



## **UNIVERSIDAD AUTÓNOMA DE ENCARNACIÓN**

### **PROTOTIPO FUNCIONAL DE UNA APLICACIÓN MÓVIL PARA ALERTAS DE LOS SENSORES DE UN AUTOMÓVIL CON OPENXC**

Carlos Emanuel Mathiasen

Tesis presentada a la Facultad de Ciencia, Arte y Tecnología como requisito para la obtención del título de Licenciado/a en Análisis de Sistemas Informáticos.

**Encarnación, Paraguay**

**Octubre, 2016**

Carlos Emanuel Mathiasen

**PROTOTIPO FUNCIONAL DE UNA APLICACIÓN MÓVIL PARA ALERTAS DE LOS  
SENSORES DE UN AUTOMÓVIL CON OPENXC**

Tesis presentada a la Facultad de Ciencia, Arte y  
Tecnología, Universidad Autónoma de  
Encarnación (UNAE) como requisito para la  
obtención del título de Licenciado/a en Análisis de  
Sistemas Informáticos.

Línea de investigación: Reingeniería de  
Negocios.

Orientador/a: Ing. Hugo Sendoa

**Encarnación, Paraguay**

**Octubre, 2016**



**Autorizo la reproducción total o parcial de este trabajo, por cualquier medio convencional o electrónico, para fines de estudio e investigación, siempre que sea citada la fuente.**

Los hechos e ideas expresados en este trabajo de investigación son de responsabilidad exclusiva del autor.

### **FICHA CATALOGRÁFICA**

Mathiasen, Carlos Emanuel (2016). Prototipo Funcional De Una Aplicación Móvil Para Alertas De Los Sensores De Un Automóvil Con Openxc.

Encarnación, Universidad Autónoma de Encarnación, 113 p.

Orientador: Ing. Hugo Sendoa

Tesis presentada a la Facultad de Ciencia, Arte y Tecnología, Universidad Autónoma de Encarnación (UNAE) como requisito para la obtención del título de Licenciado/a en Análisis de Sistemas Informáticos.

Línea de investigación: Reingeniería de Negocios.

Palabras claves: OpenXC, Android, Automóvil, Celular.

CARLOS EMANUEL MATHIASSEN

**PROTOTIPO FUNCIONAL DE UNA APLICACIÓN MÓVIL PARA ALERTAS DE LOS  
SENSORES DE UN AUTOMÓVIL CON OPENXC**

Tesis presentada a la Facultad de Ciencia, Arte y Tecnología, Universidad Autónoma de Encarnación (UNAE) como requisito para la obtención del título de Licenciado/a en Análisis de Sistemas Informáticos.

Línea de Investigación: Ingeniería Automotriz.  
Orientador/a: Ing. Hugo Sendoa

Aprobado en Encarnación, Paraguay, el 31/10/2016

Calificación: .... (en números) y letras.....

**Sínodo Examinador**

Prof. \_\_\_\_\_

Firma: \_\_\_\_\_

Prof. \_\_\_\_\_

Firma: \_\_\_\_\_

Prof. \_\_\_\_\_

Firma: \_\_\_\_\_



Dedico esta tesis a:

Yani, mi compañera incondicional, que estuvo  
conmigo día a día estos últimos años ayudándome  
en todo lo que necesitaba

Agradezco a:

Mis padres en primer lugar por estar conmigo y aguantarme durante toda mi vida universitaria. Mis hermanas, que siempre estuvieron presentes. Y a todos esos profesores, tutores y amigos profesionales que día a día me hacen ser un poco más profesional.

Esta es la filosofía de UNIX: Escribe programas que hagan una cosa y la hagan bien. Escribe programas para trabajar junto a otros. Escribe programas para manejar texto, porque este es la interfaz universal. (Raymond, 2003)

## TABLA DE CONTENIDO

<b>Prototipo Funcional de una Aplicación Móvil para Alertas de Los Sensores de un Automóvil con OpenXC .....</b>	<b>4</b>
Preguntas Específicas de Investigación .....	5
Objetivos .....	5
General .....	5
Específicos .....	5
Justificación .....	6
 <b>Revisión de Literatura .....</b>	 <b>7</b>
Bases Teóricas .....	7
El Automóvil .....	7
Historia .....	7
Etapa de la Invención .....	7
Etapa Veterana .....	7
Etapa de Latón o Eduardiana .....	8
Etapa de Época(Vintage) .....	8
Etapa Pre-Guerra .....	8
Etapa Post Guerra .....	8
Etapa Moderna .....	8
Avances de la Tecnología del Automóvil .....	9
1908 – Modelo T .....	9
1911 – Arranque Eléctrico .....	9
1925 – Encendedor de Cigarrillos .....	9
1929 – Frenos en las 4 ruedas .....	10
1930 – Radio .....	10
1934 – Barras de Suspensión .....	10
1949 – Aerocar .....	10
1956 – Dirección Asistida .....	10
1969: Caja Automática de 3 velocidades .....	10
1970: Reproductor de Cassettes .....	11
1988: Air Bags .....	11
1996:On Board Disgnostics .....	11
On Board Disgnostics(OBD): Sistema de Diagnóstico a Bordo .....	13
Historia .....	13
Interfaces estándares .....	15
OBD-II Protocolos de Señal .....	18
OBD-II Formas de comunicación .....	22
Aplicaciones del OBD .....	25
Herramientas de escaneo de mano .....	25
Instrumentos suplementarios para el conductor .....	29
Internet of Things .....	30
Orígenes .....	30
Definición .....	30
Utilidades .....	30

OpenXC Platform .....	32
Vehicle Interface (VI) Firmware. ....	33
Librería Python. ....	33
Librería Android y Enabler.....	33
Librería IOS. ....	34
Formato de mensajes.....	34
JSON: .....	34
Protocol Buffers (protobuf): .....	34
<b>Metodología o Materiales y Métodos .....</b>	<b>35</b>
Definición del Tipo y Diseño de Investigación. ....	35
Descripción de la Población.....	35
Descripción de técnicas e instrumentos de recolección, medición, procesamiento y análisis de los datos .....	36
Procedimientos de aplicación de instrumentos .....	36
Delimitación.....	37
Alcances.....	37
Limitaciones.....	38
<b>Resultados y Discusión .....</b>	<b>39</b>
Vehículos Ford donde se realizaron pruebas de OpenXC por la comunidad de desarrolladores. ....	39
Vehículos Ford donde se realizaron pruebas de OpenXC propias. ....	40
Vehículos de distintas marcas donde se realizaron pruebas de OpenXC propias.....	40
Datos relevantes para el Prototipo. ....	41
Lista de reparaciones que incluye el Prototipo .....	45
Requisitos del Prototipo .....	46
Bosquejos de Configuración de Datos de Usuario. ....	46
Bosquejos de pantallas principales de la aplicación .....	50
.....	51
Selección de entorno tecnológico .....	52
Entorno de Desarrollo .....	52
Versión de Plataforma Android. ....	53
Gestor de Base de Datos. ....	53
Herramientas Colaborativas y Control de Versiones.....	54
Modelo de Datos.....	54
Plan de Desarrollo.....	55
Metodología Aplicada.....	55
Desarrollo de la Aplicación. ....	56
Descripción General del Prototipo.....	56
Cálculo del kilómetro de alerta .....	65
Ejemplo de uso de la fórmula para el cálculo de kilómetro de alerta .....	65
Pruebas y Correcciones.....	66
<b>Conclusión .....</b>	<b>69</b>
<b>Recomendaciones .....</b>	<b>71</b>

<b>Lista de Referencias.....</b>	<b>72</b>
Anexos .....	78
Anexo 1: Enlaces a los proyectos OpenXC.....	78
VI Firmware:.....	78
Librería Python .....	78
Librería Android y Enabler:.....	78
Librería IOS: .....	78
Formato de Mensajes:.....	78
Anexo 2 .....	79
Parameters ID para modo \$01 y \$02.....	79
Parameters ID para modo \$05.....	84
Parameters ID para mode \$06.....	85
Parameters ID para modo \$09.....	88
Anexo 3: Datos de Pruebas para Emular el Hardware de OpenXC.....	90
Anexo 4: Vehículos donde se realizaron pruebas de OpenXC por la comunidad de desarrolladores.....	91
Anexo 5: Encuesta realizada a personal de talleres mecánicos .....	97

## LISTA DE TABLAS

TABLA 1 - SAE J1850 PWM .....	19
TABLA 2 - SAE J1850 VPW.....	19
TABLA 3 - ISO 9141-2 .....	20
TABLA 4 - ISO 14230 KWP2000.....	21
TABLA 5 - ISO 15765 CAN .....	21
TABLA 6 - VEHÍCULOS TESTEADOS CON OPENXC POR LA COMUNIDAD DE DESARROLLADORES .....	39
TABLA 7 - VEHÍCULOS TESTEADOS CON OPENXC QUE SON MARCA FORD .....	40
TABLA 8 - VEHÍCULOS TESTEADOS CON OPENXC QUE NO SON MARCA FORD ....	40
TABLA 9 - LISTA DE REPARACIONES QUE INCLUYE EL PROTOTIPO .....	45
TABLA 10 - VERSIÓN DE PLATAFORMA ANDROID.....	53

## LISTA DE GRÁFICOS

GRÁFICO 1 - ENCUESTA A MECÁNICOS: ¿CADA CUÁNTOS KILÓMETROS SE HACE EL CAMBIO DE ACEITE? .....	41
GRÁFICO 2 - ENCUESTA A MECÁNICOS: ¿CADA CUÁNTOS KILÓMETROS SE HACE EL CAMBIO DE FILTRO DE ACEITE? .....	41
GRÁFICO 3 - ENCUESTA A MECÁNICOS: ¿CADA CUÁNTOS KILÓMETROS SE HACE EL SOPLADO DE FILTRO DE AIRE? .....	42
GRÁFICO 4 - ENCUESTA A MECÁNICOS: ¿CADA CUÁNTOS KILÓMETROS SE HACE EL CAMBIO DE FILTRO DE AIRE? .....	42
GRÁFICO 5 - ENCUESTA A MECÁNICOS: ¿CADA CUÁNTOS KILÓMETROS SE HACE EL CAMBIO DE FILTRO DE POLEN? .....	42
GRÁFICO 6 - ENCUESTA A MECÁNICOS: ¿CADA CUÁNTOS KILÓMETROS SE HACE EL CAMBIO DE ESCOBILLAS DEL LIMPIA PARABRISAS? .....	43
GRÁFICO 7 - ENCUESTA A MECÁNICOS: ¿CADA CUÁNTOS KILÓMETROS SE HACE EL CAMBIO DE BUJÍAS? .....	43
GRÁFICO 8 - ENCUESTA A MECÁNICOS: ¿CADA CUÁNTOS KILÓMETROS SE HACE EL CAMBIO DE PASTILLAS DE FRENO? .....	43
GRÁFICO 8 - ENCUESTA A MECÁNICOS: ¿CADA CUÁNTOS KILÓMETROS SE HACE EL CAMBIO DE CORREA DE DISTRIBUCIÓN? .....	44
GRÁFICO 8 - ENCUESTA A MECÁNICOS: ¿CADA CUÁNTOS KILÓMETROS SE HACE EL CAMBIO DE NEUMÁTICOS? .....	44
GRÁFICO 11 - ENCUESTA A MECÁNICOS: ¿CADA CUÁNTOS KILÓMETROS SE HACE EL CAMBIO DE FUSIBLES Y LÁMPARAS? .....	44

GRÁFICO 12 - ENCUESTA A USUARIOS: ¿TIENES AUTO? .....	67
GRÁFICO 13 - ENCUESTA A USUARIOS: ¿ CADA CUÁNTOS KILÓMETROS LO LLEVAS AL SERVICE? .....	68
GRÁFICO 14 - ENCUESTA A USUARIOS: ¿ UTILIZARÍAS UNA APLICACIÓN MÓVIL QUE TE AVISE CUANDO ES TIEMPO DE CAMBIAR ALGO EN TU AUTO? .....	68

## LISTA DE FIGURAS

FIGURA 1 - BOSQUEJO DE CONECTOR ALDL .....	15
FIGURA 2 - CONECTOR ALDL .....	15
FIGURA 3 - CONECTOR M-OBD .....	16
FIGURA 4 - CONECTOR OBD-II .....	17
FIGURA 5 - CÓDIGO OBD-II .....	23
FIGURA 6 - ELM 327 - LECTOR OBD-II.....	25
FIGURA 7 - OPENXC VI - LECTOR OBD-II.....	26
FIGURA 8 - ESCANER AUTEL MAXISCAN MS300.....	27
FIGURA 9 - ESCANER INNOVA 3100 .....	28
FIGURA 10 - ESCANER U480 CAN.....	29
FIGURA 11 - INTRANET OF THINGS.....	31
FIGURA 12 - INTERNET OF THINGS .....	31
FIGURA 13 - BOSQUEJO DE APLICACIÓN - PRIMERA PANTALLA DE CONFIGURACIÓN .....	46
FIGURA 14 - BOSQUEJO DE APLICACIÓN - SEGUNDA PANTALLA DE CONFIGURACIÓN .....	47
FIGURA 15 - BOSQUEJO DE APLICACIÓN - TERCERA PANTALLA DE CONFIGURACIÓN .....	47
FIGURA 16 - BOSQUEJO DE APLICACIÓN - CUARTA PANTALLA DE CONFIGURACIÓN .....	48
FIGURA 17 - BOSQUEJO DE APLICACIÓN - QUINTA PANTALLA DE CONFIGURACIÓN .....	48

FIGURA 18 - BOSQUEJO DE APLICACIÓN - SEXTA PANTALLA DE CONFIGURACIÓN .....	49
FIGURA 19 - BOSQUEJO DE APLICACIÓN - PANTALLA PRINCIPAL .....	50
FIGURA 20 - BOSQUEJO DE APLICACIÓN - PANTALLA DE ALERTAS.....	51
FIGURA 20 - BOSQUEJO DE APLICACIÓN - PANTALLA AUTO.....	50
FIGURA 22 - MODELO DE DATOS DE APLICACIÓN .....	55

## **LISTA DE ABREVIATURAS**

IoT	Internet of Things
ORM	Object-Relational Mapping
OEA	Organización de los Estados Americanos
VI	Vehicle Interface
APK	Android Application Package

**PROTOTIPO FUNCIONAL DE UNA APLICACIÓN MÓVIL PARA ALERTAS DE LOS  
SENSORES DE UN AUTOMÓVIL CON OPENXC**

Autor: Carlos Emanuel Mathiasen

Orientador/a: Ing. Hugo Sendoa

Línea de Investigación: Ingeniería Automotriz

**RESUMEN**

Este trabajo aborda la creación de un “Prototipo Funcional de una Aplicación Móvil Para Alertas de los Sensores de un Automóvil con OpenXC” debido al poco conocimiento que se tiene actualmente sobre esta materia a nivel nacional, existiendo una escasez del desarrollo de aplicaciones de este tipo en la zona, que exploten toda la información que un automóvil brinda y aprovechando la api liberada de OpenXC. El objetivo de esta investigación fue desarrollar un prototipo de aplicación móvil para el envío de alertas tempranas de reparación de un automóvil utilizando sus sensores y OpenXC, un hardware y software abierto desarrollado por la compañía de automóviles Ford. Fue desarrollado en el lenguaje nativo de Android, Java, conectándose vía Bluetooth con el automóvil. Para gestionar los datos del usuario en el dispositivo móvil se utilizó la base de datos SQLite 3 con el ORM ActiveAndroid, logrando una integración automóvil-smartphone propósito de este prototipo.

Finalmente evidenciamos un aumento de información brindaba al usuario por distintas interfaces que pueden ser aprovechadas para generar aplicaciones útiles de uso diario y que nos permiten mediante la información temprana reducir riesgos de desperfectos y elevar la vida útil del automóvil utilizando sus sensores aplicados a dispositivos móviles con Android.

PALABRAS CLAVE: OpenXC, Android, Automóvil, Celular.

**EL TITULO DE LA OBRA TRADUCIDA AL GUARANÍ**

Apohára: Carlos Emanuel Mathiasen

Sambyhyhára: .....

Tembikuaareka rape:.....

**ÑEMOMBYKY**

Ko tembiapo oñemoñepyrū ojejapohaguã ko mba'e oñembohérava “Prototipo Funcional de una Aplicación Móvil Para Alertas de los Sensores de un Automóvil con OpenXC”, pe tembikuua sai rupi ko mba'e rehegua tetã Paraguáipe ha pokã rupi japopyrã peíchagua ko jerére, ojeporu pe API OpenXC onohēvaekue. Ojejapohaguã ejeipuru JAVA ñe'ē, péa ha'e, umi pymbyry popeguáva oguerekóva Android peguarã ha eñemonguetáva mbayrukuérandive Bluetooth rupive.

PALABRAS CLAVE: OpenXC, Android, Automóvil, Celular.

## FUNCTIONAL PROTOTYPE OF A MOBILE APPLICATION FOR ALERTS OF CAR'S SENSORS USING OPENXC

Author: Carlos Emanuel Mathiasen

Advisor: Ing. Hugo Sendoa

Research Line: Automotive Engineering

### SUMMARY

This paper deals with the development of a "FUNCTIONAL PROTOTYPE OF A MOBILE APPLICATION FOR ALERTS of CAR'S SENSORS USING OPENXC" owing to the fact that the little knowledge about this at national level, showing a lack of application development in the country, exploiting all the information that a car provides and using the API released from OpenXC. The objective of this research was to develop a prototype mobile application for sending early warnings of car repairing using its sensors and OpenXC, open hardware and open source developed by the Ford Motor Company. The prototype was developed on android's native language, Java, using Bluetooth port to connect with the car. To deal with user data in the device, use SQLite 3 with Active Android ORM, making a perfect integration between car and mobile. Finally evidence an increase of information provided to user from different interfaces that can take advantage to generate a lot of applications, that can permit, using the car information, brought down risks and raise the useful life of the cars.

*KEY WORDS:* OpenXC, Android, Car, Mobile.

## **Prototipo Funcional de una Aplicación Móvil para Alertas de Los Sensores de un Automóvil con OpenXC**

Hoy en día el automóvil se está convirtiendo cada vez más en una necesidad para poder transportarse desde un lugar a otro. Y a medida que la tecnología avanza, los automóviles traen más y más sensores para darle una información más precisa al conductor, desde un sensor que avise que las puertas no están bien cerradas hasta uno que nos dice el ángulo actual de las ruedas a la hora de estacionar, pasando por los más conocidos como los del tanque de combustible, la velocidad actual, las RPM actuales, entre otros. Y del lado opuesto tenemos a la persona inexperta que no conoce mucho sobre todo lo que su vehículo necesita, desde mantenimientos cada cierta cantidad de kilómetros, cambio de cubiertas, cambio de aceite, filtros de aire y nafta, pastillas de freno entre tantos otros.

Por estas razones existe la necesidad de crear una aplicación móvil que pueda conectarse con los sensores del automóvil y dar alertas al usuario de los momentos para realizar tareas de mantenimiento en su vehículo.

En vista a esta situación surge el siguiente planteamiento: **¿Es posible crear una aplicación móvil que se conecte al vehículo y pueda dar información precisa para que el usuario pueda llevar su auto al taller a tiempo utilizando OpenXC?**

### Preguntas Específicas de Investigación

¿Cuáles son los requerimientos de la aplicación móvil?

¿Cuál es la arquitectura de hardware y software necesaria para la aplicación móvil para que pueda leer los sensores del automóvil?

¿Cuáles son los resultados del desarrollo de una aplicación móvil encargada de dar alertas tempranas para el mantenimiento del automóvil?

¿Cuáles son los métodos de gestión de calidad?

¿Cuáles son los resultados obtenidos luego de aplicar los métodos de gestión de calidad?

### Objetivos

**General.** Desarrollar un prototipo funcional de aplicación móvil para el envío de alertas tempranas de reparación de un automóvil utilizando sus sensores y OpenXC.

#### Específicos.

- Investigar y analizar el hardware que utilizan los automóviles hoy en día para brindar información al usuario.
- Diseñar la arquitectura de hardware y software necesarios para poder crear la aplicación móvil.
- Desarrollar el prototipo funcional de una aplicación móvil que pueda leer la información brindada por el automóvil para el envío de alertas tempranas de reparación.
- Realizar las pruebas pertinentes con emuladores de sistemas OBD-II y realizar ajustes necesarios según los resultados.

## Justificación

Hoy en día con todo el avance de la tecnología, se conoce muy poco sobre todo el hardware que brindan los automóviles y es muy poco explotado aún. Existen aplicaciones que muestran todos los datos que puede brindar un vehículo, pero no le dan un tratamiento que sea de interés al usuario a esa información, por lo que no tienen tanto éxito, excepto para las personas más expertas en el tema.

Esta propuesta pretende hacer llegar información útil al conductor del vehículo y que desconoce sobre todo el mantenimiento que los automóviles necesitan, dándole la posibilidad de poder tener en un estado óptimo su vehículo.

## Revisión de Literatura

### Bases Teóricas

#### El Automóvil.

**Historia.** Los automóviles aparecieron por primera vez en la segunda mitad del siglo XVIII y han ido evolucionando desde entonces para llegar a los que conocemos hoy en día. La historia del automóvil se divide en siete etapas (López Echegoyán, 2012).

*Etapa de la Invención.* Nicolas Joseph Cugnot era un escritor e inventor francés que inventó un automóvil de vapor que fue construido para desplazar grandes piezas de artillería. Cugnot llamó a la máquina Fardier. El Fardier comenzó a circular por las calles de París en 1769, un año después se construyó un segundo modelo más grande que el primero y más potente que podía arrastrar 4'5 toneladas de peso a una velocidad de 4 km/h. (Martinez, 2012).

En 1891, Panhard & Levassor desarrolla el System Panhard, según el cual el motor se dispone en la parte delantera del vehículo, accionando las ruedas traseras. Este principio de construcción se impuso poco a poco y fue aplicado por la mayoría de los fabricantes.(IES Castillo de la Yedra, n.d.)

*Etapa Veterana.* Ya a finales del siglo XIX y principios del siglo XX había empezado a gestarse el auge por la producción masiva de automóviles en Francia y Estados Unidos, siendo las precursoras las francesas Panhard et Levassor (1889), y Peugeot (1891) y Henri Ford con su compañía Ford Motor Company en los Estados Unidos. Es este último que propone un sistema totalmente innovador que consistía en una cadena de montaje que le permitió alcanzar cifras de fabricación impensables.(IES Castillo de la Yedra, n.d.)

*Etapa de Latón o Eduardiana.* Recibe su nombre por el uso muy a menudo de latón para fabricar las carrocerías. En esta etapa los vehículos todavía recordaban a las antiguas carrozas tiradas por caballos. Comprende desde el año 1908 hasta el año 1918.(Martinez, 2012)

*Etapa de Época(Vintage).* Abarcó la década entre el final de la Primera Guerra Mundial y la caída de la bolsa que comenzó con la Gran Depresión. Los vehículos empezaron a ser cerrados en las zonas para los pasajeros, y la comodidad era mayor. Los motores eran cada vez más potentes. (Audet, 2006)

*Etapa Pre-Guerra.* Comenzó con la Depresión en 1930 y terminó después de la Segunda Guerra Mundial en 1948. Se popularizaron los guardabarros integrados y los cuerpos completamente cerrados. Alas, luces y estribos también se convierten en parte de la carrocería del coche.(Audet, 2006)

*Etapa Post Guerra.* Desde el inicio de la recuperación de la Segunda Guerra Mundial (1948) hasta la etapa Moderna. Etapa caracterizada por el desarrollo de autos más rápidos más seguros y eficientes.(IES Castillo de la Yedra, n.d.)

*Etapa Moderna.* Se Caracteriza por el desarrollo de motores más seguros y eficientes y menos contaminantes. Empiezan a desarrollarse vehículos híbridos (motores eléctricos y a combustión) y empieza la carrera por desarrollar motores cada vez más potentes y autos inteligentes con autonomía propia.(IES Castillo de la Yedra, n.d.)

### *Avances de la Tecnología del Automóvil.*

El automóvil en sus más de doscientos años de historia ha ido evolucionando a pasos agigantados, agregando cada vez más y más tecnología a la simple máquina pensada para trasportar a las personas de un lugar a otro. Hoy en día existen autos que se manejan autónomamente utilizando inteligencia artificial y con la más mínima intervención humana. (Palin et al., 2012). A continuación se detallan algunos cambios relevantes en la historia del automóvil (O'Brien, 2015), para luego avanzar sobre el tema realmente importante: On Board Diagnostic Port.

*1908 – Modelo T.* También conocido como el "Tin Lizzie", cambió la forma en que los estadounidenses vivían, trabajaban y viajaban. Con los revolucionarios avances en la línea de montaje de Henry Ford en la fabricación de automóviles, hacen que el Modelo T sea el primer coche al alcance de la mayoría de las personas. (History.com Staff, 2010)

*1911 – Arranque Eléctrico.* Desarrollado por Charles Franklin Kettering en conjunto con Edward A. Deeds, consistía en un aparato distribuidor que llevaba la corriente de alto voltaje hacia cada una de las bujías. Este sistema fue instalado por primera vez en coches Cadillac y generalizado rápidamente a toda la industria automovilística norteamericana.(biografiasyvidas.com Staff, 2004)

*1925 – Encendedor de Cigarrillos.* Aprovechando la energía eléctrica, recién implementada se desarrollaron los llamados Encendedores “Inalámbricos” de Cigarrillos para el auto.(Laukkonen, 2016)

*1929 – Frenos en las 4 ruedas.* Fue la primera vez que se pusieron frenos individuales en cada rueda para poder tener un frenado más seguro.(O’Brien, 2015)

*1930 – Radio.* Por solo 130 dólares americanos se podía conseguir un aparato revolucionario para poder escuchar transmisiones de radio desde la comodidad del automóvil. (O’Brien, 2015)

*1934 – Barras de Suspensión.* Empiezan a aparecer las barras de suspensión individuales para cada rueda, haciendo más confortable el andar. (O’Brien, 2015)

*1949 – Aerocar.* Se crea el primer auto que también puede volar en el mundo, lo realizó la Moulton B. Taylor.(AeroCar, 1999)

*1956 – Dirección Asistida.* La dirección asistida tiene una larga historia, comenzando a finales del siglo 19. Pero es en la década del ‘50 donde 1 de cada 4 autos la trae incorporada.(Patrascu, 2010)

*1969: Caja Automática de 3 velocidades.*

Si bien las transmisiones automáticas venían de la década del 20(mistertransmission.com, 2000), tuvieron una pausa durante la segunda guerra mundial. Se hicieron fuertes ya al finalizar la década del 50 y principios de ‘60 donde muchas marcas la incluían de forma estándar en sus vehículos (O’Brien, 2015).

*1970: Reproductor de Cassettes.* El despliegue de casetes permitió uno de los grandes logros de la humanidad: la cinta de la mezcla. Este desarrollo también anunció la creación del mercado de accesorios reproductores de casetes de cinta-marca de Alpine y Pioneer, entre otros(JUSTIN BERKOWITZ, 2010).

*1988: Air Bags.* El sensor de choque de los automóviles fue inventado en 1968 por Allen Breed (Second Chance Garage, n.d.). Pero recién en 1988, la compañía de automóviles Chrysler se convierte en la primera que ofrece el sistema de airbags como un estándar en sus automóviles (Bellis, 2016).

*1996: On Board Diagnostics.* Un sistema de diagnóstico a bordo, fue implementado en todos los nuevos vehículos en Estados Unidos para controlar la emisión de gases tóxicos de los vehículos (California Environmental Protection Agency, 2016).

La industria automotriz continúa avanzando día a día ofreciendo cada vez mejores prestaciones para el usuario final. Mejoras en la seguridad, en la manejabilidad y en la comodidad de la persona que va dentro del vehículo mientras este está en marcha, son pilares que todas las compañías directa o indirectamente lo tienen y en base a eso aparecen nuevas ideas y mejoras día a día.

Hoy en día conocemos vehículos autómatas (Google, n.d.), vehículos totalmente eléctricos (Palin et al., 2012) y quizás el día de mañana podamos ver ciudades futuras donde los autos tengan

autopistas aéreas(Bisson, 1997), o que su combustible no sea más que el desecho de nuestros hogares (Gardner, 1989),

Tratar de explotar esta tecnología para utilizar los datos que nos brinda para hacer aún mejor el estar del usuario en el vehículo es el propósito de este libro.

***On Board Diagnostics(OBD): Sistema de Diagnóstico a Bordo.***

*Historia.* El Instituto NAPA de Tecnología del Automóvil, NIAT por sus siglas en inglés (Technology, 1998) nos brinda una breve reseña de cómo fue evolucionando el ODB a lo largo del tiempo:

1960: Se forma en California, Estados Unidos, el Control de Contaminación de Vehículos Motorizados (MVPCB por sus siglas en inglés) para regular la emisión de gases del caño de escape.

1966: Los controles de emisión de gases eran obligatorios para los vehículos vendidos en el estado de California y debían implementar la válvula de ventilación PCV.

1968: El congreso de Estados Unidos adopta la medida tomada por California dos años antes y lo hace de carácter federal para todos los vehículos vendidos en el país.

1968: Se forma la Junta de Recursos del Aire de California (CARB por sus siglas en inglés), responsable por la vigilar y mantener la calidad del aire en niveles saludables en todo el estado.

1970: El congreso de Estados Unidos vuelve a tomar las ideas de California para crear la Agencia de Protección Ambiental (EPA por sus siglas en inglés).

1971: Todos los nuevos vehículos son obligados a conocer la emisión de gases a la atmósfera por primera vez.

1972: Se introduce la válvula de recirculación de gases de escape o válvula EGR.

1974: El Congreso de Estados Unidos retrasa la ejecución de la obligatoriedad de los estándares de dióxido de carbono para darle mayor tiempo a las empresas para que se acomoden a las nuevas regulaciones.

1975: Aparece por primera vez el convertidor catalítico, encargado del control y la reducción de los gases.

1978 - 1981: Se empieza a ejecutar la obligatoriedad de los estándares en todos los nuevos vehículos.

1981: Todos los vehículos empezar a seguir los estándares, se conoce el convertidor catalítico de 3 vías y los sensores de oxígeno, con estos nuevos sistemas fue necesario empezar a implementar una computadora de diagnóstico a bordo.

1983: Programas de Inspección y Mantenimiento fueron establecidos en 64 ciudades de Estados Unidos, los vehículos eran obligados a pasar por estas evaluaciones para evitar gran cantidad de contaminación.

1985: CARB, EPA y la Sociedad de Ingenieros Automotrices (SAE) trabajan en conjunto para estandarizar y regular los sistemas de emisión de gases.

1988: OBD-I es obligatorio para todos los modelos vendidos en 1988 en California. Ese mismo año, California propone un conjunto de regulaciones que hoy en día se conoce como OBD-II.

1989: California adopta el estándar OBD-II, el objetivo es tener el sistema en todos los vehículos para 1996.

1990: Se realiza el mismo programa pero a nivel nacional en Estados Unidos.

1994: El primer vehículo con OBD-II es producido.

1996: Se supone que todos los vehículos ya vienen con OBD-II incorporado.

2001: La Unión Europea hace que el estándar EOBD sea obligatorio en todos los vehículos a gasolina que se venden en Europa (EU, 1998).

2008: Todos los vehículos vendidos en Estados Unidos a partir del 2008 deben seguir los estándares de la ISO 15765-4 (“EPA—APPROVED IOWA REGULATIONS Iowa citation Title State effective date EPA approval date Explanation Iowa Department of Natural Resources

Environmental Protection Commission [567] Chapter 20—Scope of Title—Definitions—Forms—  
Rules of Practice The definit,” 2005)

2010: HDOBD (Heavy Duty OBD) es obligatorio para motores no comerciales.

### *Interfaces estándares.*

A continuación, se listan las interfaces estándares de diagnóstico a bordo.

ALDL. Fue un estándar de la compañía General Motors para sus vehículos entre 1982 y 1986, antes de la estandarización del OBD-II (ws6transam.org, n.d.). Contaba con un conector de 12 pines (Ver Figuras 1 y 2).

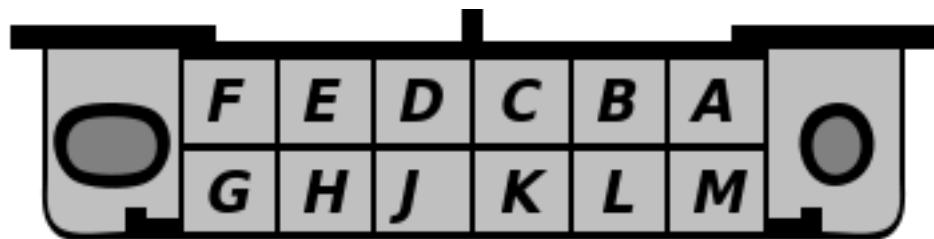


Figura 1 - Bosquejo de Conector ALDL



Figura 2 - Conector ALDL

M-OBD. También conocido como Multiplex OBD es un estándar desarrollado por Toyota, previo a OBD-II. Se utilizaba el mismo conector, pero se necesitaba un cable y un software propiedad de Toyota (Ver Figura 3).

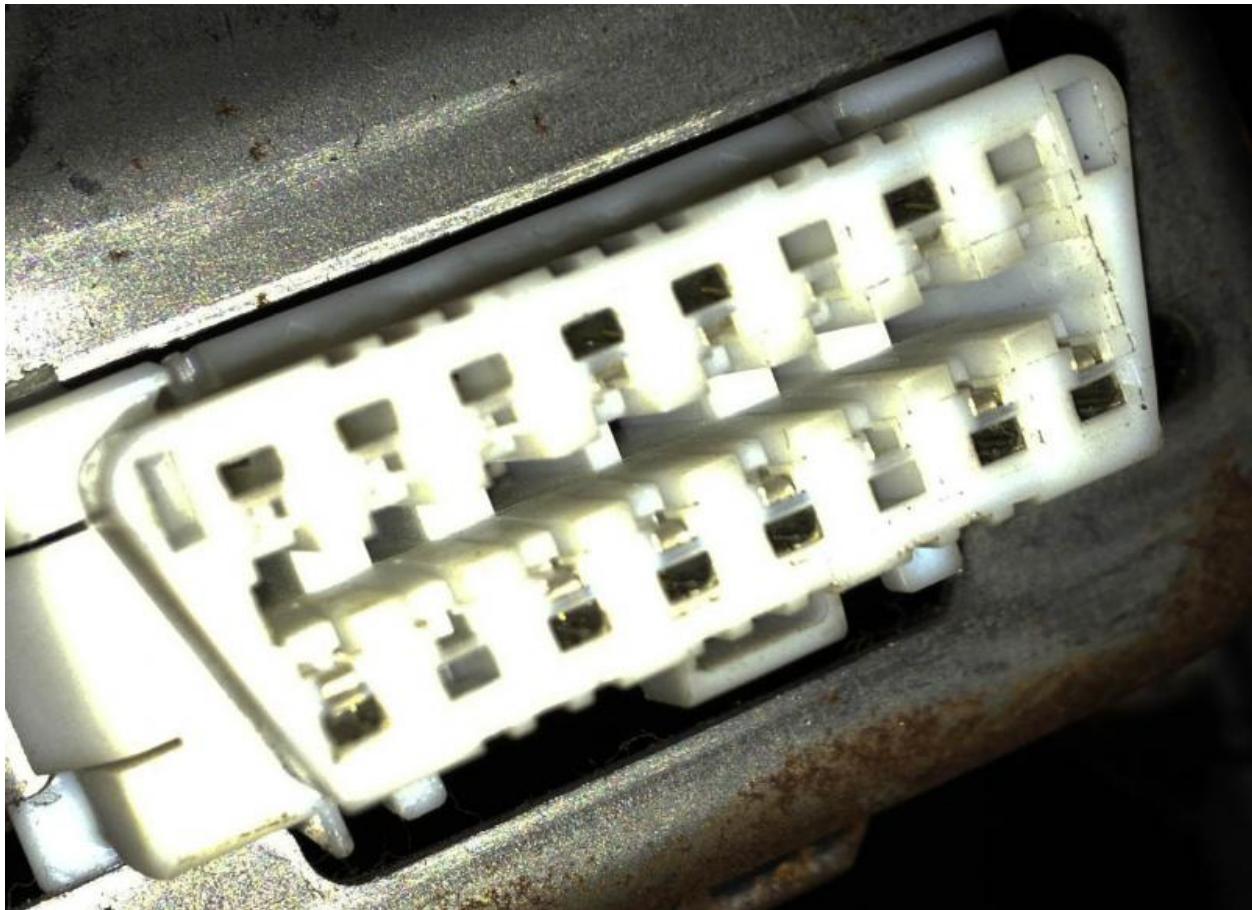


Figura 3 - Conector M-OBD

OBD-I. Se limitaban a la vigilancia de sólo unos pocos componentes relacionados con las emisiones y que no estaban calibrados para un determinado nivel de rendimiento de la emisión (dieselnet.com, 2010).

OBD-1.5. Fue utilizado también por General Motors entre 1991 y 1995, en el momento en que hacían la transición entre ALDL y OBD-II, incluso OBD 1.5 es solo un nombre simbólico, ya

que en los manuales solo se lo nombra como ALDL u OBD-II. Utilizaba el mismo circuito que ALDL, pero utilizando el puerto de 16 pines de OBD-II.

OBD-II. Es el estándar que se utiliza hoy en día en casi el 100% de los automóviles (Ver figura 4). Es la mejora del OBD-I. El estándar OBD-II especifica el tipo de conector que se debe utilizar, SAE J1962 (Engineers, 2001); los protocolos de señalización eléctricos y el formato de mensajería. De esta manera 1 solo dispositivo podría analizar y escanear distintos modelos de automóviles, sin necesidad de hacer nada extra. Si bien existe un estándar en la codificación los fabricantes pueden agregar más mensajes a este estándar propios de sus vehículos.



Figura 4 - Conector OBD-II

EOBD. Es un estándar europeo que se encuentra en vigencia desde el 2001 para vehículos a gasolina y desde 2003 para vehículos con motor diesel (EU, 1998). El estándar es muy similar al OBD-II incluso se utiliza el mismo conector, el SAE J1962.

EOBD2. No es realmente un estándar, ni mucho menos una mejora de EOBD. Sus siglas significan: “Enhanced OBD 2”, es decir una mejora de OBD-2, donde cada empresa agrega cosas al estándar OBD-II y lo lanza al mercado con este nombre. Cada nueva funcionalidad que pueda tener, depende de cada fabricante, al igual que el cómo poder leerla la información que nos brinda (Gendan.co.uk, n.d.).

JOBD. Es una versión del estándar OBD-II para autos que se venden en Japón.

ADR 79/01 & 79/02. Es un estándar Australiano, que se rige bajo los mismos principios que OBD-II utilizando incluso el mismo conector SAE J1962.

*OBD-II Protocolos de Señal.* Existen 5 diferentes protocolos de comunicación disponibles en el estándar OBD-II. Al igual que otras cosas, los fabricantes pueden utilizar el que les parezca más conveniente (Sarkfun, n.d.).

### SAE J1850 PWM.

Tabla 1 - SAE J1850 PWM

*Funcionalidad*   *Descripción*

<i>BUS +</i>	Pin 2
<i>BUS -</i>	Pin 10
<i>12V</i>	Pin 16
<i>GND</i>	Pins 4, 5
<i>Bus State</i>	Activo cuando BUS+ está en HIGH, BUS- está en LOW
<i>Maximum Signal Voltage</i>	5V
<i>Minimum Signal Voltage</i>	0V
<i>Number of Bytes</i>	12
<i>Bit Timing</i>	'1' bit - 8uS, '0' bit - 16uS, Start of Frame - 48uS

### SAE J1850 VPW.

Tabla 2 - SAE J1850 VPW.

*Funcionalidad*   *Descripción*

<i>BUS +</i>	Pin 2
<i>12V</i>	Pin 16
<i>GND</i>	Pins 4, 5
<i>Bus State</i>	LOW
<i>Maximum Signal Voltage</i>	+7V

<i>Decision Signal Voltage</i>	+3.5V
<i>Minimum Signal Voltage</i>	0V
<i>Number of Bytes</i>	12
<i>Bit Timing</i>	'1' bit -HIGH 64uS, '0' bit -HIGH 128uS, Start of Frame - HIGH 200uS

ISO 9141-2.

Tabla 3 - ISO 9141-2

<i>Funcionalidad</i>	<i>Descripción</i>
<i>Línea K (Bidireccional)</i>	Pin 7
<i>Línea L (Unidireccional, opcional)</i>	Pin 15
<i>12V</i>	Pin 16
<i>GND</i>	Pins 4, 5
<i>Bus State</i>	Línea K High. Bus Activo cuando el manejo es LOW
<i>Maximum Signal Voltage</i>	+12V
<i>Minimum Signal Voltage</i>	0V
<i>Number of Bytes</i>	Mensajes: 260, Datos: 255
<i>Bit Timing</i>	UART: 10400bps, 8-N-1

## ISO 14230 KWP2000.

Tabla 4 - ISO 14230 KWP2000.

	<i>Funcionalidad</i>	<i>Descripción</i>
<i>Línea K (Bidireccional)</i>	Pin 7	
<i>Línea L (Unidireccional, opcional)</i>	Pin 15	
<i>12V</i>	Pin 16	
<i>GND</i>	Pin 4, 5	
<i>Bus State</i>	Se activa cuando el manejo es LOW	
<i>Maximum Signal Voltage</i>	+12V	
<i>Minimum Signal Voltage</i>	0V	
<i>Number of Bytes</i>	Datos: 255	
<i>Bit Timing</i>	UART: 10400bps, 8-N-1	

## ISO 15765 CAN.

Tabla 5 - ISO 15765 CAN

	<i>Funcionalidad</i>	<i>Descripción</i>
<i>CAN HIGH (CAN H)</i>	Pin 6	
<i>CAN LOW (CAN L)</i>	Pin 14	
<i>12V</i>	Pin 16	
<i>GND</i>	Pins 4, 5	
<i>Bus State</i>	Activo cuando CANH es High	
<i>CANH Señal de Voltaje</i>	+3,5V	
<i>CANL Señal de Voltaje</i>	+1.5V	

<i>Maximum Signal Voltage</i>	CANH = +4.5V, CANL = +2.25V
<i>Minimum Signal Voltage</i>	CANH = +2.75V, CANL = +0.5V
<i>Number of Bytes</i>	CANH = +4.5V, CANL = +2.25V
<i>Bit Timing</i>	250kbit/sec or 500kbit/sec

### *OBD-II Formas de comunicación*

Según la norma ISO 15031 existen diez modos de comunicación para poder leer y obtener datos del OBD-II. Con cada uno de estos modos se pueden mandar distintos parámetros al vehículo, llamados “Parameters ID” o PID(utilsobdfacile.com, n.d.). Estos PID están descriptos en el Anexo 2 de este libro, donde existe una descripción de cada uno según el modo seleccionado.

Mode \$01. Es usado para saber que qué sensores están activos en el momento en que se está por hacer el escaneo.

Mode \$02. Encargado de mostrar los datos de “Freeze Frame” o instantáneos de una falla. Cuando ocurre una falla se guardan todos los datos de ese momento en “Freeze Frame”

Mode \$03. Este modo se encarga de mostrar los códigos de error cargados y con los que se podrá detectar exactamente qué está pasando o ha pasado (Ver figura 5).

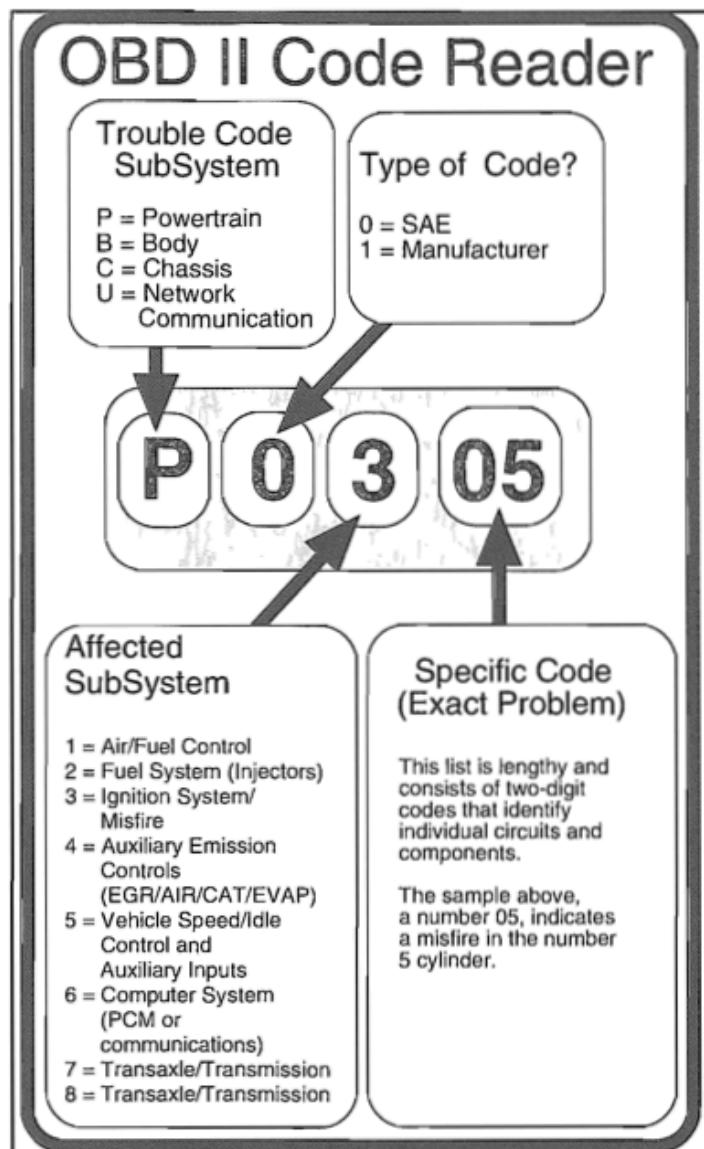


Figura 5 - Código OBD-II

Mode \$04. Es utilizado para limpiar los datos almacenados y apagar el indicador de falla del motor.

Mode \$05. Brinda información de los resultados de diagnósticos del sensor de oxígeno del vehículo. Para los vehículos que utilizan el Protocolo CAN, este modo ha caído en desuso, por lo que se debe recurrir al modo \$06.

Mode \$06. Este modo muestra los resultados de diagnósticos automáticos realizados en el sistema que no están sujetos a una constante revisión.

Mode \$07. Se utiliza para acceder a los datos de la primera vez que se manejó el vehículo una vez reseteado el sistema, para ver que realmente se haya arreglado el problema.

Mode \$08. Permite poder realizar evaluaciones externas totalmente controladas por el usuario que lo está realizando.

Mode \$09. Es el encargado de brindar información respecto al vehículo como el número de identificación y valores de calibración de todos los sensores.

Mode \$0A. Este modo sirve para almacenar los códigos de falla de una manera permanente, sin verse afectado por el modo \$04, es el propio sistema el que limpia los datos de este modo una vez que haya terminado de correr sus propios diagnósticos.

### ***Aplicaciones del OBD.***

Si bien en un principio se pensó en OBD como un sistema de diagnóstico a bordo para poder controlar las emisiones de gases tóxicos a la atmósfera, hoy podemos observar que es mucho más que eso y esa información se puede aprovechar para hacer otras cosas. A continuación, se nombran algunos ejemplos de sistemas que actualmente explotan esta información y posibles sistemas para un futuro no muy lejano.

*Herramientas de escaneo de mano.* Existen una gran cantidad de herramientas de escaneo, desde las más sencillas que solo muestran los códigos de error que ofrece el OBD-II, hasta escáneres más complejos y de varios miles de dólares que sirven para realizar la mayoría de las funciones que permite el puerto.

Conectores que necesitan de otro dispositivo y software para operar: Son dispositivos que dependen de algún hardware y software externo para brindar la información al usuario.

Uno de los productos más conocidos y recomendados para escaneo de vehículos es el ELM327 (Ver figura 6), que es un dispositivo sirve para conectarse a todos los protocolos de OBD-II. (ELMElectronics, 2005)



Figura 6 - ELM 327 - Lector OBD-II

El otro dispositivo, si bien no es conocido debido a que no está aún en producción es OpenXC (Ver figura 7). OpenXC es toda una plataforma de desarrollo para realizar aplicaciones móviles para automóviles. Desde librerías para Android, hasta emuladores de OBD-II que pueden correr en servidores web, como en aplicaciones Android.



Figura 7 - OpenXC VI - Lector OBD-II

Conectores que no necesitan de otro dispositivo y software para operar: Estos dispositivos son más utilizados en los talleres para realizar diagnósticos más completos del estado de un automóvil. A continuación se nombran algunos de los más populares (EZVID, 2015):

Autel MaxiScan MS300 es un escáner sencillo, robusto y fiable lo utilizan las personas que quieren llevar a cabo sus propios controles y los talleres mecánicos que lo utilizan para gran cantidad de marcas (EZVID, 2015).



Figura 8 - Escaner Autel MaxiScan MS300

INNOVA 3100, es un escáner para ser utilizado por una persona con mayores conocimientos en el tema, pero sin embargo sigue siendo fácil de manejar gracias a su pantalla integrada (EZVID, 2015).



Figura 9 - Escáner INNOVA 3100

U480 CAN es otro escáner más pensado para las personas que quieren ser sus propios mecánicos, muy fácil de usar. No utiliza todos los modos del estándar, pero sí tiene el de reseteo de errores (EZVID, 2015).



Figura 10 - Escaner U480 CAN

#### *Instrumentos suplementarios para el conductor.*

Hoy en día OBD es más que un estándar de diagnóstico, existen todo tipo de entusiastas que día a día tratan de hacer suplementos para el automóvil utilizando sus sensores y de esa manera brindar un mayor confort a los ocupantes del vehículo, por ejemplo, calibrando el aire acondicionado automáticamente según la temperatura externa y la interna. O haciendo un dispositivo para Internet of Things, que se encargue de mandar datos en tiempo real a una central para que, por ejemplo, las empresas de seguro puedan saber en tiempo real cuando se presenta un desperfecto en el auto.

Su aplicación tiene como límite la imaginación humana.

### **Internet of Things.**

**Orígenes.** La frase “Internet of things” o “IoT” fue acuñada hace unos 16 años por los fundadores del original MIT Auto-ID Center, con mención especial a Kevin Ashton en 1999 y David L. Brock en 2001. (Sundmaeker, Guillemin, & Friess, 2010). Actualmente el MIT Auto-ID Center es conocido como Auto-ID Labs y es el fundador de la primera conferencia de IoT a nivel mundial que reúne anualmente a investigadores y líderes de opinión.(Auto-ID, 2016). La siguiente conferencia se realizará a finales de este año en Alemania.(Conference, 2016).

**Definición.** IoT es un sistema de dispositivos de cómputos, mecánicos y digitales, objetos, animales y/o personas que tienen asignados un identificador único y tienen la habilidad de transferir datos a través de una red sin la necesidad de una interacción de humano a humano o de humano a computador.(Rouse, 2014).

**Utilidades.** En estos días existen un gran número de dispositivos, sensores entre otras cosas que están habilitados para conectarse a internet, mediante gateways, servidores proxies, etc . Esta es la topología conocida como Intranet de las cosas en lugar de IoT (Emad-ul-Haq et al., 2015), ver Figura 11. Pero esta intranet se está extendiendo más allá para poder conectarse directamente a internet sin necesidad alguna de utilizar gateways, proxies para poder realizar la conexión, ver Figura 12. Las posibilidades solo tienen un límite en la imaginación humana, desde dispositivos médicos como marcapasos y desfibriladores (Stachel, Sejdic, Ogirala, & Mickle, 2013) hasta el desarrollo de ciudades inteligentes como el caso de Songdo en Corea del Sur (Songdo, 2015), pasando por la industria automotriz con autos como el último modelo de Tesla Motors, Model S (Palin et al., 2012).



Figura 11 - Intranet of things

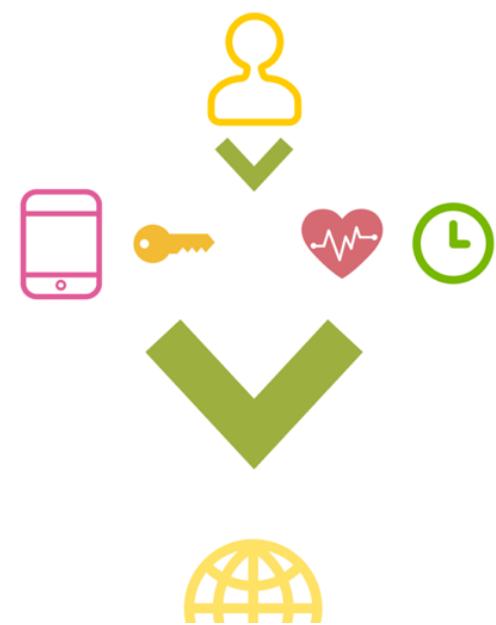


Figura 12 - Internet of Things

## OpenXC Platform

OpenXC es una combinación de hardware y software de código abierto que permite ampliar el vehículo con aplicaciones personalizadas y módulos conectables. Utiliza herramientas estándar, bien conocidas para abrir una gran cantidad de datos del vehículo para los desarrolladores.

OpenXC es una API para el automóvil - mediante la instalación de un módulo de hardware pequeño para leer y traducir las métricas de la red interna de un coche, los datos son accesibles desde la mayoría de las aplicaciones Android utilizando la biblioteca OpenXC.

Los investigadores de Ford Motor Company se unieron para crear una forma estándar de creación de software y hardware del mercado de accesorios para vehículos. Cada nuevo coche está lleno de ordenadores y electrónica, y hay un creciente interés en la conexión de la salida de estos sistemas a las aplicaciones de terceros y la web. Muchas empresas ofrecen herramientas para engancharse a la interacción del conductor, pero en su mayor parte tienen disponibilidad limitada para los aficionados y desarrolladores.

Los vehículos son indiscutiblemente cada vez más y más conectados, pero hasta ahora, el consumidor no ha tenido mucho control sobre las capacidades exactas de su coche. Muy pocos sistemas de información y entretenimiento son actualizables por el usuario, y gracias al ritmo extremadamente rápido de la electrónica de consumo y la Web (especialmente en comparación con los largos ciclos de desarrollo en la industria del automóvil) la nueva generación puede

convertirse rápidamente en la última generación. Esto se aplica tanto al hardware como al software, pero en un vehículo esto no suele ser tan así. Un automóvil de 2007 con un lector de DVD aún existe y es probable que lo sigan utilizando. Sin embargo, los primeros teléfonos con una antena 2G para poder tener internet móvil, fueron rápidamente desplazados por el siguiente modelo. El promedio de vida de un vehículo hoy es de 13 años; un teléfono inteligente es de 6 a 9 meses.

Toda la plataforma de OpenXC es de código abierto, por lo que se puede acceder a su código fuente y realizar cambios si se creyeran necesarios. En el anexo se podrán encontrar los enlaces para cada proyecto.

La plataforma está dividida en cinco proyectos que en su conjunto forman OpenXC.

**Vehicle Interface (VI) Firmware.** Es un software que se ejecuta en los microcontroladores, los cuales se conectan a los buses CAN del vehículo para poder brindar una versión genérica para interfaces usb/bluetooth.

**Librería Python.** Si bien OpenXC está pensado para utilizar en desarrollos móviles, para poder hacer pruebas más rápidas sin configurar mucho el entorno, existe la librería escrita en python donde por línea de comandos y conectados por un cable USB al hardware se pueden ver todos los datos que arroja el vehículo en tiempo real.

**Librería Android y Enabler.** Esta es la librería encargada de conectar por medio de bluetooth un dispositivo android con el hardware de OpenXC, es una librería escrita en Java y que se puede agregar fácilmente a un proyecto android utilizando los medios convencionales.

Enabler es una aplicación de prueba para la plataforma, al instalar la aplicación y conectarla por bluetooth con el hardware ya se pueden ver todos los datos que se reciben.

**Librería IOS.** Al igual que la librería para Android, también hay una para dispositivos Iphone/Ipod, escrita en Objective-C.

**Formato de mensajes.** Es una documentación donde se detalla como los datos son enviados desde el hardware y cuáles son los valores máximos y mínimos que pueden llegar a retornar. También muestra una especificación para crear un archivo con datos de prueba

La información se envía en dos formatos:

**JSON:** es un formato de intercambio de datos ligero. Es fácil leerlo y escribirlo para los seres humanos. Es fácil para las máquinas analizar y generar. Se basa en un subconjunto del lenguaje de programación JavaScript, estándar ECMA-262 3<sup>a</sup> Edición - diciembre de 1999. JSON es un formato de texto que es completamente independiente del lenguaje (Json.org, n.d.).

**Protocol Buffers (protobuf):** Es un protocolo de mecanismo extensible de plataforma neutral independiente del idioma de Google para serializar datos estructurados (Google, 2008). En el anexo se puede observar un archivo de ejemplo en formato JSON que sirve para emular datos enviados por el hardware.

## Metodología o Materiales y Métodos

### Definición del Tipo y Diseño de Investigación.

El diseño de la investigación es exploratorio. Ya que la misma se efectúa sobre un tema u objeto desconocido o poco estudiado, por lo que sus resultados constituyen una visión aproximada de dicho objeto, es decir, un nivel superficial de conocimientos (Arias, 2006).

El tipo de investigación seleccionado es aplicada. Ésta se caracteriza por su interés en la aplicación, utilización y consecuencias prácticas de los conocimientos. La investigación aplicada busca el conocer para hacer, para actuar, para construir, para modificar (GRAJALE G, 2000).

Este tipo está orientada a conocer las necesidades que no están siendo satisfechas. La misma hace uso de los conocimientos o teorías o de investigación básica para resolver un problema existente como así también implementar la solución (Baptista, Fernandez Collado, & Hernandez Zampieri, 1991).

### Descripción de la Población.

Se ha considerado como población a cualquier persona que tenga acceso a un automóvil y a un teléfono celular con bluetooth incorporado, en distintas partes del mundo.

Se ha compartido un resumen de la investigación y un video del prototipo en funcionamiento para determinar el grado de aceptación que podría tener una aplicación de este tipo en la sociedad y determinar un promedio de cantidad de kilómetros que recorren antes de llevar el auto al servicio técnico para poder realizar los cambios.

## **Descripción de técnicas e instrumentos de recolección, medición, procesamiento y análisis de los datos**

Se ha realizado encuestas a personal de talleres mecánicos, procesando las informaciones obtenidas y realizando las mediciones correspondientes, se ha comparado con bibliografía de diferentes autores determinando el contenido de la aplicación.

Para poder medir el grado de aceptación del prototipo de aplicación desarrollada, se realizó una encuesta online a personas alrededor del mundo, ofreciéndoles un resumen de la investigación y un video con el prototipo funcionando.

## **Procedimientos de aplicación de instrumentos**

Se ha confeccionado una encuesta en base a bibliografías de diversos autores.

La encuesta se aplicó a personal de talleres mecánicos visitando su lugar habitual de trabajo.

A partir del resultado de la encuesta se procedió a llenar la base de datos del prototipo con los cambios más comunes que las personas deben realizar a sus vehículos y cada cuanto tiempo deberían hacerlo.

Se han realizado bocetos para determinar los requerimientos del software.

Una vez finalizado el prototipo se realizó un video donde se muestran las funcionalidades que posee. Con este video y con un resumen de la investigación se realizó una encuesta online para poder determinar la utilidad que pueda llegar a tener una aplicación en producción.

## Delimitación

**Alcances.** El software se utilizará utilizando una plataforma brindada por la empresa Ford para poder leer los sensores de sus autos. Esta plataforma es aún un prototipo por lo que la aplicación no puede salir a producción una vez terminada.

Lista de Pantallas que tendrá el Prototipo.

- Se podrá utilizar la aplicación en teléfonos que soporten el Sistema operativo Android.
- Pantalla de bienvenida inicial:
  - El usuario podrá configurar la aplicación, tanto la conexión con el vehículo como las alertas utilizando pantallas en forma de Wizard que le llevaran paso a paso a configurarla.
- Pantalla de inicio:
  - Información de alertas actuales y las siguientes, con la opción de marcar como realizadas.
  - Cantidad de Kilómetros actuales del vehículo.
- Pantalla actual del Auto:
  - Velocidad actual del vehículo.
  - Esquema del automóvil donde mostrará el estado de las puertas (abiertas, cerradas) y el ángulo actual de las ruedas.
- Pantalla de Alertas:
  - Mostrará las alertas activas actualmente y la opción de habilitar y desactivar alertas.
- Pantalla de Configuración de Alerta
  - El usuario podrá modificar la cantidad de KM de cada alerta.
- La aplicación informará utilizando el Sistema de notificaciones de Android y según el odómetro del vehículo, cuando es necesario un cambio en el automóvil.

Se realizarán pruebas con los sensores del auto en 2 niveles. Primeramente, utilizando el software de emulación brindado por Ford y como segundo nivel se realizarán pruebas reales en automóviles, utilizando el hardware ofrecido por la misma empresa.

## **Limitaciones**

- No tendrá más características de las ya mencionadas anteriormente.
- No se podrá utilizar en teléfonos que no tengan Android como sistema operativo.
- El usuario podrá realizar el registro de un auto a la vez. No existiendo garantía de que los datos anteriores se mantengan.
- Será solo una aplicación para pruebas, ya que se utiliza OpenXC Platform que aún no está disponible para el público.

## Resultados y Discusión

De acuerdo a la investigación realizada se obtuvieron los siguientes resultados

**Vehículos Ford donde se realizaron pruebas de OpenXC por la comunidad de desarrolladores.**

A continuación, se detalla la lista de vehículos donde fue probado OpenXC Platform, estas pruebas fueron hechas por desarrolladores de todo el mundo, los resultados de esta prueba se pueden ver en el Anexo 4 de este libro.

*Tabla 6 - Vehículos testeados con OpenXC por la comunidad de desarrolladores*

<b>Modelo</b>	<b>Año</b>
<i>C-Max Hybrid</i>	2013
<i>C-Max Energi</i>	2013
<i>E350</i>	2011
<i>E350</i>	2012
<i>Escape</i>	2013
<i>Explorer</i>	2012
<i>Explorer</i>	2013
<i>F-150</i>	2011
<i>Falcon</i>	2013
<i>Figo</i>	2010
<i>Focus</i>	2009
<i>Focus</i>	2012
<i>Focus</i>	2013
<i>Focus Electric</i>	2013
<i>Focus ST</i>	2013
<i>Fiesta</i>	2012
<i>Fusion</i>	2006
<i>Fusion</i>	2013
<i>Mustang</i>	2005
<i>Mustang</i>	2012
<i>Taurus</i>	2013

<i>Territory</i>	2013
<i>MKZ</i>	2013
<i>Police Interceptor</i>	2004

### Vehículos Ford donde se realizaron pruebas de OpenXC propias.

A continuación, se detallan los vehículos marca Ford donde se realizaron pruebas propias y tuvieron un resultado favorable conectándose bien a la aplicación Enabler de OpenXC y el Prototipo de Bilen.

Tabla 7 - Vehículos testeados con OpenXC que son marca Ford

<i>Modelo</i>	<i>Año</i>
<i>Focus – Automático</i>	2014
<i>Kuga</i>	2011

### Vehículos de distintas marcas donde se realizaron pruebas de OpenXC propias.

Para propósitos de esta investigación también se hicieron pruebas con el hardware en vehículos de otras marcas, pero sin tener un resultado favorable como era de esperarse.

A continuación, se muestra la lista de vehículos donde se hicieron las pruebas.

Tabla 8 - Vehículos testeados con OpenXC que no son marca Ford

<i>Modelo</i>	<i>Año</i>
<i>Renault Clio</i>	2014
<i>Chevrolet Astra</i>	2010

### Datos relevantes para el Prototipo.

De acuerdo a diferentes informaciones obtenidas en internet, en charlas con personal de talleres mecánicos y a la encuesta desarrollada a los mismos, que se puede ver en el Anexo 5 de este libro, se pudo decidir cuáles eran los datos que deberían estar en el prototipo.

A continuación, se listan los resultados de las preguntas de la encuesta al personal de los talleres mecánicos, que en total fueron 3.

¿Cada cuántos kilómetros se hace el cambio de aceite? (3 respuestas)

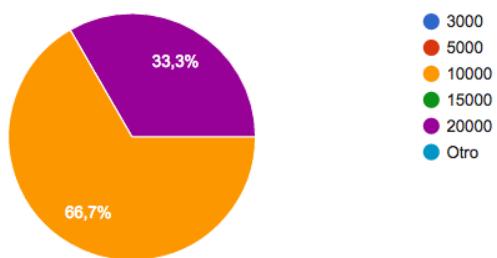


Gráfico 1 - Encuesta a Mecánicos: ¿Cada cuántos kilómetros se hace el cambio de aceite?

¿Cada cuántos kilómetros se hace el cambio de filtro de aceite? (3 respuestas)

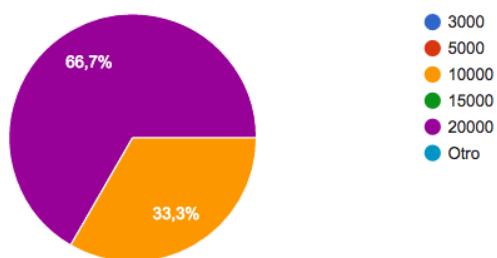


Gráfico 2 - Encuesta a Mecánicos: ¿Cada cuántos kilómetros se hace el cambio de filtro de aceite?

¿Cada cuántos kilómetros se hace el soplado de filtro de aire? (3 respuestas)

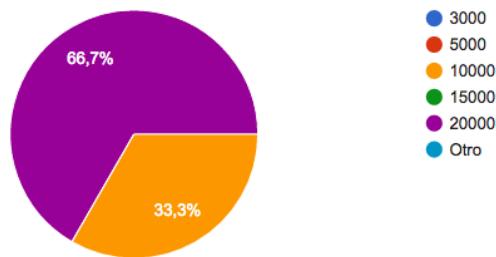


Gráfico 3 - Encuesta a Mecánicos: ¿Cada cuántos kilómetros se hace el soplado de filtro de aire?

¿Cada cuántos kilómetros se hace el cambio de filtro de polen? (3 respuestas)

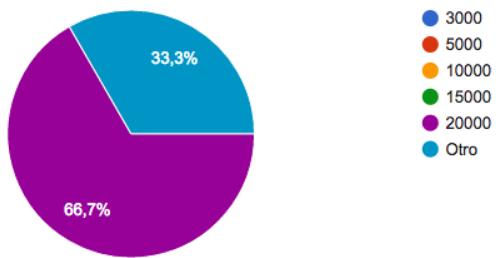


Gráfico 5 - Encuesta a Mecánicos: ¿Cada cuántos kilómetros se hace el cambio de filtro de polen?

¿Cada cuántos kilómetros se hace el cambio de filtro de aire? (3 respuestas)

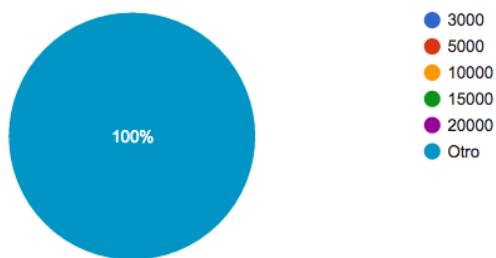


Gráfico 4 - Encuesta a Mecánicos: ¿Cada cuántos kilómetros se hace el cambio de filtro de aire?

**¿Cada cuántos kilómetros se hace el cambio de escobillas del limpia parabrisas?**

(2 respuestas)

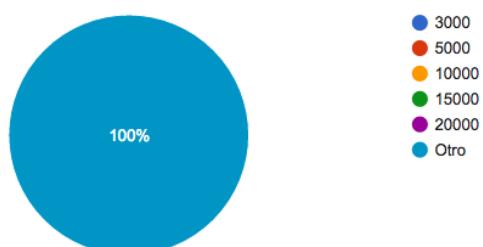


Gráfico 6 - Encuesta a Mecánicos: ¿Cada cuántos kilómetros se hace el cambio de escobillas del limpia parabrisas?

**¿Cada cuántos kilómetros se hace el cambio de bujías?** (2 respuestas)

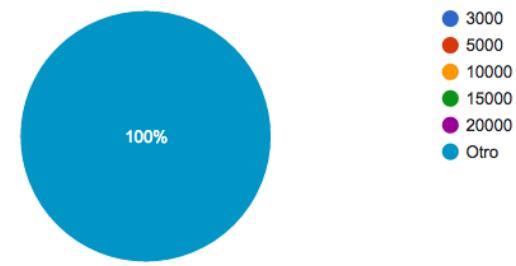


Gráfico 7 - Encuesta a Mecánicos: ¿Cada cuántos kilómetros se hace el cambio de bujías?

**¿Cada cuántos kilómetros se hace el cambio de pastillas de freno?**

(3 respuestas)

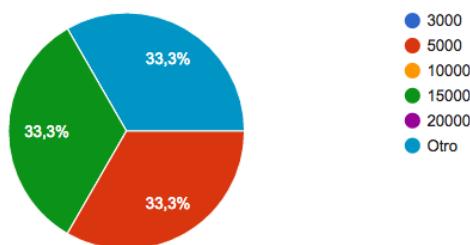


Gráfico 8 - Encuesta a Mecánicos: ¿Cada cuántos kilómetros se hace el cambio de pastillas de freno?

**¿Cada cuántos kilómetros se hace el cambio de la correa de distribución?**  
(3 respuestas)

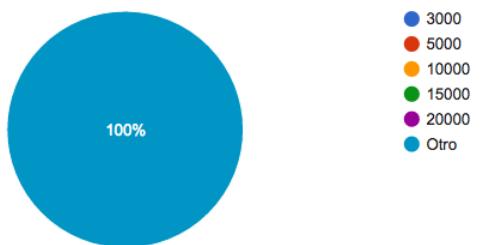


Gráfico 9 - Encuesta a Mecánicos: ¿Cada cuántos kilómetros se hace el cambio de correa de distribución?

**¿Cada cuántos kilómetros se hace el cambio de neumáticos?** (3 respuestas)

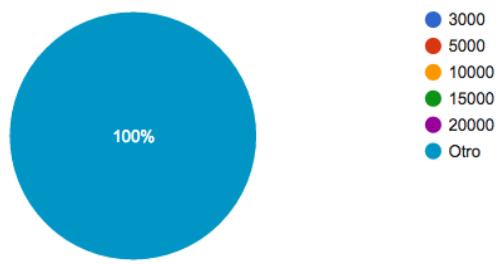


Gráfico 11 - Encuesta a Mecánicos: ¿Cada cuántos kilómetros se hace el cambio de neumáticos?

**¿Cada cuántos kilómetros se hace el cambio de Fusibles y Lámparas?**  
(3 respuestas)

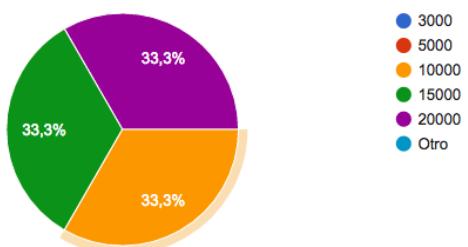


Gráfico 10 - Encuesta a Mecánicos: ¿Cada cuántos kilómetros se hace el cambio de fusibles y lámparas?

### **Lista de reparaciones que incluye el Prototipo**

Luego de ver los resultados de la encuesta y buscar información recurriendo a diversos autores, se pudo definir la lista de datos que aparecerían en el Prototipo (Ver Tabla 9).

*Tabla 9 - Lista de reparaciones que incluye el Prototipo*

<b>Reparación</b>	<b>Cantidad de KM</b>
<i>Limpia Parabrisas</i>	30000
<i>Líquido Limpia Parabrisas</i>	5000
<i>Filtro de Aire</i>	10000
<i>Filtro de Combustible</i>	15000
<i>Pastillas de Freno</i>	5000
<i>Alineación y Balanceo</i>	2000
<i>Cubiertas</i>	20000
<i>Calibración de Neumáticos</i>	1000
<i>Filtro de Aceite</i>	10000
<i>Líquido Refrigerante</i>	10000

## Requisitos del Prototipo

De acuerdo al relevamiento realizado y consultando a distintas fuentes de información, se pudo determinar los requisitos finales del prototipo.

La aplicación se divide en dos partes, la primera se ejecuta 1 sola vez al instalar la aplicación y se encarga de configurar los datos del usuario y la conexión por bluetooth con el hardware. La segunda parte consta de las pantallas necesarias para utilizar la aplicación día a día, donde se puede ver en tiempo real el estado de ciertos sensores, así como configurar las alertas que se supongan necesarias.

## Bosquejos de Configuración de Datos de Usuario.

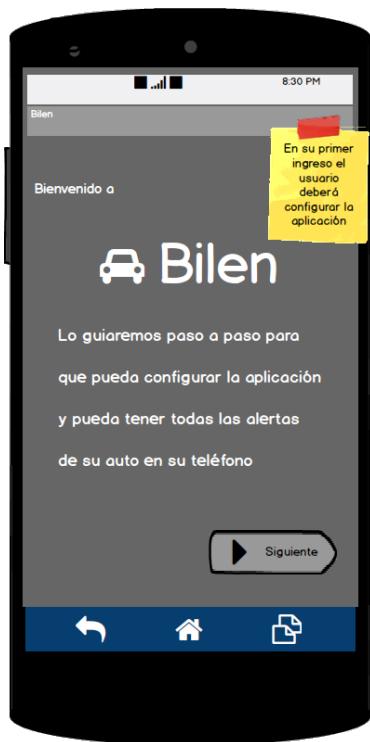


Figura 13 - Bosquejo de Aplicación -

Primera pantalla de configuración



Figura 14 - Bosquejo de Aplicación -

Segunda pantalla de configuración

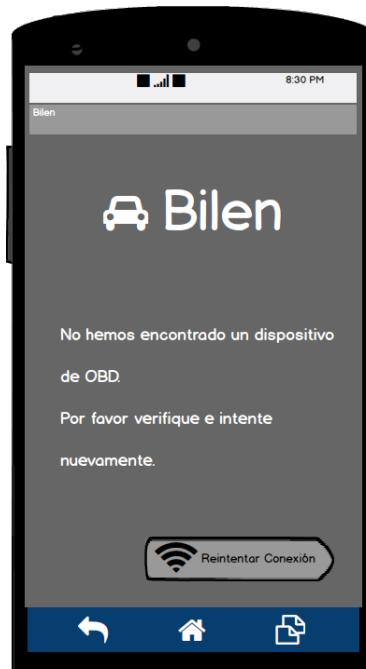


Figura 15 - Bosquejo de Aplicación -

Tercera pantalla de configuración



Figura 16 - Bosquejo de Aplicación -

Cuarta pantalla de configuración



Figura 17 - Bosquejo de Aplicación -

Quinta pantalla de configuración



Figura 18 - Bosquejo de Aplicación -

Sexta pantalla de configuración

**Bosquejos de pantallas principales de la aplicación**

Figura 19 - Bosquejo de Aplicación -

Pantalla Principal

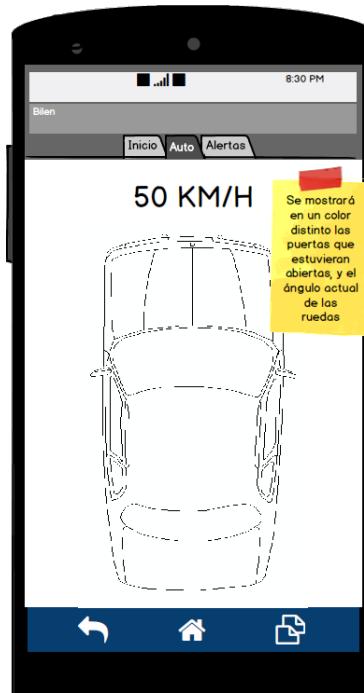


Figura 20 - Bosquejo de Aplicación -

Pantalla Auto

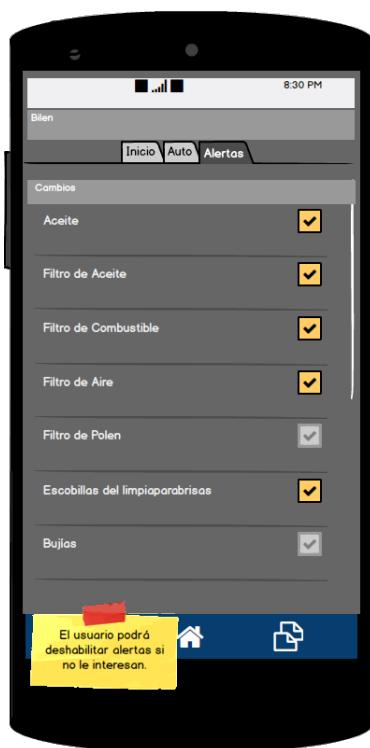


Figura 21 - Bosquejo de Aplicación -

Pantalla de Alertas

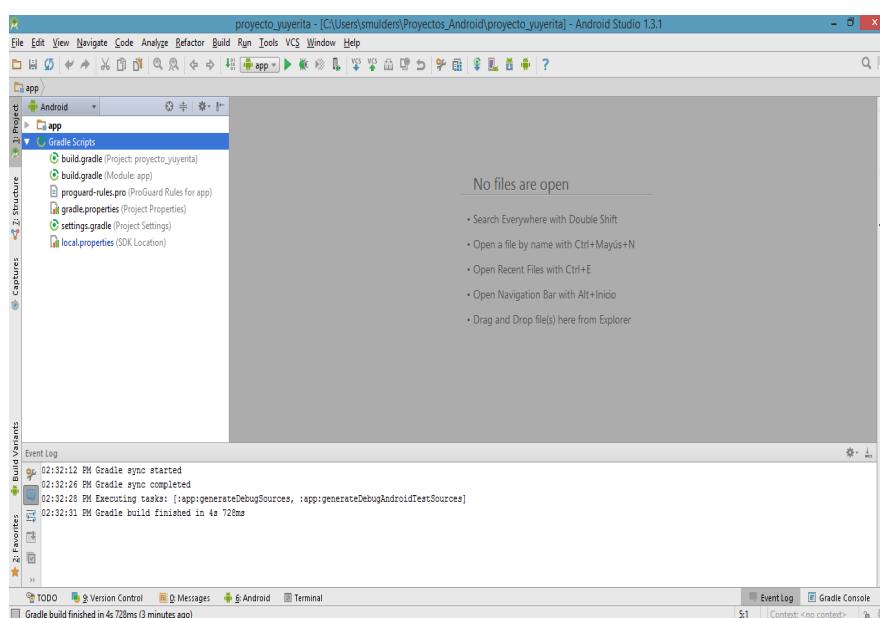
## Selección de entorno tecnológico

Se ha decidido desarrollar el prototipo para dispositivos móviles Android, debido a que al momento de empezar esta investigación la única librería testeada para utilizar OpenXC era precisamente la librería para Android. Para llevar a cabo el desarrollo se ha utilizado el siguiente entorno:

**Entorno de Desarrollo.** Se utilizó el IDE de desarrollo brindado por Google para poder crear aplicaciones móviles en Android denominado Android Studio.

Este IDE cuenta con las siguientes características. (Acerca de Android, 2015).

- Utiliza Gradle como herramienta para automatizar tareas como las tareas de compilación, testeo y empaquetado del proyecto.
- Permite generar APK (Application Application Package) para poder distribuir el software.
- Posee plantillas prediseñadas que facilitan la creación de aplicaciones.
- Editor de interfaz de usuario fácil de usar con la opción de arrastrar y soltar componentes.
- Posee integración con varios softwares emuladores de dispositivos Android, como Genymotion.



**Versión de Plataforma Android.** Se ha desarrollado la aplicación para los dispositivos móviles que posean una versión igual o superior a la 4.0.3 Ice Cream Sandwich. Al momento de empezar a desarrollar la aplicación las versiones de Android que dominaban el mercado eran las versiones 4.X, siendo la 4.0.3 la primera versión en ese rango. De acuerdo a la siguiente tabla, la cual se obtuvo de la página oficial de desarrolladores de Android, se puede justificar lo antes dicho.

Tabla 10 - Versión de Plataforma Android.

Versión	Nombre	API	Distribución
2.2	Froyo	8	0.1%
2.3.3 - 2.3.7	Gingerbread	10	1.5%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	1.4%
4.1.x	Jelly Bean	16	5.6%
4.2.x		17	7.7%
4.3		18	2.3%
4.4	KitKat	19	27.7%
5.0	Lollipop	21	13.1%
5.1		22	21.9%
6.0	Marshmallow	23	18.7%

**Gestor de Base de Datos.** Para la gestión de los datos en la aplicación se ha utilizado el motor de base de datos embebido que vienen en todos los dispositivos Android (Android, 2016). SQLite es una biblioteca que implementa, sin servidor y sin configuración, un motor de base de datos SQL transaccional autónomo (“About SQLite,” 2016).

Para tener una capa de abstracción a la base de datos y el código de la aplicación se utilizó un ORM (Object Relational Mapping) llamado Active Android. Este nos ayuda a escribir consultas

a la base de datos sin necesidad de utilizar SQL, sino a través de mensajes entre objetos (Pardo, 2010).

**Herramientas Colaborativas y Control de Versiones.** Para poder llevar un mejor control del versionado del código y la documentación de este trabajo de investigación, se utilizó GIT. GIT es un controlador de versiones de fácil manejo y con un gran potencial a la hora de poder crear ramas de desarrollo.

Se crearon 2 repositorios, el primero dedicado pura y exclusivamente a todo lo referido a documentación y otro dedicado al código fuente del Prototipo.

El servicio que se utilizó para alojar estos repositorios es Bitbucket, debido a que es un servicio gratuito y con un límite muy alto para almacenamiento de archivos grandes.

**Modelo de Datos.** Como se explicó en apartados anteriores, para llevar a cabo la gestión de los datos desde el código se utilizó Active Android, pero antes de escribir código se hizo un análisis de cuáles serían las entidades relacionales que se utilizarían en la aplicación.

La aplicación cuenta con 2 tipos de datos, los estáticos son los ya cargados en la aplicación y que no se pueden modificar, entre estos datos tenemos la lista de reparaciones antes mencionada; también tenemos los datos del automóvil que el usuario ingresa al instalar por primera vez la aplicación. Como datos dinámicos el estado actual de los sensores, que no se persistirán, sino que serán leídos en tiempo real y las alertas que el usuario configure para que le avisen cuando debe realizar un cambio. Sabiendo esto, se ha procedido a diagramar el modelado relacional de la aplicación móvil.

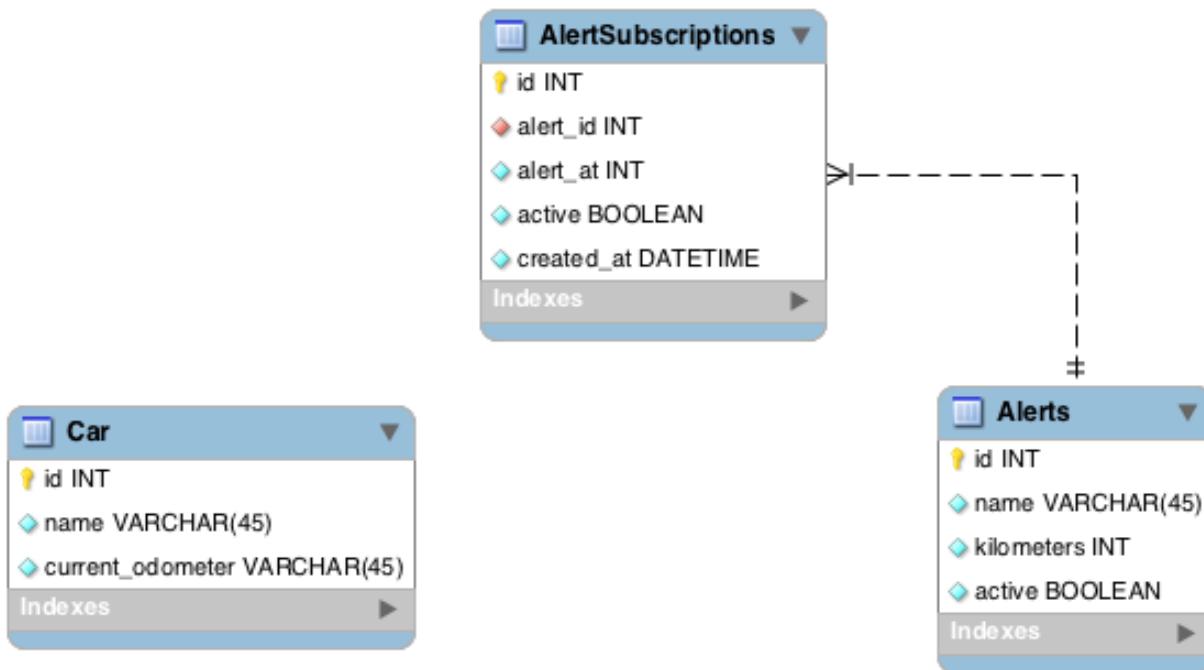


Figura 22 - Modelo de Datos de Aplicación

**Metodología Aplicada.** Para llevar a cabo el proyecto de desarrollo de software se hay procedido a utilizar la metodología ágil llamada Kanban.

Kanban es un sistema muy simple que se basa principalmente en: Visualización del flujo de trabajo, Limitación del trabajo en proceso, Medición y Gestión del Flujo y Mejora incremental. Tal como lo mencionan los autores de esta metodología, son justamente los métodos simples los que generan los menores trastornos y generalmente los más sustentables beneficios. Esta metodología puede verse como un sistema transparente y visual de limitación del trabajo en curso y arrastre (pull) del valor. Ayuda a entregar valor promoviendo el flujo y exponiendo los cuellos de botella, las colas, el impacto de los defectos, los costos económicos y por lo tanto el desperdicio. Kanban, tal como uno de sus mayores exponentes David Anderson afirma, ha demostrado ser útil en equipos que realizan desarrollo Ágil de software, pero también están

ganando fuerza en equipos que utilizan métodos más tradicionales. Kanban dentro de un proyecto o en un área tiene que ser adaptado a su contexto y a sus características específicas (Gimson, 2012).

**Desarrollo de la Aplicación.** Una vez instalado y configurado el IDE y el entorno de desarrollo se ha procedido a realizar la aplicación de la siguiente manera.

En un primer paso se procedió a organizar el código en paquetes para poder tener bien separada cada capa de la aplicación. Para ello se crearon 4 paquetes:

- activities: Dentro de este paquete se encuentran todas las clases que dan funcionalidad a la interfaz de usuario
- adapters: En este paquete se pueden encontrar todas las clases que hacen de puente entre las vistas que contienen listas y los datos que se quieren mostrar en esa lista.
- fragments: Un fragment es una clase de Android que sirve para posicionar la vista de distintas formas según el dispositivo que se utiliza y el tamaño de su pantalla. La aplicación fue pensada en pestañas, donde cada una de las pestañas es un fragment.
- fragments.wizzard: En este paquete se encuentran todos los fragments para las pantallas que aparecen al instalar la aplicación.
- models: En este paquete se encuentran las clases que utilizan Active Android y hacen de nexo entre SQLite y el resto de las clases de la aplicación.

En un segundo paso se procedió a crear todas las vistas de la aplicación, para poder presentar una demostración del funcionamiento de la aplicación con datos estáticos.

Finalmente se agregó la librería de OpenXC Android en el proyecto para poder realizar pruebas utilizando los emuladores de la plataforma OpenXC.

### ***Descripción General del Prototipo.***

Configuración de Datos de Usuario. Una vez instalada la aplicación en el dispositivo móvil, la primera pantalla que aparece es una pantalla de bienvenida para poder ir configurando la

aplicación. Esta primera parte está compuesto por una serie de pasos que van explicando al usuario que hacer para poder tener la aplicación funcionando y correctamente conectada al dispositivo del hardware.

La primera pantalla es simplemente de bienvenida, con un botón para empezar a completar los pasos de configuración.

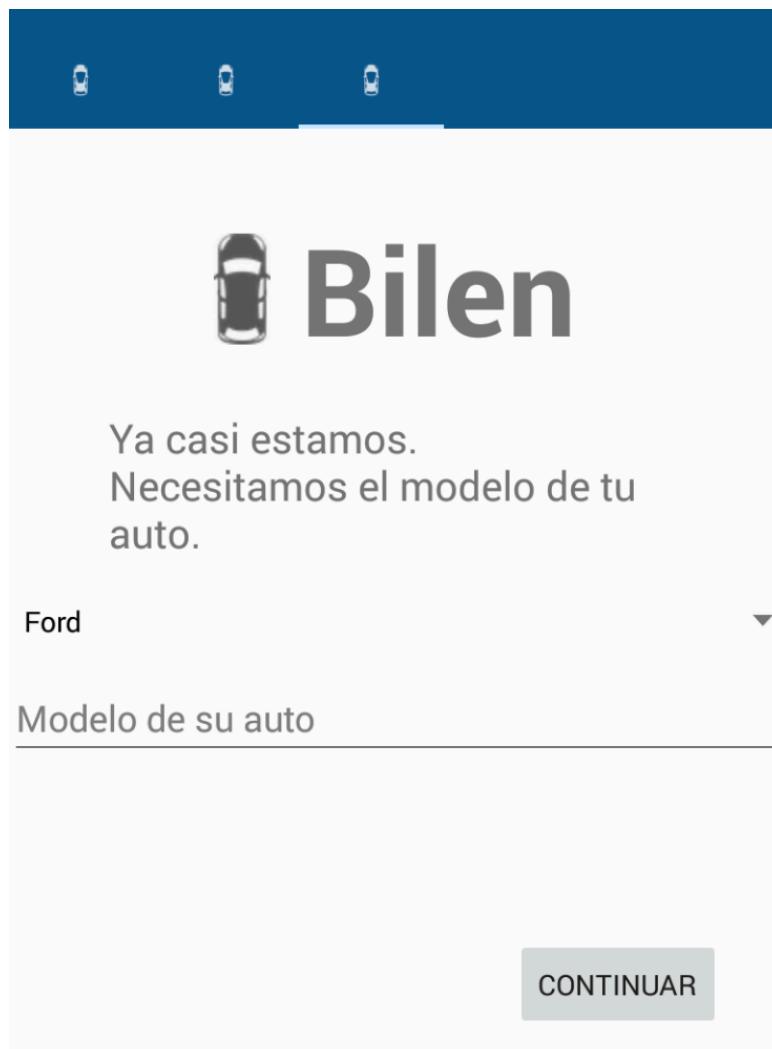


La segunda Pantalla es la pantalla encargada de conectar la aplicación a través del bluetooth del dispositivo móvil con el hardware de OpenXC. En caso de que la aplicación no pueda conectarse, volverá a recargar la misma pantalla pidiendo que vuelva a intentar conectar. Este es el paso más importante de todos, si no se puede completar, la aplicación no se terminará de configurar y no se podrá utilizar.



La tercera pantalla es la encargada de guardar la información del vehículo al que se va a conectar la aplicación, para realizar esta configuración, el usuario tiene un formulario con dos campos.

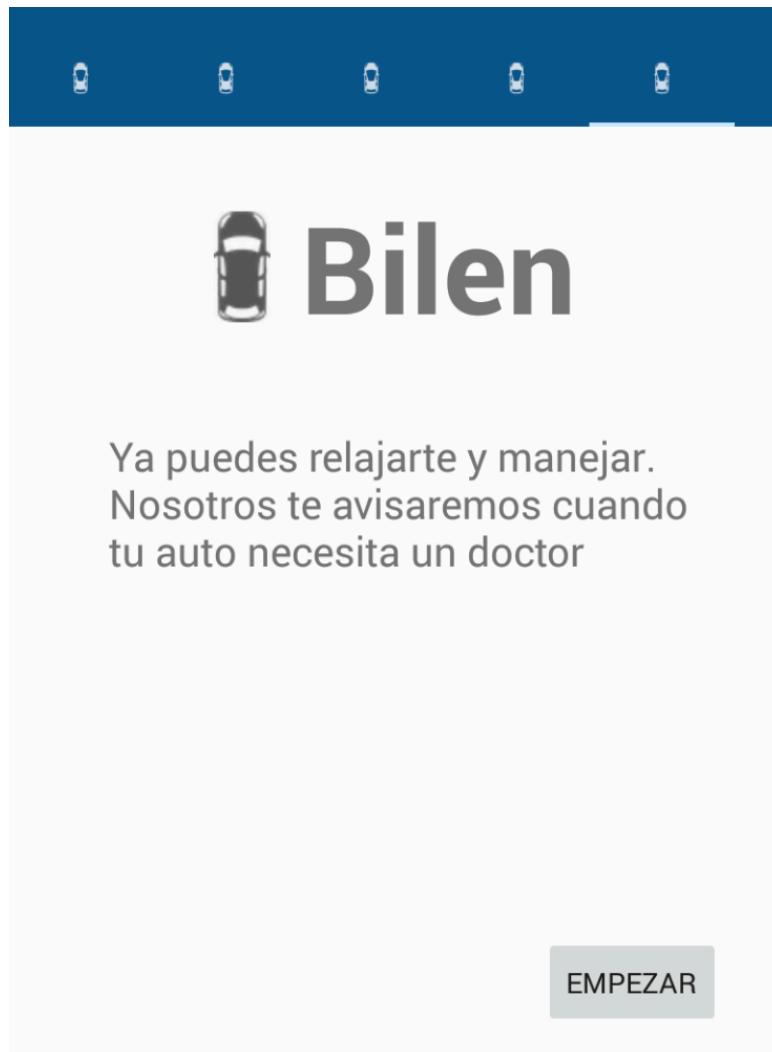
En el primero tiene que elegir la marca del vehículo al cual se va a conectar el hardware. Y el segundo campo el modelo del mismo. Estos datos sirven solo para guardar la información y mostrarlas en pantallas posteriores, no se utiliza para validar si OpenXC podrá conectarse.



La cuarta pantalla es donde el usuario va a poder decidir qué alertas quiere recibir. Se observa una lista con todas las reparaciones pre cargadas en la aplicación. El usuario tiene que ir marcando SI/NO de acuerdo a cual quiera y cual no. Estas alertas pueden volver a configurarse más adelante en una pantalla de la aplicación principal.



La quinta y última pantalla, está encargada de dar la finalización nomas del todo el proceso de configuración de la aplicación. Cuenta con un mensaje y un botón para poder acceder a la aplicación principal.



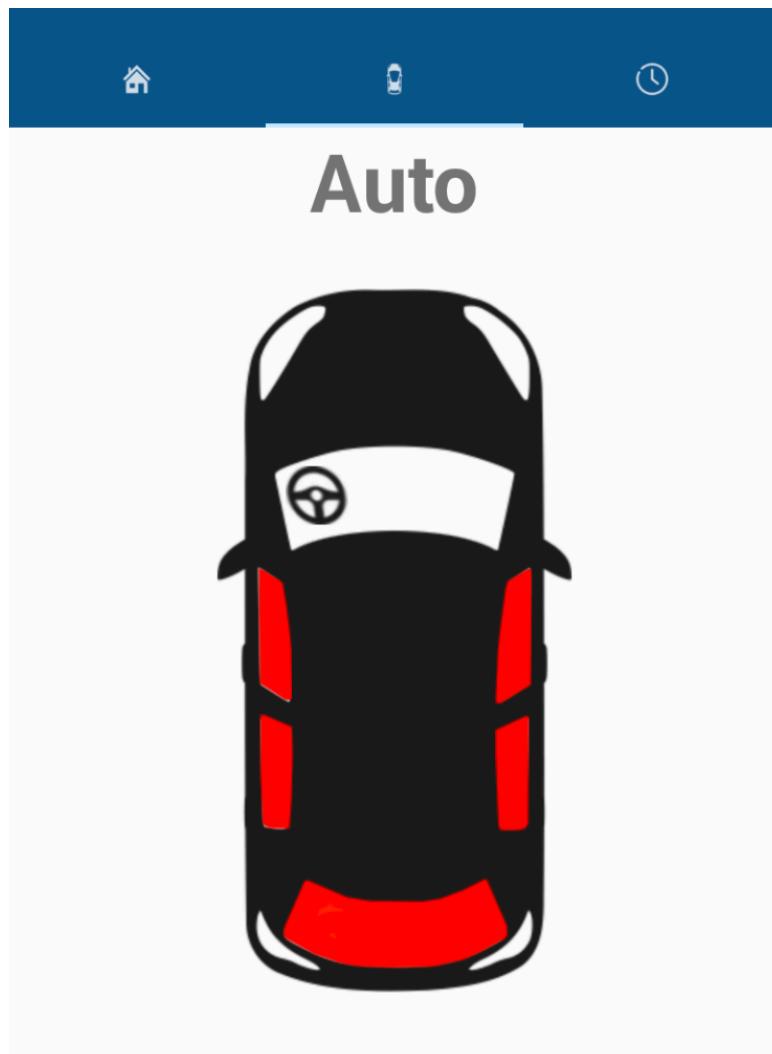
Una vez finalizado el proceso de configuración se accede a la aplicación principal.

La misma cuenta solo de 3 pantallas dispuestas en forma de pestañas para poder acceder rápidamente a cada una de ellas.

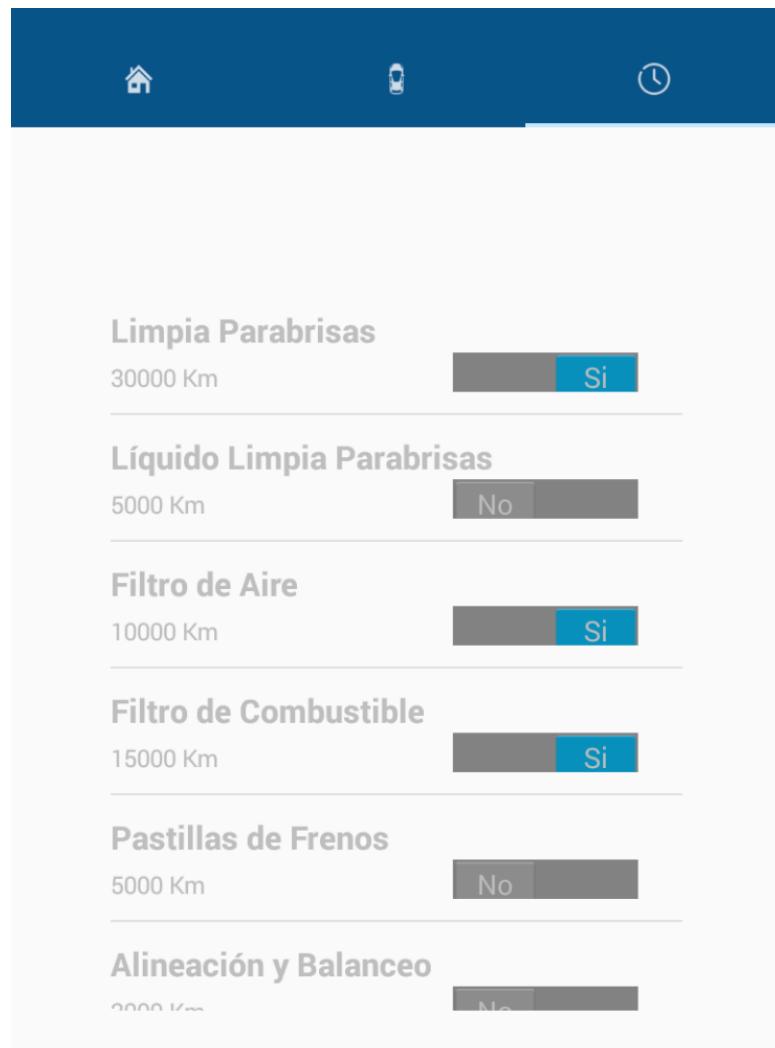
En la primera, llamada Home, se pueden ver los datos del vehículo, así como una lista de las siguientes reparaciones a realizar. En esta pantalla se marcan las reparaciones como completadas una vez que se hayan realizado.



En la segunda pantalla o Auto, se pueden observar datos en tiempo real de los sensores. Se escuchan constantemente 3 sensores: El estado de las puertas, la posición del volante y la velocidad actual. Se puede observar la figura de un auto donde si alguna de sus aberturas se encontrara abierta, se mostrará en la aplicación en color rojo.



La tercera y última pantalla de la aplicación, es la pantalla de Alertas, en ella se pueden activar y desactivar las alertas que se configuraron anteriormente en el proceso de configuración



### ***Cálculo del kilómetro de alerta***

El prototipo está pensado para que alerte al usuario, sin la necesidad de que tenga que configurar nada.

Se desarrolló una fórmula matemática que se encarga de calcular el próximo valor de kilómetro donde la alerta tendrá lugar.

La fórmula utiliza como constante el valor del intervalo en kilómetros de cada reparación y el odómetro actual al momento de leer la información como valor variable.

- Odómetro Actual(OA)
- Constante de Alerta(CA)
- Siguiente Cambio (SC) = OA + (CA - (OA % CA))

La fórmula suma al valor actual del odómetro (OA) lo que falta para el siguiente cambio. Esto último se calcula sacando el módulo de la división entre el OA y la Constante de Alerta(CA) y restándolo de la CA.

Ejemplo de uso de la fórmula para el cálculo de kilómetro de alerta

- OA = 1450
- CA = 1000
- SC = 1450 + (1000 - (1450 % 1000))
- SC = 1450 + (1000 - 450)
- SC = 1450 + 550
- SC = 2000

## Pruebas y Correcciones

Se realizaron pruebas de caja negra, a medida que se iba avanzando en el desarrollo. Las mismas consistían en hacer una recorrida completa por la aplicación cuando se realizaba un cambio muy grande, como por ejemplo agregar una nueva pantalla.

Las pruebas se realizaron tanto en emuladores de dispositivos Android como en dispositivos reales, siempre conectados por cable a través de la computadora, ya que no se generó un archivo APK, debido a que solo se trata de un prototipo.

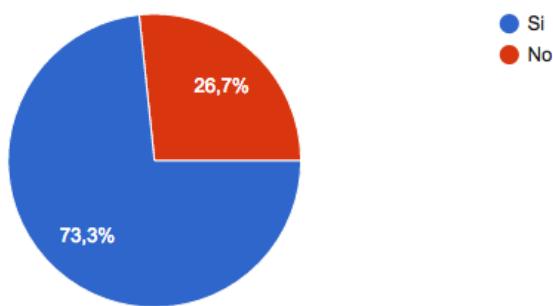
Una vez finalizado el desarrollo se procedió a realizar un video con el prototipo en funcionamiento, utilizando los emuladores de OpenXC. Junto a este video y un resumen de la investigación se llevó a cabo una encuesta online con una población de 75 individuos, los cuales tienen o tuvieron alguna vez acceso a un auto y/o a un teléfono celular con bluetooth.

El objetivo de la encuesta era poder observar el grado de aceptación que tendría una aplicación de este tipo en la sociedad para en un futuro poder desarrollar una aplicación para ponerla en producción.

La encuesta solo cuenta con tres simples preguntas:

La primera de ellas tenía el objetivo de saber qué cantidad de encuestados realmente tenían auto y si no lo poseían, no se podían contestar las dos siguientes, solo era útil para poder tener un poco de control sobre la población ya que la misma la podía realizar cualquiera que tenga acceso al link.

Como se puede observar en la imagen, solo 73,3% (55 Individuos) tienen un auto en su poder. El 26,7% restante ya no es válido para seguir contestando la encuesta ya que sus respuestas podrían llegar a ser simplemente una respuesta al azar.

**¿Tienes un auto? (75 respuestas)***Gráfico 12 - Encuesta a Usuarios: ¿Tienes auto?*

La segunda pregunta iba más referida a la misma que se le hizo al personal de los talleres mecánicos, pero de una forma más general. El objetivo de esta pregunta era poder corroborar que lo que dicen los mecánicos es semejante a lo que saben y/o piensan los conductores de vehículos. Según lo que se pudo investigar en diversos autores y con testimonios de los mecánicos un auto debe entrar al taller cada 10.000 Km aproximadamente si es un auto 0Km y cada 5.000 si es un auto con una edad mayor a los 5 años. De las 55 personas que contestaron esta pregunta el 75% contestó en el mismo rango que dicen los expertos. Esto comprueba una de las hipótesis que se tenía al principio de esta investigación. La forma de controlar el estado de un vehículo es según cuantos kilómetros recorrió desde la última vez que estuvo en el taller mecánico.

**¿Cada cuantos kilómetros lo llevas al service?** (55 respuestas)

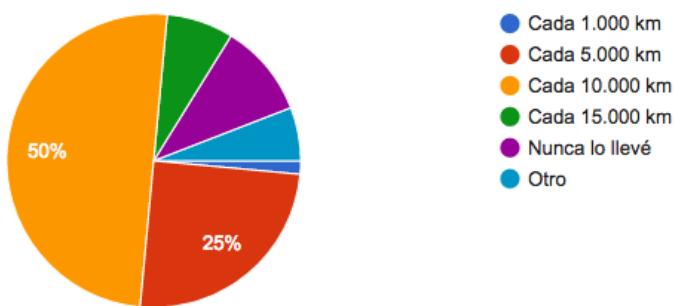


Gráfico 13 - Encuesta a Usuarios: ¿Cada cuántos kilómetros lo llevas al service?

La tercera y última pregunta no tuvo un resultado favorable. Se preguntaba si utilizarían alguna aplicación que les recordara cuando deben llevar el auto al taller, solo el 62,5% de los encuestados contestó que sí. Es un número bastante bajo sabiendo que el 100% de los que contestaron tenían un auto. Sin embargo, es útil para poder saber hacia donde seguir cuando se quiera llevar a producción este proyecto.

**¿Utilizarías una aplicación móvil que te avise cuando es tiempo de cambiar algo en tu auto?**

(55 respuestas)

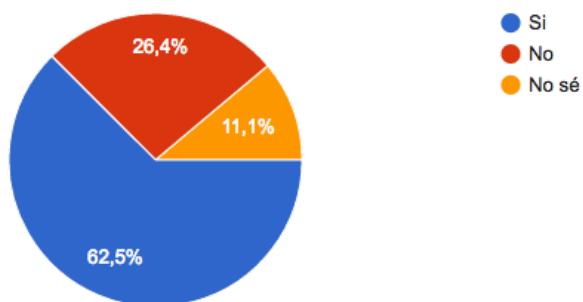


Gráfico 14 - Encuesta a Usuarios: ¿Utilizarías una aplicación móvil que te avise cuando es tiempo de cambiar algo en tu auto?

## Conclusión

Con el trabajo de campo realizado para recopilar información, se ha determinado que en nuestra zona, haciendo referencia a la zona Norte de Argentina y sur de Paraguay, existe una manera fácil de saber cuándo un vehículo debe ir al taller a realizar alguna reparación. Si bien no son datos exactos existe un consenso bastante homogéneo para poner un valor en kilómetros al desgaste de alguna parte de los automóviles. Esto no solo es un dato que lo saben los mecánicos, sino también los dueños de los autos, lo que hace que cualquiera con el mínimo de conocimientos pueda llevar el auto al taller.

Sin embargo, también se concluyó que hay gente que no conoce mucho sobre esta homogeneidad de datos y es a esta gente que una aplicación que le avise cuando deberían repararlo les sería muy útil.

Está clara la tendencia de ir día a día mejorando la tecnología de los autos. Sin embargo, la tecnología utilizada en esta investigación existe desde finales de los 80 y principios de los 90, por lo que se concluye que el grado de ignorancia en el tema es muy elevado. Los motivos de tal ignorancia pueden ser de los más variados, empezando por el principal, la falta de incentivo económico para realizar investigaciones sobre el tema hasta el simple hecho de creer que acá en la zona no existen autos tecnológicos que nos brinden todo tipo de información.

Se pudo diseñar y desarrollar un prototipo de software el cual mediante una conexión Bluetooth puede leer en tiempo real los datos que brinda el sistema OBD-II de un auto marca Ford, pero que solo se limitó a esta marca por el software y hardware que se utilizó para llevar a cabo la investigación, pero que con los cambios adecuados puede ser utilizable en todas las marcas de vehículos.

Para medir el grado de aceptación de una aplicación de este tipo en la sociedad futura se realizó una encuesta aplicada a un total de 75 personas, donde solo 55 fueron capaces de contestar la pregunta más importante y que hacía referencia precisamente a si les sería útil una aplicación de este tipo. Solo un poco más del 62% de los encuestados contestaron que sí. Al haber sido una encuesta cerrada, no se pudo determinar por qué el 48% restante no está interesado en un software de este tipo.

Personalmente concluyo que esta investigación logró su cometido al demostrar que existe mucha tecnología por ser explotada aún y quizás se piensa que no existe o que incluso ya está quedando en desuso sin que siquiera la hayamos conocido.

## Recomendaciones

Se recomienda como líneas futuras de investigación, llevar este prototipo a un nivel más alto y poder utilizarlo en cualquier auto que se rija bajo el estándar OBD-II, que hoy en día ya son casi el 100% de los vehículos fabricados después de 1996.

Sobre funcionalidades del software se podrían agregar:

- Geolocalización del vehículo en tiempo real utilizando el GPS que vienen incorporados en algunos autos.
- Enviar datos a la nube utilizando la conexión de internet del dispositivo móvil, para que un mecánico pueda evaluar el estado del auto sin necesidad de estar frente al mismo.
- Desarrollar la aplicación para dispositivos IOS.
- Alertar al usuario por fallas en el vehículo en tiempo real, dando una información detallada del problema.
- Guardar un historial en la nube de las veces que se llevó el auto a reparación.
- Hacer más configurable la aplicación para que el usuario diga el kilometraje exacto en el que hizo un cambio.

## Lista de Referencias

- About SQLite. (2016). Retrieved from <https://www.sqlite.org/about.html>
- Acerca de Android. (2015). Android. Retrieved from  
<https://developer.android.com/intl/es/about/versions/android-5.1.html>
- AeroCar. (1999). AeroCar. Retrieved from <http://www.aerocar.com/>
- Android. (2016). Saving Data in SQL Databases | Android Developers. Retrieved from <https://developer.android.com/training/basics/data-storage/databases.html>
- Arias, F. G. (2006). *El Proyecto de Investigación. Introducción a la metodología científica* (5th ed.). Venezuela.
- Audet, M. (2006). Antique and Vintage Cars Overview. Retrieved from [http://antiques.lovetoknow.com/Antique\\_Cars](http://antiques.lovetoknow.com/Antique_Cars)
- Auto-ID. (2016). Auto-ID Labs. Retrieved May 28, 2016, from  
[http://autoidlabs.org/wordpress\\_website/](http://autoidlabs.org/wordpress_website/)
- Baptista, L., Fernandez Collado, C., & Hernandez Zampieri, R. (1991).  
*Metodología de la investigación*. Mexico.
- Bellis, M. (2016). Automotive Airbag History and Invention. Retrieved from  
[http://inventors.about.com/od/astartinventions/a/air\\_bags.htm](http://inventors.about.com/od/astartinventions/a/air_bags.htm)
- biografiasyvidas.com Staff. (2004). Charles Franklin Kettering. Retrieved from  
<http://www.biografiasyvidas.com/biografia/k/kettering.htm>
- Bisson, T. (1997). *The Fifth Element: A Novel*. (Harpercollins (Mm), Ed.) (1997th

- ed.). Retrieved from <https://www.amazon.es/Fifth-Element-Novel-Terry-Bisson/dp/0061058386>
- California Environmental Protection Agency. (2016). On-Board Diagnostics Program. Retrieved from  
<https://www.arb.ca.gov/msprog/obdprog/obdprog.htm>
- Conference, I. (2016). IoT 2016 - Internet of Things 2016. Retrieved May 28, 2016, from <http://www.iot-conference.org/iot2016/>
- Developers. (2008). OpenXC Ford Vehicle Compatibility - Hojas de Cálculo de Google. Retrieved from <https://docs.google.com/spreadsheets/d/1hOBl9tFwR1KRFXfeaHTAddwJuSGx5Ir1ET4N2zWAiE/edit#gid=0>
- dieselnet.com. (2010). Emission Standards: USA: On-Board Diagnostics (California). Retrieved from  
[https://www.dieselnet.com/standards/us/obd\\_ca.php](https://www.dieselnet.com/standards/us/obd_ca.php)
- ELMElectronics. (2005). OBD – Elm Electronics. Retrieved from  
<https://www.elmelectronics.com/products/ics/obd/>
- Emad-ul-Haq, Q., Aboalsamh, H., Belghith, A., Hussain, M., Abdul, W., Dahshan, M. H., & Ghouzali, S. (2015). Challenges and solutions for internet of things driven by IPv6. *KSII Transactions on Internet and Information Systems*, 9(12), 4739–4758. <http://doi.org/10.3837/tiis.2015.12.001>
- Engineers, S. of A. (2001). SAE J1962: Diagnostic Connector Equivalent to

- ISO/DIS 15031, (1), 1806–5. Retrieved from  
<https://law.resource.org/pub/us/cfr/ibr/005/sae.j1962.2002.pdf>
- EPA—APPROVED IOWA REGULATIONS Iowa citation Title State effective date EPA approval date Explanation Iowa Department of Natural Resources Environmental Protection Commission [567] Chapter 20—Scope of Title—Definitions—Forms—Rules of Practice The definit. (2005). *Federal Register*, 70(243). Retrieved from  
<http://www.stat.gov.az/source/environment/indexen.php>
- EU. (1998). Directive 98/69/EC of the European Parliament and of the Council, 31(October 1998), 1–127.
- EZVID. (2015). Top Ten OBD2 Scanners 2015 | Best OBD2 Car Scanners Bluetooth. Retrieved from <https://www.ezvid.com/top-ten-obd2-scanners>
- Gardner, C. S. (1989). *Back to the Future Part II*. USA: Berkley; Mt edition (November 1, 1989).
- Gandan.co.uk. (n.d.). What is EOBD, EOBD2 and OBDII? :: Gandan Automotive Products. Retrieved from [http://www.gandan.co.uk/article\\_5.html](http://www.gandan.co.uk/article_5.html)
- Gimson, L. L. (2012). *Metodologías ágiles y Desarrollo basado en conocimiento*. (Quinta). México.
- Google. (n.d.). Google Self-Driving Car Project. Retrieved from  
<https://www.google.com/selfdrivingcar/>

Google. (2008). Protocol Buffers - Google's data interchange format.

GRAJALE G, T. (2000). Tipos de Investigacion. *IUPuebla*, 4.

<http://doi.org/10.1590 / S0124-00642009000200014>

History.com Staff. (2010). Model T. Retrieved from

<http://www.history.com/topics/model-t>

IES Castillo de la Yedra. (n.d.). HISTORIA DEL AUTOMÓVIL Etapa de la invención. Retrieved from [http://tecnologia-lcp.orgfree.com/Documentos/Historia del automovil.pdf](http://tecnologia-lcp.orgfree.com/Documentos/Historia%20del%20automovil.pdf)

Json.org. (n.d.). JSON. Retrieved from <http://www.json.org/>

JUSTIN BERKOWITZ. (2010). The History of Car Radios - Feature - Car and Driver. Retrieved from <http://www.caranddriver.com/features/the-history-of-car-radios>

Laukkonen, J. (2016). From Car Cigarette Lighter To 12V Socket. Retrieved from <http://cartech.about.com/od/Power/a/12v-Socket-Or-Cigarette-Lighter-Receptacle.htm>

Martinez, J. (2012). Etapa de la Invención. Retrieved from

<http://eliniciodeloscoches.blogspot.com.ar/2012/11/etapa-de-la-invencion.html>

mistertransmission.com. (2000). Transmission History Basics | Mister Transmission. Retrieved from <https://www.mistertransmission.com/a-brief-history-of-the-automatic-transmission>

- O'Brien, J. (2015). Tech Time Machine: Car Tech. Retrieved June 26, 2016, from  
<http://mashable.com/2015/01/05/car-tech-ces/#ifXgnuTPakqU>
- OpenXC, P. (2008). Supported Vehicles - OpenXC. Retrieved from  
<http://openxcplatform.com/hardware/vehicles.html>
- Palin, R., Johnston, V., Johnson, S., D'Hooge, A., Duncan, B., & Gargoloff, J. I. (2012). The Aerodynamic Development of the Tesla Model S - Part 1: Overview. *SAE International*, 1, 177. <http://doi.org/10.4271/2012-01-0177>
- Pardo, M. (2010). Activeandroid by pardom. Retrieved from  
<http://www.activeandroid.com/>
- Patrascu, D. (2010). POWER STEERING. Retrieved from  
<http://www.autoevolution.com/news/history-of-the-steering-wheel-20109.html>
- Raymond, E. S. (2003). *The art of unix programming. System.*  
<http://doi.org/10.1002/chin.200327184>
- Rouse, M. (2014). What is Internet of Things (IoT)? - Definition from WhatIs.com. Retrieved May 28, 2016, from  
<http://internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT>
- Sarkfun, T. (n.d.). Getting Started with OBD-II - learn.sparkfun.com. Retrieved from <https://learn.sparkfun.com/tutorials/getting-started-with-obd-ii>
- Second Chance Garage, L. (n.d.). Cushioning the Blow: History of Automotive Airbags. Retrieved from <http://www.secondchancegarage.com/public/history-airbags>

of-airbags.cfm

Songdo, M. T. B. in. (2015). Songdo, South Korea: the world's first smart city - in pictures. *The Guardian*. Retrieved from <http://www.theguardian.com/cities/2014/dec/22/songdo-south-korea-world-first-smart-city-in-pictures%5Cnfiles/576/songdo-south-korea-world-first-smart-city-in-pictures.html>

Stachel, J. R., Sejdic, E., Ogirala, A., & Mickle, M. H. (2013). The impact of the internet of Things on implanted medical devices including pacemakers, and ICDs. In *Conference Record - IEEE Instrumentation and Measurement Technology Conference* (pp. 839–844).

<http://doi.org/10.1109/I2MTC.2013.6555533>

Sundmaeker, H., Guillemin, P., & Friess, P. (2010). *Vision and challenges for realising the Internet of Things. ... the Internet of Things.*

<http://doi.org/10.2759/26127>

Technology, N. I. of A. (1998). *OBD II and Second Generation Scan Tools*. NAPA Institute of Automotive Technology.

utilsobdfacile.com. (n.d.). OBD Modes and configuration (PID) - Outils OBD Facile. Retrieved from <http://www.utilsobdfacile.com/obd-mode-pid.php>

ws6transam.org. (n.d.). ALDL revealed. Retrieved from <http://www.ws6transam.org/ALDL.html>

## Anexos

### Anexo 1: Enlaces a los proyectos OpenXC.

**VI Firmware:** <https://github.com/openxc/vi-firmware>

**Librería Python:** <https://github.com/openxc/openxc-python>

**Librería Android y Enabler:** <https://github.com/openxc/openxc-android>

**Librería IOS:** <https://github.com/openxc/openxc-ios-framework>

**Formato de Mensajes:** <https://github.com/openxc/openxc-message-format>

## Anexo 2

A continuación, se muestran las tablas de parámetros que se pueden enviar al OBD-II dependiendo del modo, se muestran en inglés la mayoría de las entradas debido a que es la forma en que se conocen cada uno de estos parámetros.

### Parameters ID para modo \$01 y \$02.

PID	No.	Descripción
00	00	Lista de Parámetros aceptados
01	01	Estado antes de la última limpieza
02	02	Código de Error de "freeze frame"
03	03	Estado del Sistema de Combustible
04	04	Porcentage de Carga del motor
05	05	Temperatura del Motor en grados centígrados
06	06	Short-term fuel % trim bank 1
07	07	Long-term fuel % trim bank 1
08	08	Short-term fuel % trim bank 2
09	09	Long-term fuel % trim bank 2
0A	10	Fuel pressure in kPa
0B	11	Intake manifold absolute pressure in kPa
0C	12	Engine speed in rpm
0D	13	Vehicle speed in kph
0E	14	Timing advance on cylinder 1 in degrees
0F	15	Intake air temperature in °C
10	16	Air flow measured by the flowmeter in g/s
11	17	Throttle position in %

12	18	Status of the secondary intake circuit
13	19	O2 sensor positions bank/sensor
14	20	Oxygen sensor volts bank 1 sensor 1/td>
15	21	Oxygen sensor volts bank 1 sensor 2
16	22	Oxygen sensor volts bank 1 sensor 3
17	23	Oxygen sensor volts bank 1 sensor 4
18	24	Oxygen sensor volts bank 2 sensor 1
19	25	Oxygen sensor volts bank 2 sensor 2
1A	26	Oxygen sensor volts bank 2 sensor 3
1B	27	Oxygen sensor volts bank 2 sensor 4
1C	28	OBD computer specification
1D	29	O2 sensor positions bank/sensor
1E	30	Auxiliary input status
1F	31	Run time since engine start
20	32	List of PIDs supported (range 33 to 64)
21	33	Distance travelled with MIL on in kms
22	34	Relative fuel rail pressure in kPa
23	35	Fuel rail pressure in kPa
24	36	O2 sensor (extended range) bank 1, sensor 1 (lambda and volts)
25	37	O2 sensor (extended range) bank 1, sensor 2 (lambda and volts)
26	38	O2 sensor (extended range) bank 1, sensor 3 (lambda and volts)
27	39	O2 sensor (extended range) bank 1, sensor 4 (lambda and volts)
28	40	O2 sensor (extended range) bank 2, sensor 1 (lambda and volts)
29	41	O2 sensor (extended range) bank 2, sensor 2 (lambda and volts)
2A	42	O2 sensor (extended range) bank 2, sensor 3 (lambda and volts)
2B	43	O2 sensor (extended range) bank 2, sensor 4 (lambda and volts)

2C	44	EGR in %
2D	45	EGR error in %
2E	46	Evaporation purge in %
2F	47	Fuel level in %
30	48	Number of warning(s) since faults (DTC) were cleared
31	49	Distance since faults (DTC) were cleared.
32	50	Evaporation system vapour pressure in Pa
33	51	Barometric pressure in kPa
34	52	O2 sensor (extended range) bank 1, sensor 1 (lambda and volts)
35	53	O2 sensor (extended range) bank 1, sensor 2 (lambda and volts)
36	54	O2 sensor (extended range) bank 1, sensor 3 (lambda and volts)
37	55	O2 sensor (extended range) bank 1, sensor 4 (lambda and volts)
38	56	O2 sensor (extended range) bank 2, sensor 1 (lambda and volts)
39	57	O2 sensor (extended range) bank 2, sensor 2 (lambda and volts)
3A	58	O2 sensor (extended range) bank 2, sensor 3 (lambda and volts)
3B	59	O2 sensor (extended range) bank 2, sensor 4 (lambda and volts)
3C	60	Catalyst temperature in °C bank 1, sensor 1
3D	61	Catalyst temperature in °C bank 2, sensor 1
3E	62	Catalyst temperature in °C bank 1, sensor 2
3F	63	Catalyst temperature in °C bank 2, sensor 1
40	64	List of PIDs supported (range 65 to 96)
41	65	Monitor status this drive cycle
42	66	Control module voltage in V
43	67	Absolute engine load
44	68	Equivalent fuel/air mixture request
45	69	Relative throttle position in %

46	70	Ambient air temperature in °C
47	71	Absolute throttle position B in %
48	72	Absolute throttle position C in %
49	73	Accelerator pedal position D in %
4A	74	Accelerator pedal position E in %
4B	75	Accelerator pedal position F in %
4C	76	Commanded throttle actuator in %
4D	77	Engine run time since MIL on in min
4E	78	Engine run time since faults cleared in min
4F	79	External test equipment no. 1 configuration information
50	80	External test equipment no. 2 configuration information
51	81	Fuel type used by the vehicle
52	82	Ethanol fuel %
53	83	Absolute evaporation system vapour pressure in kPa
54	84	Evaporation system vapour pressure in Pa
55	85	Short-term O2 sensor trim bank 1 and 3
56	86	Long-term O2 sensor trim bank 1 and 3
57	87	Short-term O2 sensor trim bank 2 and 4
58	88	Long-term O2 sensor trim bank 2 and 4
59	89	Absolute fuel rail pressure in kPa
5A	90	Relative accelerator pedal position in %
5B	91	Battery unit remaining life (hybrid) in %
5C	92	Engine oil temperature in °C
5D	93	Fuel injection timing in °
5E	94	Fuel consumption in litre/hr
5F	95	Fuel consumption in litre/hr

60	96	List of PIDs supported (range 97 to 128)
61	97	Driver demand: torque percentage (%)
62	98	Final engine torque percentage (%)
63	99	Engine torque reference in Nm
64	100	Engine torque data in %
65	101	Auxiliary inputs / outputs
66	102	Flowmeter sensor
67	103	Engine water temperature in °C
68	104	Air temperature sensor in °C
69	105	Commanded EGR and EGR error
6A	106	Commanded Diesel intake air flow control and relative intake air flow position
6B	107	Recirculation gas temperature in °C
6C	108	Commanded throttle actuator control and relative throttle position
6D	109	Fuel pressure control system
6E	110	Injection pressure control system
6F	111	Turbocharger compressor inlet pressure in kPa
70	112	Boost pressure control in kPa
71	113	Variable Geometry turbo (VGT) control
72	114	Wastegate control
73	115	Exhaust pressure in kPa
74	116	Turbocharger RPM
75	117	Turbocharger A temperature in °C
76	118	Turbocharger B temperature in °C
77	119	Charge air cooler temperature in °C
78	120	Exhaust Gas temperature (EGT) Bank 1
79	121	Exhaust Gas temperature (EGT) Bank 2

7A	122	Diesel particulate filter (DPF) bank 1
7B	123	Diesel particulate filter (DPF) bank 2
7C	124	Diesel Particulate filter (DPF) temperature
7D	125	NOx NTE control area status
7E	126	PM NTE control area status
7F	127	Engine run time
80	128	List of PIDs supported (range 129 to 160)

### Parameters ID para modo \$05

PID	No.	Description
00	00	List of PIDs supported (range 01 to 32)
01	01	Rich to lean sensor threshold voltage
02	02	Lean to rich sensor threshold voltage
03	03	Low voltage used to calculated passage time
04	04	High voltage used to calculated passage time
05	05	Rich to lean calculated passage time
06	06	Lean to rich calculated passage time
07	07	Minimum sensor voltage during test cycle
08	08	Maximum sensor voltage during test cycle
09	09	Time between sensor transitions
0A	10	Sensor period
0B	11	Reserved for future use

**Parameters ID para mode \$06**

PID	No.	Description
0	0	List of PIDs supported (range 01h to 20h)
1	1	Exhaust gas sensor bank 1 - sensor 1
2	2	Exhaust gas sensor bank 1 - sensor 2
3	3	Exhaust gas sensor bank 1 - sensor 3
4	4	Exhaust gas sensor bank 1 - sensor 4
5	5	Exhaust gas sensor bank 2 - sensor 1
6	6	Exhaust gas sensor bank 2 - sensor 2
7	7	Exhaust gas sensor bank 2 - sensor 3
8	8	Exhaust gas sensor bank 2 - sensor 4
9	9	Exhaust gas sensor bank 3 - sensor 1
0A	10	Exhaust gas sensor bank 3 - sensor 2
0B	11	Exhaust gas sensor bank 3 - sensor 3
0C	12	Exhaust gas sensor bank 3 - sensor 4
0D	13	Exhaust gas sensor bank 4 - sensor 1
0E	14	Exhaust gas sensor bank 4 - sensor 2
0F	15	Exhaust gas sensor bank 4 - sensor 3
10	16	Exhaust gas sensor bank 4 - sensor 4
20	32	List of PIDs supported (range 21h to 40h)
21	33	Catalytic bank 1
22	34	Catalytic bank 2
23	35	Catalytic bank 3
24	36	Catalytic bank 4
31	49	EGR bank 1
32	50	EGR bank 2

33	51	EGR bank 3
34	52	EGR bank 4
35	53	VVT bank 1
36	54	VVT bank 2
37	55	VVT bank 3
38	56	VVT bank 4
39	57	EVAP (Cap Off / 0.150")
3A	58	EVAP (0.090")
3B	59	EVAP (0.040")
3C	60	EVAP (0.020")
3D	61	Flux de purge
40	64	List of PIDs supported (range 41h to 60h)
41	65	Heated exhaust gas sensor bank 1 - sensor 1
42	66	Heated exhaust gas sensor bank 1 - sensor 2
43	67	Heated exhaust gas sensor bank 1 - sensor 3
44	68	Heated exhaust gas sensor bank 1 - sensor 4
45	69	Heated exhaust gas sensor bank 2 - sensor 1
46	70	Heated exhaust gas sensor bank 2 - sensor 2
47	71	Heated exhaust gas sensor bank 2 - sensor 3
48	72	Heated exhaust gas sensor bank 2 - sensor 4
49	73	Heated exhaust gas sensor bank 3 - sensor 1
4A	74	Heated exhaust gas sensor bank 3 - sensor 2
4B	75	Heated exhaust gas sensor bank 3 - sensor 3
4C	76	Heated exhaust gas sensor bank 3 - sensor 4
4D	77	Heated exhaust gas sensor bank 4 - sensor 1
4E	78	Heated exhaust gas sensor bank 4 - sensor 2

4F	79	Heated exhaust gas sensor bank 4 - sensor 3
50	80	Heated exhaust gas sensor bank 4 - sensor 4
60	96	List of PIDs supported (range 61h to 80h)
61	97	Heated catalyst bank 1
62	98	Heated catalyst bank 2
63	99	Heated catalyst bank 3
64	100	Heated catalyst bank 4
71	113	Secondary air 1
72	114	Secondary air 2
73	115	Secondary air 3
74	116	Secondary air 4
80	128	List of PIDs supported (range 81h to A0h)
81	129	Fuel system bank 1
82	130	Fuel system bank 2
83	131	Fuel system bank 3
84	132	Fuel system bank 4
85	133	Turbo pressure bank 1
86	134	Turbo pressure bank 2
90	144	Nox Absorber bank 1
91	145	Nox Absorber bank 2
98	152	Catalytic NOx bank 1
99	153	Catalytic NOx bank 2
A0	160	List of PIDs supported (range A1h to C0h)
A1	161	General misfire monitoring data
A2	162	Misfire data cylinder 1
A3	163	Misfire data cylinder 2

A4	164	Misfire data cylinder 3
A5	165	Misfire data cylinder 4
A6	166	Misfire data cylinder 5
A7	167	Misfire data cylinder 6
A8	168	Misfire data cylinder 7
A9	169	Misfire data cylinder 8
AA	170	Misfire data cylinder 9
AB	171	Misfire data cylinder 10
AC	172	Misfire data cylinder 11
AD	173	Misfire data cylinder 12
AE	174	Misfire data cylinder 13
AF	175	Misfire data cylinder 14
B0	176	Misfire data cylinder 15
B1	177	Misfire data cylinder 16
B2	178	PM filter bank 1
B3	179	PM filter bank 2
C0	192	List of PIDs supported (range C1h to E0h)
E0	224	List of PIDs supported (range E1h to FFh)
E1-FF	225-255	Manufacturer specific

### Parameters ID para modo \$09

PID	No.	Description
00	00	List of PIDs supported (range 01h to 20h)
01	01	VIN message count
02	02	VIN (vehicle identification number)
03	03	Calibration ID message count

04	04	Calibration IDs
05	05	CALIB verification numbers message count
06	06	Calibration verification number
07	07	IPT message count
08	08	In-use performance tracking (IPT)
09	09	ECU name message count
0A	10	ECU name
0B	11	In-use performance tracking
0C-FF	12-255	ISO/SAE reserved

### Anexo 3: Datos de Pruebas para Emular el Hardware de OpenXC

```

1 {"timestamp": 1351181576.64078, "name": "engine_speed", "value": 714.0}
2 {"timestamp": 1351181576.64578, "name": "vehicle_speed", "value": 0.0}
3 {"timestamp": 1351181576.6507802, "name": "brake_pedal_status", "value": true}
4 {"timestamp": 1351181576.6557803, "name": "latitude", "value": 42.291595}
5 {"timestamp": 1351181576.6607804, "name": "longitude", "value": -83.237617}
6 {"timestamp": 1351181576.6657805, "name": "fuel_consumed_since_restart", "value": 0.07345}
7 {"timestamp": 1351181576.6707807, "name": "steering_wheel_angle", "value": 8.0868}
8 {"timestamp": 1351181576.6757808, "name": "fuel_consumed_since_restart", "value": 0.073475}
9 {"timestamp": 1351181576.680781, "name": "steering_wheel_angle", "value": 8.0868}
10 {"timestamp": 1351181576.685781, "name": "fine_odometer_since_restart", "value": 0.0}
11 {"timestamp": 1351181576.690781, "name": "steering_wheel_angle", "value": 11.1633}
12 {"timestamp": 1351181576.6957812, "name": "steering_wheel_angle", "value": 11.1633}
13 {"timestamp": 1351181576.7007813, "name": "parking_brake_status", "value": false}
14 {"timestamp": 1351181576.7057815, "name": "headlamp_status", "value": false}
15 {"timestamp": 1351181576.7107816, "name": "windshield_wiper_status", "value": false}
16 {"timestamp": 1351181576.7157817, "name": "steering_wheel_angle", "value": 11.1633}
17 {"timestamp": 1351181576.7207818, "name": "door_status", "value": "driver", "event": false}
18 {"timestamp": 1351181576.725782, "name": "door_status", "value": "passenger", "event": true}
19 {"timestamp": 1351181576.730782, "name": "door_status", "value": "rear_left", "event": false}
20 {"timestamp": 1351181576.7357821, "name": "door_status", "value": "rear_right", "event": false}
21 {"timestamp": 1351181576.7357821, "name": "door_status", "value": "boot", "event": false}
22 {"timestamp": 1351181576.7407823, "name": "odometer", "value": 0.00}
23 {"timestamp": 1351181576.7457824, "name": "high_beam_status", "value": false}
24 {"timestamp": 1351181576.7507825, "name": "steering_wheel_angle", "value": 11.1633}
25 {"timestamp": 1351181576.7557826, "name": "parking_brake_status", "value": false}
26 {"timestamp": 1351181576.7607827, "name": "headlamp_status", "value": false}
27 {"timestamp": 1351181576.7657828, "name": "steering_wheel_angle", "value": 11.1633}
28 {"timestamp": 1351181576.770783, "name": "fine_odometer_since_restart", "value": 0.066071}
29 {"timestamp": 1351181576.775783, "name": "steering_wheel_angle", "value": 11.1633}
30 {"timestamp": 1351181576.7807832, "name": "parking_brake_status", "value": false}
31 {"timestamp": 1351181576.7857833, "name": "headlamp_status", "value": false}
32 {"timestamp": 1351181576.7907834, "name": "steering_wheel_angle", "value": 11.1633}
33 {"timestamp": 1351181576.7957835, "name": "steering_wheel_angle", "value": 11.1633}
34 {"timestamp": 1351181576.8007836, "name": "fine_odometer_since_restart", "value": 0.066071}
35 {"timestamp": 1351181576.8057837, "name": "parking_brake_status", "value": false}
36 {"timestamp": 1351181576.8107839, "name": "headlamp_status", "value": false}
37 {"timestamp": 1351181576.815784, "name": "steering_wheel_angle", "value": 11.1633}
38 {"timestamp": 1351181576.820784, "name": "steering_wheel_angle", "value": 11.1633}
39 {"timestamp": 1351181576.8257842, "name": "parking_brake_status", "value": false}
40 {"timestamp": 1351181576.8307843, "name": "headlamp_status", "value": false}
41 {"timestamp": 1351181576.8357844, "name": "parking_brake_status", "value": false}
42 {"timestamp": 1351181576.8407845, "name": "headlamp_status", "value": false}
43 {"timestamp": 1351181576.8457847, "name": "steering_wheel_angle", "value": 11.1633}
44 {"timestamp": 1351181576.8507848, "name": "parking_brake_status", "value": false}
45 {"timestamp": 1351181576.855785, "name": "headlamp_status", "value": false}
46 {"timestamp": 1351181576.860785, "name": "steering_wheel_angle", "value": 11.1633}
47 {"timestamp": 1351181576.8657851, "name": "fine_odometer_since_restart", "value": 0.066071}
48 {"timestamp": 1351181576.8707852, "name": "steering_wheel_angle", "value": 11.1633}
49 {"timestamp": 1351181576.8757854, "name": "parking_brake_status", "value": false}
50 {"timestamp": 1351181576.8807855, "name": "headlamp_status", "value": false}
51 {"timestamp": 1351181576.8857856, "name": "steering_wheel_angle", "value": 11.1633}
52 {"timestamp": 1351181576.8907857, "name": "steering_wheel_angle", "value": 11.1633}
53 {"timestamp": 1351181576.8957858, "name": "fine_odometer_since_restart", "value": 0.066071}
54 {"timestamp": 1351181576.900786, "name": "parking_brake_status", "value": false}

```

**Anexo 4: Vehículos donde se realizaron pruebas de OpenXC por la comunidad de desarrolladores.**

A continuación, se detalla la lista de vehículos soportados según (OpenXC, 2008). Estos datos se encuentran publicados en una planilla de Spreesheet (Developers, 2008), que es mantenida por la comunidad de desarrolladores.

	<b>Año</b>	<b>Ángulo del Volante</b>	<b>Velocidad de Motor</b>	<b>Posición de transmisión</b>
C-Max Hybrid	2013	CAN1	CAN1	no
C-Max Energi	2013	CAN1	CAN1	no
E350	2011	no	CAN1	CAN1
E350	2012	no	CAN1	CAN1
Escape	2013	CAN1	CAN1	CAN1
Explorer	2012	CAN1	CAN1	CAN1
Explorer	2013	CAN1	CAN1	CAN1
F-150	2011	CAN1	CAN1	CAN1
Falcon	2013	CAN1	CAN1	CAN1
Figo	2010	no	CAN1	CAN1, solo caja automática
Focus	2009	no	CAN1	CAN1, solo caja automática
Focus	2012	CAN1	CAN1	CAN1
Focus	2013	CAN1	CAN1	CAN1
Focus Electric	2013	CAN1	CAN1	CAN1
Focus ST	2013	CAN1	CAN1	CAN1
Fiesta	2012	CAN1	CAN1	CAN1
Fusion	2006	no	CAN1	CAN1
Fusion	2013	CAN1	CAN1	CAN1
Mustang	2005	no	CAN1	CAN1

Mustang	2012	CAN1	CAN1	CAN1
Taurus	2013	CAN1	CAN1	CAN1
Territory	2013	CAN1	CAN1	CAN1
MKZ	2013	CAN1	CAN1	CAN1
Police Interceptor	2004	no	no	no

	Año	Posición de caja de cambios	Estado de Ignición	Estado Pedal de Freno	Estado de Estacionamiento
C-Max Hybrid	2013	CAN1	CAN1	CAN1	CAN1
C-Max Energi	2013	CAN1	CAN1	CAN1	CAN1
E350	2011		CAN1	CAN1	CAN1
E350	2012		CAN1	CAN1	CAN1
Escape	2013	CAN1	CAN1	CAN1	CAN1
Explorer	2012		CAN1	CAN1	CAN1
Explorer	2013		CAN1	CAN1	CAN1
F-150	2011		CAN1	CAN1	CAN1
Falcon	2013	CAN1	CAN1	CAN1	CAN1
Figo	2010		CAN2-1	CAN1	CAN1
Focus	2009		CAN1	CAN1	CAN1
Focus	2012	CAN1	CAN1	CAN1	CAN1
Focus	2013	CAN1	CAN1	CAN1	CAN1
Focus Electric	2013	CAN1	CAN1	CAN1	CAN1
Focus ST	2013	CAN1	CAN1	CAN1	CAN1

Fiesta	2012		CAN1	CAN1	CAN1
Fusion	2006		CAN1	CAN1	CAN1
Fusion	2013		CAN1	CAN1	no
Mustang	2005		CAN1	no	CAN1
Mustang	2012		CAN1	CAN1	CAN1
Taurus	2013		CAN1	CAN1	CAN1
Territory	2013	CAN1	CAN1	CAN1	CAN1
MKZ	2013		CAN1	CAN1	no
Police					
Interceptor	2004	no	no	no	no

	Año	Posición Pedal del Acelerador	Transmisión, Torque	Velocidad
C-Max Hybrid	2013	CAN1	CAN1	CAN1
C-Max Energi	2013	CAN1	CAN1	CAN1
E350	2011	CAN1	CAN1	CAN1
E350	2012	CAN1	CAN1	CAN1
Escape	2013	CAN1	CAN1	CAN1
Explorer	2012	CAN1	no	CAN1
Explorer	2013	CAN1	CAN1	CAN1
F-150	2011	CAN1	CAN1	CAN1
Falcon	2013	CAN1		CAN1
Figo	2010	CAN1	no	CAN1
Focus	2009	CAN1	CAN1	CAN1

Focus	2012	CAN1	CAN1	CAN1
Focus	2013	CAN1	CAN1	CAN1
Focus Electric	2013	CAN1	CAN1	CAN1
Focus ST	2013	CAN1	CAN1	CAN1
Fiesta	2012	CAN1	CAN1	CAN1
Fusion	2006	CAN1	CAN1	CAN1
Fusion	2013	CAN1	CAN1	CAN1
Mustang	2005	CAN1	CAN1	CAN1
Mustang	2012	CAN1	CAN1	CAN1
Taurus	2013	CAN1	CAN1	CAN1
Territory	2013	CAN1		CAN1
MKZ	2013	CAN1	CAN1	CAN1
Police Interceptor	2004	no	no	no

	Año	Combustible consumido desde el reinicio	Estado de puertas	Estado Limpiaparabrisas	Odómetro
C-Max Hybrid	2013	CAN1	CAN1	CAN2-1	CAN2-1
C-Max Energi	2013	CAN1	CAN1	CAN2-1	CAN2-1
E350	2011	CAN1	CAN2-1	CAN1	CAN1, comienza en 0 en cada reinicio
E350	2012	CAN1	CAN2-1	CAN1	CAN1, , comienza en 0 en cada reinicio
Escape	2013	CAN1	CAN1	CAN1	CAN1
Explorer	2012	CAN1	CAN1	no	CAN1
Explorer	2013	CAN1	CAN1	no	CAN1
F-150	2011	CAN1	CAN2-1	CAN2-1	CAN1, , comienza en 0 en cada reinicio

Falcon	2013	CAN1	CAN1	CAN2	CAN1, , comienza en 0 en cada reinicio
Figo	2010	CAN1		CAN1	CAN1, , comienza en 0 en cada reinicio
Focus	2009	CAN1	CAN2-1	no	CAN1, , comienza en 0 en cada reinicio
Focus	2012	CAN1	CAN1	CAN1	CAN1
Focus	2013	CAN1	CAN1	CAN1	CAN1
Focus Electric	2013	CAN1	CAN1	CAN2-1	CAN2-1
Focus ST	2013	CAN1			CAN2-1
Fiesta	2012	CAN1	CAN1	CAN1	CAN1
Fusion	2006	CAN1	CAN2-1	no	CAN1
Fusion	2013	CAN1	CAN1	CAN1	CAN1
Mustang	2005	CAN1	CAN2-1	no	CAN1, , comienza en 0 en cada reinicio
Mustang	2012	CAN1	CAN2-1	CAN2-1	CAN1
Taurus	2013	CAN1	CAN1	no	CAN1
Territory	2013	CAN1	CAN1	CAN2	CAN1, , comienza en 0 en cada reinicio
MKZ	2013	CAN1	CAN1	CAN1	CAN1
Police					
Interceptor	2004	no	no	no	no

	Año	Estado de Luz	Nivel de Combustible	Latitud	Longitud	Botón de evento
C-Max Hybrid	2013	CAN2-1	CAN2-1	CAN1	CAN1	no
C-Max Energi	2013	CAN2-1	CAN2-1	CAN1	CAN1	no
E350	2011	CAN2-1	CAN1	no	no	no

E350	2012	CAN2-1	CAN1	no	no	no
Escape	2013	CAN1	CAN1	CAN2-1	CAN2-1	no
Explorer	2012	CAN1	CAN1	CAN2-1	CAN2-1	no
Explorer	2013	CAN1	CAN1	CAN2-1	CAN2-1	no
F-150	2011	CAN2-1				no
Falcon	2013	CAN2	CAN1	no	no	no
Figo	2010		CAN1	no	no	no
Focus	2009	CAN2-1	CAN1			no
Focus	2012	CAN1	CAN1	CAN2-1	CAN2-1	no
Focus	2013	CAN1	CAN1	CAN2-1	CAN2-1	no
Focus Electric	2013	CAN2-1	CAN2-1	CAN1	CAN1	no
Focus ST	2013	CAN2-1	CAN2-1	CAN1	CAN1	no
Fiesta	2012	CAN1	CAN1	no (tested without SYNC)	no (tested without SYNC)	no
Fusion	2006	CAN2-1	no	no	no	no
Fusion	2013	CAN1	CAN1	CAN1	CAN1	no
Mustang	2005	no	no	no	no	no
Mustang	2012	CAN2-1	CAN1	CAN2-1	CAN2-1	no
Taurus	2013	CAN1	CAN1	CAN2-1	CAN2-1	no
Territory	2013	CAN2	CAN1	no	no	no
MKZ	2013	CAN1	CAN1	CAN1	CAN1	no
Police						
Interceptor	2004	no	no	no	no	no

**Anexo 5: Encuesta realizada a personal de talleres mecánicos**

A continuación, se detallan las preguntas realizadas a personal de tres talleres diferentes de la ciudad de Posadas (Argentina).

- 1 ¿Cada cuántos kilómetros se hace el cambio de aceite?
- 2 ¿Cada cuántos kilómetros se hace el cambio de filtro de aceite?
- 3 ¿Cada cuántos kilómetros se hace el soplado de filtro de aire?
- 4 ¿Cada cuántos kilómetros se hace el cambio de filtro de aire?
- 5 ¿Cada cuántos kilómetros se hace el cambio de filtro de polen?
- 6 ¿Cada cuántos kilómetros se hace el cambio de escobillas del limpia parabrisas?
- 7 ¿Cada cuántos kilómetros se hace el cambio de bujías?
- 8 ¿Cada cuántos kilómetros se hace el cambio de neumáticos?
- 9 ¿Cada cuántos kilómetros se hace el cambio de la correa de distribución?
- 10 ¿Cada cuántos kilómetros se hace el cambio de pastillas de freno?
- 11 ¿Cada cuántos kilómetros se hace el cambio de Fusibles y Lámparas?
- 12 ¿Qué otras cosas se pueden controlar?