



Cylock:

**Proyecto
Final Diploma**

Autor:

Carlos Gregorio
Martín Pérez

Director:

Jesús Tomás
Gironés

**TESINA PARA LA
OBTENCIÓN DEL TÍTULO
DE:**

**Diploma de Especialización
en Desarrollo de
Aplicaciones Android**

Septiembre de
2019

Contenido

Cylock:.....	1
Autor:.....	1
Director:.....	1
Diploma de Especialización en Desarrollo de Aplicaciones Android.....	1
Introducción.....	3
Descripción del problema.....	3
Objetivos.....	3
Motivación.....	4
Tecnologías utilizadas.....	5
Arquitectura de la aplicación.....	6
Esquema del diseño.....	6
Modelo de datos.....	10
Vistas.....	11
Conclusiones.....	14
Anexos.....	15
Código fuente GitHub.....	15
SIM900 Datasheet.....	15

Introducción

Descripción del problema

Los robos de bicicletas en las grandes ciudades suponen un importante problema debido no solo a las consecuencias económicas de éste percance, sino también a la preocupación que levanta en sus usuarios.

Los métodos o sistemas antirrobo a menudo no suponen más que una medida disuasoria contra los ladrones. Los más típicos son los candados y cadenas así como otros elementos menos directos como BiciRegistro, la “Red de Ciudades por la Bicicleta” que cuenta con la participación de la policía local y de los propios Ayuntamientos. Permite registrar el vehículo tras lo que ofrecen unas pegatinas para colocarlas de tal manera que aunque se quite, el número de registro quede plasmado con pintura ultravioleta en el cuadro de la bici.

Sea cual fuere el método no existe en el mercado nada efectivo que permita ubicar la bicicleta en tiempo real para realizar una denuncia en caso de robo. Permitiendo a las autoridades competentes obtener una trazabilidad de la ubicación y así facilitar la detención del ladrón y la devolución del vehículo.

Objetivos

Desarrollar una aplicación basada en el concepto Internet of Things (IoT) y un dispositivo embebido que irá colocado en la bicicleta de manera discreta para evitar que sea detectado.

Por un lado, la app deberá permitir activar la alarma para recibir las coordenadas de la bicicleta en caso de robo. Como valor añadido se aportarán las funcionalidades clásicas de una aplicación deportiva estudiando la posibilidad de permitir que la app obtenga y calcule las rutas realizadas así como otros datos de interés para el usuario (distancia mensual recorrida, gasolina ahorrada, histórico de rutas etc.).

Por otro lado, si se dispone de tiempo suficiente se desarrollará el dispositivo emisor de coordenadas que será capaz de conectarse a un servidor MQTT (Firebase o similar) para que éste se encargue de notificar los distintos eventos que lance este dispositivo a la aplicación.

Por todo ello, se considera priorizar el desarrollo de la app con un banco de datos de prueba, que serán generados de forma automática para poder comprobar su funcionamiento.

Motivación

La reciente mudanza a una ciudad grande, como es Madrid, y la adquisición de una bicicleta, para realizar desplazamientos por la ciudad sin favorecer a la creciente contaminación, han hecho aparente la necesidad de una App con las características propuestas en esta tesina, que de solución a la problemática de robos.

Además, al tratarse de un producto novedoso, no se descarta estudiar las posibles vías de negocio que pueda ofrecer comercializar el dispositivo embebido. Esto daría un punto de seguridad al usuarios y así fomentaría un medio de transporte más sostenible.

Tecnologías utilizadas

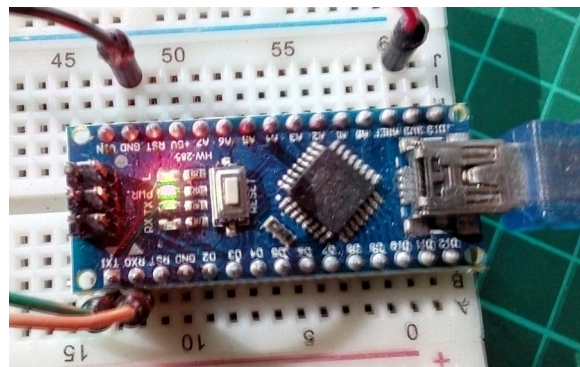
Se ha intentado, ante todo, desarrollar el software de manera limpia, buscando la sencillez y claridad. Procurando que éste sea mantenible y escalable en el tiempo. Para ello se han empleado prácticas propuestas en asignaturas como Arquitectura software y diseño.

Por otro lado, dado que la aplicación se comunicará con un dispositivo embebido, se enfocará el problema desde el paradigma Internet de las cosas *IoT*.

Además se intentará reducir tanto la complejidad de uso como la curva de aprendizaje de los usuarios, buscando una interfaz limpia, atractiva y sencilla a través de los elementos gráficos. Incluyendo un tutorial inicial si fuera necesario.

Dado que la aplicación debería ubicar la bicicleta en un mapa se ha optado por emplear el Open Street Map (OSM) en lugar del propio facilitado por Google (Google Maps). Esto permite reducir costos por uso de datos. Además, dispone de un mapa específico para bicicletas que indica carriles bici y otros elementos interesantes para el usuario como vías rurales o rutas alternativas. La librería Android que lo implementa es *osmdroid*.

Como se comentará más adelante, el dispositivo embebido, usará la tecnología GSM/GPRS para la obtención de coordenadas y también poder comunicarse con el servidor. Además del microcontrolador Arduino Nano para gestionar su funcionamiento.



Arquitectura de la aplicación

Esquema del diseño

En primera instancia se barajaron todas las posibilidades para conectar el dispositivo incrustado en la bicicleta con la aplicación Android. Dispositivos Bluetooth, GSM, GPRS etc.

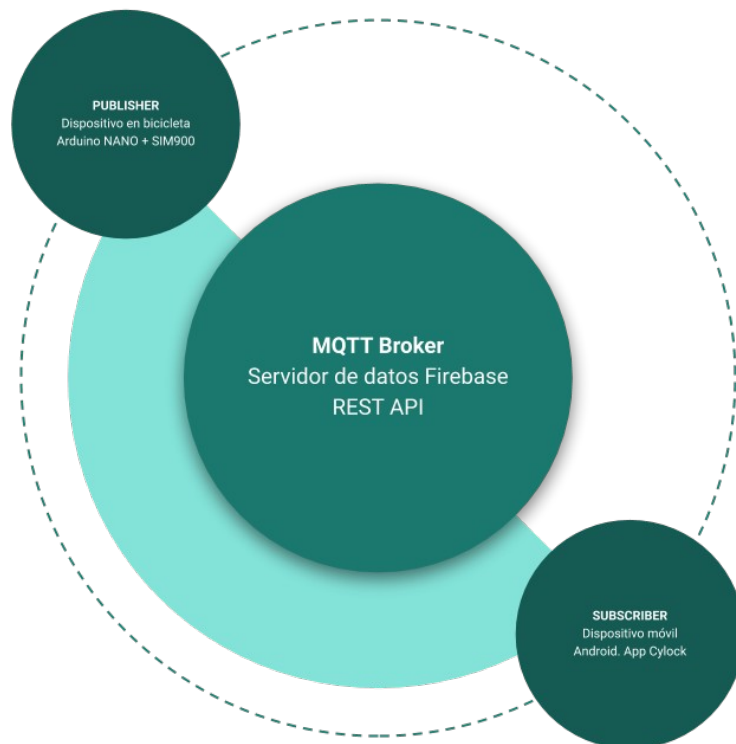


Figura 1: Esquema de tecnologías

Se tomó la decisión de usar la tarjeta SIM900 pese a necesitar una SIM (con tarifa de datos) porque ofrecía una tecnología GSM/GPRS. Esto facilita la obtención de las coordenadas y la conectividad con la red para crear la arquitectura MQTT. Ver anexos para obtener más detalles del microcontrolador.

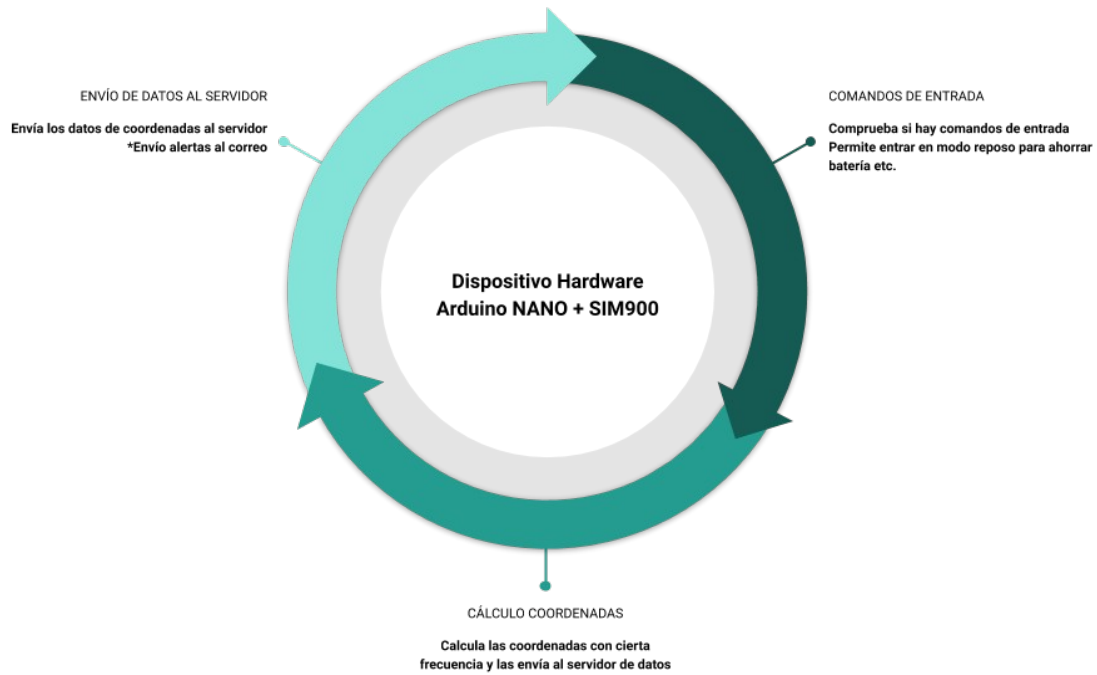


Figura 2: Esquema de funcionamiento del dispositivo en la bicicleta

Como contraposición, la potencia de cálculo así como la problemática de tener que programar SIM900 con comandos AT (*AT+commands*) mediante un dispositivo *FTDI*, impulsaron a añadir a éste conjunto una capa superior para controlarlo: añadir el Arduino NANO. Dicha capa se encargaría de gestionar los estados del dispositivo (su estado de reposo, su estado de emisión, el estado de las baterías etc). Ambos dispositivos irían conectados juntos con una batería como se muestra en la siguiente imagen.

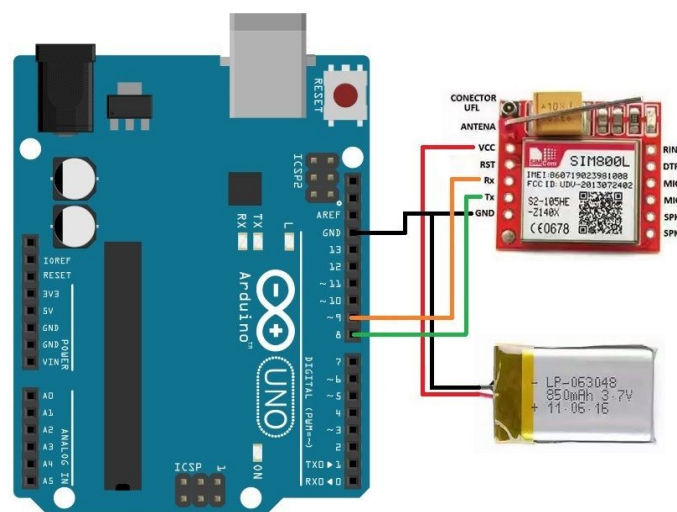


Figura3: Diagrama de conexión Arduino + SIM900 + batería

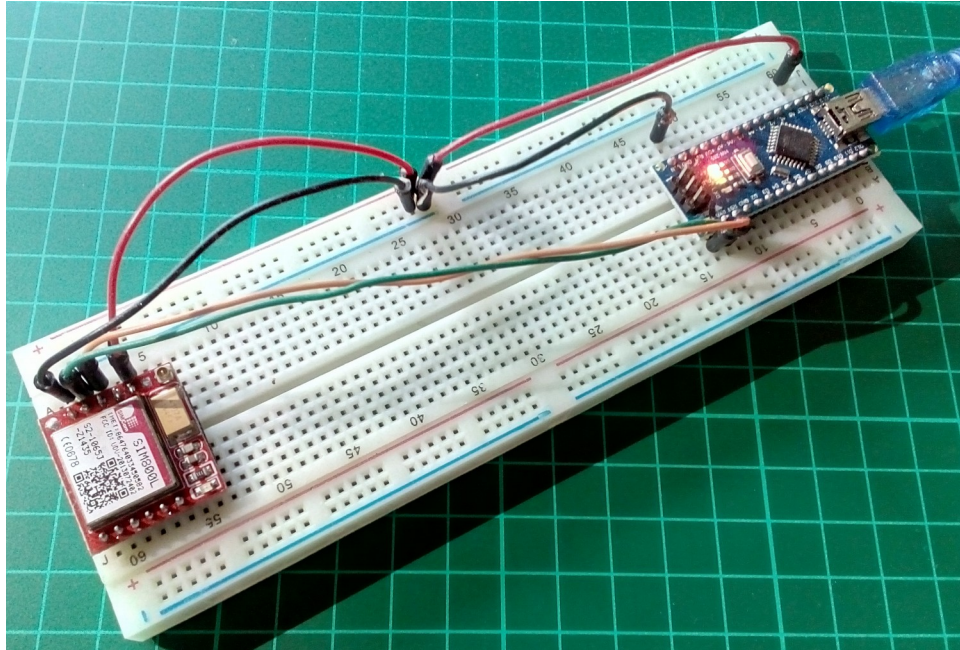


Figura 4: Laboratorio de pruebas Arduino Nano + SIM900

Como ya se ha mencionado, se han centrado los esfuerzos en el desarrollo de una aplicación Android que permitiera la captura de los datos emitidos por el dispositivo embebido y que se conectara con el servidor MQTT cuando se activara la alarma desde la pantalla principal. Su funcionamiento a grandes rasgos queda plasmado en el siguiente diagrama de flujo:

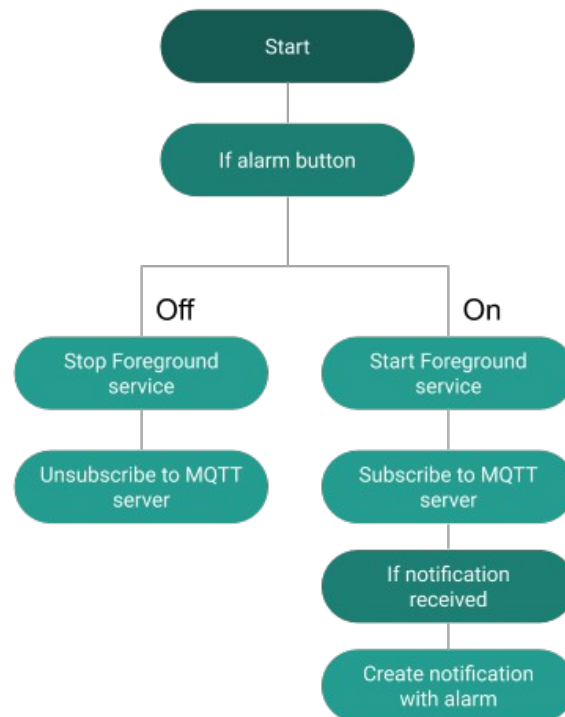


Figura 4: Diagrama de flujo de la aplicación

Debido a que se empezará el desarrollo por la aplicación móvil, el servicio creado solamente pondrá en marcha un temporizador que simulará un evento de robo con un plazo máximo de un minuto. En el futuro ésta lógica debería cambiar para que el programa se conecte en modo suscriptor al servidor MQTT que reciba los datos de la bicicleta.

Modelo de datos

Para la implementación del modelo de datos “ruta” se ha optado por heredar el existente en la librería *osmdroid* para representar poli-líneas en el mapa *Polyline*, que almacena un conjunto de geopuntos (*Geopoint*), la distancia recorrida además de ciertas características relacionadas con su representación en el mapa como color de la línea, ancho etc. La nueva clase creada *CpathRoute* contiene además de esto, métodos para obtener el geopunto central (para poder centrar el mapa en una ruta seleccionada), obtener su distancia total, y fechas de inicio, fin e incluso la duración del recorrido.

Estas características empleadas para mostrar el histórico de rutas así como otras estadísticas como la distancia recorrida en el mes en curso o el combustible ahorrado en el mismo período de tiempo.

Por otro lado la ubicación actual de la bicicleta es almacenada en un *DataManager* como un tipo de datos también contendio en la librería de OSM *Geopoint* que consta de los atributos necesarios para localizar un punto en concreto en el mapa (latitud, longitud etc.)

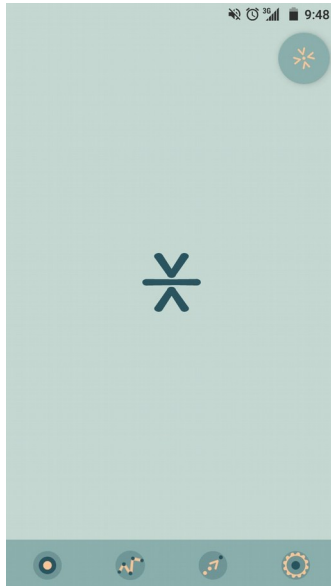
El mencionado *DataManager* no es más que un contenedor que almacena una lista de *CpathRoute* y un *Geopoint*, con el histórico de rutas realizadas y el punto geométrico en el que se encuentra la bicicleta respectivamente, los métodos necesarios para manejar los datos almacenados.

Los servicios web usados en el desarrollo de este proyecto han sido los propios de OSM. Por otro lado, emplear Firebase para montar un servidor MQTT rápido que permita conectar la aplicación con el dispositivo embebido.

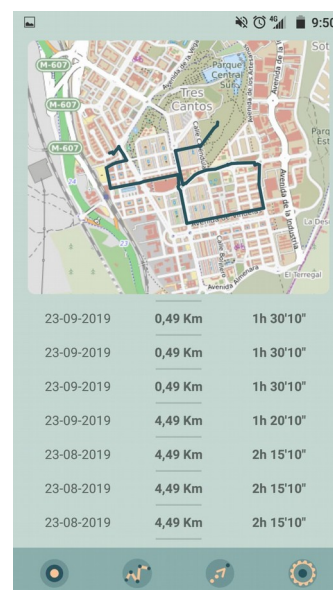
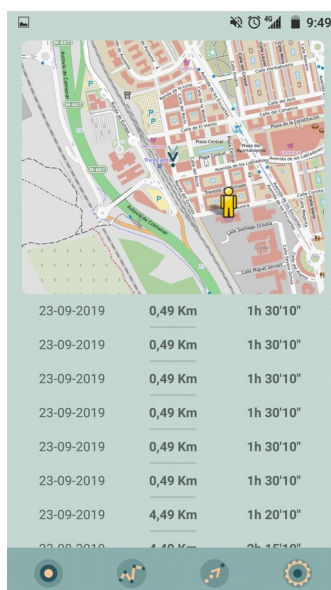
Como se venía comentando, por falta de tiempo, se ha optado por crear un *CexampleData* que contiene una serie de rutas y ubicaciones simuladas.

Vistas

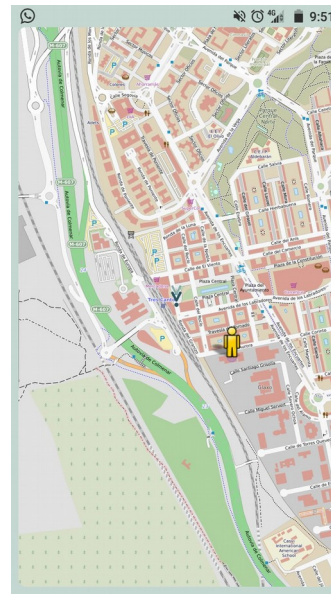
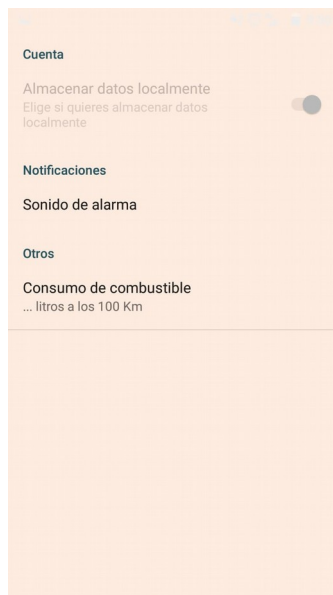
1. Vista principal (*HomeFragment*) 2. Estadísticas (*StatsFragment*)



3. Historial de rutas (*HistoryFragment*)



4. Preferencias (*PreferenceActivity*) 5. Mapa (*MapActivity*)



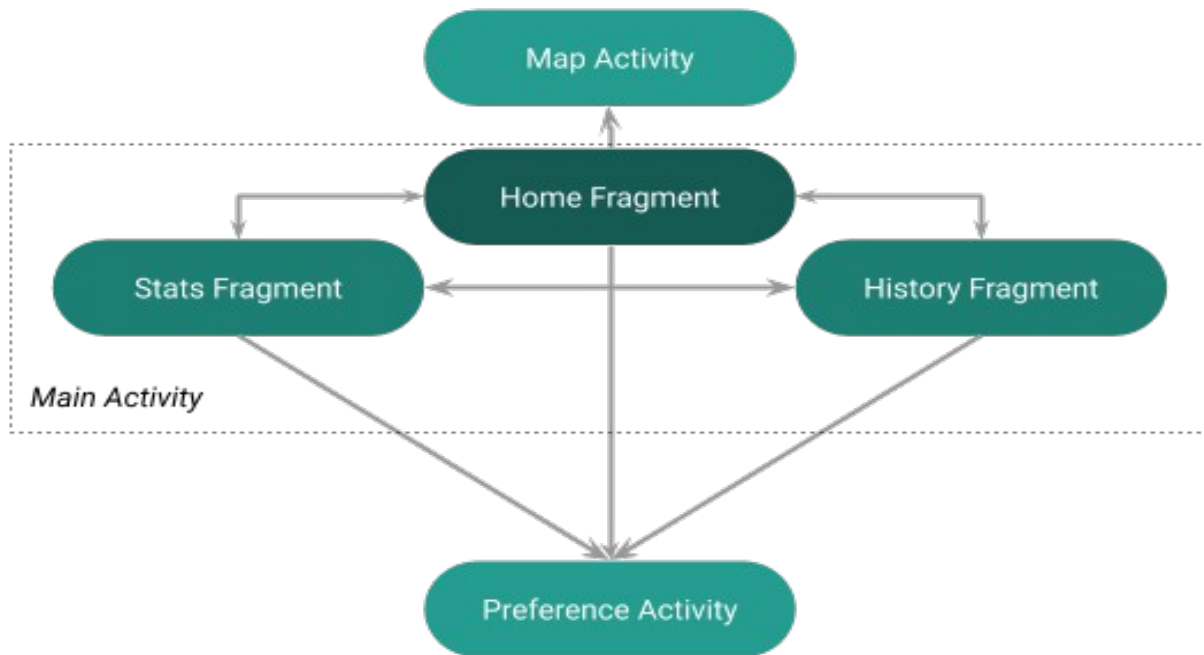
Finalmente las notificaciones que crea la aplicación. En primera instancia la notificación de posible robo del vehículo, en segunda la notificación asociada al *foregroundservice* que establece conexión con el servidor MQTT.



Todos las las vistas son accesibles desde la vista principal (*HomeFragment*) desde el menú inferior y desde cualquier otra vista de tipo *fragment* ya que pertenecen a la misma actividad. Las vistas Preferencias y Mapa pertenecen a actividades independientes que no contienen menú inferior, y por lo tanto, solo permitirán pulsar el botón atrás de Android para volver a la vista anterior.

Para el caso de las notificaciones, la asociada al servicio, abrirá directamente la vista principal (*HomeFragment*) mientras que la notificación de posible robo, abrirá directamente la vista Mapa con la ubicación de la bicicleta en tiempo real.

A continuación se plasma dicho funcionamiento en el diagrama de navegación entre vistas:



Conclusiones

El objetivo principal del desarrollo de este proyecto, crear una aplicación antirrobo para vehículos de movilidad tales como bicicletas, se ha completado con éxito. No así el desarrollo del dispositivo embebido complementario que se incrustaría en el vehículo.

Por tanto queda abierta la línea de implementación de este sistema así como el *profiling* y estudio de la optimización del mismo para obtener un producto cuyas características hardware correspondan a los objetivos fijados. Evitando sobredimensionar los componentes empleados para cubrir dichas necesidades.

No obstante, esta primera versión del proyecto deja en consideración algunas mejoras como un botón para comenzar una nueva ruta, algunas mejoras en la interfaz de usuario o la resolución de posibles *bugs* que puedan detectarse.

Anexos

Código fuente GitHub

Se adjunta el enlace al repositorio de GitHub además se incluyen las fuentes en el entregable del proyecto junto a esta tesina.

<https://github.com/Carlosmape/CyLock>

SIM900 Datasheet

Se adjunta el enlace de descarga así como una copia del fichero en PDF en el entregable del proyecto:

ftp://imall.iteadstudio.com/IM120417009_IComSat/DOC_SIM900_Hardware%20Design_V2.00.pdf