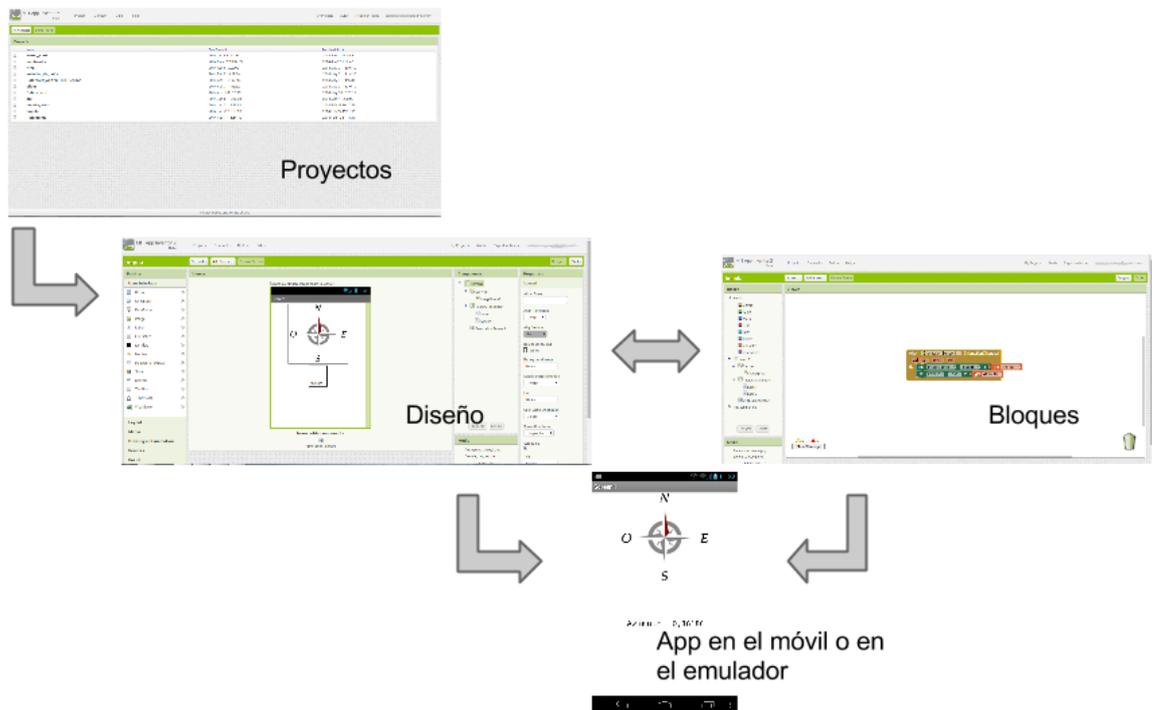


Figura 4 MIT AppInventor [APPI-2016]

1.3.5 Visual Studio 2013

Visual Studio 2013 es el IDE de programación por excelencia de la plataforma .NET y, por ende, de las aplicaciones que corren en los dispositivos del ecosistema de Microsoft.

Trasciende las necesidades específicas del desarrollo al construir, junto con otros servidores, un completo y complejo sistema integrado de gestión de programación de aplicaciones informáticas. [VS-2016]

1.3.6 GMap.Net

Gmap.Net es una librería para usar google maps en .NET. Su amplitud de posibilidades es inmensa. Se puede crear tanto mapas dinámicos como estáticos.

A diferencia de otras librerías para trabajar con mapas, Gmap permite la creación de eventos para poder dotar de mayor funcionalidad a la aplicación. Gracias a esto, se pueden generar marcadores e interactuar con ellos, o hacer polígonos con información dinámica, etc...

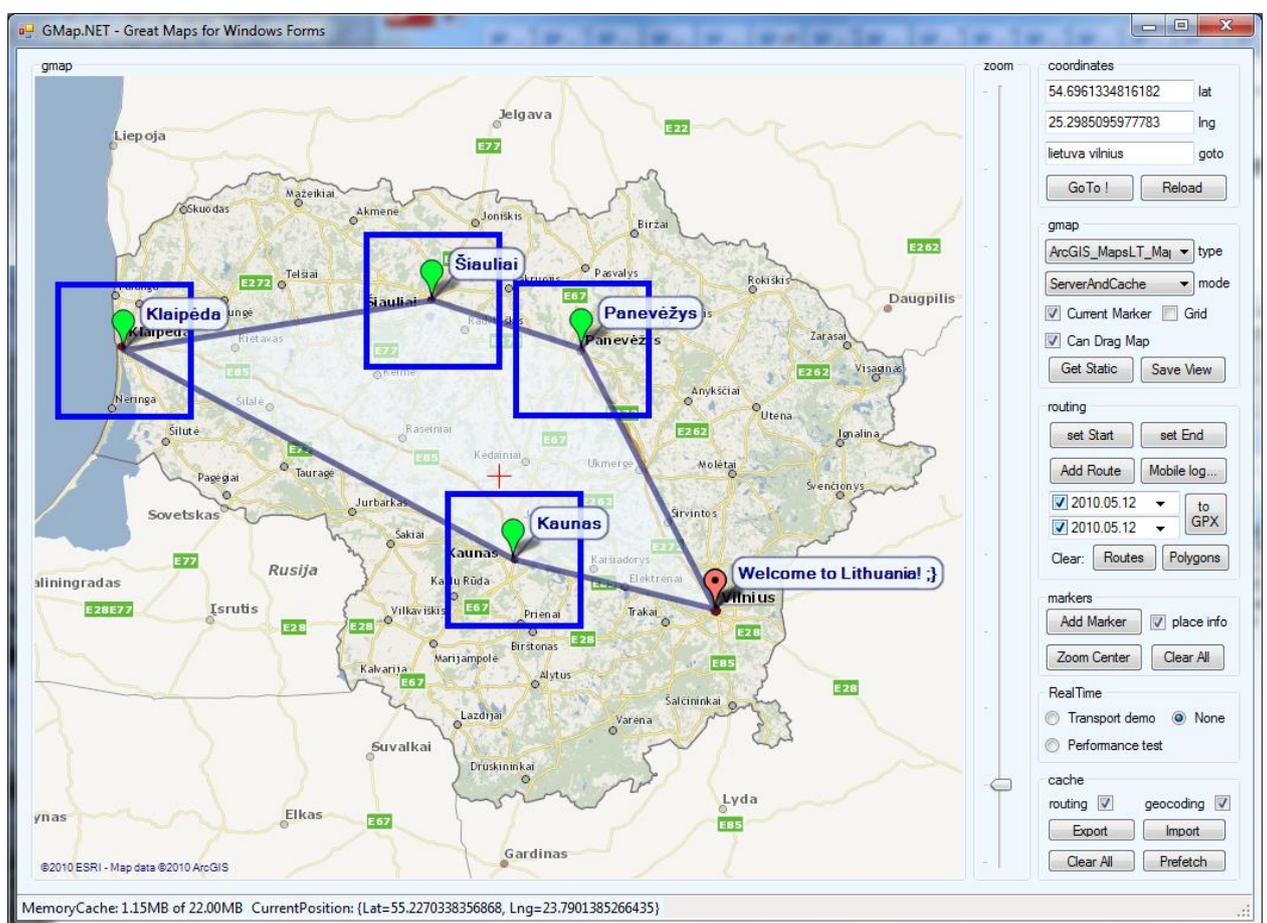


Figura 5 GMap.Net

1.3.7 XInput.NET

Librería para aplicaciones .Net, que permite controlar un mando de la Xbox 360.

Incluye funciones para poder asignar funcionalidad a los diversos botones del mando y conocer su estado.

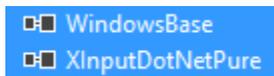


Figura 6 XInput.Net

1.3.8 TinyGPS

Librería para Arduino, que facilita funciones para gestionar y configurar el módulo GPS.

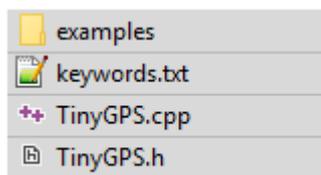


Figura 7 TinyGPS

1.3.9 WIFIESP

Librería para Arduino, que facilita funciones para gestionar y configurar el módulo WI-FI ESP8266

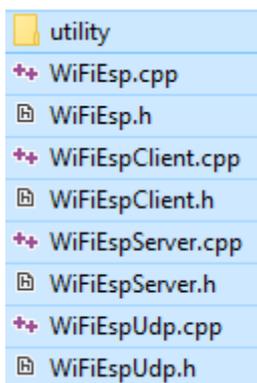


Figura 8 WIFIESP

1.3.10 Sistema de comunicación

En este apartado se explican los diferentes módulos que se han tenido en cuenta para el sistema de comunicación del proyecto. Así como las ventajas y desventajas que proporciona cada uno.

1.3.10.1 Mando RC

Las emisoras capaces de emitir en la frecuencia de 2,4 Ghz utilizan un protocolo de escalonamiento de espectro **Direct Sequence Spread Spectrum (DSSS)**. Esta tecnología permite que numerosas emisoras y sus correspondientes receptores usen sin problemas un mismo canal.

Ventajas

- Las emisoras y receptores provistos de 2,4 Ghz son poco sensibles a interferencias.
- Se tratan de emisoras digitales que disponen de memoria para guardar las configuraciones de varios modelos y también se puede cambiar la forma en la que interactúan con los servos (estos han de ser digitales también).
- Sólo se emplea un canal por lo que una vez ligado el receptor con su emisora correspondiente no existe posibilidad de interferir con otro usuario, como ocurre en FM.

Inconvenientes

DSSS es una buena tecnología de comunicaciones para radiocontrol, el problema es que se necesita un espectro de frecuencias amplio para una transmisión estable. Para radiocontrol no quedó más remedio que elegir una de las frecuencias libres y gratuitas, la 2,4 Ghz usada también por dispositivos WI-FI, Bluetooth, etc... y que es gratis porque no interesa para fines comerciales ya que coincide con la frecuencia de resonancia de la molécula del agua. Esto quiere decir que las emisiones en 2,4 Ghz se ven muy atenuadas por la niebla y la humedad del ambiente (esta frecuencia no sirve para radioenlaces a distancia para el gobierno o empresas privadas fuertes). Así pues estamos con una buena tecnología de transmisión pero con una frecuencia pésima, invadida por múltiples dispositivos y además poco eficiente para la transmisión de señales. **[VDRC-2016]**

1.3.10.2 Bluetooth

Es una tecnología de ondas de radio de corto alcance (2.4 Ghz de frecuencia) cuyo objetivo es el simplificar las comunicaciones entre dispositivos informáticos, como

ordenadores móviles, teléfonos móviles, otros dispositivos de mano y entre estos dispositivos e Internet. También pretende simplificar la sincronización de datos entre los dispositivos y otros ordenadores.

Permite comunicaciones, incluso a través de obstáculos, a distancias de hasta unos 10 metros. Esto significa que, por ejemplo, puedes permitir escuchar música en formato MP3 en diferentes lugares de la casa. También sirve para crear una conexión a Internet inalámbrica desde tu portátil usando tu teléfono móvil. Un caso aún más práctico es el poder sincronizar libretas de direcciones, calendarios etc. en tu PDA, teléfono móvil, ordenador de sobremesa y portátil automáticamente y al mismo tiempo. [B-2016]



Figura 9 Funcionamiento Bluetooth [B-2016]

Ventajas

- Conectividad inalámbrica

Inconvenientes

- Velocidad de transmisión muy lenta
- Cuando es usado inadecuadamente, podemos recibir mensajes y archivos indeseados.
- Limitado radio de acción entre los periféricos (10 metros aprox).
- Límite entre la cantidad de **periféricos** que podemos usar. Los adaptadores Bluetooth solo permiten hasta 7 equipos **pareado** (término usado para definir los equipos que se pueden sincronizar y comunicar entre sí).
- Gasta mucha energía de la batería, cuando está en el modo visible.

[VDB-2016]

1.3.10.3 WI-FI

Es un mecanismo de conexión de dispositivos electrónicos de forma **inalámbrica**. Los dispositivos habilitados con WI-FI (como un ordenador, un televisor inteligente, una videoconsola, un móvil o un reproductor de música) pueden conectarse a internet a través de un punto de acceso de red inalámbrica. Dicho punto de acceso tiene un alcance de unos veinte metros en interiores, distancia que es mayor al aire libre.

WI-FI es una marca de la WI-FI Alliance, la organización comercial que adopta, prueba y certifica que los equipos cumplen con los estándares 802.11 relacionados a redes inalámbricas de área local. [W-2016]

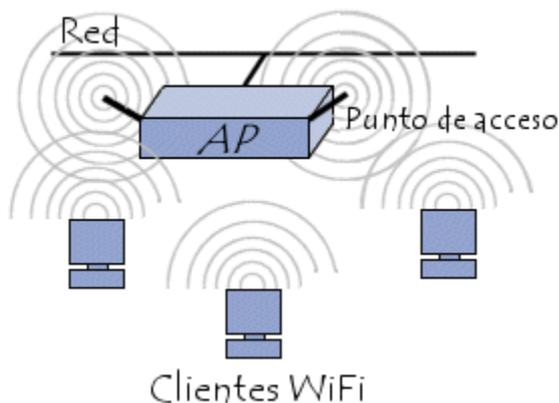


Figura 10 Funcionamiento WI-FI [W-2016]

Ventajas

- Conectividad inalámbrica
- Elección de entre varias señales libres o con seguridad

Inconvenientes

- Muchos dispositivos de diferentes marcas no son absolutamente compatibles con la tecnología WI-FI, lo cual también repercute en la velocidad.
- Su radio de acción es limitado (20 metros para interior aproximadamente).
- La cercanía de dos o más puntos de acceso puede afectar la calidad de la comunicación; esto se da especialmente en edificios donde muchas personas utilizan esta tecnología simultáneamente.
- Dispone de baja seguridad.

1.3.10.4 GSM

GSM, o sistema Global para comunicación móvil, es una red de telefonía móvil mundial estándar.

En una red GSM, la terminal del usuario se llama **estación móvil**. Una estación móvil está constituida por una tarjeta **SIM** (*Módulo de identificación de abonado*), que permite identificar de manera única al usuario y a la terminal móvil, o sea, al dispositivo del usuario (normalmente un teléfono portátil).

Las terminales (dispositivos) se identifican por medio de un número único de identificación de 15 dígitos denominado **IMEI** (*Identificador internacional de equipos móviles*). Cada tarjeta SIM posee un número de identificación único (y secreto) denominado **IMSI** (*Identificador internacional de abonados móviles*). Este código se puede proteger con una clave de 4 dígitos llamada *código PIN*.

Por lo tanto, la tarjeta SIM permite identificar a cada usuario independientemente de la terminal utilizada durante la comunicación con la estación base. Las comunicaciones entre una estación móvil y una estación base se producen a través de un vínculo de radio, por lo general denominado **interfaz de aire** (o en raras ocasiones, *interfaz Um*). [GSM-2016]

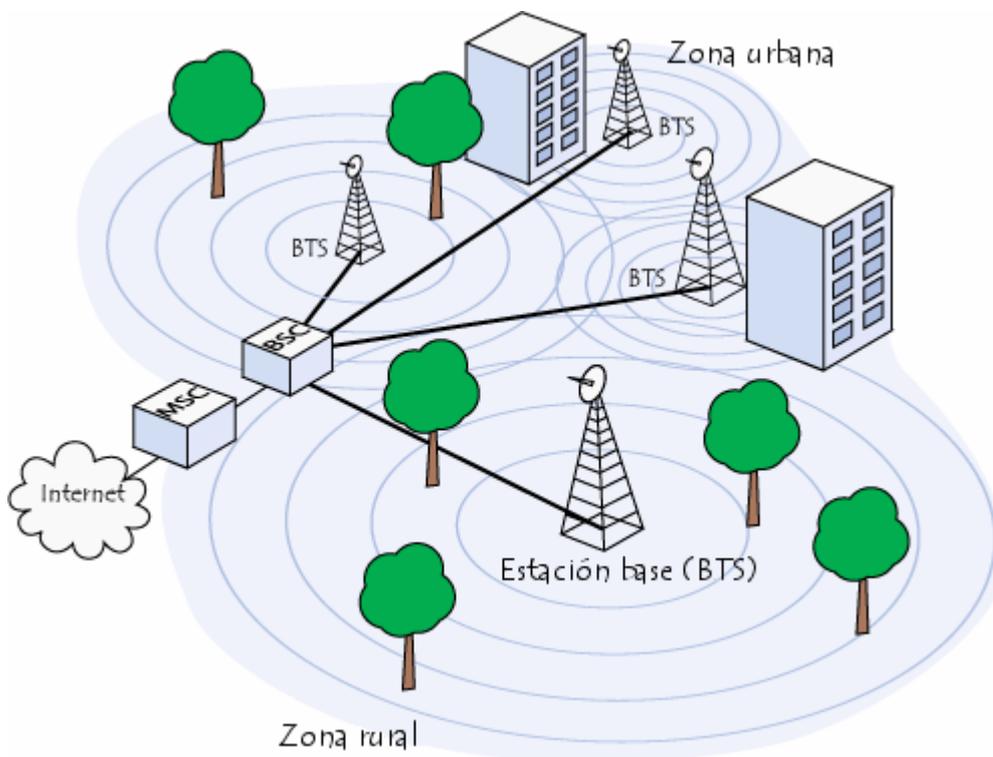


Figura 11 Funcionamiento GSM [GSM-2016]

Ventajas

- Identificación de los usuarios de los móviles mediante el uso de un chip incorporado en el teléfono denominado simcard.
- Debido a que es posible almacenar la información en una tarjeta denominada SIM en vez del dispositivo como sucedía anteriormente, se facilita el traspaso de diferentes servicios (Mensajería, Agenda) de un Móvil a otro sin mayores problemas
- Extensa área de cobertura, gracias a su capacidad para roaming internacional (Es posible tener cobertura y realizar y recibir llamadas desde diferentes partes del mundo con el mismo dispositivo y misma línea).
- Provee claridad y calidad en los servicios de voz.
- Permite la transmisión y recepción de información multimedia.
- Ofrece múltiples servicios como: Mensajería de Texto, Identificador de Llamada, llamada entrante, llamada saliente, llamada en espera, correo de voz, entre otros.
- Debido a su amplio uso alrededor del mundo, existe una gran variedad de equipos para la escogencia de los usuarios.

Inconvenientes

- Una de las principales desventajas de GSM es que gran cantidad de usuarios comparten el mismo ancho de banda, y con una cantidad suficiente de estos la transmisión puede encontrar interferencias.
- Puede presentar interferencia con determinados dispositivos electrónicos.
- Requiere un número considerado de radios base para garantizar una gran cobertura.

[VDGSM-2016]

1.3.11 Sistemas de recolección de datos

Después de definir los sistemas de comunicación, se procede a mostrar los sistemas de recolección de datos implementados y probados en el proyecto.

1.3.11.1 Ultrasonidos

El sensor de ultrasonidos es un dispositivo de medición de distancia que se basa en las propiedades magnetostrictivas de determinados materiales. Una lámina de material magnetostrictivo o membrana tiene la propiedad de deformarse mecánicamente y generar ultrasonidos al ser excitada por una corriente eléctrica. El efecto contrario también se produce, es decir, que una vibración mecánica produce una corriente eléctrica. Por tanto, estos sensores emiten una radiación ultrasónica que rebota en los obstáculos del entorno y captan los ecos recibidos.

Los sensores de ultrasonidos se utilizan para medir distancias, que a partir de la excitación de la membrana magnetostrictiva con una serie de impulsos eléctricos, se genera un tren de ondas ultrasónicas. Por otro lado, la membrana magnetostrictiva, recibe los ecos de las emisiones recibidas y las transforma en impulsos eléctricos. [SU-2016]

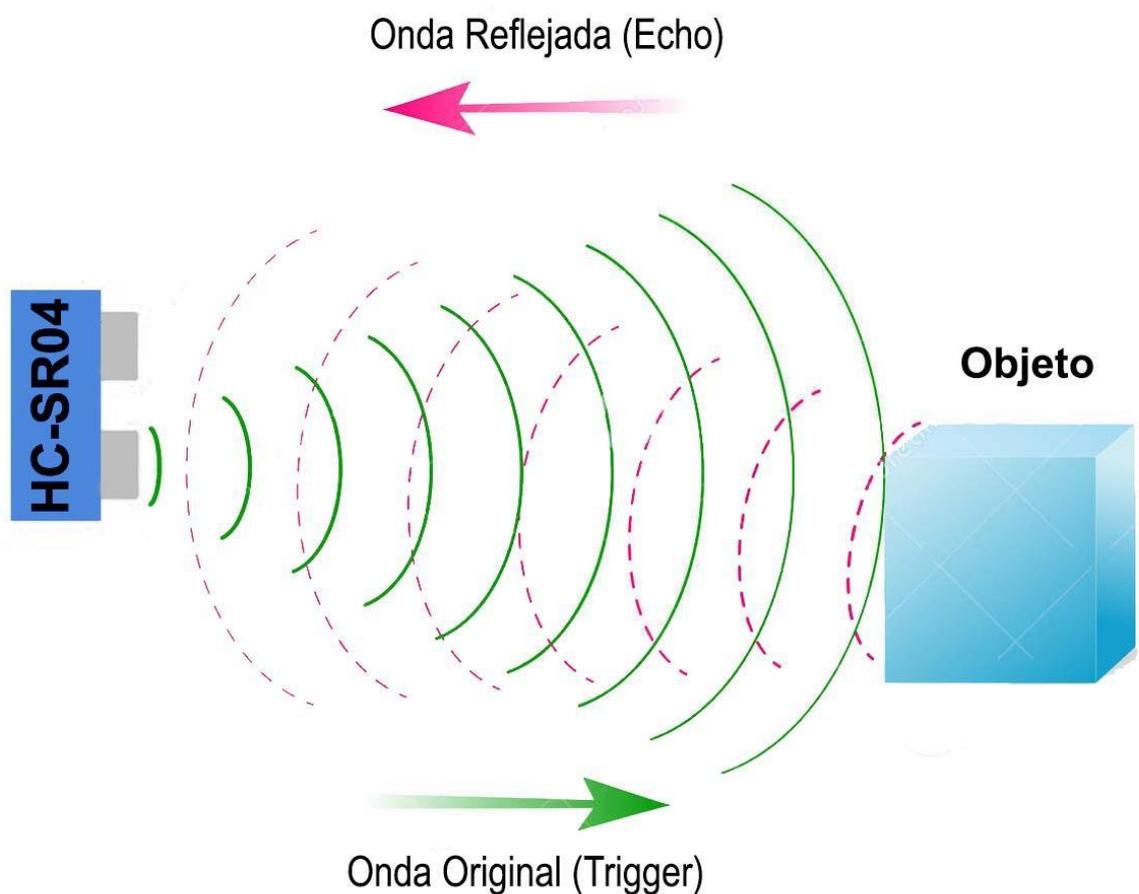


Figura 12 Funcionamiento Ultrasonidos [SU-2016]

1.3.11.2 Cámara IP

Una Cámara IP (también conocidas como cámaras Web o de Red) es una videocámara especialmente diseñada para enviar las señales (video, y en algunos casos

audio) a través de Internet desde un explorador (por ejemplo el Internet Explorer) o a través de un concentrador (un HUB [H-D-2016] o un SWITCH [SW-D-2016]) en una Red Local (LAN [L-D-2016])

En las cámaras IP pueden integrarse aplicaciones como detección de presencia (incluso el envío de mail si detectan presencia), grabación de imágenes o secuencias en equipos informáticos (tanto en una red local o en una red externa (WAN [W-D-2016]), de manera que se pueda comprobar por qué ha saltado la detección de presencia y se graben imágenes de lo sucedido. [CIP-2016]

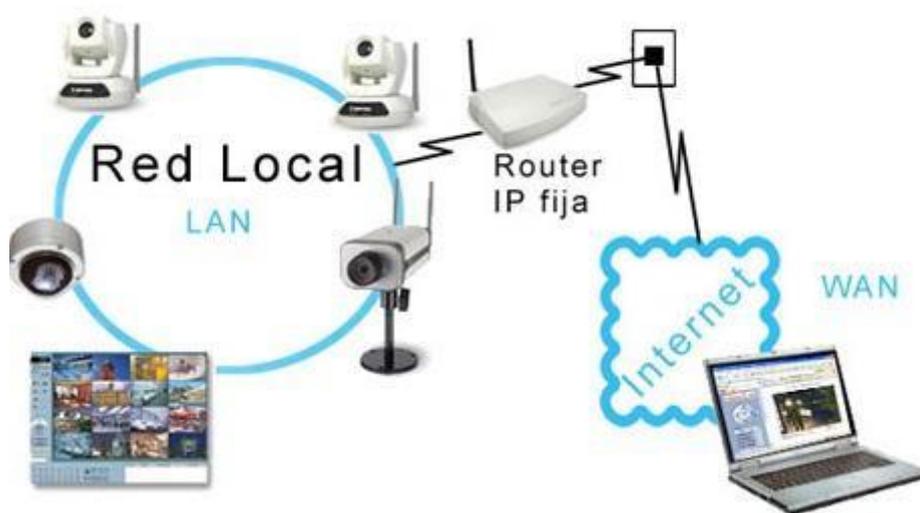


Figura 13 Funcionamiento Cámara IP [CIP-2016]

1.3.11.3 GPS

El sistema de posicionamiento global (GPS) es un sistema que permite determinar en toda la Tierra la posición de un objeto (una persona, un vehículo) con una precisión de hasta centímetros (si se utiliza GPS diferencial), aunque lo habitual son unos pocos metros de precisión. El sistema fue desarrollado, instalado y empleado por el Departamento de Defensa de los Estados Unidos. Para determinar las posiciones en el globo, el sistema GPS se sirve de 24 satélites y utiliza la trilateración.

El GPS funciona mediante una red de 24 satélites en órbita sobre el planeta Tierra, a 20 200 km de altura, con trayectorias sincronizadas para cubrir toda la superficie de la Tierra. Cuando se desea determinar la posición, el receptor que se utiliza para ello localiza automáticamente como mínimo tres satélites de la red, de los que recibe unas señales indicando la identificación y la hora del reloj de cada uno de ellos. Con base en estas señales, el aparato sincroniza el reloj del GPS y calcula el tiempo que tardan en llegar las señales al equipo, y de tal modo mide la distancia al satélite mediante el método de trilateración inversa, el cual se basa en determinar la distancia de cada satélite al punto de medición. Conocidas las distancias, se determina fácilmente la propia posición relativa respecto a los satélites. [GPS-2016]

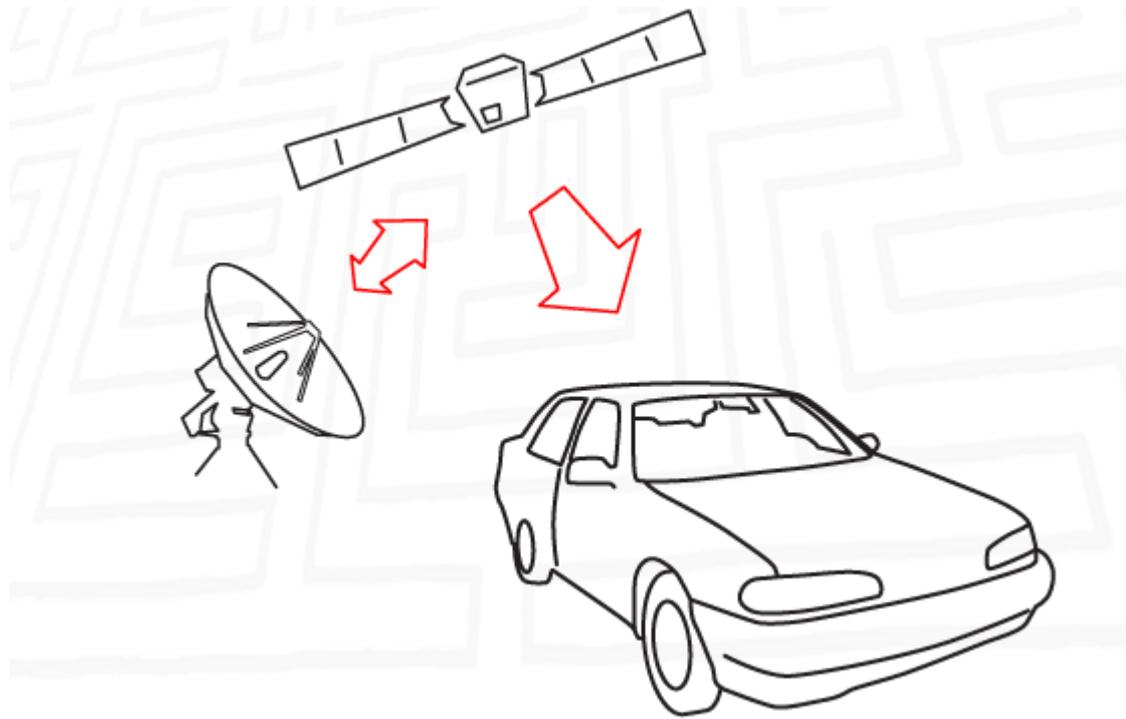


Figura 14 Funcionamiento GPS [GPS-F-2016]

1.3.11.4 MAP

Para exportar las coordenadas del recorrido realizado y las rutas de las imágenes capturadas se emplea el formato GPX. Este formato es un esquema XML [XML-D-2016] pensado para transferir datos GPS entre aplicaciones. Se puede usar para describir puntos (waypoints), recorridos (tracks), y rutas (routes). [GPX-2016]

Una vez obtenido el fichero, se puede importar en cualquier visualizador gps (las fotos no se verán si no se usa un visualizador instalado en el dispositivo que contenga las fotos).

Para este proyecto, vamos a emplear un móvil como Cámara IP y punto AP [AP-D-2016] para transmitir la información y realizar el Streaming [ST-D-2016] . Aparte, se utilizará un módulo GPS para saber en todo momento la posición de vehículo y realizar el mapa.

El módulo ultrasonidos, aunque estuvo incorporado en algún prototipo previo, se decidió mantener al margen ya que implicaba un excesivo tiempo a invertir para obtener las respuestas deseadas.

1.3.12 Elección de Tecnologías de comunicación y de captación de datos

Para la comunicación entre dispositivos se han barajado las siguientes posibilidades descritas a continuación. Siendo la tecnología WIFI la seleccionada para este proyecto. Esto se debe a los siguientes motivos:

- Ya está incorporado en todos los dispositivos a emplear.
- El WI-FI tiene un mayor alcance que el Bluetooth.
- Con el WI-FI puedo establecer múltiples conexiones.
- El Bluetooth suele dar problemas a la hora de realizar la conexión.

- Es el único sistema de comunicación de los estudiados que permite realizar todas las comunicaciones necesarias para realizar las funcionalidades indicadas sin necesidad de añadir otros sistemas de comunicación auxiliares.

* Se ha descartado el uso de GSM (**que en realidad sería la idónea en el escenario del proyecto**) por falta de presupuesto.

Para los sistemas de captación de datos, **se ha excluido** el módulo ultrasonidos debido a la excesiva carga de trabajo que supone su implementación y correcto funcionamiento. Esta variante a tener en cuenta, descarta la opción de emplear el vehículo captador de datos de forma autónoma.

El resto de Tecnologías de captación de datos mencionadas en el apartado [1.3.11](#) se incorporarán al proyecto.

1.3.13 Sistemas de control

Dado que desde un inicio ya se han establecido los sistemas de control, su aparición en este y el resto de apartados de este documento serán casi nulos.

Como sistemas de control, se hacen referencia a los dispositivos para los que se van a desarrollar las aplicaciones encargadas de controlar y gestionar el vehículo.

Los Sistemas de control empleados en el proyecto MapCar son:

- Un dispositivo móvil: Donde se instalará la aplicación desarrollada en Android mediante Android Studio, y permitirá controlar y obtener las coordenadas del vehículo mediante la tecnología WI-FI.
- Un Ordenador: Preferentemente un portátil con Sistema Operativo Windows 7,8 o 10. En él se instalará la aplicación que, mediante un mando de la xbox360 permitirá

controlar el vehículo, dibujar el recorrido que realice, generar un mapa dinámico y visualizar el movimiento del mismo.

1.4 Metodología

Para llevar a cabo el desarrollo del software y el prototipo captador de datos, se ha decidido en la medida de lo posible una **metodología iterativa**. Donde, partiendo de un prototipo minimalista, se añadirán nuevas funcionalidades por iteración.

1.4.1 Metodología Iterativa

Dado que, antes de comenzar este proyecto ya existía un prototipo del vehículo en funcionamiento (desarrollado por mí mismo), se consideró interesante ir ampliando (de manera incremental) las funcionalidades a lo largo de iteraciones cortas.

Este método es en cierta manera similar al desarrollo ágil propuesto por **Scrum**

[S-D-2016] , pero con algunas diferencias importantes ya que no existe un equipo de trabajo para realizar el desarrollo. No hay un cliente y las únicas partes implicadas son el propio desarrollador y el director del proyecto.

Inicialmente, se realiza una especulación sobre las posibles fases del proyecto y las semanas de inicio y fin de cada iteración.

Al inicio de cada iteración se establecen las nuevas funcionalidades a implementar, la fecha de finalización de esta y la siguiente reunión a realizar seguido de una demostración de lo realizado hasta la fecha.

Para una mayor comprensión de las funcionalidades implementadas y una correcta ubicación de las mismas, se crearán diferentes diagramas de casos de uso junto a su correspondiente flujo de eventos. Estos diagramas y flujo de eventos serán no exhaustivos y su objetivo será detallar las funcionalidades implementadas.

2 Planificación

2.1 Alcance

El objetivo de este proyecto es desarrollar un prototipo de un vehículo y un software captador de datos que permita una posterior explotación de los mismos.

Para ello se busca completar los siguientes objetivos:

- Desarrollar un prototipo captador de datos.
- Programar una aplicación móvil para controlar el prototipo.
- Programar una aplicación de ordenador para controlar el prototipo, visualizar el streaming de la cámara y realizar mapas del recorrido realizado junto con los datos captados.
- Desarrollar una estructura de datos para exportar el mapa realizado.

2.1.1 Descripción de los objetivos concretos del proyecto

2.1.1.1 Gestión de adquisiciones

- Arduino RC Car con FPV Camera [FPV-2016]: Aplicación en c# que permite controlar con un mando de la box un vehículo mediante puerto serie en Arduino. También incorpora funcionalidad vídeo streaming mediante IP.

2.1.1.2 Licencias de los productos adquiridos

- Arduino RC Car con FPV Camera
 - Reconocimiento – NoComercial – CompartirIgual (by-nc-sa)

2.1.1.3 Licencia del producto final

- **Reconocimiento – NoComercial – CompartirIgual (by-nc-sa):** No se permite un uso comercial de la obra original ni de las posibles obras

derivadas, la distribución de las cuales se debe hacer con una licencia igual a la que regula la obra original.



Figura 15 Licencia CC BY_NC_SA [L-F-2016]

2.1.2 Restricciones del proyecto

- Se empleará mínimo un dispositivo Android como controlador del vehículo.
- El proyecto incluirá un módulo GPS además de sus otros módulos de captación de datos.

2.1.3 WBS/EDT

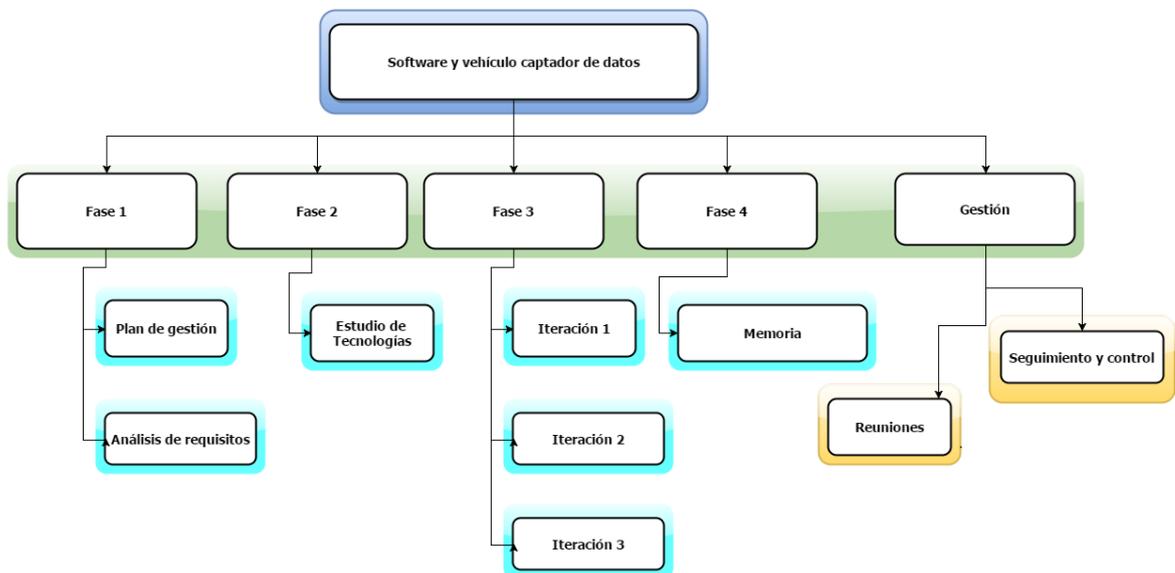


Figura 16 EDT

2.1.4 Riesgos

- Durante la ejecución del proyecto, se encuentra una época de exámenes de otras asignaturas que complica el trabajo del equipo. El nivel de trabajo en estos periodos debe ser menor.
- Se puede dar el caso de suspender alguna de las asignaturas, retrasando considerablemente la finalización del proyecto.

2.2 Periodos de realización de las tareas

El proyecto se divide inicialmente en 22 semanas, siendo las siguientes:

S1 (15 feb - 21 feb)	S2 * (22 feb - 28 feb)	S3 (29 feb - 06 Mar)	S4 (07 Mar - 13 Mar)
S5 (14 Mar - 20 Mar)	S6 (21 Mar - 27 Mar)	S7 (28 Mar - 03 Abr)	S8 (04 Abr - 10 Abr)
S9 (11 Abr - 17 Abr)	S10 * (18 Abr - 24 Abr)	S11 (25 Abr - 01 May)	S12 (02 May - 08 May)
S13 (09 May - 15 May)	S14 * (16 May - 22 May)	S15 ** (23 May - 29 May)	S16 (30 May - 05 Jun)
S17 (06 Jun - 12 Jun)	S18 (13 Jun - 19 Jun)	S19 (20 Jun - 26Jun)	S20 (27 Jun - 03 Jul)
S21 (04 Jul - 10 Jul)	S22 (11 Jul - 17 Jul)		

Tabla 1 Semanas del proyecto

*Fin de trabajo y proyectos: Semanas a las que se le dedicará poco tiempo al proyecto.

**Exámenes: Semanas a las que se le dedicará poco tiempo al proyecto.

2.2.1 Relacionados con el objeto de proyecto en sí

Fase 1, La realización de la primera tarea, ayuda a tener en cuenta objetivos, prioridades y plazos de entrega de cada tarea individual para un buen desarrollo del proyecto.

Una vez realizada una buena planificación, se procede a realizar las siguientes actividades descritas en este documento.

Fase 2, Estudio de tecnologías. Para proceder con el desarrollo del prototipo y el software primero es necesario conocer las diversas tecnologías de las que se dispone y escoger las más adecuadas.

Fase 3, Desarrollo del producto. Esta fase está compuesta de varias iteraciones destinadas al desarrollo del prototipo y los softwares de captación de datos.

Iteración 1. Implementar los sistemas de comunicación.

Iteración 2. Implementar los sistemas de captación de datos.

Iteración 3. Desarrollar el Mapa que mostrará el recorrido realizado por el vehículo y los datos captados.

Fase 4, Memoria.

2.2.2 Dependencias entre tareas

- La realización de la primera fase, ayuda a tener en cuenta objetivos, prioridades y plazos de entrega de cada tarea individual para un buen desarrollo del proyecto. Una vez realizada una buena planificación, se procede a realizar las siguientes tareas descritas en este documento.
- La Fase 2 define las diversas tecnologías a emplear en la siguiente fase de este proyecto.
- La Fase 3 determina el producto final.
- En la Fase 4 se genera un documento que albergue toda la información y conocimiento recopilado en el transcurso de las anteriores fases.

2.2.3 Diagrama de Gantt

	Fase 1	Fase 2	Fase 3	Fase 4	Gestión
S1					
S2					
S3					
S4					
S5					
S6					
S7					

S8					
S9					
S10					
S11					
S12					
S13					
S14					
S15					
S16					
S17					
S18					

Tabla 2 Diagrama Gantt

2.2.4 Hitos en el desarrollo del proyecto

	Fin Fase 1	Fin Fase 2	Fin Fase 3	Fin Fase 4	Matriculación del PFG	Subir el PFG a ADDI	Defensa del PFG
S1							
S2	◆						
S3							
S4		◆					
S5							
S6							
S7							
S8							
S9							
S10							

S11							
S12							
S13			◆				
S14							
S15							
S16							
S17							
S18				◆	◆		
S19						◆	
S20							
S21							◆
S22							◆

Tabla 3 Hitos

2.2.5 Estimación de dedicación de cada una de las tareas

Fases	SUB-TAREA	Previsión	
		Horas	Semanas
Fase 1	Plan de Gestión	7h	S1 - S2
	Análisis de Requisitos	5h	S1 - S2
Fase 2	Estudio de tecnologías	24h	S3 – S4

Fase 3	Iteración 1	Sistema de comunicación	Implementación Módulo RC	7h	S5
			Implementación módulo Bluetooth	5h	S5
			Construcción APK Bluetooth	6h	S6
			Implementación módulo WIFI	12h	S6
			Construcción APK WIFI	16h	S7
	Iteración 2	Sistema de captación de datos	Diseño e implementación de aplicación pc para el control del coche mediante WIFI	22h	S8
			Implementación módulo GPS	10h	S8
			Implementar Sistema de streaming vídeo	7h	S9
			Capturar fotos y mostrarlas	2h	S10
			Añadir opciones a la cámara	2h	S10
	Iteración 3	Mapa	Mostrar el recorrido que realiza el coche	12h	S10
			Generar fichero con el recorrido y la fotos obtenidas	8h	S10 - S11
	Fase 4	Memoria		100h	S14 - S18

Gestión	Seguimiento y Control	1h	S1-S2
		2h	S3 – S4
		2h	S5 - S13
		2h	S14 - S18
TOTAL		284h	

Tabla 4 Estimación de horas/tarea

Se estima que la dedicación para este proyecto será de 300 horas en total como máximo.

2.3 Sistema de información

Para el sistema de información se ha generado una carpeta en Google Drive llamada “Software y vehículo captador de datos”, la cual contiene las siguientes subcarpetas;

- “Activos”: carpeta en la que se almacena la documentación que se va generando durante la elaboración del producto y en la gestión del mismo. Aquí también se contiene todo lo necesario para la elaboración del producto final. Toda esta documentación se divide en diferentes subcarpetas:
 - “Código”: aquí se almacena los backups [B-D-2016] de las aplicaciones desarrolladas.
 - “Imágenes”: Imágenes necesarias para la memoria.
 - “Plantillas”: se almacenan todas las plantillas que se generen para la elaboración del proyecto.
 - “Programas”: Programas y drivers necesarios para el funcionamiento de la aplicación.
 - “Videos”: Almacena todos los videos a introducir en el producto.
- “Gestión”: se guardan todos los entregables del proyecto, junto con sus respectivas revisiones, también todos los documentos relacionados con la gestión del proyecto. Seguimiento y control, planificación etc...

- “*Actas*”: carpeta en la que se van almacenando las actas generadas en las diferentes reuniones. El formato del nombre de cada uno de los documentos generados es: *Acta año/mes/día*
- “*Informes*”: documentos de interés para ayudar con la culminación de proyecto
- “*Producto*”: se guardan los entregables finales.

La elección de la utilización de Google Drive es su facilidad para almacenar, compartir y gestionar el contenido.

3 Requisitos

En este capítulo, se van a detallar los requisitos que componen el desarrollo de la fase 3.

Para detallar los requisitos, el método seguido ha sido, a lo largo de las diferentes iteraciones, implementar los módulos con sus correspondientes funcionalidades seguido de la definición de sus casos de uso y los flujos de eventos expresados en UML.

Cada iteración va acompañado del prototipo resultante de la finalización de la iteración anterior para una mejor comprensión.

3.1 Iteraciones

El número de iteraciones establece un grado de prioridad requerido para una correcta implementación.

A continuación se muestran los detalles de las diferentes iteraciones que conforman este proyecto:

Iteración 1. Implementar los sistemas de comunicación.

Iteración 2. Implementar los sistemas de captación de datos.

Iteración 3. Desarrollar el Mapa que mostrará el recorrido realizado por el vehículo y los datos captados.

3.1.1 Iteración 1

Implementación de los sistemas de comunicación.

3.1.1.1 Prototipo Actual

Partimos **únicamente** de un coche prototipo minimalista realizado previamente al inicio del proyecto. La estructura de dicho vehículo se muestra en la **Figura 17 Prototipo inicial** y consta de los componentes básicos (1 motor, 4 ruedas, 1 chasis, 1 servo de dirección, 1 batería, 1 controlador de motores L298n y 1 placa Arduino uno R3).

Dicho prototipo carece de un sistema de control, por lo que si se enciende, está programado para moverse constantemente hacia adelante.

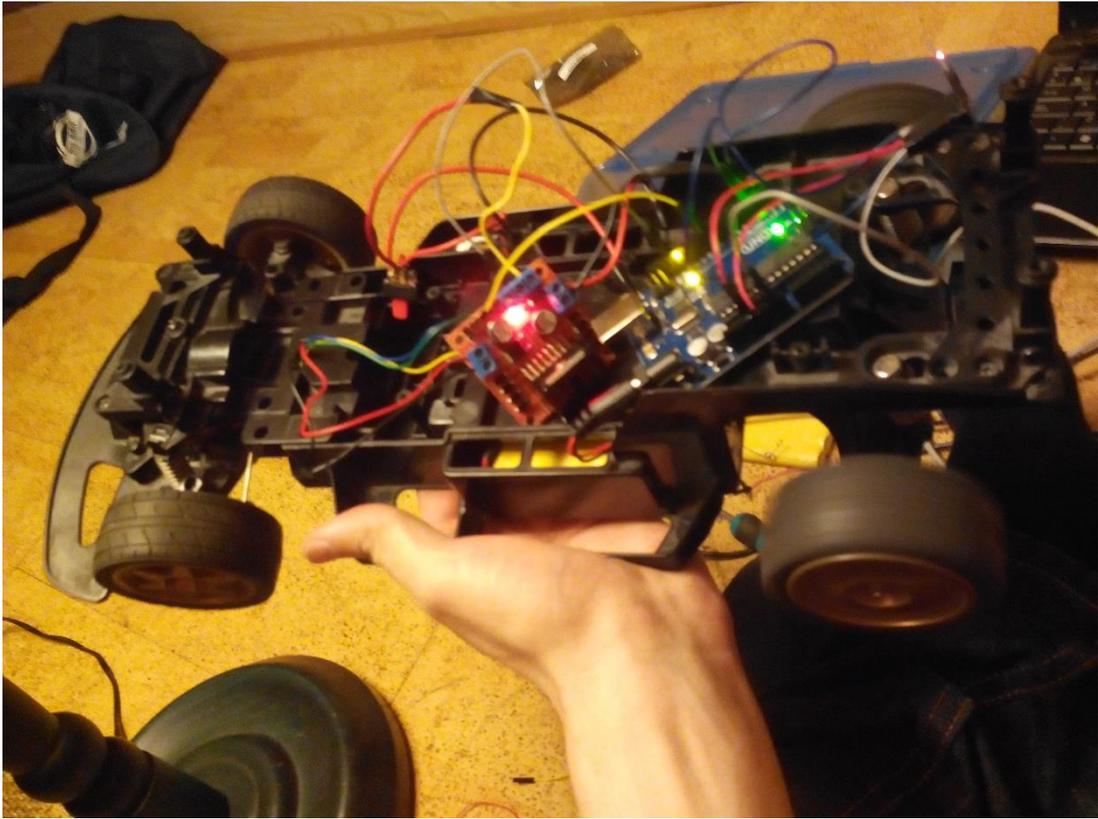


Figura 17 Prototipo inicial

3.1.1.2 Casos de uso y flujo de eventos

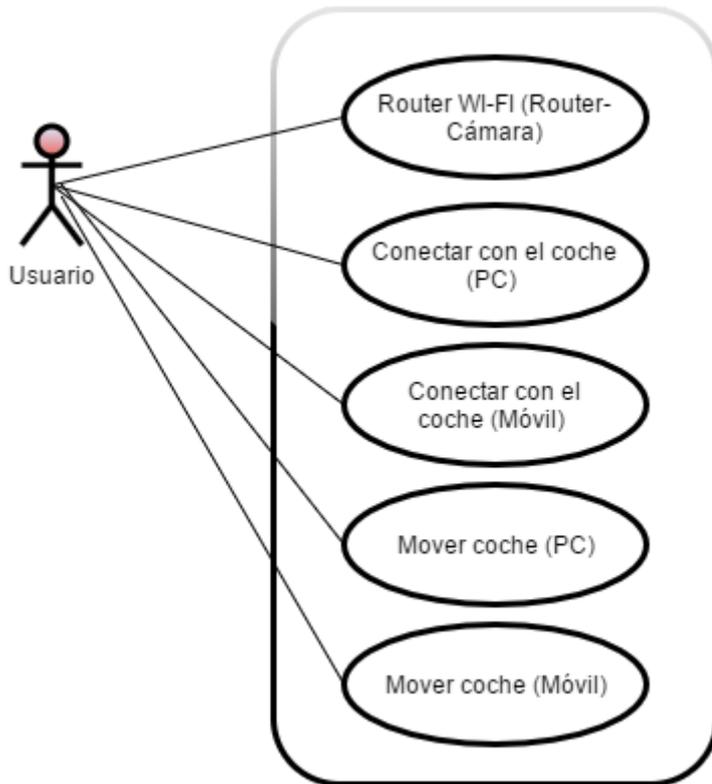


Figura 18 Caso de uso iteración 1

Casos de uso	Flujo de eventos
Router WI-FI (Router-Cámara)	1. Encender la opción de “Zona con cobertura inalámbrica”.
Conectar con el Coche (PC)	Previo: Encender el router WI-FI y el coche. 1. Conectarse a la red WIFI del Coche 2. Introducir la IP y el puerto del coche (Se muestra en “Zona con cobertura inalámbrica” del Router-Cámara) y conectarse.
Conectar con el Coche (Móvil)	Previo: Encender el router WI-FI y el coche. 1. Conectarse a la red WIFI del Coche 2. Introducir la IP y el puerto del coche (Se muestra en “Zona con cobertura inalámbrica” del Router-Cámara) y conectarse.
Mover coche (PC)	Previo: Encender el router WI-FI, el coche y conectar el coche a la aplicación PC por TCP. 1. El usuario usa los controles del mando para establecer

	<p>velocidad, ángulo de giro y dirección del vehículo.</p> <p>2. El coche recibe por WIFI los parámetros y se mueve en función de las órdenes recibidas.</p>
Mover coche (Móvil)	<p>Previo: Encender el router WI-FI, el coche y conectar el coche a la aplicación Móvil-Mando por TCP.</p> <p>1. El usuario usa el Slidebar, el acelerómetro y los botones para establecer velocidad, ángulo de giro y dirección del vehículo.</p> <p>2. El coche recibe por WIFI los parámetros y se mueve en función de las órdenes recibidas.</p>

Tabla 5 Flujo de eventos iteración 1

3.1.2 Iteración 2

Implementación de los sistemas de captación de datos.

3.1.2.1 Prototipo Actual

En esta iteración ya disponemos de un vehículo controlable y capaz de comunicarse con cada aplicación mediante WI-FI (En esta iteración, hemos sustituido el Arduino uno R3 por un Arduino WEMOS D1 que incorpora un módulo WI-Fi ESP8266).

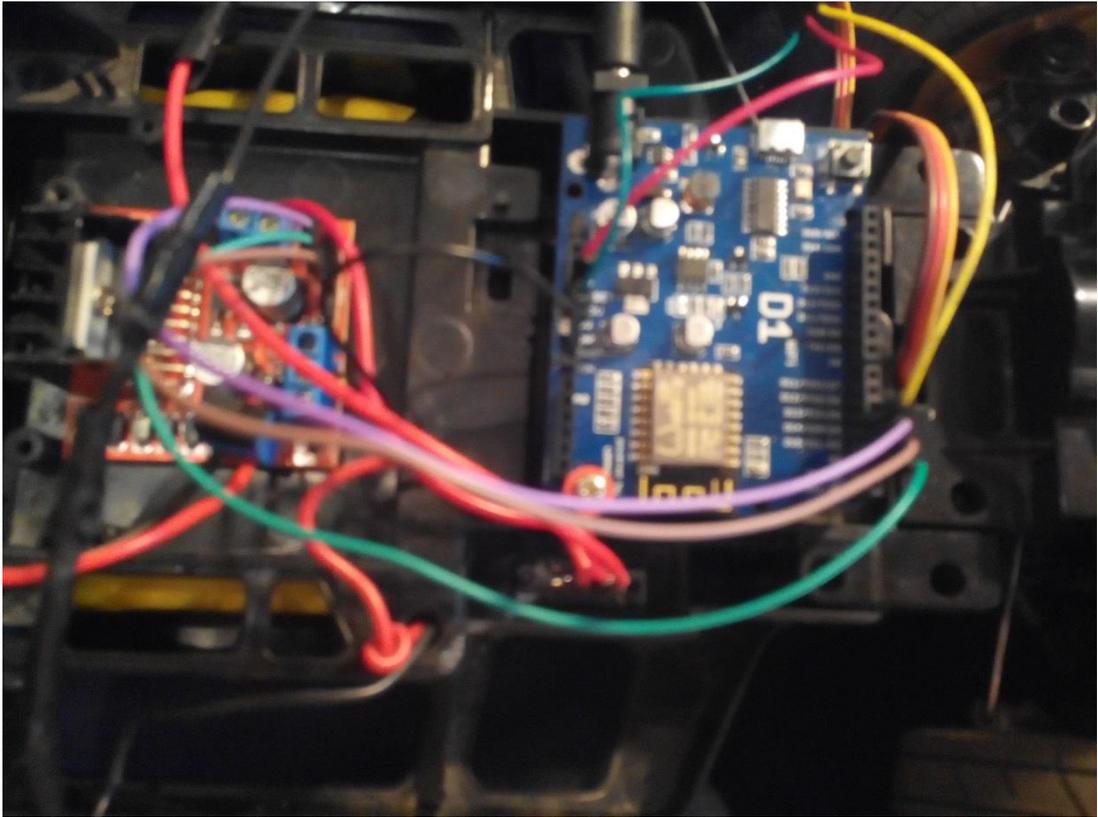


Figura 19 Prototipo inicio iteración 2

3.1.2.2 Casos de uso y flujo de eventos

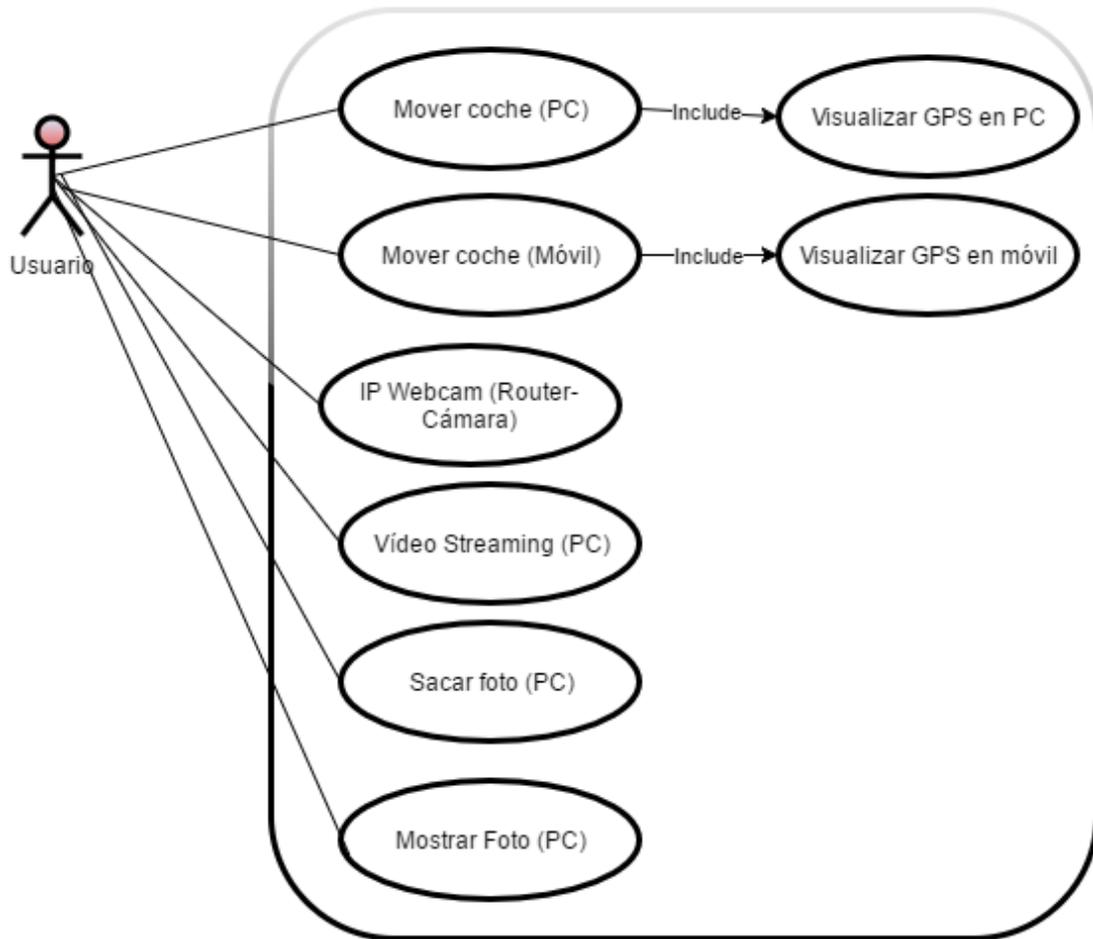


Figura 20 Caso de uso iteración 2

Casos de uso	Flujo de eventos
IP WebCam	1. Encender el programa IP Webcam
Vídeo Streaming (Aplicación PC)	Previo: Encender IP Webcam. 1. Conectarse a la IP de la Cámara IP 2. Se muestra en la aplicación el Streaming de la cámara
Sacar foto (PC)	Previo: Encender IP Webcam y hacer Video Streaming. 1. Pulsar el botón de sacar Foto

	<ol style="list-style-type: none"> 2. La Cámara envía la foto capturada a la aplicación 3. La aplicación almacena la foto en un directorio.
Mostrar fotos (PC)	<ol style="list-style-type: none"> 1. El usuario pulsa el botón de mostrar directorio. 2. La aplicación muestra la carpeta con las fotos obtenidas y los mapas exportados
Visualizar GPS	<p>Previo: Tener la aplicación conectada al coche</p> <ol style="list-style-type: none"> 3. El coche envía las coordenadas a la aplicación 4. La aplicación muestra las coordenadas por pantalla.

Tabla 6 Flujo de eventos iteración 2

3.1.3 Iteración 3

Desarrollo del Mapa que mostrará el recorrido realizado por el vehículo y los datos captados.

3.1.3.1 Prototipo Actual

Antes de comenzar la iteración 3 ya disponemos de un prototipo con sistemas de comunicación y captación de datos como se muestra en la **Figura 21 Prototipo inicio iteración 3**.



Figura 21 Prototipo inicio iteración 3

3.1.3.2 Casos de uso y flujo de eventos

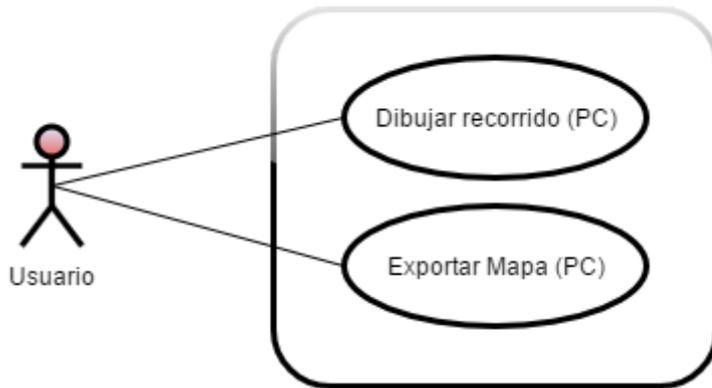


Figura 22 Caso de uso iteración 3

Figura CU3-F-2016:

Casos de uso	Flujo de eventos
Dibujar recorrido (PC)	<p>Previo: Tener la aplicación conectada al coche y tener coordenadas validas</p> <ol style="list-style-type: none"> 1. La aplicación recibir coordenadas y dibuja en el mapa las coordenadas interconectadas entre sí por una línea.
Exportar mapa (PC)	<p>Previo: Tener la aplicación conectada al coche y tener coordenadas validas Conectarse a la IP de la Cámara IP</p> <ol style="list-style-type: none"> 1. EL usuario pulsa el botón de “Exportar mapa” 2. Se genera un fichero “Fecha”.GPX en el directorio de la aplicación.

Tabla 7 Flujo de eventos iteración 3

3.2 Resultado Final

Una vez terminado el proceso de ejecución de las diferentes iteraciones, se tiene como resultado un prototipo que cubre todas las necesidades establecidas previamente

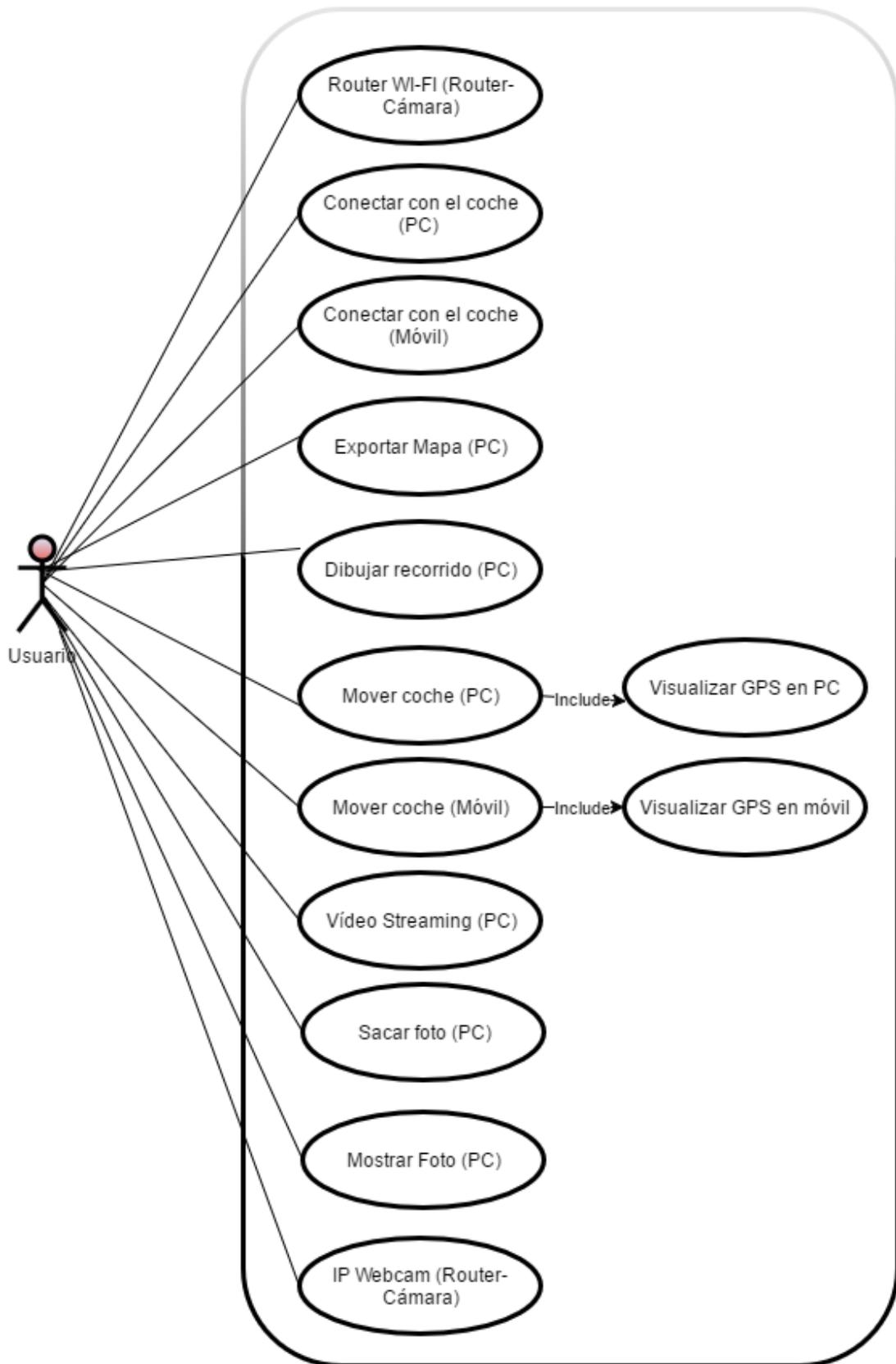


Figura 23 Caso de uso final

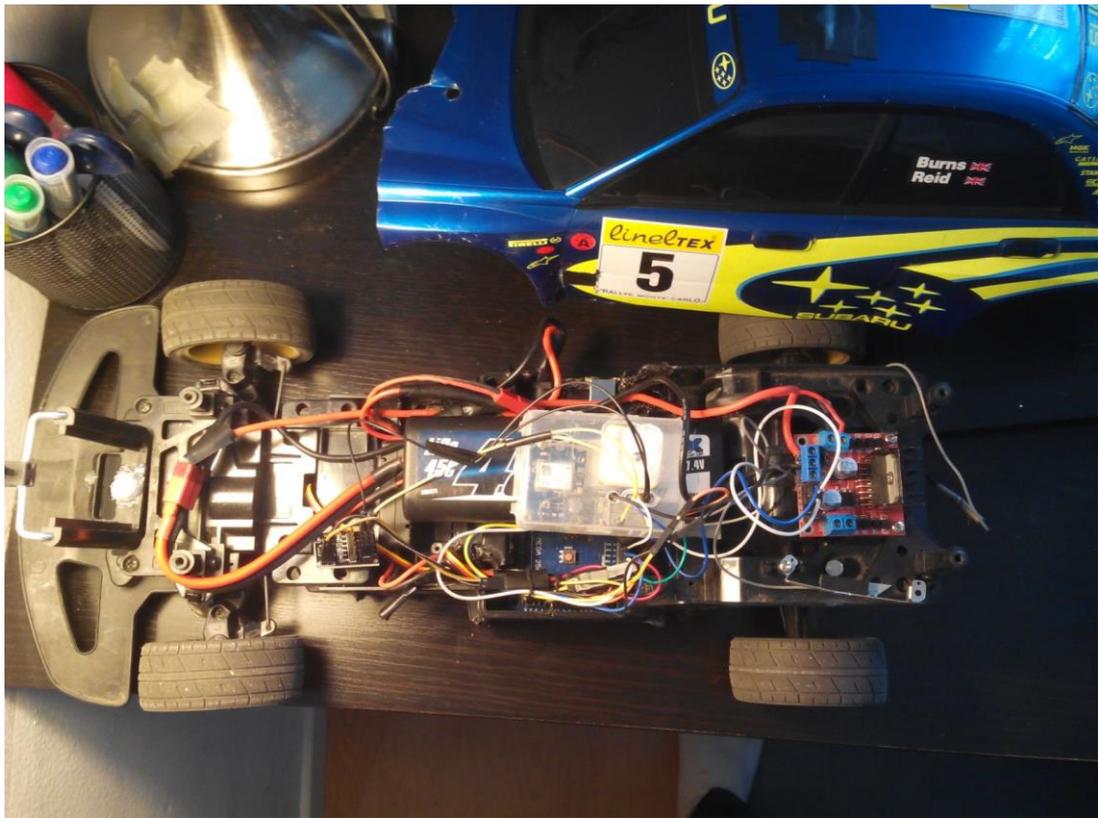


Figura 24 Prototipo final

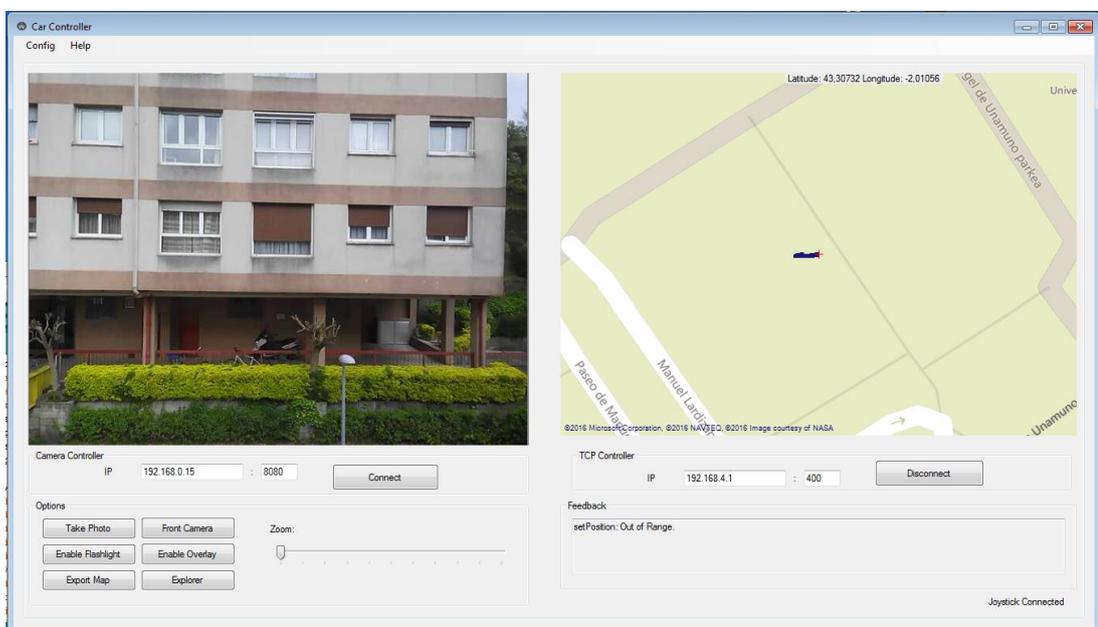


Figura 25 Aplicación de PC (Interfaz de control)

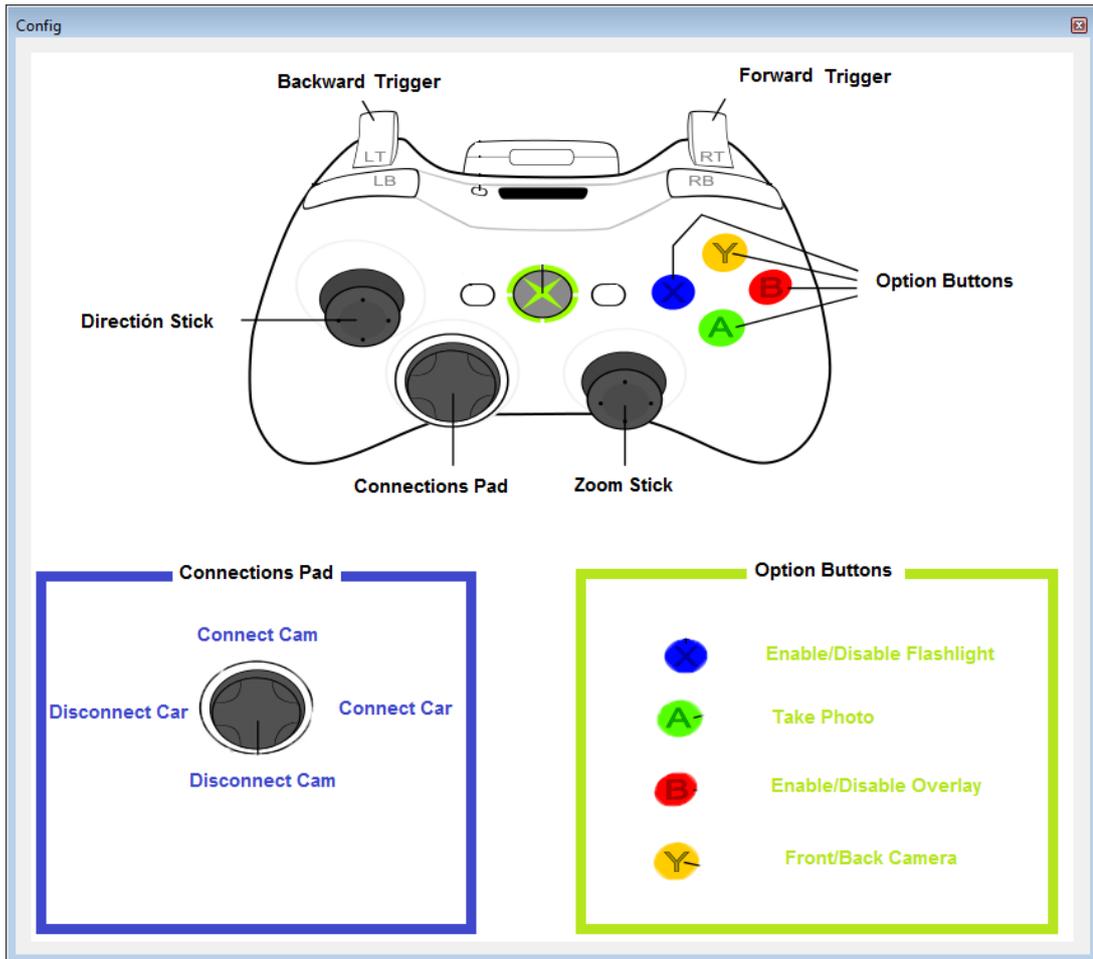


Figura 26 Aplicación de PC (Interfaz de configuración)



Figura 27 Aplicación de PC (Interfaz de ayuda)



Figura 28 Aplicación móvil (Interfaz de control)



Figura 29 Aplicación móvil (Interfaz de conexión)

4 Diseño

Este capítulo va a explicar la arquitectura que he aplicado tanto a nivel hardware como software.

Después se muestra el modelo de datos con el que se alojan los elementos que forman los mapas generados.

Para terminar, se muestra un diagrama de secuencia de varias funcionalidades para entender la aplicación.

4.1 Arquitectura software

Antes de nada, es preciso aclarar los diferentes dispositivos y los nombres que han adquirido para evitar confusiones.

Para poder intercomunicar el coche con la aplicación de ordenador y la aplicación móvil **se emplea un dispositivo móvil** que además incorpora una aplicación para poder usarse como cámara IP. Este dispositivo recibe el nombre de **Router – Cámara**.

En caso de necesitar testear el vehículo se ha desarrollado una aplicación móvil que permite, en el dispositivo Android en el que se instale controlar el vehículo. Este dispositivo recibe el nombre de Mando móvil y **no es el mismo** que el Router – Cámara. Tampoco es lo mismo el Mando móvil que el mando de xbox360 que se conecta de manera inalámbrica o por cable a la aplicación pc para controlar el coche.

Por ende y para hacer mención al ordenador hablamos de PC y para el vehículo: coche, prototipo o vehículo mismamente.

4.2 Arquitectura hardware

El primer diseño de conexiones que se muestra a continuación, corresponde al prototipo inicial del que he partido para realizar este proyecto.

La siguiente figura representa las tecnologías y la arquitectura seguida para la realización de proyecto.

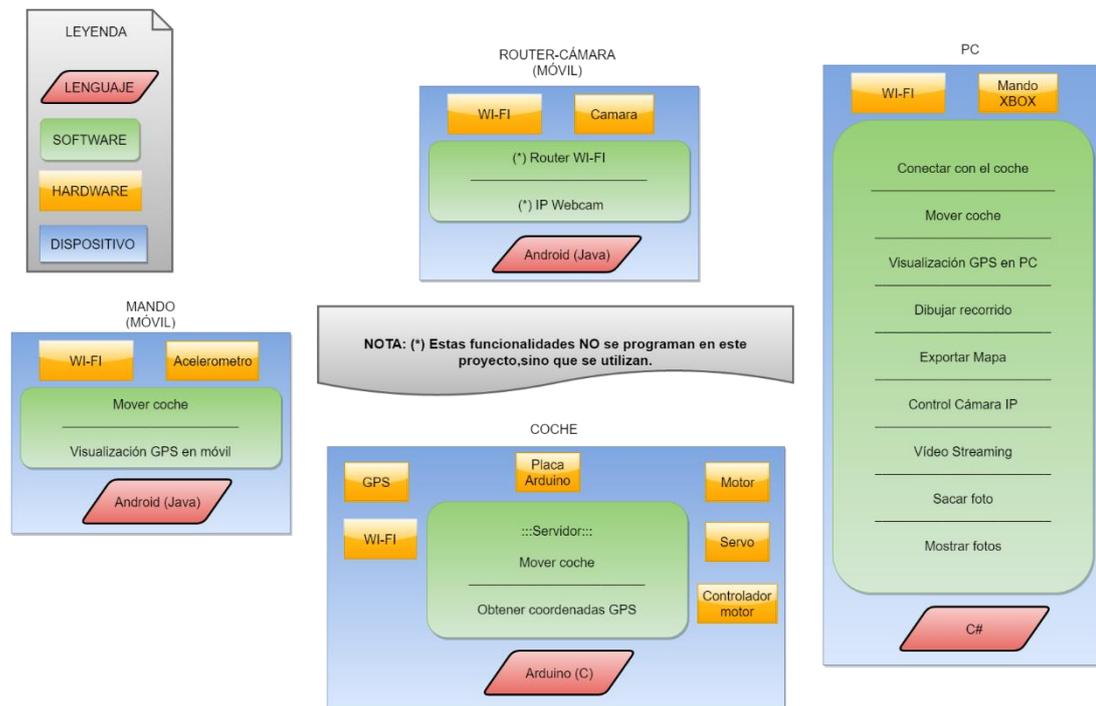


Figura 30 Arquitectura Software

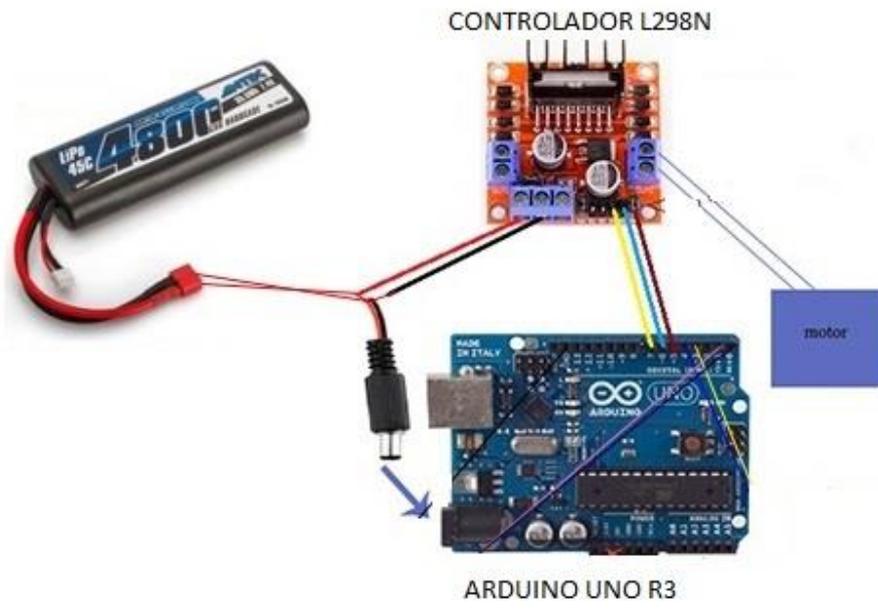


Figura 31 Arquitectura hardware inicial

Después de varias modificaciones, el esquema de conexiones resultante es:

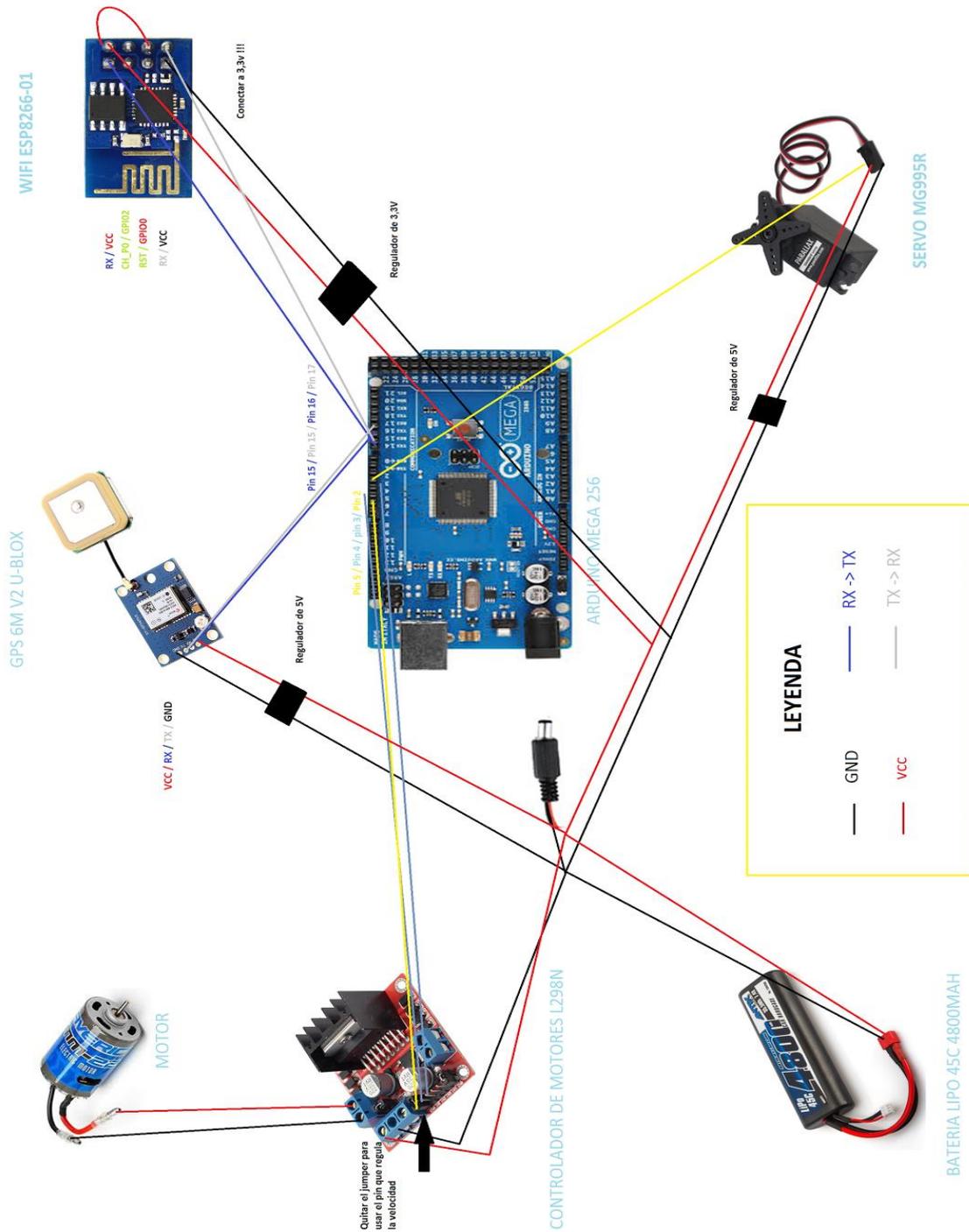


Figura 32 Arquitectura hardware final

4.3 Modelo de datos

Para almacenar las rutas de las fotos tomadas y las coordenadas del recorrido realizado se utiliza un esquema XML pensado para transferir datos GPS entre aplicaciones denominado GPX.

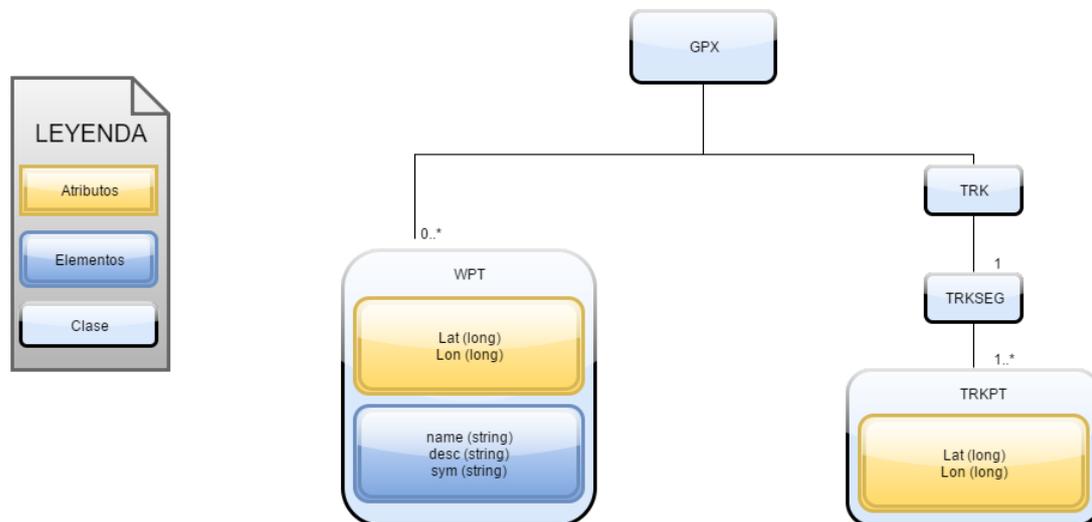


Figura 33 Estructura del esquema GPX

La clase WPT corresponde a los puntos en donde se han sacado las fotos.

La clase TRKPT corresponde a los puntos por donde se ha ido moviendo el coche.

4.4 Diagramas de secuencia

Los diagramas de secuencia, son utilizados para modelar la interacción entre objetos en un sistema. Un diagrama de secuencia muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo. Los diagramas de secuencia, detallan la implementación del sistema para un escenario concreto incluyendo los objetos y clases que se usan para implementar el escenario concreto.

4.4.1 Conectar con el coche

Las siguientes figuras muestran los diagramas de secuencia correspondientes a la conexión entre aplicación y vehículo.

La **Figura 34** Diagrama de secuencias Conectar con el coche (Aplicación PC) muestra el diagrama correspondiente a la aplicación de PC, en cambio, la **Figura 35** Diagrama de

secuencias Conectar con el coche (Aplicación Móvil)muestra el diagrama correspondiente a la aplicación móvil.

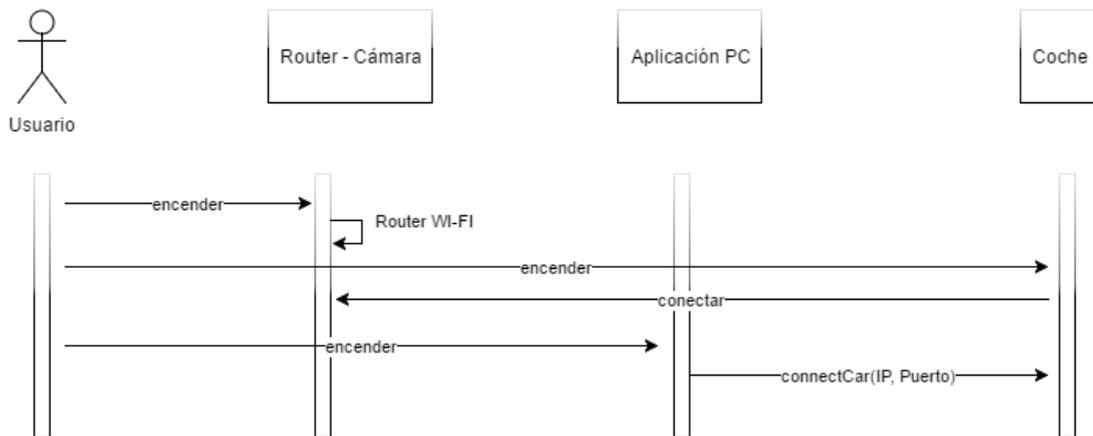


Figura 34 Diagrama de secuencias Conectar con el coche (Aplicación PC)

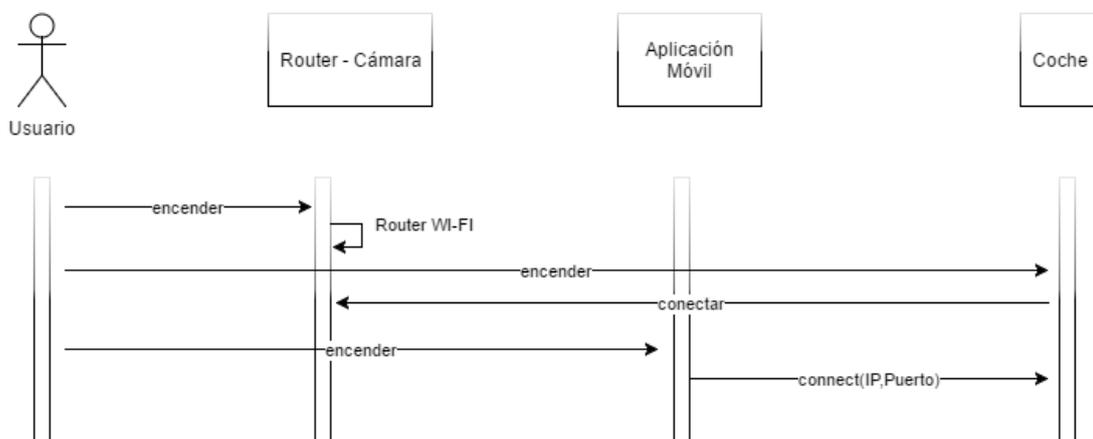


Figura 35 Diagrama de secuencias Conectar con el coche (Aplicación Móvil)

4.4.2 Mover coche y obtener coordenadas

Las siguientes figuras muestran los diagramas de secuencia correspondientes al sistema en el que el usuario mueve el coche y recibe las coordenadas GPS.

La Figura 36 Diagrama de secuencia Mover coche y obtener coordenadas (Aplicación PC)muestra el diagrama correspondiente a la aplicación de PC, en cambio, la Figura 37

Diagrama de secuencias Mover coche y obtener coordenadas (Aplicación Móvil) muestra el diagrama correspondiente a la aplicación móvil.

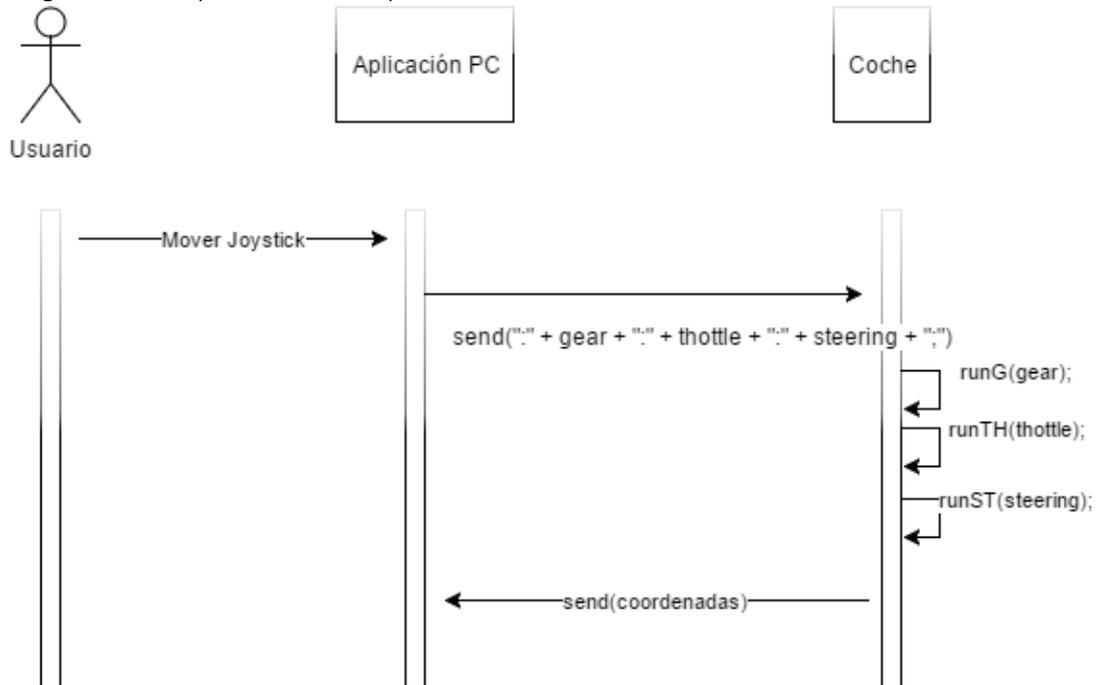


Figura 36 Diagrama de secuencia Mover coche y obtener coordenadas (Aplicación PC)

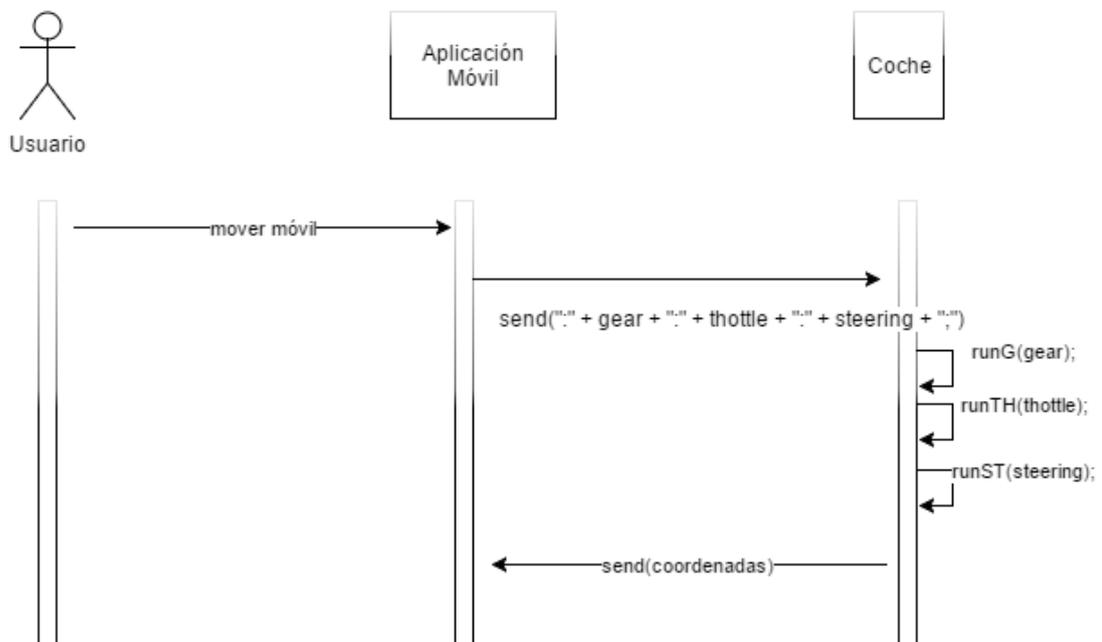


Figura 37 Diagrama de secuencias Mover coche y obtener coordenadas (Aplicación Móvil)

4.4.3 Vídeo Streaming

La **Figura 38** Diagrama de secuencias Vídeo streaming (Aplicación PC) muestra los diagramas de secuencias correspondientes al sistema en el que la aplicación PC obtiene el streaming vídeo.

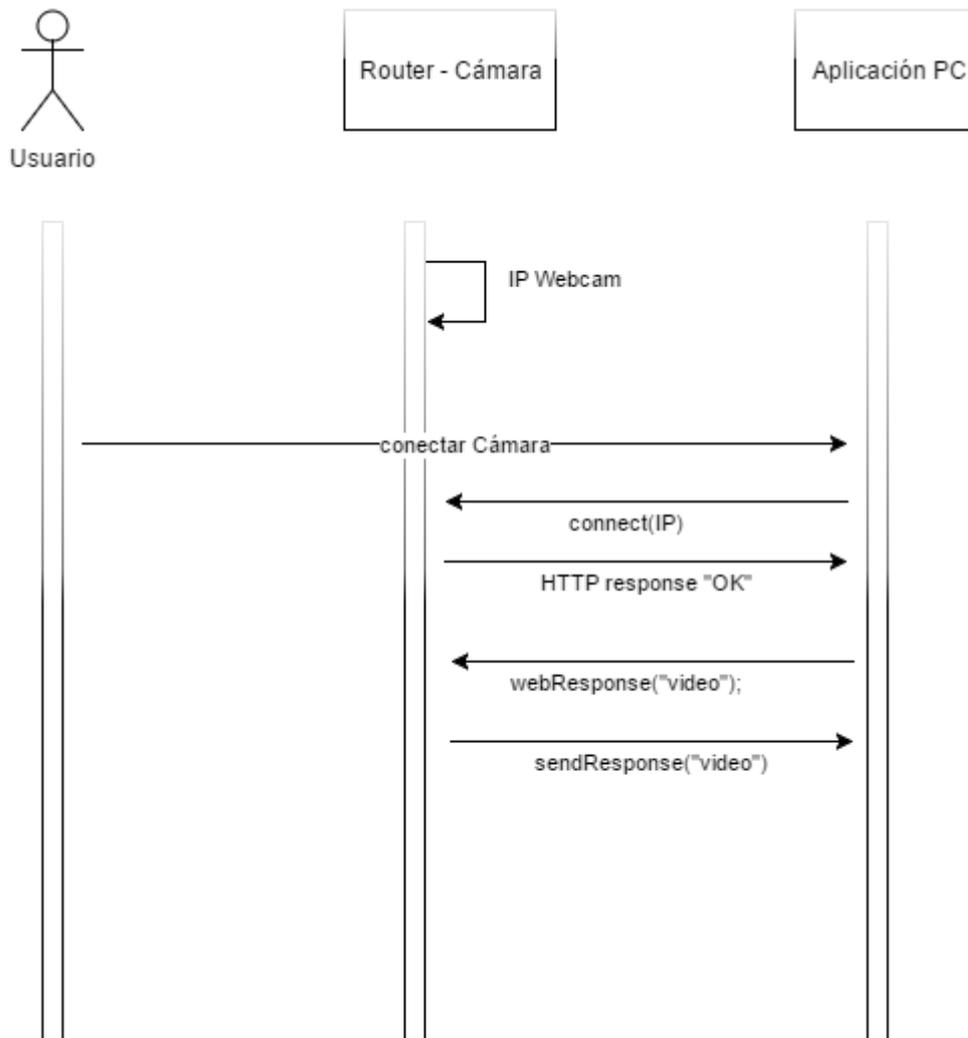


Figura 38 Diagrama de secuencias Vídeo streaming (Aplicación PC)

5 Implementación

En este capítulo se explica cómo se ha estructurado la implementación para entender el patrón de arquitectura seguido en el desarrollo de la aplicación y se mencionan los diferentes errores y soluciones que han ido surgiendo a lo largo de proyecto.

5.1 Detalles de implementación

A continuación se muestran los detalles de implementación más destacados de cada dispositivo empleado.

5.1.1 Router- Cámara

El router-cámara, dispone de 2 aplicaciones de terceros que he implementado en el móvil para poder usar el móvil como cámara IP (aplicación IP Webcam) y como router (Aplicación Zona WI-Fi incorporada en los móviles Android) a la vez.

5.1.2 Móvil

Para controlar el movimiento del vehículo empleo el Acelerómetro que viene incorporado en el propio móvil. Su implementación se muestra en la **Figura 39** Acelerómetro.

```
@Override
public void onSensorChanged(SensorEvent evento) {
    synchronized (this){
        float[] masData;
        int y;
        if(evento.sensor.getType() == Sensor.TYPE_ACCELEROMETER) {
            masData = evento.values; //masData[0] == x, masData[1] == y, masData[2] == z
            y = (int) masData[1];
            if (position !=y) { // Dont send repeat values
                steering = Integer.toString(y);
                //accelerometer.setText("Steering: " + steering);
                if (( y <= 6) && ( y >= -6)) {
                    sendCMD();
                }
                position = y;
            }
        }
    }
}
```

Figura 39 Acelerómetro

Cuando el móvil, situado de manera horizontal, se mueve hacia la posición Y+ o Y- se comprueba que la nueva Y no es igual a la vieja Y. Esto permite limitar la cantidad de mensajes que se envían hacia el coche para evitar así un cuello de botella.

También se comprueba que la Y es menor o igual que 6 o mayor o igual que -6 para evitar un excesivo giro y que el control de vehículo sea más cómodo.

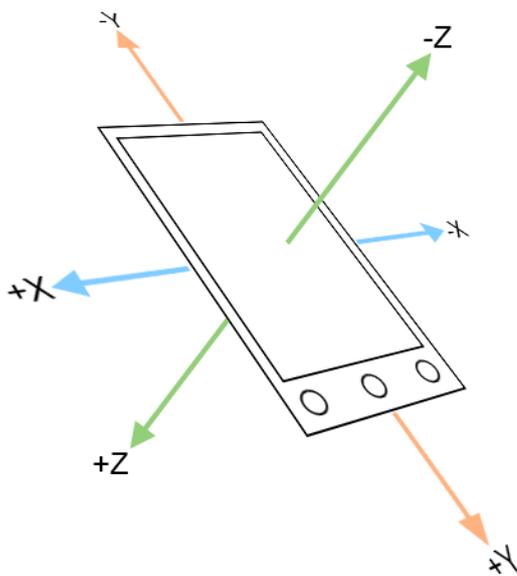


Figura 40 Posiciones móvil [A-F-2016]

5.1.3 PC

En la aplicación de PC, he creído conveniente destacar la implementación del recorrido.

Antes de nada es preciso mencionar que, para ello, se emplean dos funciones además de los objetos contenedores de las rutas, marcas y del mapa:

1) Objetos contenedores:

```
private GMapOverlay overlay;  
private GMapRoute route;  
private GMarkerGoogle marker;
```

2) addMarker(String text, Coords coords): Dado una coordenada (latitud y longitud) y un texto (la ruta de la imagen que queremos almacenar en dicho marcador), genera un marcador en el controlador del mapa y lo guarda en el overlay.Markers

```

//Add marker to the map
public void addMarker(String text, Coords coords)
{
    if (!coords.isWrongCoords())
    {
        Console.WriteLine("Marker");

        this.dispatcher.Invoke(DispatcherPriority.Normal, (Action)() =>
        {
            marker = new GMarkerGoogle(coords.getPosition(), GMarkerGoogleType.green);
            marker.ToolTipText = text;
            overlay.Markers.Add(marker);
            addRoute(coords);
        }));
    }
}

```

Figura 41 addMarker

3) addRoute(Coords coords): Dado una coordenada (latitud y longitud), almacena dicha ubicación en un array de rutas (overlay.Routes) y **dibuja dicho array en el mapa, borrando el anterior.**

```

//Add new position to the map, and connect existing points
public void addRoute(Coords coords)
{
    if (!coords.isWrongCoords())
    {
        Console.WriteLine("Route");
        this.dispatcher.Invoke(DispatcherPriority.Normal, (Action)() =>
        {
            if (!isRepeatLastCoord(coords.getPosition())) // dont save last same coords
            {
                route.Points.Add(coords.getPosition());
                overlay.Routes.Add(route);
            }
        }));
    }
}

```

Figura 42 addRoute

Este método es un poco engorroso, y me ha dado muchos problemas de eficiencia, ya que cuanto mayor es la ruta, mayor es el esfuerzo que realiza la aplicación para dibujarla y a su vez mayor es el tiempo de respuesta del resto de funciones que se ejecutan en el programa (movimiento, botones, etc..)

Para actualizar las rutas (addRoute) se emplea un timer de 5 segundos encargado de ejecutar la función SI las coordenadas son correctas (si no son latitud = 1000, longitud = 1000 o si no son iguales a las anteriores coordenadas).

En cambio, los marcadores (addMarker) solo se añaden cuando se presiona el botón de sacar foto, ya sea en el mando o/y en la interfaz de la propia aplicación.

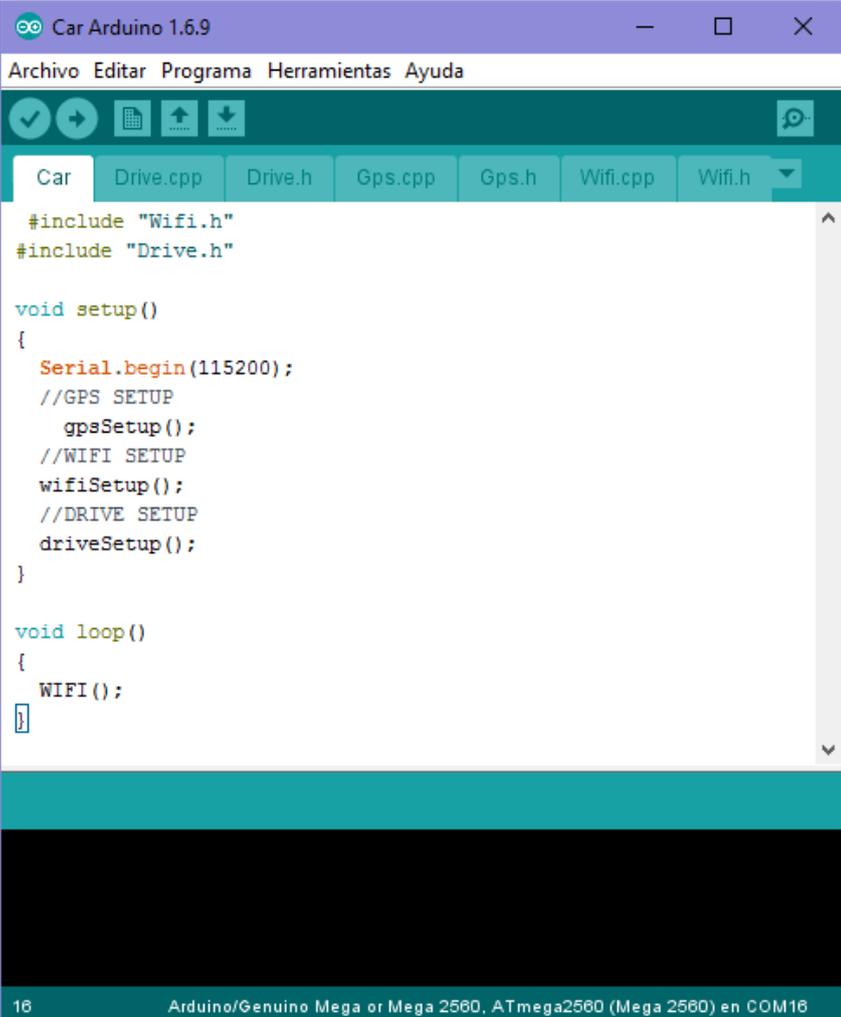
5.1.4 Coche

El coche cumple la función de servidor, ya que se encarga de recoger peticiones de la aplicación móvil o PC y reaccionar en función de la misma.

Está programado en .ino (una variante de c) mediante la herramienta ArduinoIDE.

Una vez desarrollado el programa, este se carga en la tarjeta Arduino quien se encarga de comunicarse entre los diversos módulos instalados.

En la **Figura 43** Arduino main se muestran los ficheros que componen el coche.



```
Car Arduino 1.6.9
Archivo Editar Programa Herramientas Ayuda
Car Drive.cpp Drive.h Gps.cpp Gps.h Wifi.cpp Wifi.h
#include "Wifi.h"
#include "Drive.h"

void setup()
{
  Serial.begin(115200);
  //GPS SETUP
  gpsSetup();
  //WIFI SETUP
  wifiSetup();
  //DRIVE SETUP
  driveSetup();
}

void loop()
{
  WIFI();
}
```

16 Arduino/Genuino Mega or Mega 2560, ATmega2560 (Mega 2560) en COM16

Figura 43 Arduino main

Carino, es el fichero principal que, primero realiza la función Setup() y luego, la función loop() donde se mantiene iterando constantemente hasta cerrar el controlador Arduino.

```
void WIFI()
{
  client = server.available();
  while (client) { //When receive data...
    String trash = client.readStringUntil(':');
    String g = client.readStringUntil(':');
    int th = client.readStringUntil(':').toInt();
    int st = client.readStringUntil(':').toInt();
    //Serial.println(":"+g+":"+th+":"+st+");");
    runG(g);
    runTH(th);
    runST(st);
    client.print(getCoords());
  }
  calm();
}
```

Figura 44 Función WI-FI de coche

En la función WIFI(), se realizan las tareas de movimiento y de envío de coordenadas.

La aplicación, ya sea wifi o móvil envía la siguiente cadena de datos:

```
(":" + gear + ":" + thottle + ":" + steering + ");");
```

Siendo gear (dirección), thottle (velocidad), steering (giro).

5.2 Problemas y soluciones

1. **Problema:** Cuando el mando (Android) manda muchos datos al coche (Arduino), este, se satura dando una excepción de un intento de escribir en un espacio de memoria restringido. Este fallo ocurre por un Split [SP-D-2016] en el Arduino que, forma un cuello de botella en los mensajes entrantes provocando dicha excepción y como consecuencia de ello, la desconexión y el reinicio inmediato del WI-FI.

Solución: Usar la función .readStringUntil() para separar las cadenas de texto recibidas.

2. **Problema:** El WI-FI no se comunica a una velocidad fluida con el Software.

Solución: Incrementar los baudios.

3. **Problema:** Solo se puede tener un SoftwareSerial o sino se ralentiza excesivamente el Arduino.

Solución: Pasar a un modelo más potente de Arduino.

4. **Problema:** La cámara abre múltiples sesiones.

Solución: Añadir un delay [D-D-2016] después de conectar la cámara.

5. **Problema:** Al cargar excesivas rutas en el controlador del mapa, se ralentiza el movimiento del coche y el streaming de la cámara.

Solución: Evitar guardar coordenadas repetidas e incrementar el intervalo en el que se recarga el controlador.

6. **Problema:** El programa se ralentiza con cada acción.

Solución: Añadir threads [T-D-2016] a todas las funciones con alto coste de ejecución.

7. **Problema:** El Gps da coordenadas incorrectas de vez en cuando.

Solución: añadir al GPS un temporizador de 2 segundos.

8. **Problema:** Errores de objetos en uso a la hora de agregar rutas al controlador de mapa.

Solución: Añadir dispatchers.invoke a las funciones que guardan y actualizan el mapa.

9. **Problema:** El uso de interrupciones en Arduino, genera problemas en la comunicación entre aplicación y vehículo.

Solución: No usar interruptores.

6 Pruebas

Este apartado recoge y explica las diversas pruebas realizadas para la optimización y correcto funcionamiento del prototipo MapCar.

6.1 Prototipo de pruebas

Para estudiar los diversos sistemas de comunicación y recolección de datos, he realizado un prototipo de pruebas, para trabajar con los módulos por separado sin tener que realizar cambios en el vehículo a emplear.

La **Figura 45** muestra el prototipo de pruebas con un montaje básico. (Un Arduino Uno R3, un controlador de motores L298n, 3 ruedas, un chasis, 2 motores y un módulo WI-FI).

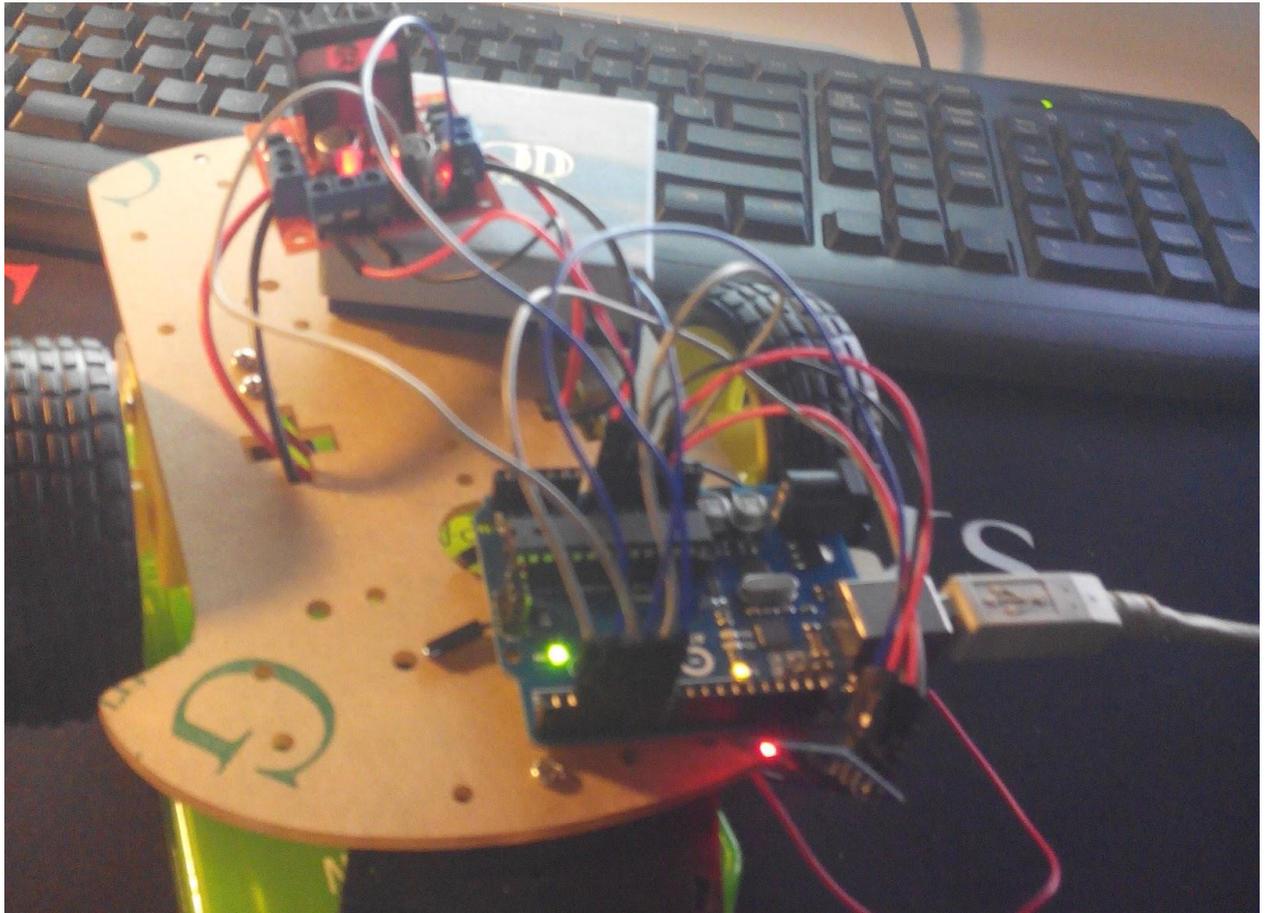


Figura 45 Prototipo de pruebas

6.2 Pruebas en las funcionalidades

Las pruebas de software realizadas, mayormente son de tipo caja negra (se tienen en cuenta los valores de entrada del usuario que interactúa con la aplicación y se comprueba que el resultado de las operaciones sea el esperado).

Muchas funciones no tienen feedback cuando se produce error, ya que se pretende que el usuario este pendiente únicamente de la cámara cuando se controla el vehículo. Únicamente se han añadido feedback a los errores de conexión.

6.2.1 Prueba P1: Conexión coche (PC)

Prueba 1: El usuario intenta conectarse al vehículo dejando vacío el campo de la IP. En este caso la aplicación debe mostrar mensaje de error.

Entrada: IP vacía.

Salida esperada: TCP connect: System.Net.Sockets.SocketException

Salida real: TCP connect: System.Net.Sockets.SocketException

Prueba 2: El usuario intenta conectarse al vehículo dejando vacío el campo del puerto. En este caso la aplicación debe mostrar mensaje de error.

Entrada: Puerto vacío.

Salida esperada: TCP connect: System.Net.Sockets.SocketException

Salida real: SE CIERRA LA APLICACIÓN

Prueba 3: El usuario introduce correctamente los datos pero el vehículo esta desconectado.

Entrada: IP y puerto correcto.

Salida esperada: TCP connect: System.Net.Sockets.SocketException

Salida real: TCP connect: System.Net.Sockets.SocketException

Prueba 4: El usuario introduce correctamente los datos y la aplicación se conecta al vehículo.

Entrada: IP y puerto correcto.

Salida esperada: Controlar el vehículo.

Salida real: controlar el vehículo.

6.2.2 Prueba Explorar directorio:

Prueba 1: El usuario pulsa el botón de explorer sin antes haber exportado un mapa o haber sacado una foto.

Entrada: Presionar botón Explorer.

Salida esperada: First you need take photo or export map!

Salida real: First you need take photo or export map!

Prueba 1: El usuario pulsa el botón de explorer habiendo exportado un mapa y/o haber sacado una foto.

Entrada: Presionar botón Explorer.

Salida esperada: Se abre la carpeta contenedora.

Salida real: Se abre la carpeta contenedora.

6.2.3 Prueba Visualización GPS (PC):

Prueba 1: El usuario lleva el coche por una zona sin cobertura GPS (después de haberlo tenido encendido y que la aplicación envíe coordenadas válidas).

Entrada: -

Salida esperada: Latitude: 1000 Longitude: 1000

Salida real: Latitude: 1000 Longitude: 1000

Prueba 2: El usuario lleva el coche por una zona sin con coordenadas validas después de mandarlo por una zona sin cobertura GPS.

Entrada: -

Salida esperada: coordenadas buenas.

Salida real: coordenadas buenas.

6.2.4 Prueba Dibujar recorrido:

Prueba 1: El usuario no tiene el coche vinculado a la aplicación.

Entrada: -

Salida esperada: No se dibuja nada.

Salida real: No se dibuja nada.

Prueba 2: El usuario tiene el coche vinculado a la aplicación pero no hay coordenadas válidas.

Entrada: -

Salida esperada: No se dibuja nada.

Salida real: No se dibuja nada.

Prueba 3: El usuario tiene el coche vinculado a la aplicación y hay coordenadas válidas.

Entrada: -

Salida esperada: Se dibuja nuevamente la ruta y las marcas.

Salida real: Se dibuja nuevamente la ruta y las marcas.

6.2.5 Prueba Exportar Mapa:

Prueba 1: El usuario pulsa el botón de exportar mapa cuando la cámara no está conectada.

Entrada: Pulsar botón Export Map.

Salida esperada: -

Salida real: -

Prueba 2: El usuario pulsa el botón de exportar mapa cuando la cámara está conectada.

Entrada: Pulsar botón sacar foto.

Salida esperada: -

Salida real: -

Prueba 3: El usuario pulsa el botón de exportar mapa cuando la cámara está conectada y el coche está vinculado a la aplicación pero aún no han salido coordenadas válidas.

Entrada: Pulsar botón sacar foto.

Salida esperada: -

Salida real: -

Prueba 4: El usuario pulsa el botón de exportar mapa cuando la cámara está conectada y el coche está vinculado a la aplicación y han salido coordenadas válidas.

Entrada: Pulsar botón sacar foto.

Salida esperada: Exportar mapa.

Salida real: Exportar mapa.

6.2.6 Prueba Control Cámara:

Prueba 1: El usuario pulsa el botón de sacar foto cuando la cámara no está conectada.

Entrada: Pulsar botón sacar foto.

Salida esperada: -

Salida real: -

Prueba 2: El usuario pulsa el botón de sacar foto cuando la cámara está conectada.

Entrada: Pulsar botón sacar foto.

Salida esperada: Foto.

Salida real: Foto.

6.2.7 Prueba Vídeo Streaming:

Prueba 1: El usuario intenta conectarse a la cámara dejando vacío el campo de la IP. En este caso la aplicación debe mostrar mensaje de error.

Entrada: IP vacía.

Salida esperada: Camera streaming: System.NullReferenceException

Salida real: Camera streaming: System.NullReferenceException

Prueba 2: El usuario intenta conectarse a la cámara dejando vacío el campo del puerto. En este caso la aplicación debe mostrar mensaje de error.

Entrada: Puerto vacío.

Salida esperada: Camera streaming: System.NullReferenceException

Salida real: Camera streaming: System.NullReferenceException

Prueba 3: El usuario introduce correctamente los datos y pero la cámara esta desconectada.

Entrada: IP y puerto correcto.

Salida esperada: Camera streaming: System.NullReferenceException

Salida real: Camera streaming: System.NullReferenceException

Prueba 4: El usuario introduce correctamente los datos y la aplicación se conecta a la cámara del vehículo.

Entrada: IP y puerto correcto.

Salida esperada: Streaming vídeo.

Salida real: Streaming vídeo.

6.2.8 Prueba Sacar foto:

Prueba 1: El usuario pulsa el botón de sacar foto cuando la cámara no está conectada.

Entrada: Pulsar botón Take Photo.

Salida esperada: nada.

Salida real: nada.

Prueba 2: El usuario pulsa el botón de sacar foto cuando la cámara está conectada.

Entrada: Pulsar botón Take Photo.

Salida esperada: Foto.

Salida real: Foto.

6.2.9 Prueba Mostrar foto:

Prueba 1: El usuario clic en un marcador.

Entrada: Pulsar botón sacar foto.

Salida esperada: Se muestra la foto en el controlador del mapa.

Salida real: Se muestra la foto en el controlador del mapa.

6.2.10 Prueba Conexión coche (móvil):

Prueba 1: El usuario intenta conectarse al vehículo dejando vacío el campo de la IP. En este caso la aplicación debe mostrar mensaje de error.

Entrada: IP vacía.

Salida esperada: Coudn't get I/O for the connection. Connection refused.

Salida real: Coudn't get I/O for the connection. Connection refused.

Prueba 2: El usuario intenta conectarse al vehículo dejando vacío el campo del puerto. En este caso la aplicación debe mostrar mensaje de error.

Entrada: Puerto vacío.

Salida esperada: Coudn't get I/O for the connection. Connection refused.

Salida real: SE CIERRA LA APLICACIÓN

Prueba 3: El usuario introduce correctamente los datos pero el vehículo esta desconectado.

Entrada: IP y puerto correcto.

Salida esperada: Coudn't get I/O for the connection. Conection refused.
Salida real: Coudn't get I/O for the connection. Conection refused.

Prueba 4: El usuario introduce correctamente los datos y la aplicación se conecta al vehículo.

Entrada: IP y puerto correcto.

Salida esperada: Controlar el vehículo.

Salida real: controlar el vehículo.

6.2.11 Prueba Visualización GPS (móvil):

Prueba 1: El usuario lleva el coche por una zona sin cobertura GPS (después de haberlo tenido encendido y que la aplicación envié coordenadas válidas).

Entrada: -

Salida esperada: Latitude: 1000 Longitude: 1000

Salida real: Latitude: 1000 Longitude: 1000

Prueba 2: El usuario lleva el coche por una zona sin con coordenadas validas después de mandarlo por una zona sin cobertura GPS.

Entrada: -

Salida esperada: Coordenadas buenas.

Salida real: Coordenadas buenas.

6.3 Pruebas en el hardware

6.3.1 Prueba de autonomía



Figura 46 Batería 800mAh [BA-F-2016]

Dado que partimos de la batería de un coche RC (con una autonomía inicial de 25 minutos), y que, dicho vehículo ha sido dotado de nuevos módulos y funcionalidades, se ve preciso determinar si la autonomía de la batería es óptima para este proyecto.

En caso de que la duración a máximo rendimiento sea inferior a 15 minutos, será necesario buscar módulos que gasten menos batería, optimizar las funcionalidades para limitar el tiempo en que se usan los diversos módulos o en el peor de los casos, (ya que supone una inversión monetaria) adquirir una batería de mayor mAh.

Resultado: Usando el coche a máximo rendimiento (empleando todas las funcionalidades descritas en el apartado de Diseño) la duración de la batería en terreno nivelado es de 8 minutos, mientras que si se controla en terreno desnivelado la duración puede ser inferior a 5 minutos.

Dado que cambiando módulos y/u optimizando el software se conseguiría un aumento de la autonomía poco considerable, se ha visto preciso cambiar a una batería con mucha más autonomía.



Figura 47 Batería 4800mAh [BB-F-2016]

Una vez sustituida la batería por la mostrada en la Figura, y realizada nuevamente la prueba por el mismo circuito que con la primera batería, la autonomía aumenta hasta más de 25 minutos en máximo rendimiento.

7 Seguimiento y control

A continuación se muestra un desglose de las tareas realizadas, cuándo se han llevado a cabo y el tiempo empleado en las mismas.

Tal como se había notificado en el apartado de riesgos, el proyecto ha sufrido una alteración de la planificación inicial, aumentando la duración de las mismas 5 semanas.

Fases	SUB-TAREA			Previsión		Realización	
				Horas	Semanas	Horas	Semanas
Fase 1	Plan de Gestión			7h	S1 - S2	5h	S1 - S2
	Análisis de Requisitos			5h	S1 - S2	5h	S1 - S2
Fase 2	Estudio de tecnologías			24h	S3 – S4	22h	S3 – S4
Fase 3	Iteración 1	Sistema de comunicación	Implementación Módulo RC	7h	S5	8h	S5
			Implementación módulo Bluetooth	5h	S5	6h	S5
			Construcción APK Bluetooth	6h	S6	6h	S6
			Implementación módulo WIFI	12h	S6	15h	S6
			Construcción APK WIFI	16h	S7	16h	S7
	Iteración 2	Sistema de captación de datos	Diseño e implementación de aplicación pc para el control del coche mediante WIFI	22h	S8	18h	S8
			Implementación módulo GPS	10h	S8	7h	S8
			Implementar Sistema	7h	S9		

		de streaming vídeo			18h	S11	
		Capturar fotos y mostrarlas	2h	S10	7h	S12	
		Añadir opciones a la cámara	2h	S10	5h	S12	
	Iteración 3	Mapa	Mostrar el recorrido que realiza el coche	12h	S10	2h	S13
			Generar fichero con el recorrido y la fotos obtenidas	8h	S10 - S11	6h	S13
Fase 4	Memoria			100h	S14 - S18	104h	S14 - S23
Gestión	Seguimiento y Control			1h	S1-S2	3h	S1-S2
				2h	S3 – S4	2h	S3 – S4
				2h	S5 - S13	6h	S5 - S13
				2h	S14 - S18	2h	S14 - S23
TOTAL				284h		302h	

Tabla 8 Seguimiento y control



8.1 ¿Qué es LPS?

Línea de producción del software (LPS) o Software Product Line (SPL).

Es un conjunto de sistemas software, que comparten un conjunto común de características, las cuales satisfacen las necesidades específicas de un dominio o segmento particular de mercado, y que se desarrollan a partir de un sistema común de activos base de una manera preestablecida. [LPS-2016]

Cuando una empresa desarrolla productos para distintos clientes, por ejemplo, una empresa de teléfonos móviles que poseen diferentes equipos de acuerdo a las necesidades, surge el problema de la existencia de diferentes versiones de software lo cual complica la coordinación para el desarrollo y mantenimiento de los productos de software que comparten características similares.

En la producción de software las herramientas de desarrollo IDE y las metodologías ayudaban a agilizar y sistematizar la creación de un único producto, entonces en muchos casos se veía la necesidad de reutilizar la mayor parte del código común entre las diferentes versiones de las aplicaciones.

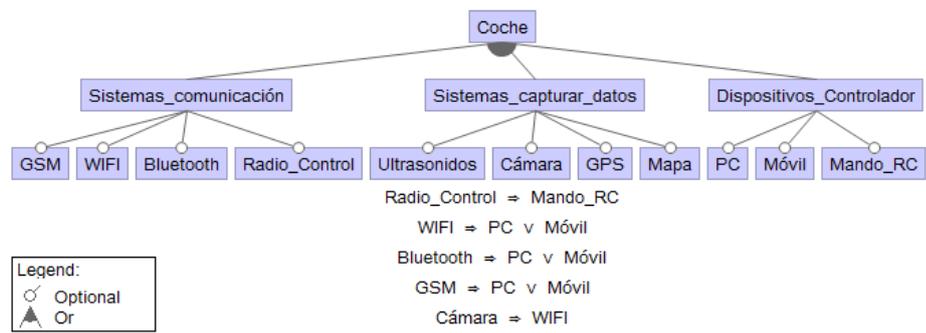
Las líneas de productos de software se centra en la creación de diferentes variaciones de un mismo producto, haciendo que la gestión de producción sea lo más eficaz y eficientemente. Permitiendo reducir costos, tiempo y software de alta calidad. [SPL-2016]

8.2 SPL en MapCar

Dado que se han estudiado varias tecnologías para el desarrollo de este proyecto y hemos visto que se pueden emplear diferentes combinaciones de módulos y dispositivos para el funcionamiento del mismo, se podría plantear el desarrollo del software del proyecto mediante líneas de producción.

Cómo me he centrado en una serie de módulos concretos, no he visto importante el uso de un SPL. Pero, en caso de necesitar diferentes configuraciones de los diversos módulos es más que recomendable la realización del SPL en el proyecto.

A continuación voy a mostrar un diagrama empleando la extensión de eclipse FeatureIDE que representa **los diferentes módulos que se pueden añadir al coche** en función de la configuración seleccionada:



- Las implicaciones en la parte inferior del diagrama representan las restricciones.

A continuación se muestra la configuración (Simple.config) que simboliza los “paquetes software” empleados en este proyecto.

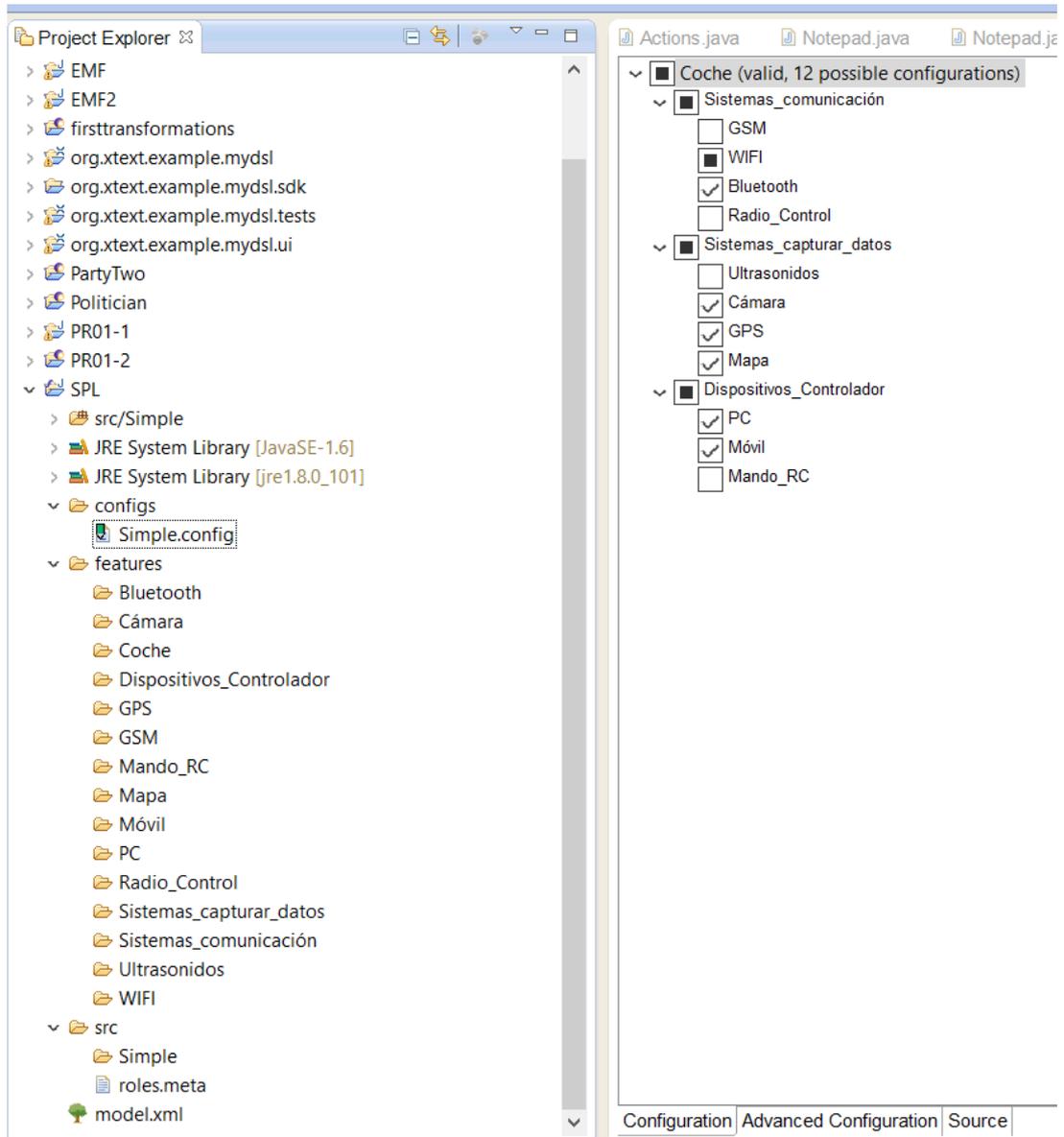


Figura 48 SPL MapCar

Además de la configuración empleada y como ya he mencionado al inicio de este capítulo, se pueden realizar diversas configuraciones dependiendo de lo que se desee añadir. En este caso he creado una configuración (Full.config) que emplea integra todos los “paquetes softwares” disponibles.

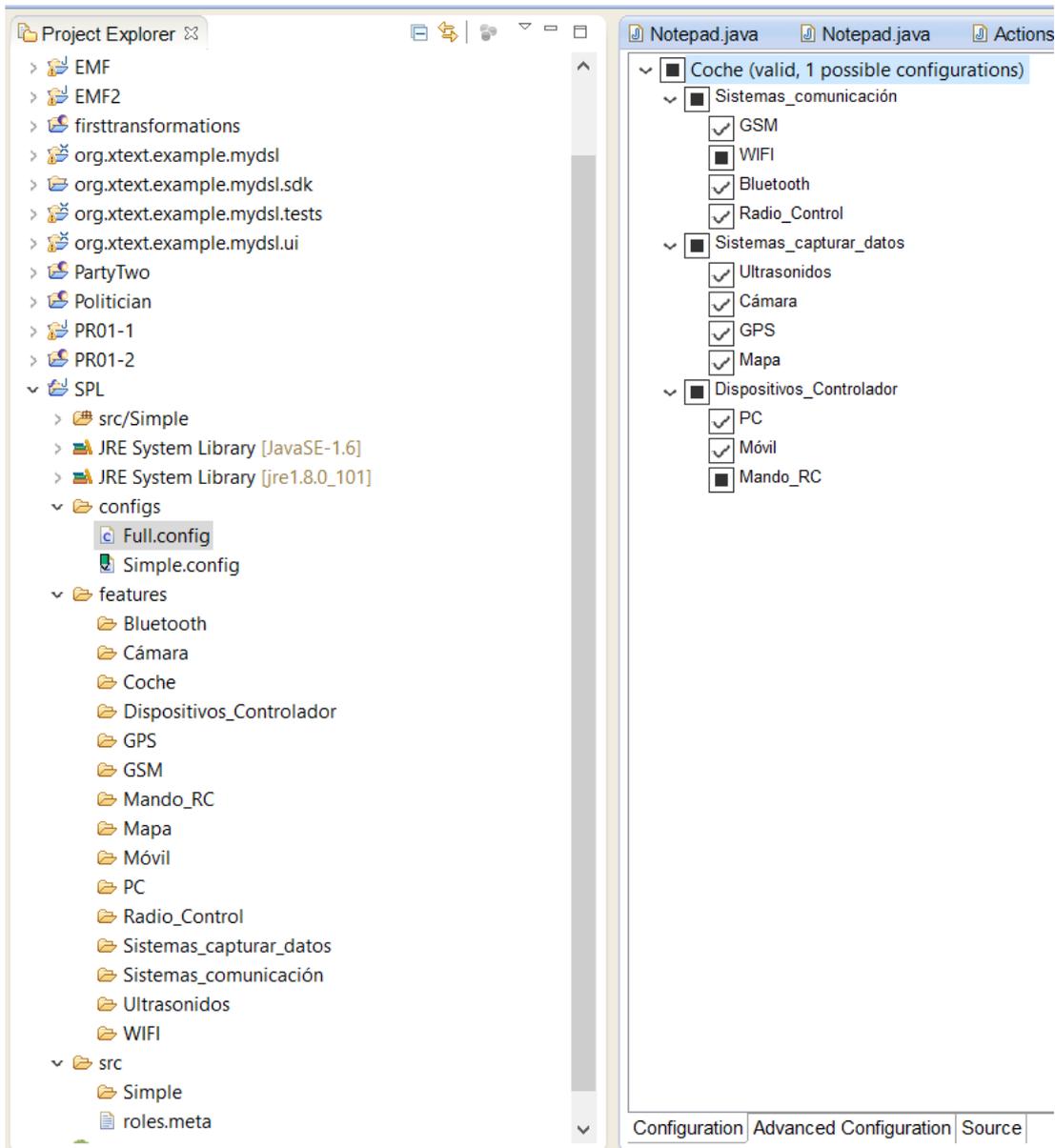


Figura 49 SPL de ejemplo

9 Conclusiones

Recapitulando, se han analizado varias tecnologías para desarrollar un prototipo de vehículo captador de datos que incorpore la una placa Arduino. Para ello, se ha realizado en primer lugar un estudio de los diferentes sistemas (comunicación, recolección de datos y

control) y posteriormente se han llevado a cabo varias pruebas con ayuda de un prototipo de pruebas para delimitar las tecnologías más idóneas para cada uno. Una vez implementados los diferentes módulos, se han realizado varias pruebas tanto software como hardware para corroborar el correcto funcionamiento de las aplicaciones y de prototipo.

Tras la realización del proyecto, y atendiendo a los resultados obtenidos, se puede observar que se ha cumplido con el totalidad de los objetivos marcados para el mismo.

Cabe recalcar, que la falta de financiación ha sido un obstáculo importante en el transcurso de proyecto, ya que en numerosas situaciones ha ocasionado la búsqueda de tecnologías alternativas a las inicialmente planteadas en las reuniones.

En caso de una buena financiación, este proyecto podría haber alcanzado un interés de mayor grado, ya que se podría haber incluido factores como autonomía, control a larga distancia y diferentes sistemas de captación de datos.

Pese a todo, y como se ha mencionado más arriba, se han cumplido todos los objetivos del proyecto y sin superar el plazo de horas establecido en el [Subcapítulo 2.2.5 Estimación de dedicación de cada una de las tareas](#) como se indica en el [capítulo 7 Seguimiento y control](#).

En cuanto a los riesgos existentes, si bien es cierto que el proyecto presenta ciertas estimaciones y una serie de datos que podrían ser clasificados de subjetivos, se ha sido muy prudente a la hora de incluirlos en el estudio.

Es necesario comprender que todo el proyecto realizado es un prototipo, ya que la implementación de un vehículo real está fuera del alcance tanto por presupuesto como falta de tiempo como varias razones más. En cambio, al ser solo un prototipo a escala reducida, se brinda la posibilidad de que en un futuro, se puedan añadir nuevas funcionalidades y emplear para proyectos de índole diferente

Por último, he de añadir, que a lo largo del proyecto he ido realizando vídeos que demuestran cada objetivo alcanzado. Dicho contenido se encuentra en el canal de youtube [Arduino Maniacos](#), y es de dominio público.

9.1 Reflexiones

Tal vez, este proyecto hubiese sido mucho más productivo y funcional si se hubiese llevado a cabo entre 2 integrantes. Uno encargado de diseñar el ámbito Software y otro el del Hardware.

Pero independientemente del coste que me ha supuesto comprender el funcionamiento de los componentes hardware que he necesitado para el proyecto, la experiencia ha sido muy gratificante y los conocimientos adquiridos muy útiles para proyectos futuros.

Por último, he de notificar, que en el [capítulo 8 LPS](#) se hace mención del significado de las líneas de producción del software y la posibilidad de su implementación en este proyecto. Aunque se ha desarrollado vagamente, no se acerca ni una mínima parte al total de sus

posibilidades, y esto es algo en lo que me hubiese gustado profundizar más en caso de haber realizado otra planificación ya que me parece uno de los temas más interesantes que he cursado a lo largo de grado.

Bibliografía

A

[A-2016] ¿Qué es Arduino?:

<http://arduino.cl/que-es-arduino/>

[A-F-2016] Acelerómetro:

http://liraelectronica.weebly.com/uploads/4/9/3/5/4935509/3249133_orig.jpg

[AIDE-2016] Instalación y descripción del Arduino IDE:

<http://elcajondeardu.blogspot.com.es/2016/03/instalacion-y-descripcion-del-arduino.html>

[APF-2016] Buscador de rutas Arduino:

<https://www.youtube.com/watch?v=YjIHb7XvTJ0>

[APPI-2016] APP Inventor:

<https://sites.google.com/site/appinventormegusta/primeros-pasos>

[AS-2016] ¿Qué es Android Studio?:

<http://academiaandroid.com/android-studio-v1-caracteristicas-comparativa-eclipse/>

B

[B-2016] ¿Qué es el Bluetooth?:

<http://www.masadelante.com/faqs/que-es-bluetooth>

[BA-F-2016] Batería 800mAh:

https://www.google.es/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=0ahUKEwjGlon61e7OAhUHPRoKHcFnDXyQjRwIBw&url=http%3A%2F%2Fwww.esmrecargable.com.ar%2Fproduct_info.php%3Fproducts_id%3D475&psig=AFQjCNEzXk-oG70Lru4pN8oB2l26AccuPw&ust=1472836540554513

[BB-F-2016] Batería 4800mAh:

<https://www.google.es/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=0ahUKEwiExb-m1u7OAhWCSxoKHdWcDuMQjRwIBw&url=http%3A%2F%2Fwww3.towerhobbies.com%2Fcgi-bin%2FWTI0001P%3FI%3DLXFWRN%26P%3D8&psig=AFQjCNERHBzCmJOoB5400qd7hD2Rvza3zw&ust=1472836668957679>

[BAUD-2016] Cambiando el baudrate del módulo bluetooth hc-06:

<https://www.squirrel-labs.net/blog/hc-06-bluetooth-module-changing-baudrate-etc/#page=1>

[BM-2016] Bing Maps:

<http://algoimagen.blogspot.com.es/2014/05/bing-maps-en-aplicaciones-net-parte-i.html>

C

[CIP-2016] Cámara IP:

<http://www.domodesk.com/a-fondo-camaras-ip>

[C-2016] Coordenadas al clicar en el mapa:

<http://stackoverflow.com/questions/14750099/location-information-where-mouse-click-on-the-map-gmap-net>

[CACN-2016] Caja blanca y caja negra:

<http://es.slideshare.net/StudentPc/software-caja-negra-y-caja-blanca>

[CPM-2016] Carpeta de proyectos con mapas:

<https://onedrive.live.com/?authkey=%21AAkpAYt7ufvGmUE&id=C6FDBF1F1FE1F985%21143&cid=C6FDBF1F1FE1F985>

[CRC-2016] Coche RC Android / Arduino. Parte I (Qué comprar y dónde hacerlo):
<http://pingpollos.com/coche-rc-android-arduino-parte-i-que-comprar-y-donde-hacerlo/>

F

[FPV-2016] Coche Arduino RC con cámara FPV:

<http://www.instructables.com/id/Arduino-RC-Car-with-FPV-Camera/?ALLSTEPS>

G

[GMAP-2016] Tutorial GMAP.NET:

<http://www.independent-software.com/gmap-net-tutorial-maps-markers-and-polygons/>

[GMAPR-2016] Repositorio GMAP.NET:

<http://greatmaps.codeplex.com/releases/view/73162>

[GMAP-F-2016] GMAP.NET:

<https://greatmaps.codeplex.com/>

[GPS-2016] ¿Qué significa GPS?:

http://www.radiofrecuencia.com/tema.php?ID=QUE_SIGNIFICA_GPS

[GPS-F-2016] GPS:

<https://uk.support.tomtom.com/euf/assets/images/public/gpsworking.gif>

[GPX-2016] GPX:

<https://es.wikipedia.org/wiki/GPX>

[GSM-2016] GSM:

<http://es.ccm.net/contents/681-estandar-gsm-sistema-global-de-comunicaciones-moviles>

I

[IPW-F-2016] IPWebcam:

https://thepretentiouspandabiswadeep.files.wordpress.com/2016/05/ipwebcam_setup.jpg?w=748

L

[L298N-2016] Tutorial: Uso de Driver L298N para motores DC y paso a paso con Arduino:
<http://electronilab.co/tutoriales/tutorial-de-uso-driver-dual-l298n-para-motores-dc-y-paso-a-paso-con-arduino/>

[L-F-2016] Licencia:

<http://cuadernodemarketing.com/wp-content/uploads/2015/01/CreativeCommons.png>

[LPS-2016] LPS:

<https://ingsoftwarei2014.wordpress.com/category/lineas-de-produccion-de-software-lps/>

M

[M2M-2016] Sistema GPS con Arduino + placa M2M:

<http://www.instructables.com/id/Sistema-GPS-con-Arduino-M2M-Shield/>

[MC-2016] Usar marcador OnMarkerClick cuando se usa MouseClick:

<http://stackoverflow.com/questions/31602748/using-gmap-net-for-c-how-do-i-use-a-marker-onmarkerclick-without-calling-map-m>

[MM-20016] Motores con memoria:

<https://forum.arduino.cc/index.php?topic=397180.0>

P

[PA-F-2016] Placa Arduino Uno r3:

<https://img.pccomponentes.com/articles/7/77672/arduino-uno-r3-atmega328.jpg>

R

[RCR-2016] De coche RC a Robot:

<http://www.instructables.com/id/RC-Car-to-Robot/?ALLSTEPS>

S

[SPL-2016] ¿Qué son los SPL?:

<https://ingsoftwarei2014.wordpress.com/category/lineas-de-produccion-de-software-lps/>

[SU-2016] Sensor de ultrasonidos:

<http://wiki.robotica.webs.upv.es/wiki-de-robotica/sensores/sensores-proximidad/sensor-de-ultrasonidos/>

U

[U-2016] Todo sobre sensores de ultrasonidos y Arduino:

<https://wngeek.wordpress.com/2013/06/11/todo-sobre-sensores-de-ultrasonidos-y-arduino/>

V

[VDB-2016] Ventajas y desventajas del Bluetooth:

<https://junihh.wordpress.com/2007/06/02/ventajas-y-desventajas-de-bluetooth/>

[VDGSM-2016] Ventajas y desventajas del GSM:

<http://www.taringa.net/posts/info/17373646/Tecnologia-GSM.html>

[VDRC-2016] Pros y contras de las emisoras a 2,4GHZ:

<http://rc.lapipadelindio.com/general/pros-contras-frecuencia-emisoras-24ghz>

[VS-2016] ¿Qué es Visual Studio2013?:

<http://www.genbetadev.com/herramientas/visual-studio-2013>

W

[W-2016] ¿Qué es WI-FI?:

<https://es.wikipedia.org/wiki/Wifi>

[WESP-2016] Repositorio WiFiEsp para el WI-FI ESP8266:

<https://github.com/bportaluri/WiFiEsp>

Diccionario

B

[B-D-2016] Backup:

En tecnologías de la información e informática es una copia de los datos originales que se realiza con el fin de disponer de un medio para recuperarlos en caso de su pérdida.

https://es.wikipedia.org/wiki/Copia_de_seguridad

D

[D-D-2016] Delay:

Pausa de una ejecución durante un período de tiempos medido generalmente en milisegundos.

E

[XML-D-2016] Esquema XML:

Es un lenguaje utilizado para describir la estructura y las restricciones de los contenidos de los documentos, de una forma muy precisa, más allá de las normas sintácticas impuestas por el propio lenguaje XML. Se consigue así una percepción del tipo de documento con un nivel alto de abstracción.

H

[H-D-2016] HUB:

Es un dispositivo que permite centralizar el cableado de una red de computadoras, para luego poder ampliarla.

L

[L-D-2016] LAN:

LAN son las siglas de Local Área Network, Red de área local. Una LAN es una red que conecta los ordenadores en un área relativamente pequeña y predeterminada (como una habitación, un edificio, o un conjunto de edificios). Las redes LAN se pueden conectar entre ellas a través de líneas telefónicas y ondas de radio.

<http://www.masadelante.com/faqs/lan>

P

[AP-D-2016] Punto AP:

Un punto de acceso inalámbrico (en inglés: *Wireless Access Point*, conocido por las siglas WAP o AP), en una red de computadoras, es un dispositivo de red que interconecta equipos de comunicación inalámbricos, para formar una red inalámbrica que interconecta dispositivos móviles o tarjetas de red inalámbricas.

https://es.wikipedia.org/wiki/Punto_de_acceso_inal%C3%A1mbrico

S

[S-D-2016] Scrum:

Scrum es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado posible de un proyecto.

<https://proyectosagiles.org/que-es-scrum/>

[SL-D-2016] Slidebar:

Es una barra de desplazamiento.

[SP-D-2016] Split:

Función encargada de dividir un dato en varios segmentos.

[ST-D-2016] Streaming:

Es la distribución digital de multimedia a través de una red de computadoras, de manera que el usuario consume el producto (generalmente archivo de vídeo o audio) en paralelo mientras que se lo descarga.

<https://es.wikipedia.org/wiki/Streaming>

[SW-D-2016] SWITCH:

Un switch es un dispositivo diseñado para resolver problemas de rendimiento en la red, debido a anchos de banda pequeños y embotellamientos.

<http://www.aprendaredes.com/dev/articulos/que-es-el-switch.htm>

T

[T-D-2016] Threads:

Un Thread es un mecanismo que permite a una aplicación realizar varias tareas a la vez de manera concurrente.

Ejecutar varios procesos a la vez, pero enfocado a ejecutar subprocesos de un mismo proceso, compartiendo el espacio de memoria entre sí.

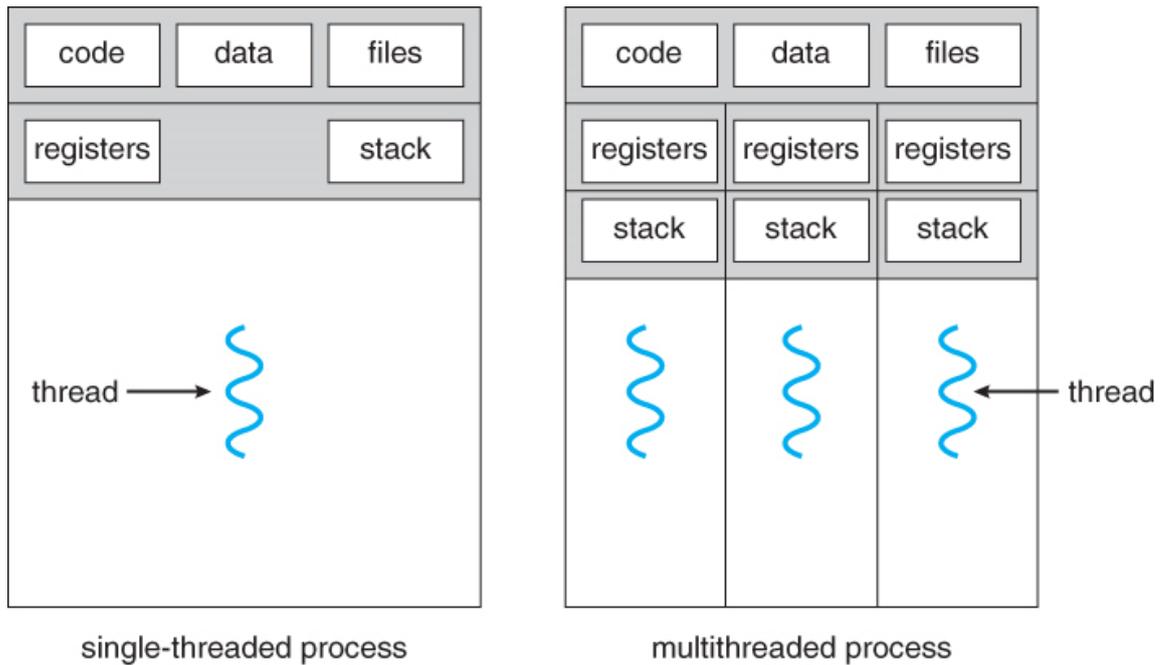


Figura 50 Threads

<https://msdn.microsoft.com/es-es/library/dn602621.aspx?f=255&MSPPErr=-2147217396>

W

[W-D-2016] WAN:

WAN es la sigla de Wide Área Network (“Red de Área Amplia”). El concepto se utiliza para nombrar a la red de computadoras que se extiende en una gran franja de territorio, ya sea a través de una ciudad, un país o, incluso, a nivel mundial.

<http://definicion.de/wan/>

Anexo A: Manual de Usuario

1. Rouer-Cámara

- Aplicación IPWebcam:

Podemos encontrarla en la Play Store de Google.

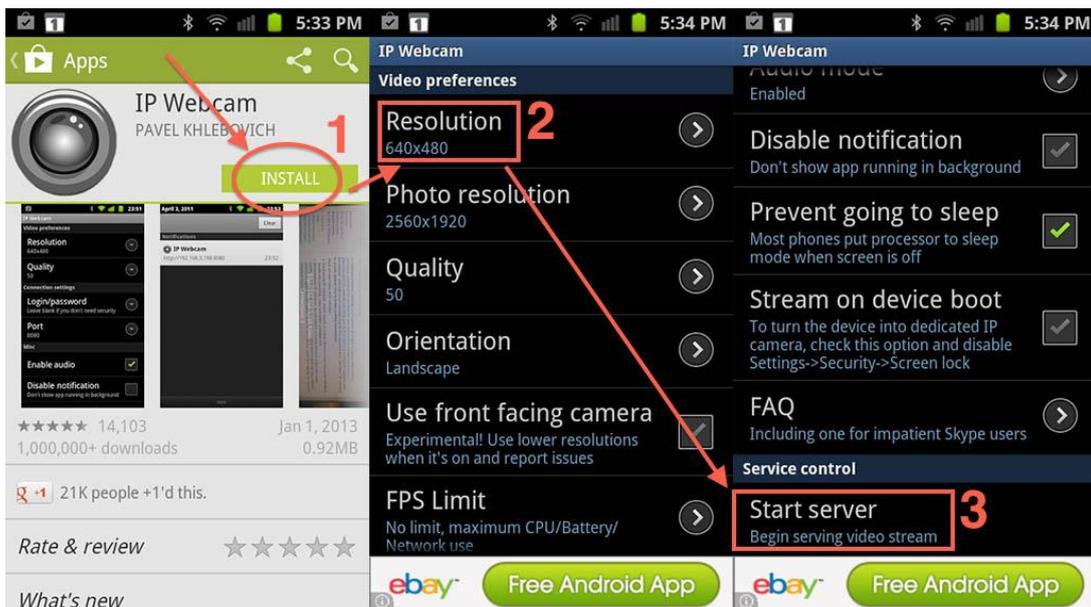


Figura 51 Funcionamiento IPWebcam [IPW-F-2016]

Una vez instalada basta con ejecutarla, seleccionar la resolución deseada y pulsar en iniciar servidor. Una vez iniciado el servidor se mostrará la IP que tendremos que introducir en la aplicación de ordenador para poder ver el Streaming.

- Aplicación Zona WI-Fi:

Integrada en móviles Android y se accede a ella a través de Ajustes -> Más -> Anclaje de red y zona Wi-Fi.

Basta con encender la aplicación e introducir el nombre de red y contraseña con la que accederán los dispositivos.

También indica la IP de cada dispositivo si presionamos encima de sus nombres una vez que se han conectados.

2. Aplicación móvil:

La aplicación Android diseñada para controlar el movimiento del coche consta de dos ventanas.

La primera, es con la realizamos todas las funciones de control del vehículo. En la segunda ventana, se añade la dirección IP y el puerto al que nos vamos a conectar (ip y puerto del coche).

Ventana de control:

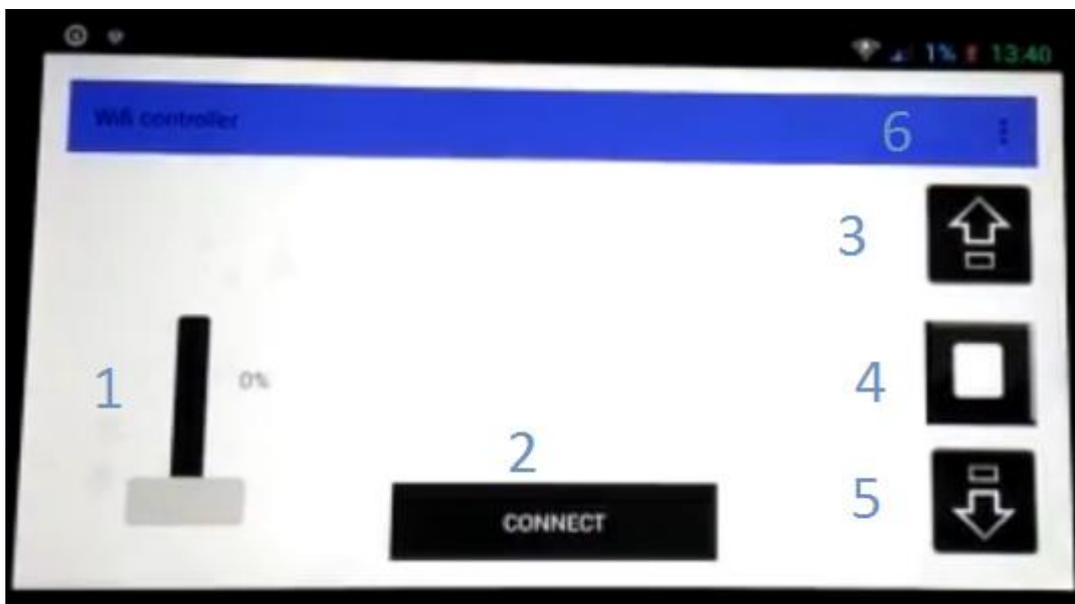


Figura 52 Aplicación móvil (Ventana de control)

1. Slidebar **[SL-D-2016]** que permite controlar la velocidad.
2. Botón para conectarnos al coche o desconectarnos de él.
3. Botón para ir hacia adelante.
4. Botón para parar el coche. Permite parar súbitamente.
5. Botón para ir hacia atrás.
6. Cambiar a la ventana de datos de conexión

Ventana de datos de conexión:

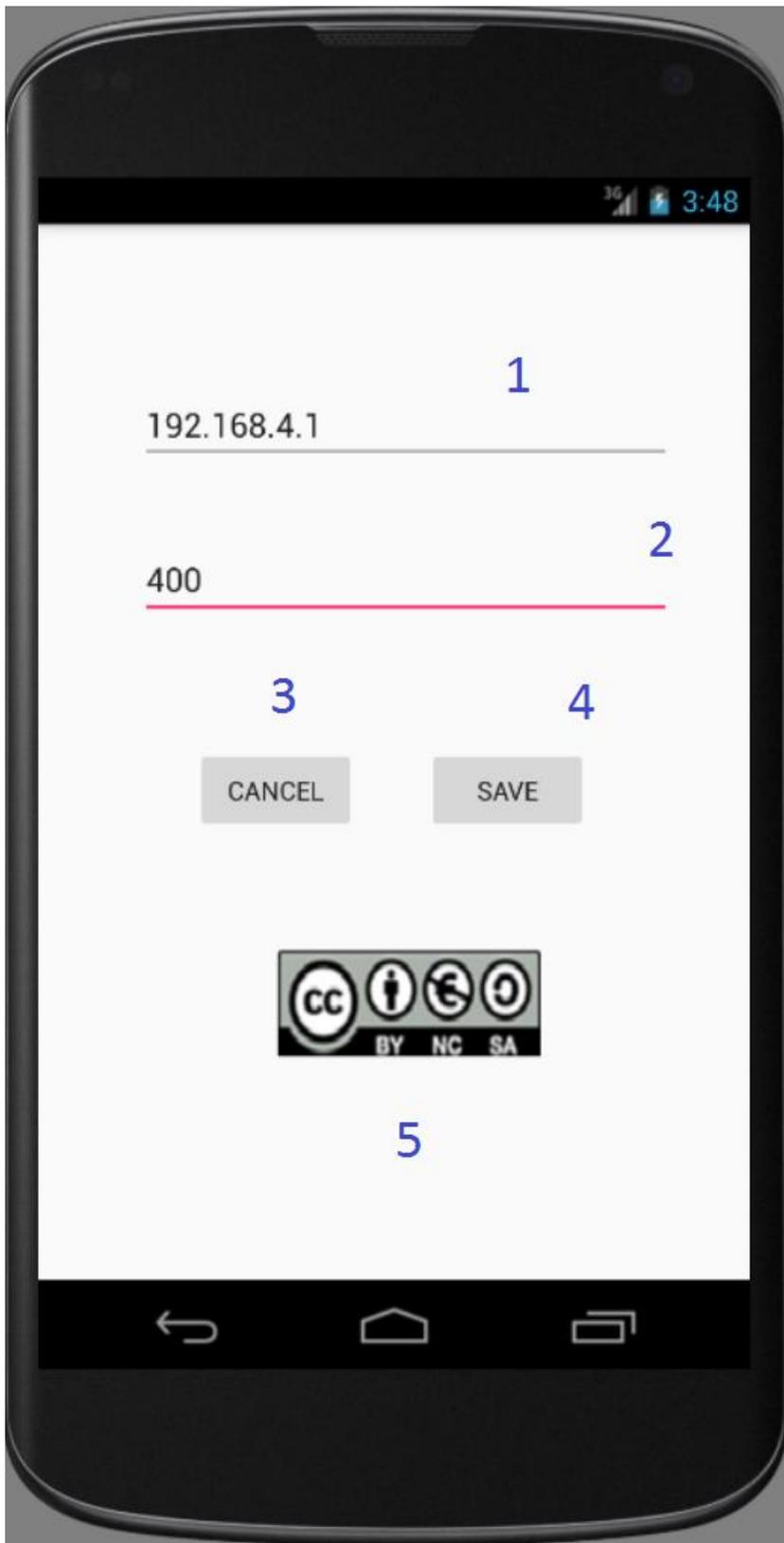


Figura 53 Aplicación móvil (Ventana de datos de conexión)

1. IP

2. Puerto
3. Botón para volver a la ventana de control sin hacer ningún cambio.
4. Botón para guardar los cambios y volver a la ventana de control.
5. Licencia BY_NC_SA

3. Aplicación ordenador:

Para desarrollar la aplicación de ordenador he partido de un software de terceros que he modificado para adaptarlo a los requisitos de mi proyecto. [Número bibliografía]

En la versión de ordenador, la interfaz está estructurada en 3 ventanas.

Alberga todas las funcionalidades para el control de vehículo, de la cámara y del mapa.

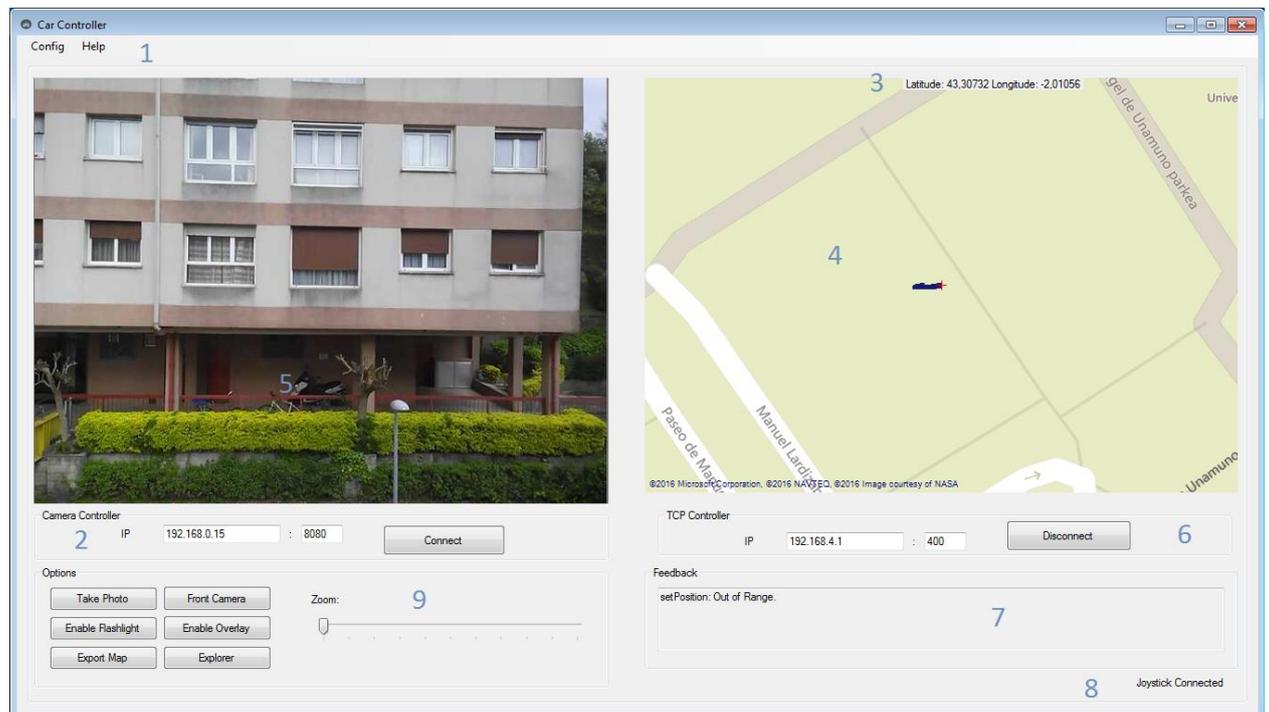


Figura 54 Aplicación Ordenador (Ventana de control)

1. Menu.
2. Camera Controller: permite conectarse a la cámara IP introduciendo su IP y puerto.
3. Etiqueta que muestra la latitud y la longitud de la ubicación del coche.
4. Controlador que muestra la ruta que va realizando nuestro coche y, en caso de pinchar en un punto en el que se ha sacado una foto, muestra dicha foto.
5. Controlador que muestra el Streaming de la cámara.
6. TCP Controller: permite conectarse al coche introduciendo su IP y puerto.
7. Feedback: Muestra los posibles errores y donde se han provocado.
8. Etiqueta que muestra cuando el joystick está conectado.

- Opciones variadas para el control de la cámara (sacar fotos, hacer zoom, etc.), exportar el mapa y explorar la carpeta donde se almacenan los datos recolectados.

Ventana de configuración:

Muestra las funcionalidades incluidas en el mando.

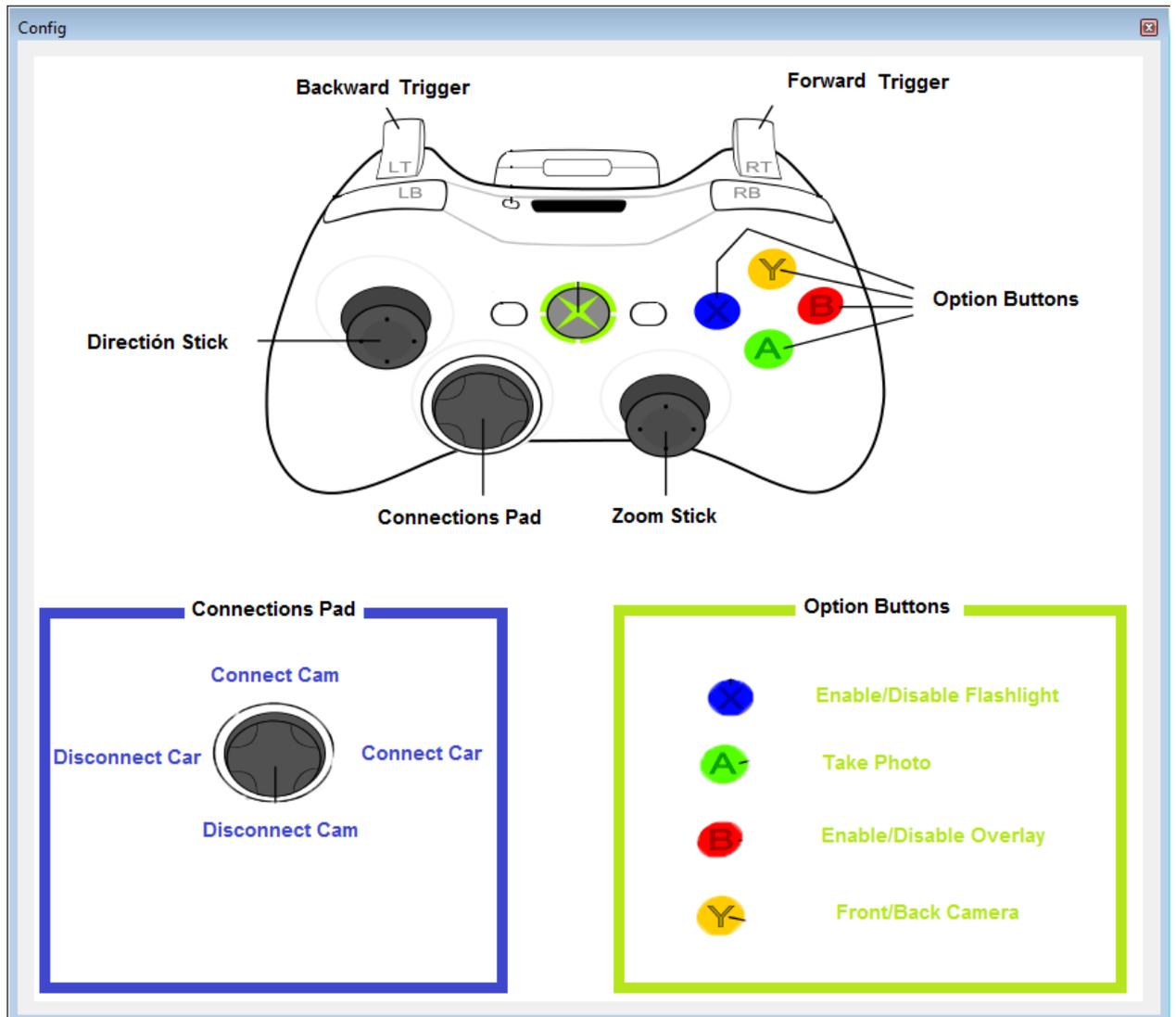


Figura 55 Aplicación Ordenador (Ventana de configuración)

Ventana de ayuda:

Muestra el creador de la aplicación, es decir mi nombre, mi email y la licencia de la aplicación.



Figura 56 Aplicación Ordenador (Ventana de ayuda)

Anexo B: Actas de reuniones

Acta de Reunión 17/02/2016

Reunidos en la Facultad de Informática de San Sebastián, el día 17 de Febrero del 2016 a las 12:45 horas, con el profesor Alfredo Goñi y el alumno Gonzalo del Palacio.

Puntos del día:

1. Toma de contacto.
2. Presentación del coche WI-FI.
3. Búsqueda de objetivos para el coche como proyecto de grado.

Resumen de decisiones tomadas:

- Se ha comentado la posibilidad de recrear un pequeño entorno en el cual el coche se maneja de manera autónoma o remota y va recopilando datos (video, audio,..).
- Funcionalidades que se podrían implementar:
 - Enviar órdenes.
 - Ofrecer información Remota
 - Captar datos
 - Extracción de información
 - Control remoto (móvil)
 - Agregar el funcionamiento actual de coche

Temas a tratar en la siguiente reunión

- Encontrar finalidades que no sean de la rama de computación o de hardware
- Definir el motivo por/para el que se precisa un coche WI-FI.
- Ver que ofrecen robots captadores de datos
- Estudiar cómo obtener localización:
 - GPS
 - WI-FI
 - MOVIL

- Calcular según se mueve el coche

Action items:

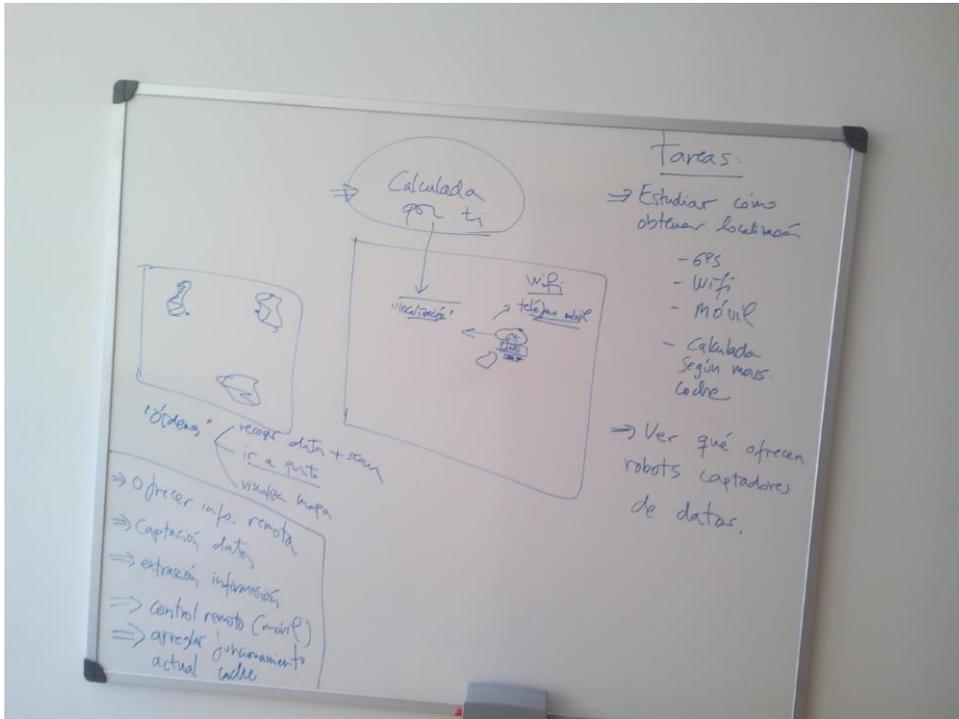


Figura 57 Captura de requisitos Acta_17/02/2016

No habiendo más asuntos que tratar, se levanta la sesión siendo las 13:15 horas del día citado, de todo lo cual doy fe como registrador.

- Fdo. Gonzalo del Palacio

Acta de Reunión 08/03/2016

Reunidos en la Facultad de Informática de San Sebastián, el día 08 de Marzo del 2016 a las 12:30 horas, con el profesor Alfredo Goñi y el alumno Gonzalo del Palacio.

Puntos del día:

1. Notificar sobre la información obtenida referente a lo acordado en la anterior sesión:
 - Encontrar finalidades que no sean de la rama de computación o de hardware
 - Definir el motivo por/para el que se precisa un coche WI-FI.
 - Ver que ofrecen robots captadores de datos
 - Estudiar cómo obtener localización:
 - GPS

- WI-FI
- MOVIL
- Calcular según se mueve el coche

2. Estructurar las funcionalidades a implementar en el proyecto.

Resumen de decisiones tomadas:

- La idea inicial de generar un coche autónomo a sido sustituida debido a la falta de experiencia y de conocimiento en dicho tema. EL control de coche será de manera remota mediante tecnología WI-FI. No se descarta la posibilidad de poder implementar la funcionalidad de coche autónomo en futuras modificaciones.
- Las funcionalidades que el coche implementa son:
 Control WI-FI mediante apk
 Control WI-FI mediante un joystick y Visualización mediante un streaming de vídeo para poder conocer la ubicación real del vehículo.
 Opción de sacar foto, tomar vídeo o grabar audio
 Conocer su ubicación exacta mediante un mapa (empleando GPS)
 Cada vez que se realice una captura de datos, en el mapa se indicará un puntero, en el que al pulsar se reproducirá el dato tomado
 Opción de realizar un mapeado y exportarlo como imagen, empleando varios ultrasonidos.

Temas a tratar en la siguiente reunión

- Definir un uso para los datos recolectados
- Volver a retocar las funcionalidades del coche

No habiendo más asuntos que tratar, se levanta la sesión siendo las 13:15 horas del día citado, de todo lo cual doy fe como registrador.

- Fdo. Gonzalo del Palacio

Acta de Reunión 17/03/2016

Reunidos en la Facultad de Informática de San Sebastián, el día 17 de Marzo del 2016 a las 12:30 horas, con el profesor Alfredo Goñi y el alumno Gonzalo del Palacio.

Puntos del día:

1. Notificar sobre la información obtenida referente a lo acordado en la anterior sesión (Definir un uso para los datos recolectados):

- Para sacar provecho de los datos obtenidos, generamos un mapa del recorrido realizado, con opción de poder interactuar con él y que nos muestre los datos recogidos en los puntos donde se han obtenido.

Resumen de decisiones tomadas:

- Se han planteado distintas cuestiones a tratar como:
 - El coche se conectara a una red WI-FI: Lo que supone, que cada vez que se quiera cambiar de red se deberán acceder a las tripas del arduino. Esto permitirá poder controlar el coche desde cualquier parte del mundo, pero perderíamos portabilidad.
 - El coche generará una red WI-FI: Cualquier emisor WI-FI podrá conectarse al coche. Obtenemos portabilidad.
 - Emplear GSM en vez de WI-FI: ¿Cuántos datos consumirá por manejar el vehículo?: ¿Se puede reducir la frecuencia con la que se envían los comandos, pero el coche reaccionará bien O perderemos movilidad?

Temas a tratar en la siguiente reunión

- Definir un uso para los datos recolectados
- Volver a retocar las funcionalidades del coche
- Enseñar una primera versión del coche:
 - Controlarlo desde un joystick y ver la ruta que realiza.
 - Controlar el coche desde el móvil (hecho): Falta adaptar la apk a cualquier pantalla de móvil.

No habiendo más asuntos que tratar, se levanta la sesión siendo las 13:15 horas del día citado, de todo lo cual doy fe como registrador.

- Fdo. Gonzalo del Palacio

Acta de Reunión 27/04/2016

Reunidos en la Facultad de Informática de San Sebastián, el día 27 de Abril del 2016 a las 12:30 horas, con el profesor Alfredo Goñi y el alumno Gonzalo del Palacio.

Puntos del día:

- Definir un uso para los datos recolectados: generar un fichero con formato x que mediante un visualizador muestre el mapa y las fotos tomadas.
- Volver a retocar las funcionalidades del coche:

- Enseñar una primera versión del coche:
 - Controlarlo desde un joystick y ver la ruta que realiza.
 - Controlar el coche desde el móvil.

Resumen de decisiones tomadas:

- Después de un rápido repaso a las funcionalidades implementadas hasta la fecha se ha decidido que, una vez implementado un fichero que almacene la ruta realizada y las fotos captadas se dará por culminado el desarrollo de este proyecto.

Temas a tratar en la siguiente reunión

- Puntos a destacar en la Memoria

No habiendo más asuntos que tratar, se levanta la sesión siendo las 13:15 horas del día citado, de todo lo cual doy fe como registrador.

- Fdo. Gonzalo del Palacio

Acta de Reunión 11/05/2016

Reunidos en la Facultad de Informática de San Sebastián, el día 11 de Mayo del 2016 a las 12:45 horas, con el profesor Alfredo Goñi y el alumno Gonzalo del Palacio.

Puntos del día:

- Puntos a destacar en la Memoria

Resumen de decisiones tomadas:

- En la reunión de hoy 11/05/2016 se han establecido los puntos iniciales de la memoria del software y vehículo captador de datos.

Temas a tratar en la siguiente reunión

- Estado actual de la memoria
- Como preparar la presentación de cara a la defensa del PFG

No habiendo más asuntos que tratar, se levanta la sesión siendo las 13:15 horas del día citado, de todo lo cual doy fe como registrador.

- Fdo. Gonzalo del Palacio

