

**Universidad Autónoma Metropolitana – Azcapotzalco**

**División de Ciencias Básicas e Ingeniería**

Ingeniería en Computación

**Proyecto Tecnológico**

**Aplicación Web para la visualización y análisis de  
datos extraídos del sistema ODBII para la gestión  
de una flota de camiones**

**Alejandro Luviano Zaldivar  
Matricula: 208202959**

**Trimestre 14 Otoño**

**Asesora  
Dra. Maricela Claudia Bravo Contreras**

Enero 2015

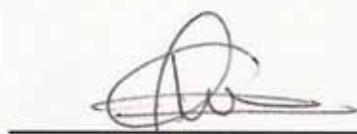
Yo, Maricela Claudia Bravo Contreras, *declaro que aprobé el contenido del presente Reporte de Proyecto de Integración y doy mi autorización para su publicación en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.*



---

Dra. Maricela Claudia Bravo Contreras  
Asesora

Yo, Alejandro Luviano Zaldivar, *doy mi autorización a la Coordinación de Servicios de Información de la Universidad Autónoma Metropolitana, Unidad Azcapotzalco, para publicar el presente documento en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.*



---

Alejandro Luviano Zaldivar  
Alumno

# Resumen

El presente documento tiene como propósito el de describir como fue realizada la aplicación web para gestionar transportes con datos obtenidos del sistema OBDII de un transporte; la propuesta para esta aplicación se enfocó el transporte a camiones, pero realmente la aplicación puede ser utilizada para gestionar datos de cualquier vehículo que cuente con un sistema OBDII. Cabe señalar que el dispositivo con el que se pretende obtener estos datos es independiente de este proyecto, el cual será el tema de otro proyecto de integración.

La aplicación web tiene como objetivo principal el despliegue de la información referente a los datos obtenidos del dispositivo; como lo es el gasto de combustible, el cual por medio de una gráfica se puede observar cómo se fue dando el consumo de éste y determinar si es adecuado o no dependiendo de la forma de la imagen; otro dato importante, son las lecturas de posicionamiento satelital; ya que la aplicación es capaz de mostrar estas lecturas en un mapa señalando el inicio de la ruta y el final de la misma, con el fin de que el usuario pueda saber si la ruta del transporte fue adecuada o no; los códigos DTC proporcionan información sobre fallas del vehículo y el dispositivo conectado a éste tendrá acceso a esta información, por lo cual el sistema proporcionara este código, la falla y la localización de la misma.

La aplicación web también incluye soporte para la gestión, en la cual se dan de alta a las empresas, usuarios que tendrán acceso al sistema y los dispositivos que obtendrán las lecturas de datos; todo esto solo lo pueden hacer los administradores de la aplicación; los usuarios pueden de dar de alta los transportes; así como los operadores de los mismos, esto también ayuda a los usuarios a tener un control sobre el personal que opera las unidades.

# Tabla de contenido

1. Introducción.....	1
2. Antecedentes y Justificación.....	1
3. Objetivos.....	2
4. Marco teórico.....	3
4.1 ¿Qué es el OBD? .....	3
4.1.1 Monitores de Emisiones OBDII.....	4
4.1.2 Acceso a la información del sistema OBDII.....	5
4.2 Modelo de desarrollo de software incremental .....	5
4.2.1 Iterativo y creciente .....	5
4.2.2 Ciclo de vida .....	6
4.2.3 Ventajas y desventajas .....	7
5. Desarrollo.....	9
5.1 Análisis.....	10
5.1.1 Modelo de Desarrollo .....	10
5.1.2 Análisis de requerimientos.....	12
5.1.3 Modelo de Dominio .....	23
5.2 Diseño.....	24
5.2.1 Modelo de Casos de Uso.....	24
5.2.2 Arquitectura de la aplicación .....	25
5.2.3 Diseño del modelo de Base de Datos .....	27
5.3 Implementación.....	28
5.3.1 Módulo de autenticación .....	29
5.3.2 Módulo de administración .....	31
5.3.3 Módulo de gestión.....	37
5.3.4 Información adicional.....	41
6. Conclusión.....	43
7. Bibliografía.....	44
8. Entregables .....	46

# 1. Introducción

Existen empresas que utilizan transporte de carga para realizar sus actividades laborales; ya sea propio o ajeno, lo cual implica que algunos operadores de este puedan realizar acciones deshonestas con los recursos proporcionados, como lo es la gasolina o el mismo transporte, facilitado por la empresa.

Los camiones actuales cuentan con un sistema OBD-II el cual está incorporado en la computadora del vehículo para monitorearlo y diagnosticar posibles fallas; pero también recupera información sobre las emisiones y los niveles de combustible; lo que facilita monitorear constantemente el estado del vehículo y del gasto de gasolina.

En este proyecto se implementará una aplicación web, la cual recibirá datos obtenidos por un dispositivo conectado al sistema OBDII del camión; éste dispositivo obtendrá la información referente al consumo de combustible, la ruta recorrida por el camión, así como posibles fallas en él (rpm, batería, etc.); esta información será enviada a la aplicación para que realice un análisis y muestre al usuario la información necesaria sobre el manejo de sus recursos.

## 2. Antecedentes y Justificación

En el transporte, el combustible tiene un alto impacto económico por lo cual es importante hacer buen uso de él y saber administrarlo de la mejor manera posible. Para tener una buena gestión de él se debe de tener un sistema de supervisión y especialmente un seguimiento del consumo en el camión. La gestión del combustible permite aprovechar cada litro de gasolina, contribuyendo a la economía de la empresa. Una adecuada gestión está ligada a una adecuada planificación de rutas y un correcto mantenimiento de los vehículos.

Actualmente en muchas empresas la regulación del consumo de gasolina y del transporte mismo se lleva a cabo de manera manual, algunas suelen llevar un control en papel o capturan los datos en algún programa para llevar el inventario de cuanto combustible gastan o el dinero destinado para ello, así como las rutas adecuadas para cada viaje; algunos operadores pueden hacer un mal uso del transporte y/o utilizarlo para actividades ajenas al trabajo o suelen sustraer el combustible en el transcurso del viaje y por ello se llegan a utilizar candados para que solo la empresa pueda suministrarlo.

## **3. Objetivos**

### **Objetivo General**

Diseñar e implementar una aplicación web para analizar y visualizar la información obtenida del dispositivo conectado al sistema OBDII de un camión.

### **Objetivos Particulares**

- Almacenar la información obtenida del dispositivo conectado al sistema OBDII del vehículo.
- Diseñar e implementar un módulo de autenticación para tener acceso al sistema.
- Diseñar e implementar un módulo para gestionar dispositivos.
- Diseñar e implementar un módulo que permita buscar y visualizar los dispositivos en uso.
- Diseñar e implementar un módulo para visualizar el gasto del combustible así como un análisis del mismo.
- Diseñar e implementar módulo que permita conocer la ruta recorrida por el vehículo.
- Diseñar e implementar un módulo que permita mostrar las posibles fallas encontradas en el vehículo.

## 4. Marco teórico

### 4.1 ¿Qué es el OBD?

*OBD (ON BOARD DIAGNOSTIC - DIAGNOSTICO A BORDO)* es una normativa que intenta disminuir los niveles de contaminación producida por los vehículos a motor.

La Comisión de Recursos del Aire de California (California Air Resources Board - CARB) comenzó la regulación de los Sistemas de Diagnóstico de a Bordo (On Board Diagnostic - OBD) para los vehículos vendidos en California, comenzando con los modelos del año 1988. La primera norma implantada fue la OBD I en 1988, donde se monitorizaban los parámetros de algunas partes del sistema como:

- La sonda lambda
- El sistema EGR y
- ECM (Modulo de control).

Una lámpara indicadora de mal funcionamiento (MIL), denominada Check Engine o Service Engine Soon, era requerida para que se iluminara y alertara al conductor del mal funcionamiento y de la necesidad de un servicio de los sistemas de control de emisiones.

Un código de falla (Diagnostic Trouble Code - DTC) era requerido para facilitar la identificación del sistema o componente asociado con la falla. Para modelos a partir de comienzos de 1994, ambos, CARB y la Agencia de Protección del Medio Ambiente (Environmental Protection Agency - EPA) aumentaron los requerimientos del sistema OBD, convirtiéndolo en el hoy conocido OBD II (2ª generación). A partir de 1996 los vehículos fabricados e importados por los USA tendrían que cumplir con esta norma. Según esto OBD II es un conjunto de normalizaciones que procuran facilitar el diagnóstico de averías y disminuir el índice de emisiones de contaminantes de los vehículos. La norma OBD II es muy extensa y está asociada a otras normas como SAE e ISO.

Estos requerimientos del sistema OBDII rigen para vehículos alimentados con gasolina, gasoil (diesel) y están comenzando a incursionar en vehículos que utilicen combustibles alternativos.

El sistema OBD II controla virtualmente todos los sistemas de control de emisiones y componentes que puedan afectar los gases de escape o emisiones evaporativas. Si un sistema o componente ocasiona que se supere el umbral máximo de emisiones o no opera dentro de las especificaciones del fabricante, un DTC (Diagnostic Trouble Code) debe ser almacenado y la lámpara MIL deberá encenderse para avisar al conductor de la falla. El sistema de diagnóstico de abordó no puede apagar el indicador MIL hasta que se realicen las correspondientes reparaciones o desaparezca la condición que provocó el encendido del indicador.

Un DTC es almacenado en la Memoria de Almacenamiento Activa (PCM Keep Alive Memory - KAM) cuando un mal funcionamiento es inicialmente detectado. En muchos casos la MIL es iluminada después de dos ciclos de uso consecutivos en los que estuvo presente la falla. Una vez que la MIL se ha iluminado, deben transcurrir tres ciclos de uso consecutivos sin que se detecte la falla para que la MIL se apague.

#### **4.1.1 Monitores de Emisiones OBDII**

Una parte importante del sistema OBDII de los vehículos, son los Monitores de Emisiones (autodiagnóstico de los elementos que intervienen en la combustión del motor y por lo tanto en las emisiones de escape), que son indicadores usados para averiguar si todos los componentes de emisiones, han sido evaluados por el sistema OBDII. Estos monitores procesan periódicamente pruebas en sistemas específicos y componentes, para asegurar que se están ejecutando dentro de límites permisibles.

Actualmente, hay 11 Monitores de Emisiones (o Monitores I/M) definidos por la Agencia de Protección Ambiental U.S (EPA). No todos los monitores están soportados por todos los vehículos y el número exacto de monitores en cada vehículo depende de la estrategia de control de emisiones de los fabricantes de motores de vehículos.

##### Monitores Continuos

Algunos de los componentes o sistemas de un vehículo se comprueban continuamente por el sistema OBDII del vehículo, mientras que otros son comprobados solo bajo condiciones específicas de operación del vehículo. Los componentes continuamente monitorizados enumerados a continuación están siempre listos:

1. Fallos del Encendido
2. Sistemas del Combustible
3. Componentes Globales (CCM)

Una vez que el vehículo se pone en marcha, el sistema OBDII está continuamente comprobando los componentes citados anteriormente, monitoriza los sensores clave del motor, vigilando los fallos de encendido del motor, y monitorizando las demandas de combustible.

##### Monitores no Continuos

A diferencia de los monitores continuos, muchas emisiones y componentes del sistema del motor, requieren que el vehículo esté funcionando bajo condiciones específicas antes de que el monitor esté listo. Estos monitores son llamados monitores no-continuos y se enumeran a continuación:

1. Sistema EGR
2. Sensores O2
3. Catalizador
4. Sistema Evaporativo
5. Calentador Sensor O2
6. Aire Secundario
7. Catalizador calentamiento
8. Sistema A/C

### **4.1.2 Acceso a la información del sistema OBDII**

Cuando el sistema almacena alguna información de error, nos indica, generalmente con una señal luminosa, que algo está funcionando incorrectamente y por tanto es aconsejable que acudamos a un taller para que revisen el automóvil. Una vez en el taller, el equipo de mecánicos, conectará nuestro automóvil un escáner o lector del sistema OBDII que le facilitara la información almacenada. A principios de los 80, cuando se extendió, el uso de este sistema de diagnosis, cada fabricante era libre de incorporar su propio conector y utilizar los códigos de error que quisiera. Esto dificultaba mucho la utilización de este sistema para la reparaciones, ya que la inversión que requería en los talleres mecánicos era altísima y poco practica (debían disponer de muchos lectores y de muchas tablas de códigos). Para que el uso de este sistema fuera practico y viable, en 1996, se llegó a un consenso entre los fabricantes y se estandarizaron los códigos y el conector. Así con un único lector de códigos y una tabla de errores, se puede diagnosticar un error en cualquier coche, independientemente del fabricante.

## **4.2 Modelo de desarrollo de software incremental**

Es un proceso de desarrollo de software creado en respuesta a las debilidades del modelo tradicional de cascada. Básicamente este modelo de desarrollo, que no es más que un conjunto de tareas agrupadas en pequeñas etapas repetitivas (iteraciones), es uno de los más utilizados en los últimos tiempos ya que, como se relaciona con novedosas estrategias de desarrollo de software y una programación extrema, es empleado en metodologías diversas. El modelo consta de diversas etapas de desarrollo en cada incremento, las cuales inician con el análisis y finalizan con la instauración y aprobación del sistema.

### **4.2.1 Iterativo y creciente**

Se planifica un proyecto en distintos bloques temporales que se le denominan iteración. En una iteración se repite un determinado proceso de trabajo que brinda un resultado más completo para un producto final, de forma que quien lo utilice reciba beneficios de este proyecto de manera creciente.

Para llegar a lograr esto, cada requerimiento debe tener un completo desarrollo en una única iteración que debe de incluir pruebas y una documentación para que el equipo pueda cumplir con todos los objetivos que sean necesarios y esté listo para ser dado al cliente. Así se evita tener riesgosas actividades en el proyecto finalizado.

Lo que se busca es que en cada iteración los componentes logren evolucionar el producto dependiendo de los completados de las iteraciones antecesoras, agregando más opciones de requisitos y logrando así un mejoramiento mucho más completo.

## **4.2.2 Ciclo de vida**

La idea principal detrás de mejoramiento iterativo es desarrollar un sistema de programas de manera incremental, permitiéndole al desarrollador sacar ventaja de lo que se ha aprendido a lo largo del desarrollo anterior, incrementando, versiones entregables del sistema. El aprendizaje viene de dos vertientes: el desarrollo del sistema, y su uso (mientras sea posible). Los pasos claves en el proceso son comenzar con una implementación simple de los requerimientos del sistema, e iterativamente mejorar la secuencia evolutiva de versiones hasta que el sistema completo esté implementado. En cada iteración, se realizan cambios en el diseño y se agregan nuevas funcionalidades y capacidades al sistema.

### ***Etapa de inicialización***

Se crea una versión del sistema. La meta de esta etapa es crear un producto con el que el usuario pueda interactuar, y por ende retroalimentar el proceso. Debe ofrecer una muestra de los aspectos claves del problema y proveer una solución lo suficientemente simple para ser comprendida e implementada fácilmente. Para guiar el proceso de iteración se crea una lista de control de proyecto, que contiene *un historial de todas las tareas que necesitan ser realizadas. Incluye cosas como nuevas funcionalidades para ser implementadas, y áreas de rediseño de la solución ya existente. Esta lista de control se revisa periódica y constantemente como resultado de la fase de análisis.*

### ***Etapa de iteración***

Esta etapa involucra el rediseño e implementación de una tarea de la lista de control de proyecto, y el análisis de la versión más reciente del sistema. La meta del diseño e implementación de cualquier iteración es ser simple, directa y modular, para poder soportar el rediseño de la etapa o como una tarea añadida a la lista de control de proyecto. El código puede, en ciertos casos, representar la mayor fuente de documentación del sistema. El análisis de una iteración se basa en la retroalimentación del usuario y en el análisis de las funcionalidades disponibles del programa. Involucra el análisis de la estructura, modularidad, usabilidad, confiabilidad, eficiencia y eficacia (alcanzar las metas). La lista de control del proyecto se modifica bajo la luz de los resultados del análisis.

### 4.2.3 Ventajas y desventajas

#### ***Ventajas***

- En este modelo los usuarios no tienen que esperar hasta que el sistema completo se entregue para hacer uso de él. El primer incremento cumple los requerimientos más importantes de tal forma que pueden utilizar el software al instante.
- Los usuarios pueden utilizar los incrementos iniciales como prototipos y obtener experiencia sobre los requerimientos de los incrementos posteriores del sistema. Existe muy pocas probabilidades de riesgo en el sistema. Aunque se pueden encontrar problemas en algunos incrementos, lo normal es que el sistema se entregue sin inconvenientes al usuario.
- Ya que los sistemas de más alta prioridad se entregan primero, y los incrementos posteriores se integran entre ellos, es muy poco probable que los sistemas más importantes sean a los que se les hagan más pruebas. Esto quiere decir que es menos probable que los usuarios encuentren fallas de funcionamiento del software en las partes más importantes del sistema.
- En el desarrollo de este modelo se da la retroalimentación muy temprano a los usuarios.
- Permite separar la complejidad del proyecto, gracias a su desarrollo por parte de cada iteración o bloque.
- El producto es consistente y puntual en el desarrollo.
- Los productos desarrollados con este modelo tienen una menor probabilidad de fallar.
- Se obtiene un aprendizaje en cada iteración que es aplicado en el desarrollo del producto y aumenta las experiencias para próximos proyectos.

#### ***Desventajas***

- La entrega temprana de los proyectos produce la creación de sistemas demasiados simples que a veces se ven un poco monótonos a los ojos del personal que lo recibe.
- La mayoría de los incrementos se harán en base de las necesidades de los usuarios. Los incrementos en si ya son estipulados desde antes de la entrega del proyecto, sin embargo hay que ver cómo se maneja el producto para ver si necesita otros cambios además de los estipulados antes de la entrega del proyecto. Este problema no se ve frecuentemente ya que la mayoría de las veces los incrementos estipulados suplen satisfactoriamente al usuario.
- Los incrementos no deben constar de muchas líneas de código ya que la idea de los incrementos es agregar accesorios al programa principal (o funcional), para que este tenga una y mil formas de desenvolverse en su tarea; llenar los incrementos de muchas líneas de código provocaría que se perdiera la objetividad o base de lo que se trata el desarrollo incremental.
- Requiere de un cliente involucrado durante todo el curso del proyecto. Hay clientes que simplemente no estarán dispuestos a invertir el tiempo necesario.

- El trato con el cliente debe basarse en principios éticos y colaboración mutua, más que trabajar cada parte independientemente, defendiendo sólo su propio beneficio.
- La entrega de un programa que es parcial pero funcional puede hacer vulnerable al programa debido a la falta de robustez en su sistema, provocando que agentes ajenos puedan interferir con el correcto funcionamiento del programa en sí.
- Infunde responsabilidad en el equipo de desarrollo al trabajar directamente con el cliente, requiriendo de profesionales sobre el promedio.

## 5. Desarrollo

Para el desarrollo de la aplicación se trató de seguir el modelo de desarrollo de software incremental; en este apartado se mostraran las etapas para la realización de la aplicación; la cual básicamente permitirá:

- Gestionar transportes y operadores de los mismos.
- Visualizar por medio de una gráfica el consumo del combustible.
- Visualizar en un mapa la ruta recorrida por el transporte.
- Encontrar alguna falla que la computadora del vehículo detecte.

Al ser una aplicación web; está tendrá distintas funcionalidades de acuerdo al tipo de usuario que acceda a ella, para obtener el tipo de usuario la aplicación contará con una página de validación en la cual el usuario introducirá su nombre de usuario y su contraseña, las cuales les fueron proporcionadas por el administrador del sistema.

La aplicación presenta dos opciones o una dependiendo el tipo de usuario correspondiente, administrar o gestión, las cuales se describirán más adelante en el documento; así como la opción de cerrar sesión si ya no se quiere trabajar en ella.

Para la elaboración de los mapas, se utiliza la API de GoogleMaps ya que actualmente es unos de los principales servidores de mapas en la web y ofrece herramientas adicionales para el usuario como tipos de mapas, zoom, etc.

### *Funcionalidad para el Administrador*

El Administrador: Son los usuarios que están a cargo del control, administración y configuración de la aplicación. Los administradores únicamente podrán realizar en la aplicación:

- Agregar, editar, eliminar y consultar las empresas en las cuales se utilizará la aplicación.
- Agregar, editar, eliminar y consultar los usuarios que tendrán acceso a la aplicación para después proporcionar de manera personal (correo o a algún otro medio de comunicación) el nombre de usuario y la contraseña para tener acceso a la aplicación.
- Agregar, editar, eliminar y consultar los dispositivos que serán montados en los vehículos para obtener los datos.

- En un momento dado, modificar las Bases de Datos.

El administrador deberá contar con su nombre de usuario y contraseña para ingresar a la aplicación.

## *Funcionalidad para el Usuario*

El usuario: Son los usuarios finales de la aplicación; los cuales podrán:

- Agregar, editar, eliminar y consultar los transportes dentro de la empresa.
- Agregar, editar, eliminar y consultar los operarios de la empresa para el transporte asignado.
- Visualizar el gasto del combustible realizado por el vehículo.
- Visualizar el mapa con la ruta realizada por el vehículo.

Cabe destacar que el usuario debe ser personal autorizado por la empresa para tener acceso a la aplicación, previo registro del administrador.

## **5.1 Análisis**

Para que se tenga un buen desarrollo de algún proyecto como este, es muy importante la parte de análisis ya que con éste se determina el alcance que puede llegar a tener el proyecto; la primer parte de este análisis se llevó a cabo en la propuesta para el proyecto en la cual se determinaron los objetivos y los problemas a resolver con la aplicación, así como los tiempos de desarrollo de cada parte de la misma; la definición de análisis en desarrollo de software es relativamente amplia, básicamente se definirá como análisis de requerimientos (estudio de los requisitos) y análisis de objetos (estudio de los objetos del dominio).

### **5.1.1 Modelo de Desarrollo**

Propiamente no se utilizó un modelo de desarrollo por el tiempo de realización de la aplicación; pero si se basa en un modelo evolutivo el cual su principal característica es adaptarse a la naturaleza evolutiva del software y en la cual se producen versiones más complejas del mismo; este modelo es el incremental ya que es una combinación del modelo lineal y el de prototipos y algunos de sus beneficios son:

- En este modelo los usuarios no tienen que esperar hasta que el sistema completo se entregue para hacer uso de él. El primer incremento cumple los requerimientos más importantes de tal forma que pueden utilizar el software al instante.
- Los usuarios pueden utilizar los incrementos iniciales como prototipos y obtener experiencia sobre los requerimientos de los incrementos posteriores del sistema.
- Existen muy pocas probabilidades de riesgo en el sistema. Aunque se pueden encontrar problemas en algunos incrementos, lo normal es que el sistema se entregue sin inconvenientes al usuario.
- Ya que los sistemas de más alta prioridad se entregan primero, y los incrementos posteriores se integran entre ellos, es muy poco probable que los sistemas más importantes sean a los que se les hagan más pruebas. Esto quiere decir que es menos probable que los usuarios encuentren fallas de funcionamiento del software en las partes más importantes del sistema
- En este modelo los usuarios no tienen que esperar hasta que el sistema completo se entregue para hacer uso de él. El primer incremento cumple los requerimientos más importantes de tal forma que pueden utilizar el software al instante.
- Los usuarios pueden utilizar los incrementos iniciales como prototipos y obtener experiencia sobre los requerimientos de los incrementos posteriores del sistema.
- Existe muy pocas probabilidades de riesgo en el sistema. Aunque se pueden encontrar problemas en algunos incrementos, lo normal es que el sistema se entregue sin inconvenientes al usuario.
- Ya que los sistemas de más alta prioridad se entregan primero, y los incrementos posteriores se integran entre ellos, es muy poco probable que los sistemas más importantes sean a los que se les hagan más pruebas. Esto quiere decir que es menos probable que los usuarios encuentren fallas de funcionamiento del software en las partes más importantes del sistema

## 5.1.2 Análisis de requerimientos

En la sección de *Introducción* se describieron algunas de las funcionalidades de la aplicación, dependiendo del tipo de usuario que ingrese a esta; para la clasificación de los requerimientos se utilizó el modelo FURPS+; pero se agruparon de distinta manera para hacer más fácil su análisis sobre la aplicación:

### 1. Sistema

- Conexión vía web para ser utilizada desde cualquier sitio con conexión a internet.
- Bajos tiempos de latencia.
- Tolerante a fallas.
- Uso de software libre.
- Adaptabilidad y facilidad de mantenimiento.
- Fácil extensibilidad.

### 2. Interfaces

- Usabilidad para el usuario.
- Sistema intuitivo.
- Solo se necesitan conocimientos para usar el navegador.
- Compatible con distintos navegadores.
- Atractivo y dinámico a la vista.
- Uso correcto de colores y botones.

### 3. Seguridad

- Autenticación de usuarios por medio de un nombre de usuario y contraseña.
- Contraseña encriptada.
- Manejo de sesiones.

### 4. Administrador

- Solo puede ingresar al módulo administrativo para gestionar empresas, usuarios y dispositivos.
- Para registrar, modificar y/o eliminar una empresa el usuario debe conocer su información (Nombre, Dirección y Teléfono).
- Para registrar, modificar y/o eliminar un usuario el administrador debe conocer su información (Nombre completo, Dirección, Teléfono, correo electrónico y tener establecido el tipo de usuario).
- Establecer el nombre de usuario y contraseña de cada usuario.
- Para registrar, modificar y/o eliminar un dispositivo el usuario debe conocer su información (dirección MAC y si es necesaria una descripción del mismo).

### 5. Usuario-Administrador y Usuario

- Únicamente el usuario-administrador puede gestionar los transportes

- y operadores de éstos.
- Para registrar, modificar y/o eliminar un transporte el usuario debe conocer su información (Marca, Modelo, Año, No. Placa, el dispositivo al que se va a signar).
- Para registrar, modificar y/o eliminar un operador el usuario debe conocer su información (Nombre completo, Dirección y a que unidad se debe asignar).
- Únicamente el usuario-administrador y el usuario tienen acceso al módulo de gestión.

## 6. Información del recorrido

- Listar los dispositivos correspondientes a la empresa ordenados por la fecha del último recorrido.
- Tener acceso a los últimos diez recorridos de cada dispositivo.
- Mostrar una gráfica con las lecturas del nivel de combustible, así como proporcionar el valor inicial y el final para determinar el gasto durante el recorrido y el rendimiento Km/l del transporte.
- Mostrar un mapa que muestre el recorrido del vehículo indicando el inicio y el fin del mismo; así como la distancia y el tiempo aproximado del recorrido.
- Proporcionar información sobre el código DTC obtenido del dispositivo y proporcionar (si fuera el caso) la localización y la causa de la falla.

### 5.1.2.1 Descripción de actores

Un actor es un ente externo que interactúa con la aplicación; en esta se encuentran tres actores principales y uno secundario (base de datos); este último se omitirá ya que está presente en cualquier interacción con la aplicación.

<b>Actor</b>	Administrador
<b>Caso de Uso</b>	Ingreso a la Aplicación, Administrar Empresa, Administrar Usuarios, Administrar Dispositivos, Salir de la Aplicación.
<b>Tipo</b>	Principal.
<b>Descripción</b>	Usuario con los privilegios para dar de alta, modificar y/o eliminar las empresas, usuarios y dispositivos a utilizar. Inicializa la configuración de la aplicación.

<b>Actor</b>	Usuario-Administrador
<b>Caso de Uso</b>	Ingreso a la Aplicación, Administrar Transporte, Administrar Operadores, Selección de Dispositivo y Fecha, Información del Recorrido, Salir de la Aplicación.
<b>Tipo</b>	Principal.
<b>Descripción</b>	Usuario con los privilegios para dar de alta, modificar y/o eliminar los transportes y operadores de su empresa. Tiene acceso a la información obtenida de los dispositivos asociados a su empresa.

<b>Actor</b>	Usuario
<b>Caso de Uso</b>	Ingreso a la Aplicación, Selección de Dispositivo y Fecha, Información del Recorrido, Salir de la Aplicación.
<b>Tipo</b>	Principal.
<b>Descripción</b>	Tiene acceso a la información obtenida de los dispositivos asociados a su empresa.

### 5.1.2.2 Descripción de casos de uso

A continuación se describen los casos de uso para el diseño de la aplicación que será descrito más adelante; cabe señalar que al hablar de usuario en alguna sección distinta a actores, se refiere a alguno de estos dentro del caso de uso.

<b>Caso de uso</b>	Ingreso a la aplicación.
<b>Actores</b>	Administrador, Usuario-Administrador, Usuario
<b>Tipo</b>	Inclusión
<b>Propósito</b>	EL usuario tenga acceso a la aplicación de manera segura.
<b>Resumen</b>	El usuario deberá ingresar su nombre de usuario y contraseña para ingresar a la aplicación.
<b>Pre-condiciones</b>	El usuario debe estar registrado en la base de datos.
<b>Post-condiciones</b>	El usuario será autenticado y podrá ingresar a la aplicación
<b>Flujo principal</b>	Se presenta la interface de inicio de sesión en la cual el usuario deberá ingresar su nombre de usuario y contraseña y presionará el botón de <i>entrar</i> .
<b>Sub-flujo</b>	(S-1): Al presionar el botón de <i>entrar</i> la aplicación validará los datos introducidos permitiendo al usuario ingresar si los datos son correctos.
<b>Excepciones</b>	(E-1): Los datos del usuario son inválidos; presentar la pantalla de nuevo informando que campo es erróneo.

<b>Caso de uso</b>	Gestionar Funciones
<b>Actores</b>	Administrador, Usuario-Administrador, Usuario.
<b>Tipo</b>	Inclusión
<b>Propósito</b>	Los usuarios tendrán la pantalla principal del sistema y podrán elegir la opción que deseen utilizar
<b>Resumen</b>	El usuario elegirá dentro de las opciones, alguna acción que desee realizar, dependiendo del tipo de usuario que corresponda.
<b>Pre-condiciones</b>	El usuario debe estar autenticado.
<b>Post-condiciones</b>	EL usuario podrá ejecutar los demás casos de uso referentes a los privilegios de acuerdo al tipo de usuario que sea.
<b>Flujo principal</b>	El usuario se encuentra dentro del menú principal una ve autenticado y deberá elegir la acción a realizar dependiendo del tipo de usuario que sea.
<b>Sub-flujo</b>	(S-1): Al presionar sobre Administrar, podrá ejecutar los casos de uso: Administrar empresa(A), Administrar usuario(A), Administrar dispositivos(A), Administrar transporte(U-A), Administrar operadores(U-A) (S-2): Al presionar sobre la opción gestionar el usuario podrá ejecutar el caso de uso Recorrido.
<b>Excepciones</b>	(E-1): Cada usuario ya tiene delimitados los privilegios o limitaciones dentro de la aplicación.

<b>Caso de uso</b>	Administrar Empresas.
<b>Actores</b>	Administrador.
<b>Tipo</b>	Exclusión.
<b>Propósito</b>	El usuario administre la información de las empresas que utilizaran la aplicación
<b>Resumen</b>	El usuario registrará, editara y/o eliminará información referente a las empresas que trabajaran con la aplicación.
<b>Pre-condiciones</b>	El usuario debe estar autenticado y haber oprimido sobre la opción <i>Empresas</i> del menú <i>Administrar</i> .
<b>Post-condiciones</b>	La información ingresada será almacenada en la BD. La aplicación muestra la lista de las empresas actualizada, una vez realizado el cambio.
<b>Flujo principal</b>	Se presenta la interface de Empresas en la cual se listan todas las empresas con su respectiva información; la aplicación solo muestra 10 empresas, si desea ver las siguientes (si es que existen) el usuario deberá presionar el número de la página que quiera consultar o el botón de siguiente o anterior según sea el caso. El usuario seleccionará una opción de configuración (nuevo, editar o eliminar) y al presionar sobre guardar o aceptar se actualizará la interface de <i>Empresas</i> .
<b>Sub-flujo</b>	(S-1): Al presionar sobre <i>Nuevo</i> aparece el modal para registrar la empresa. (S-2): Al presionar sobre el ícono de editar, se desplegará el modal para editar la información de la empresa. (S-3): Al presionar sobre el ícono de eliminar, se desplegará una advertencia y se deberá confirmar la acción.
<b>Excepciones</b>	(E-1): Únicamente el Administrador tiene acceso a este caso de uso.

<b>Caso de uso</b>	Administrar Usuarios.
<b>Actores</b>	Administrador.
<b>Tipo</b>	Exclusión.
<b>Propósito</b>	El usuario administre la información de los usuarios que utilizaran la aplicación.
<b>Resumen</b>	El usuario registrará, editara y/o eliminará información referente a los usuarios que trabajaran con la aplicación.
<b>Pre-condiciones</b>	El usuario debe estar autenticado y haber oprimido sobre la opción <i>Usuarios</i> del menú <i>Administrar</i> .
<b>Post-condiciones</b>	La información ingresada será almacenada en la BD. La aplicación muestra la lista de los usuarios actualizada, una vez realizado el cambio.
<b>Flujo principal</b>	Se presenta la interface de usuarios en la cual se listan todas los usuarios con su respectiva información; la aplicación solo muestra 10 usuarios, si desea ver los siguientes (si es que existen) el usuario deberá presionar el número de la página que quiera consultar o el botón de siguiente o anterior según sea el caso. El usuario seleccionará una opción de configuración (nuevo, editar o eliminar) y al presionar sobre guardar o aceptar se actualizará la interface de <i>Usuarios</i> .
<b>Sub-flujo</b>	(S-1): Al presionar sobre <i>Nuevo</i> aparece el modal para registrar un usuario. (S-2): Al presionar sobre el ícono de editar, se desplegará el modal para editar la información del usuario. (S-3): Al presionar sobre el ícono de eliminar, se desplegará una advertencia y se deberá confirmar la acción.
<b>Excepciones</b>	(E-1): Únicamente el Administrador tiene acceso a este caso de uso. (E-2): Para registrar un usuario debió de haber sido registrada la empresa de éste previamente.

<b>Caso de uso</b>	Administrar Dispositivos
<b>Actores</b>	Administrador.
<b>Tipo</b>	Exclusión.
<b>Propósito</b>	El usuario administre la información de los dispositivos que serán asignados a un transporte.
<b>Resumen</b>	El usuario registrará, editara y/o eliminará información referente a los dispositivos que serán asignados a un vehículo.
<b>Pre-condiciones</b>	El usuario debe estar autenticado y haber oprimido sobre la opción <i>Dispositivos</i> del menú <i>Administrar</i> .
<b>Post-condiciones</b>	La información ingresada será almacenada en la BD. La aplicación muestra la lista de los dispositivos actualizada, una vez realizado el cambio.
<b>Flujo principal</b>	Se presenta la interface de dispositivos en la cual se listan todas los dispositivos con su respectiva información; la aplicación solo muestra 10 dispositivos, si desea ver los siguientes (si es que existen) el usuario deberá presionar el número de la página que quiera consultar o el botón de siguiente o anterior según sea el caso. El usuario seleccionará una opción de configuración (nuevo, editar o eliminar) y al presionar sobre guardar o aceptar se actualizará la interface de <i>Dispositivos</i> .
<b>Sub-flujo</b>	(S-1): Al presionar sobre <i>Nuevo</i> aparece el modal para registrar un dispositivo. (S-2): Al presionar sobre el ícono de editar, se desplegará el modal para editar la información del dispositivo. (S-3): Al presionar sobre el ícono de eliminar, se desplegará una advertencia y se deberá confirmar la acción.
<b>Excepciones</b>	(E-1): Únicamente el Administrador tiene acceso a este caso de uso.

<b>Caso de uso</b>	Administrar Transporte.
<b>Actores</b>	Usuario-Administrador.
<b>Tipo</b>	Exclusión.
<b>Propósito</b>	El usuario administre la información del transporte utilizado en la empresa del cual se desee obtener información sobre su ruta.
<b>Resumen</b>	El usuario registrará, editara y/o eliminará información referente a los vehículos que serán asignados a un dispositivo.
<b>Pre-condiciones</b>	El usuario debe estar autenticado y haber oprimido sobre la opción <i>Transporte</i> del menú <i>Administrar</i> .
<b>Post-condiciones</b>	La información ingresada será almacenada en la BD. La aplicación muestra la lista de los vehículos actualizada, una vez realizado el cambio.
<b>Flujo principal</b>	Se presenta la interface de transporte en la cual se listan todas los vehículos con su respectiva información; la aplicación solo muestra 10 vehículos, si desea ver los siguientes (si es que existen) el usuario deberá presionar el número de la página que quiera consultar o el botón de siguiente o anterior según sea el caso. El usuario seleccionará una opción de configuración (nuevo, editar o eliminar) y al presionar sobre guardar o aceptar se actualizará la interface de <i>Transporte</i> .
<b>Sub-flujo</b>	(S-1): Al presionar sobre <i>Nuevo</i> aparece el modal para registrar un vehículo. (S-2): Al presionar sobre el ícono de editar, se desplegará el modal para editar la información del vehículo. (S-3): Al presionar sobre el ícono de eliminar, se desplegará una advertencia y se deberá confirmar la acción.
<b>Excepciones</b>	(E-1): Únicamente el Usuario-Administrador tiene acceso a este caso de uso. (E-1): Para registrar un transporte se le debe de asignar un dispositivo, el cual debió de haber sido registrado previamente por el Administrador.

<b>Caso de uso</b>	Administrar Operadores.
<b>Actores</b>	Usuario-Administrador.
<b>Tipo</b>	Exclusión.
<b>Propósito</b>	El usuario administre la información de los operadores del transporte utilizado en la empresa del cual se desee obtener información sobre su ruta.
<b>Resumen</b>	El usuario registrará, editara y/o eliminará información referente a los operadores que serán asignados a un transporte.
<b>Pre-condiciones</b>	El usuario debe estar autenticado y haber oprimido sobre la opción <i>Operador</i> del menú <i>Administrar</i> .
<b>Post-condiciones</b>	La información ingresada será almacenada en la BD. La aplicación muestra la lista de los operadores actualizada, una vez realizado el cambio.
<b>Flujo principal</b>	Se presenta la interface de operadores en la cual se listan todas los operadores con su respectiva información; la aplicación solo muestra 10 operadores, si desea ver los siguientes (si es que existen) el usuario deberá presionar el número de la página que quiera consultar o el botón de siguiente o anterior según sea el caso. El usuario seleccionará una opción de configuración (nuevo, editar o eliminar) y al presionar sobre guardar o aceptar se actualizará la interface de <i>Operador</i> .
<b>Sub-flujo</b>	(S-1): Al presionar sobre <i>Nuevo</i> aparece el modal para registrar un operador. (S-2): Al presionar sobre el ícono de editar, se desplegará el modal para editar la información del operador. (S-3): Al presionar sobre el ícono de eliminar, se desplegará una advertencia y se deberá confirmar la acción.
<b>Excepciones</b>	(E-1): Únicamente el Usuario-Administrador tiene acceso a este caso de uso. (E-1): Para registrar un operador se le debe de asignar un transporte, el cual debió de haber sido registrado previamente por el Usuario-Administrador.

<b>Caso de uso</b>	Recorrido.
<b>Actores</b>	Usuario-Administrador, Usuario.
<b>Tipo</b>	Exclusión.
<b>Propósito</b>	EL usuario visualizará la información referente al dispositivo asignado a un vehículo.
<b>Resumen</b>	El usuario podrá observar la información del recorrido una vez finalizado éste; teniendo acceso a recorridos anteriores para que pueda hacer un análisis de la información proporcionada.
<b>Pre-condiciones</b>	El usuario debe estar autenticado y haber oprimido sobre la opción <i>Gestión</i> del menú principal, así como haber seleccionado un dispositivo de la lista.
<b>Post-condiciones</b>	El usuario podrá visualizar la información referente al dispositivo y la fecha del recorrido.
<b>Flujo principal</b>	Se presenta la interface de recorrido en la cual al presionar un dispositivo de la lista se desplegara la información referente al dispositivo y a la fecha correspondiente; si se desea seleccionar otra fecha se debe elegir una en la opción de <i>seleccionar fecha</i> y el sistema actualizará la información correspondiente.
<b>Sub-flujo</b>	(S-1): Al presionar sobre la pestaña de <i>Información</i> se desplegaran los datos del transporte, operador y el recorrido del dispositivo y recorrido seleccionados. (S-2): Al presionar sobre la pestaña de <i>Combustible</i> se desplegara una gráfica con las lecturas de este; así como los datos del consumo del mismo. (S-3): Al presionar sobre la pestaña <i>Ruta</i> se podrá observar el mapa con la ruta trazada por el vehículo así como la distancia y el tiempo del recorrido aproximados. (S-3): Al presionar sobre la pestaña <i>Fallas</i> se podrá observar el código DTC proporcionado por el dispositivo y si fuera el caso; la localización y la causa del mismo.
<b>Excepciones</b>	(E-1): Únicamente el Usuario-Administrador y el Usuario tiene acceso a este caso de uso. (E-1): La información del recorrido debió haber sido cargada previamente a la BD por el dispositivo.

<b>Caso de uso</b>	Salir de la aplicación.
<b>Actores</b>	Administrador, Usuario-Administrador, Usuario
<b>Tipo</b>	Inclusión
<b>Propósito</b>	EL usuario cierre la sesión de manera segura.
<b>Resumen</b>	El usuario cerrará la sesión de la aplicación para mantener la confidencialidad de los datos y que solo personal autorizado tenga acceso a la aplicación.
<b>Pre-condiciones</b>	El usuario debió de haber seleccionado la opción <i>Cerrar Sesión</i> del menú principal.
<b>Post-condiciones</b>	El usuario habrá finalizado su sesión en l aplicación.
<b>Flujo principal</b>	El usuario selecciona la opción de cerrar sesión y confirma la acción dando clic en Aceptar.
<b>Sub-flujo</b>	(S-1): Al cerrar la sesión se desplegara la interface de inicio de sesión.
<b>Excepciones</b>	(E-1): Si el usuario no confirma la acción seguirá trabajando dentro de la aplicación.

### 5.1.3 Modelo de Dominio

El objetivo del análisis es la de crear una descripción del dominio desde una visión de clasificación de objetos; Este modelo del dominio fue tomado para la realización del diseño del sistema y en este caso se puede ilustrar con un diagrama de objetos; cabe destacar que la clase myconn.php es exclusivamente para realizar las operaciones sobre la BD y así que las clases que heredan sus métodos puedan interactuar de una manera más cómoda con la BD; cabe destacar que este modelo solo sirvió para el diseño del sistema. EL diagrama se puede observar en la siguiente figura:

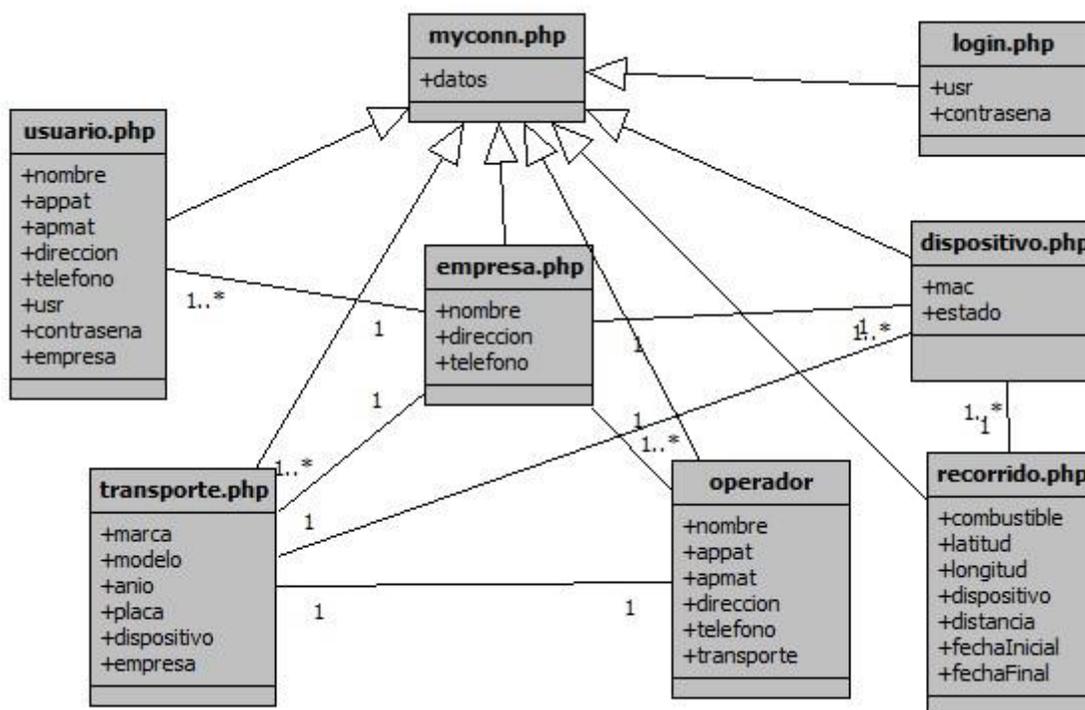


Fig.5.1.3 Modelo del Dominio

## 5.2 Diseño

En esta parte del documento se pretende mostrar la solución a a la fase de análisis para para la implementación de la aplicación.

### 5.2.1 Modelo de Casos de Uso

Dado el análisis, la elección de los actores y las relaciones entre los casos de uso; ya pueden ser establecidos en un diagrama de casos de uso para una mejor visualización de la interacción de los entes con la aplicación:

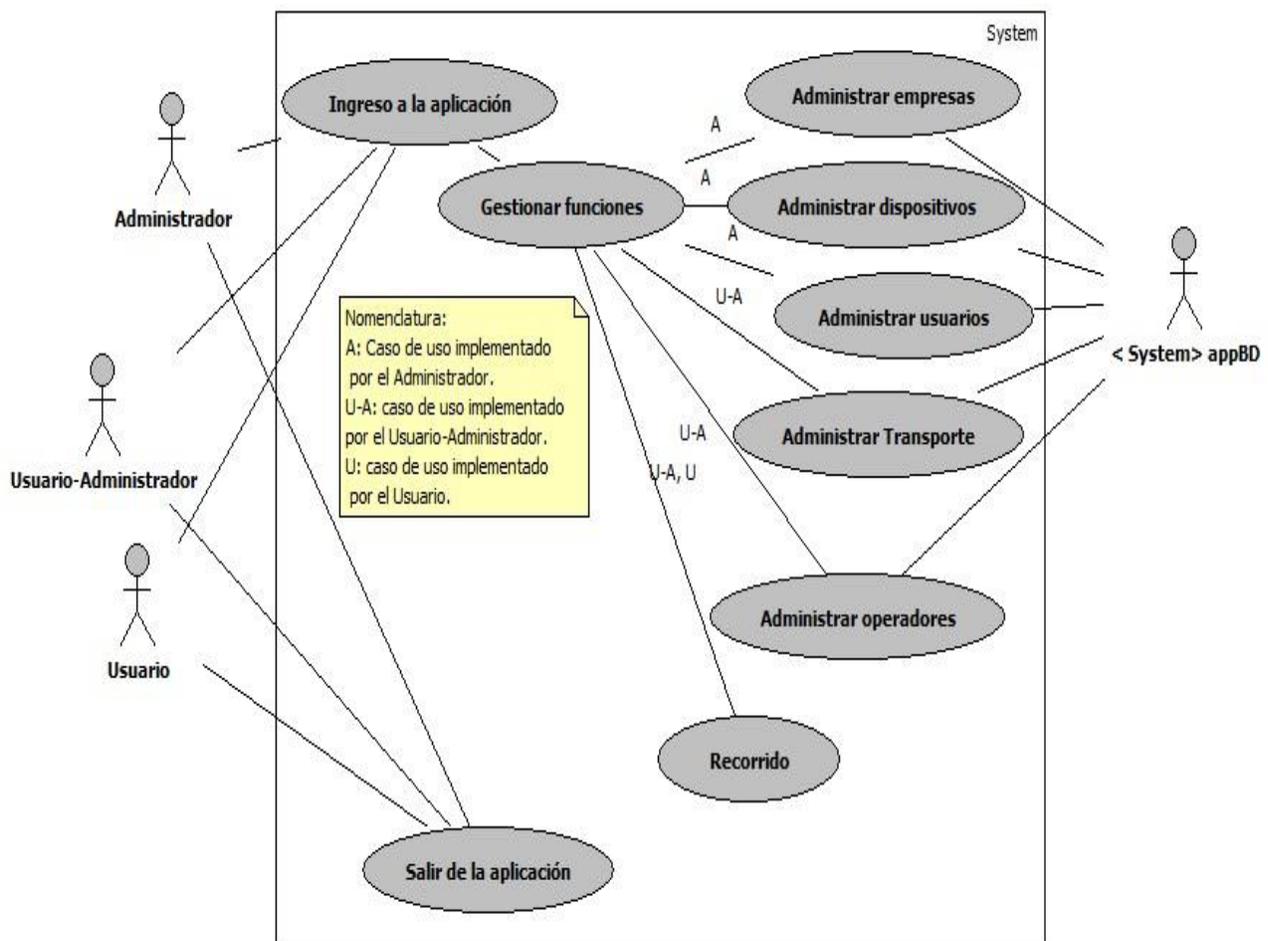


Fig.5.2.1 Modelo de casos de uso

## 5.2.2 Arquitectura de la aplicación

La aplicación estará integrada por distintos módulos y secciones, los cuales en conjunto, permitirán que los usuarios puedan realizar de manera exitosa las operaciones descritas anteriormente.

- **Lógica de la aplicación:** Es la parte central de la aplicación. En ella se establecerá la arquitectura a seguir y cómo van a interactuar los distintos módulos, la seguridad, base de datos, conexiones al servidor, etc.
- **Módulo gestión:** Básicamente es el resultado de la aplicación ya que el trabajo realizado en conjunto de los demás módulos permite que este sea ejecutado de manera adecuada. Mostrará al usuario los datos obtenidos del dispositivo conectado al sistema OBDII del vehículo, para que el usuario pueda interpretarlos.
- **Módulo Administración:** Este módulo estará dividido en cinco módulos: Módulo de administración de empresas, Módulo de administración de usuarios, Módulo de administración de dispositivos, Módulo de administración de transporte y Módulo de administración de operadores. En ellos se realizan las operaciones de registro, editar y/o eliminar información referente a cada módulo; así como listar de manera clara los datos referentes al tipo de módulo.
- **Módulo Autenticación:** El objetivo de este módulo es el de validar al usuario que quiera ingresar a la aplicación; así como crear las variables de sesión una vez dentro de la aplicación.
- **Base de Datos:** Al ser una aplicación dinámica, la base de datos es muy importante para este tipo de aplicaciones; ya que de la información almacenada depende el buen funcionamiento de la aplicación.
- **Interfaz de usuario:** Es la vista que tendrá el usuario para manipular y realizar las acciones dentro de la aplicación.

La siguiente figura muestra un diagrama de los módulos que conforman la arquitectura de la aplicación.

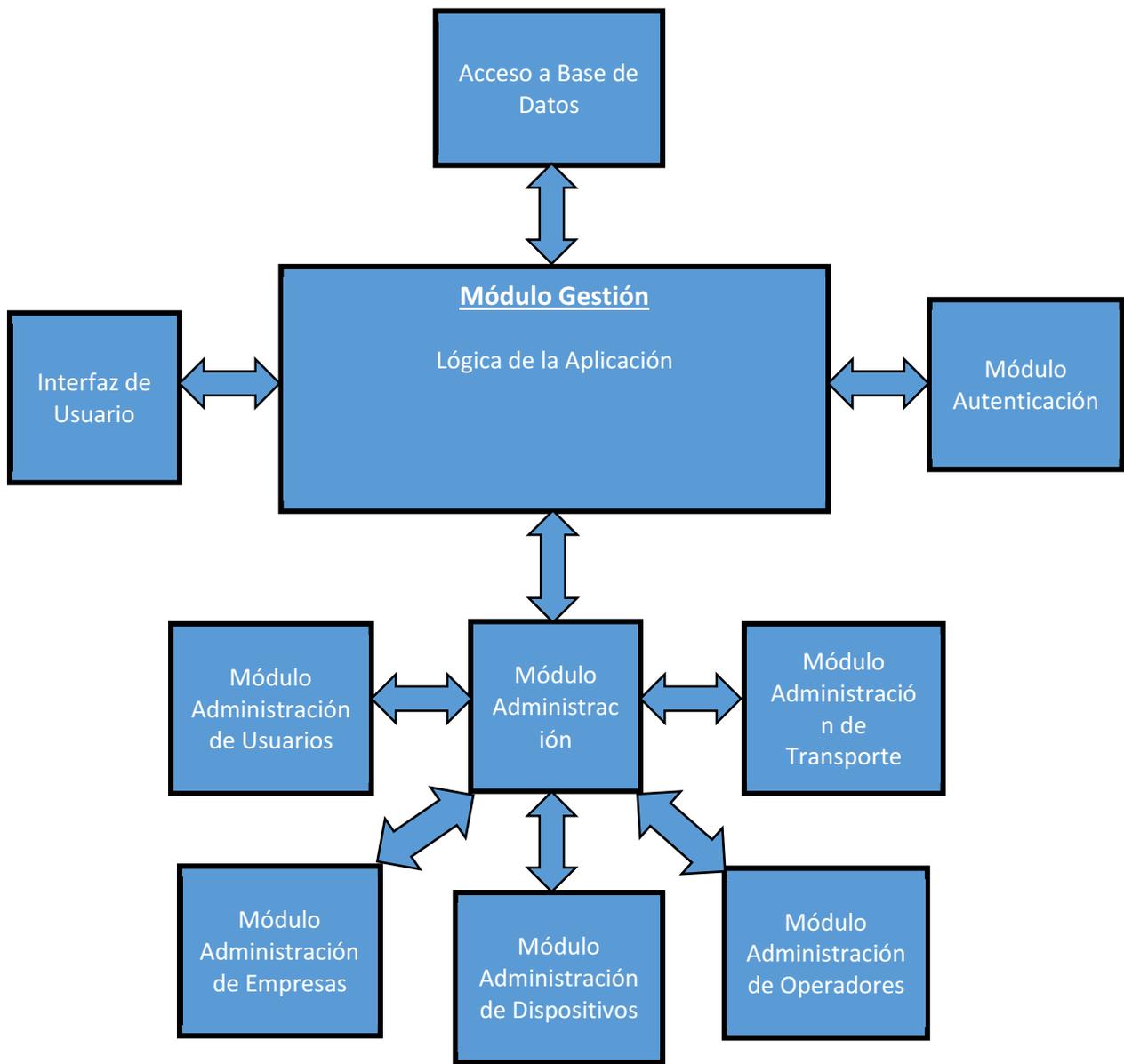


Fig.5.2.2 Arquitectura de la Aplicación.

## 5.2.3 Diseño del modelo de Base de Datos

Tomando como base el análisis principalmente en el modelo del dominio; las entidades de la base de datos son similares a las del diagrama de clases; sin embargo estas entidades van enfocada al almacenamiento de la información. La base de datos debe de cumplir con las características de una fácil administración, mantenimiento y eficiencia en el almacenamiento de la información utilizada por la aplicación.

Finalmente se construyó la base de datos que se muestra en la siguiente figura, en donde se pueden observar las entidades y el tipo de relación que hay entre ellas; cabe señalar que la tabla dtc solo es un catálogo de los códigos de error que pudiera registrar el dispositivo al obtener la información del transporte durante el recorrido.

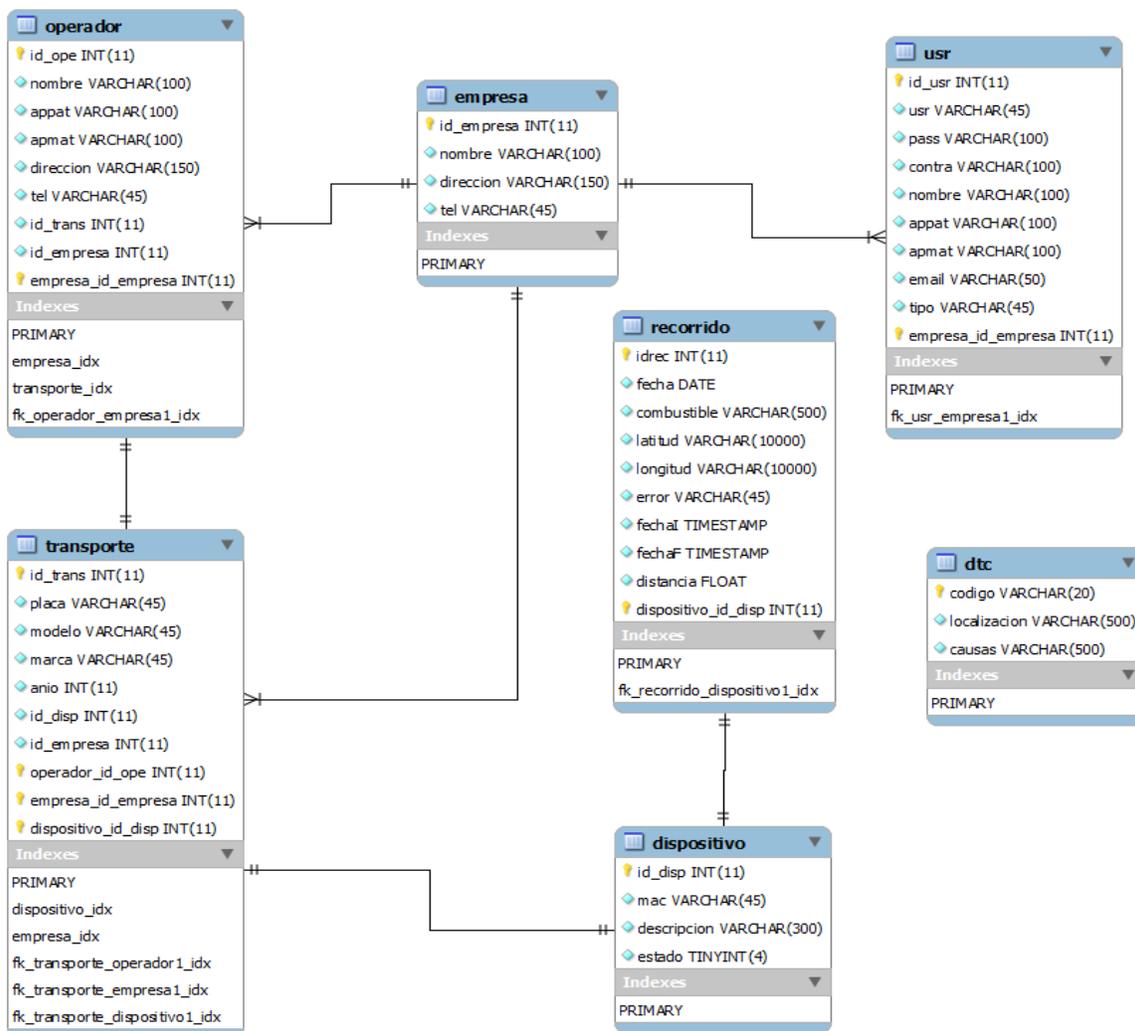


Fig.5.2.3 Modelo de Base de Datos..

## 5.3 Implementación

El ser una aplicación web implica una combinación de lenguajes y estándares propios de la red; los cuales en conjunto propician una buena funcionalidad del sistema. Para realizar la implementación de la aplicación, aunque principalmente esté basada en el lenguaje PHP; se utilizaron distintos lenguajes, librerías, técnicas, etc. para la realización del proyecto; a continuación se describe una muy breve definición de todas las herramientas/tecnologías utilizadas para la realización de la aplicación:

- **PHP:** Acrónimo de "*PHP: Hypertext Preprocessor*", es un lenguaje de 'scripting' de propósito general y de código abierto que está especialmente pensado para el desarrollo web y que puede ser embebido en páginas HTML. Su sintaxis recurre a C, Java y Perl, siendo así sencillo de aprender. La meta principal de este lenguaje es permitir a los desarrolladores web escribir dinámica y rápidamente páginas web generadas
- **HTML:** Siglas de HyperText Markup Language («lenguaje de marcas de hipertexto»), hace referencia al lenguaje de marcado para la elaboración de páginas web. Es un estándar que sirve de referencia para la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código (denominado código HTML) para la definición de contenido de una página web, como texto, imágenes, entre otros.
- **CSS:** Es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML.
- **Bootstrap:** Esencialmente Bootstrap (y su versión 3 que es la que utilizaremos) es una colección de hojas de estilo y funciones javascript auxiliares que permiten construir muy rápidamente un front-end responsive.
- **JavaScript:** es un lenguaje de programación interpretado, se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. Se utiliza principalmente en su forma del lado del cliente (client-side), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas. Todos los navegadores modernos interpretan el código JavaScript integrado en las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del Document Object Model (DOM).
- **jQuery:** Es una biblioteca de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web.

- **AJAX:** Es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (Rich Internet Applications). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, mejorando la interactividad, velocidad y usabilidad en las aplicaciones.
- **API de JavaScript de Google Maps:** Permite insertar Google Maps en páginas web; esta API está especialmente diseñada para proporcionar una mayor velocidad y que se pueda aplicar más fácilmente tanto a móviles como a las aplicaciones de navegador de escritorio tradicionales. La API proporciona diversas utilidades para manipular y para añadir contenido al mapa mediante diversos servicios, permitiéndote crear sólidas aplicaciones de mapas en un sitio web.
- **JpGraph:** Es una librería de clases orientadas a objetos para PHP, para la creación dinámica de imágenes. Con JpGraph se pueden crear gráficas complejas con un mínimo de código y con un control muy detallado.

Cabe señalar que para la realización de la aplicación se utilizó el IDE Netbeans 8 el cual fue configurado con los datos de servidor; éste es un servidor web (APACHE 2) y para la creación, administración y configuración de la base de datos se utilizó MySQL Workbench 6.

Como el proyecto va enfocado a la realización de la aplicación y para reducir tiempos, se trabajó con el Sistema Operativo Windows 8; así como el servidor independiente XAMPP 1.8.3 para Windows; lo que permite dejar de lado la configuración del servidor, así como del manejador de Base de datos (MySQL).

Como ya se mencionó anteriormente, la aplicación fue dividida en distintos módulos siguiendo el Modelo-Vista-Controlador; los cuales serán detallados a continuación; describiendo el objetivo del mismo; las clases y métodos principales para la realización de éste y la realización de la interfaz del usuario.

En esta parte sólo se mencionaran los nombres de las clases y los métodos relevantes para el funcionamiento; el código fuente completo se podrá consultar al anexo en el cd de entrega del proyecto; así como el script de la base de datos; para mas detalles consultar la sección 9 de este documento.

### 5.3.1 Módulo de autenticación

Este módulo tiene la finalidad de validar al usuario al momento de querer ingresar a la aplicación; para ello se creó la clase *login.php* la cual contiene los métodos de *Validar(nombre de usuario)* y *ValidarPass(contraseña, nombre de usuario)* en la cual se

consulta a la base de datos si estos datos corresponden a algún usuario registrado; estos métodos son utilizados en el controlador *autentica.php* en el cual se reciben los datos que vienen de la vista *login.php* los cuales son enviados por el método POST; una vez autenticado el usuario se crea la sesión y la variable de sesión `$_SESSION['id._usr']`; la cual es de gran importancia ya que contiene el valor de identificación del usuario; el cual es utilizado en varios scripts de la aplicación. A continuación se muestra la vista de *login.php* para ingresar al sistema; cabe señalar que en cualquier script de la aplicación se verifica que la sesión este creada; de lo contrario la aplicación mostrará la vista de *login.php*.

Fig.5.3.1a. Vista de *login.php*.

Una vez que el usuario sea autenticado satisfactoriamente; el usuario será re direccionado a la vista *main.php* en la cual, dependiendo el tipo de usuario mostrara una, o dos opciones como se muestran en la figuras 4.1b-1, 4.1b-2, 4.1b-3; esta vista utiliza los métodos de la clase *usuario.php*, en especial *get\_usuario(id del usuario)* para obtener los datos del usuario que accesa a la aplicación, esencialmente el tipo de usuario para saber que opción mostrar.

Fig.5.3.1b-1. Vista de *Administrador main.php*.

Fig. 5.3.1b-2. Vista de *Usuario-Administrador main.php*.

Fig. 5.3.1b-3. Vista de *Usuario main.php*.

Este script tiene un iframe en donde se cargan las vistas correspondientes a la opción seleccionada:

```
<iframe id="cont" name="cont" style="margin: 0px 0px 0px - 13px;width:102%;height: 580px; border: none"></iframe>
```

### 5.3.2 Módulo de administración

Para la realización de este módulo se utiliza una vista llamada *admin.php*; este script es cargado en el iframe de *main.php* al pulsar sobre la opción de *Administración*. Está dividido en dos partes; la parte del lado izquierdo muestra una lista con las opciones de lo que puede administrar el usuario dependiendo el tipo que sea. En la figura 4.1c-1 se muestra la lista para el Administrador y en la figura 4.1c-2 se muestra la lista para el Usuario-Administrador. Para cargar estas opciones únicamente se crea un usuario de la clase *usuario.php* para obtener el tipo de éste con el método *get\_usuario(id usuario)*.

#	Nombre	Direccion	Telefono		
1	Empresa2	perifericu sur 2345 colonia Itapan cp 14250	32435465	/	✕
2	Empresa3	perifericu sur 2345 colonia Itapan cp 14250	32435465	/	✕
3	Empresa5	perifericu sur 2345 colonia Itapan cp 14250	32435465	/	✕
4	Empresa6	perifericu sur 2345 colonia Itapan cp 14250	32435465	/	✕
5	Empresa7	perifericu sur 2345 colonia Itapan cp 14250	32435465	/	✕
6	Empresa8	perifericu sur 2345 colonia Itapan cp 14250	32435465	/	✕
7	Empresa9	perifericu sur 2345 colonia Itapan cp 14250	32435465	/	✕
8	Empresa11	perifericu sur 2345 colonia Itapan cp 14250	32435465	/	✕
9	Empresa12	perifericu sur 2345 colonia Itapan cp 14250	32435465	/	✕
10	Empresa13	perifericu sur 2345 colonia Itapan cp 14250	32435465	/	✕

Fig.5.3.2a-1. Lista para el administrador en *admin.php*.

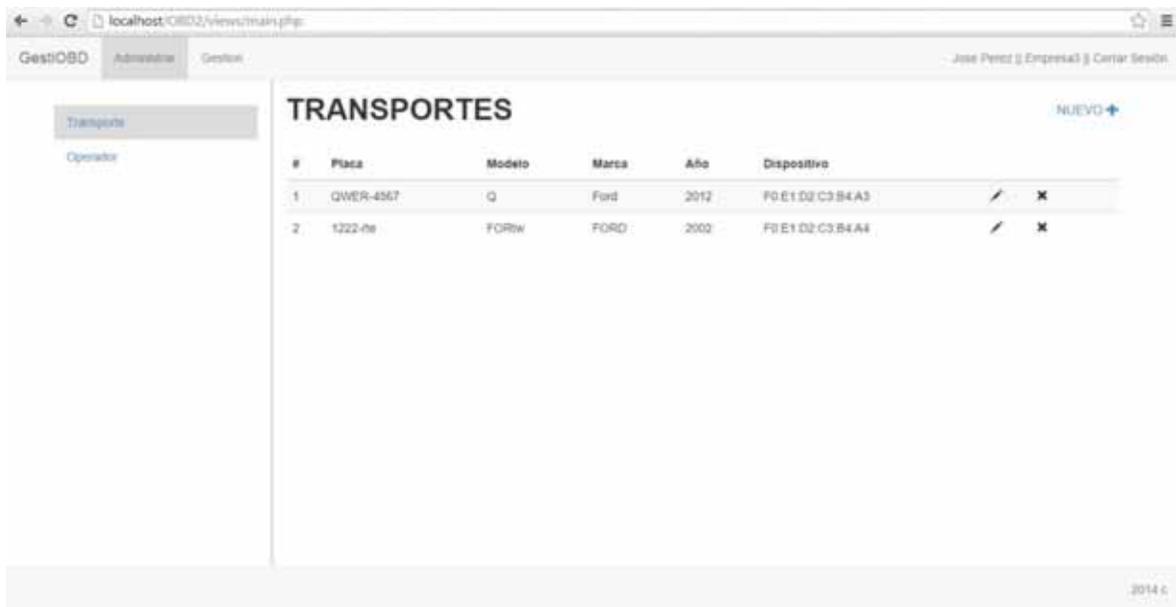


Fig.5.3.2a-2. Lista para el Usuario-administrador en admin.php.

La segunda parte del script solo contiene un div con id="list" en el que se carga otra lista correspondiente a la opción elegida que quiera administrarse; la información se carga con AJAX y básicamente es igual en las cinco distintas opciones; se lista una tabla con la información correspondiente a cada opción. Cada sub-módulo, correspondientes a Empresas, Usuarios, Dispositivos, Transporte y Operadores tienen la misma estructura. Existen las clases *empresa.php*, *usuario.php*, *dispositivo.php*, *transporte.php* y *operador.php* en las que se encuentran los métodos referentes a cada sub-módulo. En cada uno de ellos existen los métodos para listar la información como son: *Istempresa(num)*, *Istusuario(num)*, *Istdispositivo(num)*, *Isttransporte(num)* y *Istoperador(num)* respectivamente, al atributo num se refiere desde donde comienza la lista, ya que solo aparecen 10 resultados por página si se exceden los resultados aparece una lista de la siguiente página y así sucesivamente; esta parte de paginación se realiza con el método *paginacion(página activa)* el cual está presente en cada una de las clases y funciona en base a el número total de resultados que hay de cada categoría.

En cada script se encuentra una opción de nuevo; el cual al presionarse y por medio de JS con ayuda de la librería Bootstrap aparece un modal el cual es cargado por medio de AJAX dependiendo la categoría en la que se encuentre, el script *nmodals.php* recibe que categoría es la que debe mostrarse; como puede observarse en las siguientes figuras:

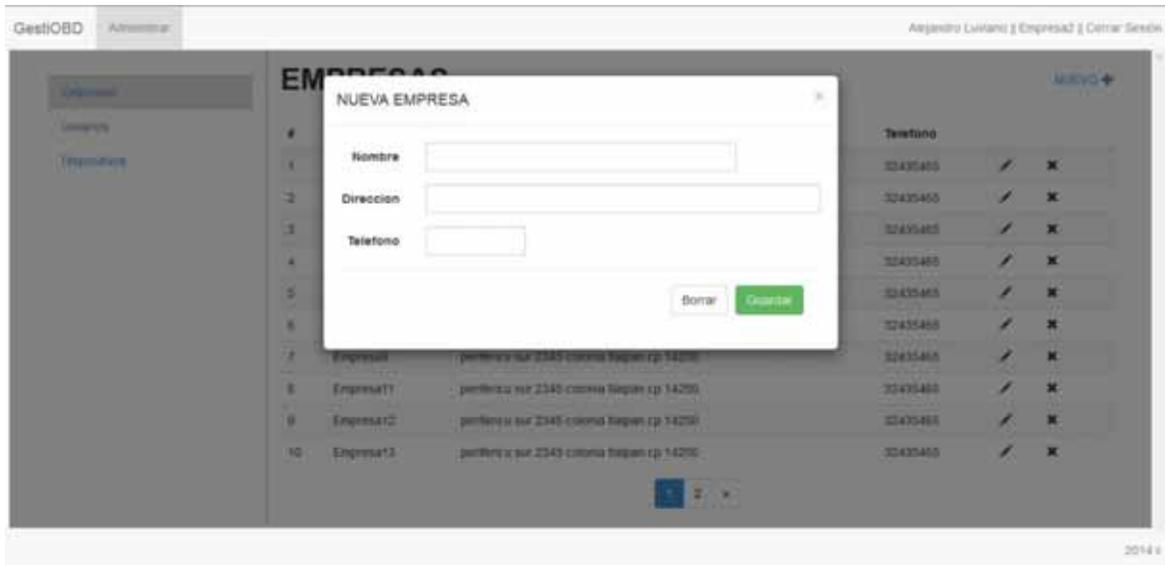


Fig.5.3.2b-1. Modal nuevo para empresa en admin.php



Fig.5.3.2b-2. Modal nuevo para usuario en admin.php



Fig.5.3.2b-3. Modal nuevo para dispositivo en admin.php

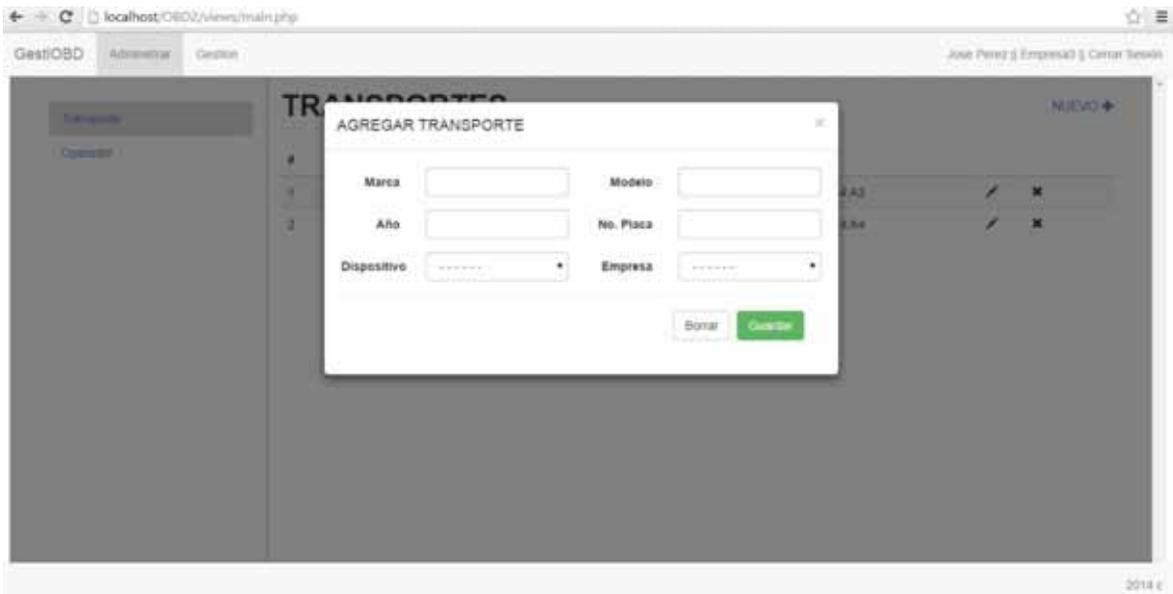


Fig.5.3.2b-4. Modal nuevo para transporte en admin.php

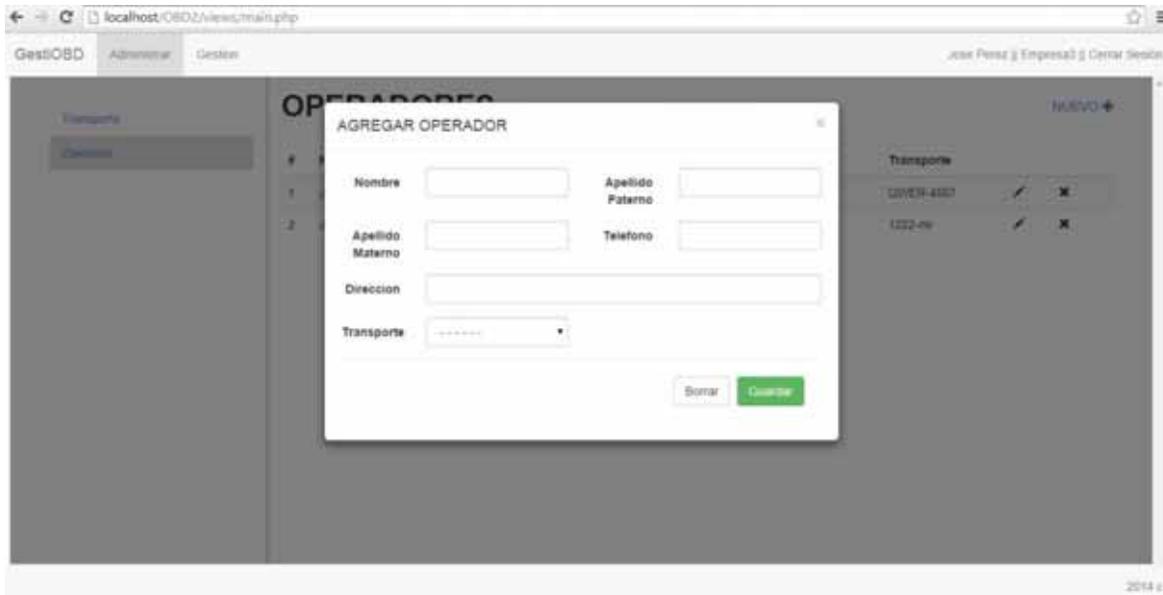


Fig.5.3.2b-5. Modal nuevo para operador en admin.php

Al presionar el botón de guardar, toda la información es enviada con el método POST al controlador *addEmpresa.php*, *addUsuario.php*, *addDispositivo.php*, *addTransporte.php* y *addOperador* según corresponda la categoría; en estos controladores se reciben los valores a ingresar a la base de datos con el método *addEmp(datos)*, *addUsr(datos)* en el que la contraseña se encripta con el método *encripta(contraseña)*, *addDisp(datos)* y *addOpe(datos)* respectivamente; también se obtiene el valor de número de página y la página activa los cuales también se obtienen con métodos descritos en cada clase; ya que el script *admin.php* recibe por medio del método GET estos valores para poder refrescar la parte de administración de cada categoría de manera adecuada; al final de cada script 'add' se redirecciona a *admin.php*; el cual además del número de página y el de página activo también recibe el valor *s*; en el cual se define que categoría estará activa por medio de javascript.

Cada fila de la lista correspondiente a cada categoría muestra la opción de editar representada por un lápiz y la opción de eliminar representado por una 'X'.

Al presionar la opción de editar; de la misma manera en que funciona el script *nmodals.php* se tiene *emodals.php*; ambos trabajan de la misma manera; solo que este último recibe el id correspondiente a la categoría y al elemento de la lista, se puede observar un ejemplo en la figura 4.1e-1; para después obtener los datos referentes a cada uno por medio del método *get\_empresa(id)*, *get\_usuario(id)*, *get\_dispositivo(id)*, *get\_transporte(id)* y *get\_operador(id)* cada uno de ellos ubicado dentro de la clase correspondiente a cada categoría. De la misma forma en la que se almacena nueva información; para la edición de ésta se utiliza de nuevo el script 'add' de cada categoría; solo que se denota por medio de la variable *accion=1* (escondida en el formulario) que se trata de una operación de update en la base de

datos y se debe de mandar el id correspondiente; una vez realizada la operación se recarga por medio de AJAX la lista.

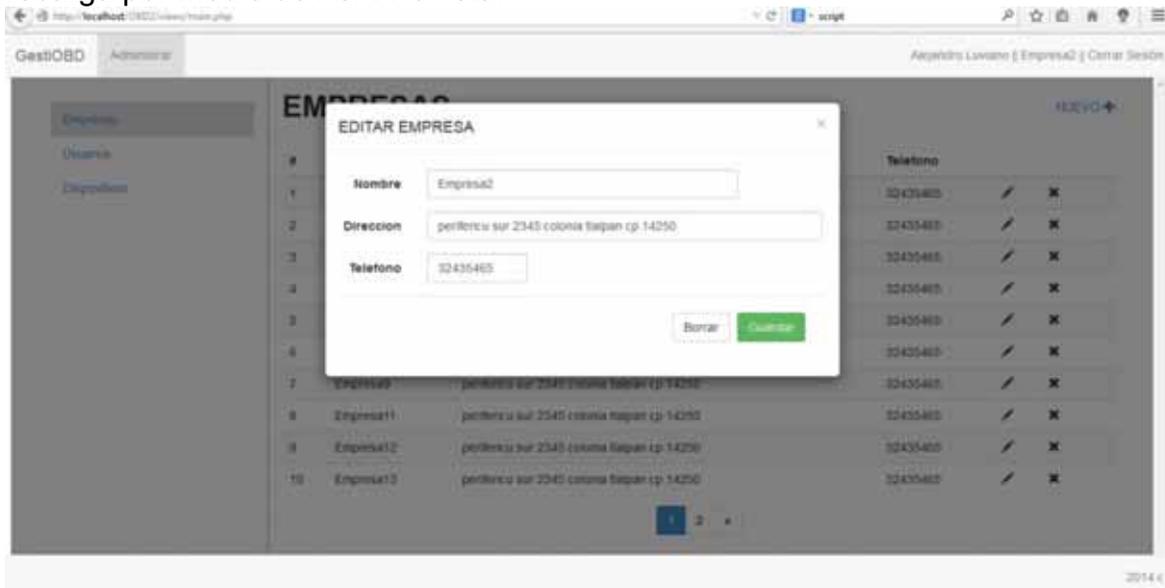


Fig.5.3.2c. Editar empresa en admin.php

Para eliminar datos referentes a un resultado de la categoría deseada; al presionar el boton de eliminar; pr medio de javascript se genera un mensaje de descición en el cual el usuario acepta o cancela la operación; en caso de aceptar se refesca el adminx(x=empresa,usuario,etc) ya que por medio de javascript en las funciones: lste(num de pagina, pagina activa, id) se manda por medio de ajax los valores de numero de pagina y la pagina activa; si se quiere eliminar se envía el id correspndiente al elemnot que desee eliminarse y pormedio del método *delEmp(id)*, *delUsr(id)*, *delDisp(id)*, *delTrans(id)* o *delOpe(id)*, según sea el caso, se elimina la informacion correspndiente al elemnto a eliminar.

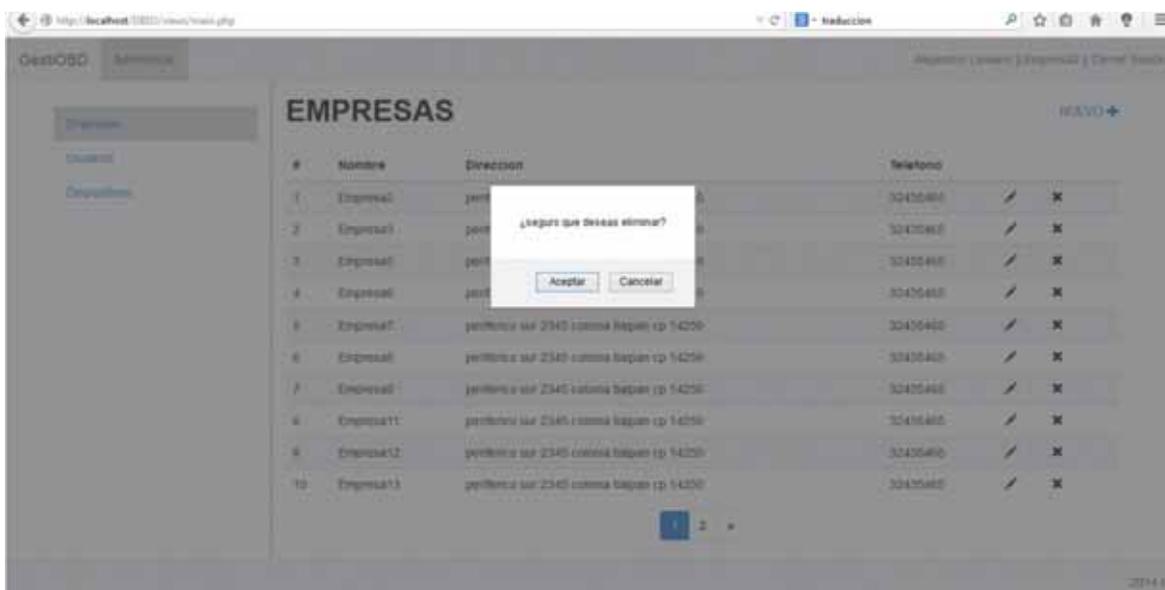


Fig.5.3.2d. Eliminar empresa en admin.php

Esto fue la parte referente al módulo de Administración; como pudo observarse todas las categorías tienen la misma lógica y por ende no difieren mucho los scripts relacionados con cada una de ellas.

### 5.3.3 Módulo de gestión

Para este módulo se creó al igual que en el anterior, la clase recorrido.php la cual contiene los métodos utilizados para la funcionalidad del mismo. Este módulo entra en función al seleccionar la opción gestión en main.php. fig 4.2a; cabe señalar que este módulo solo está disponible para tipos: *Usuarios-Administrador* y *Usuarios*.

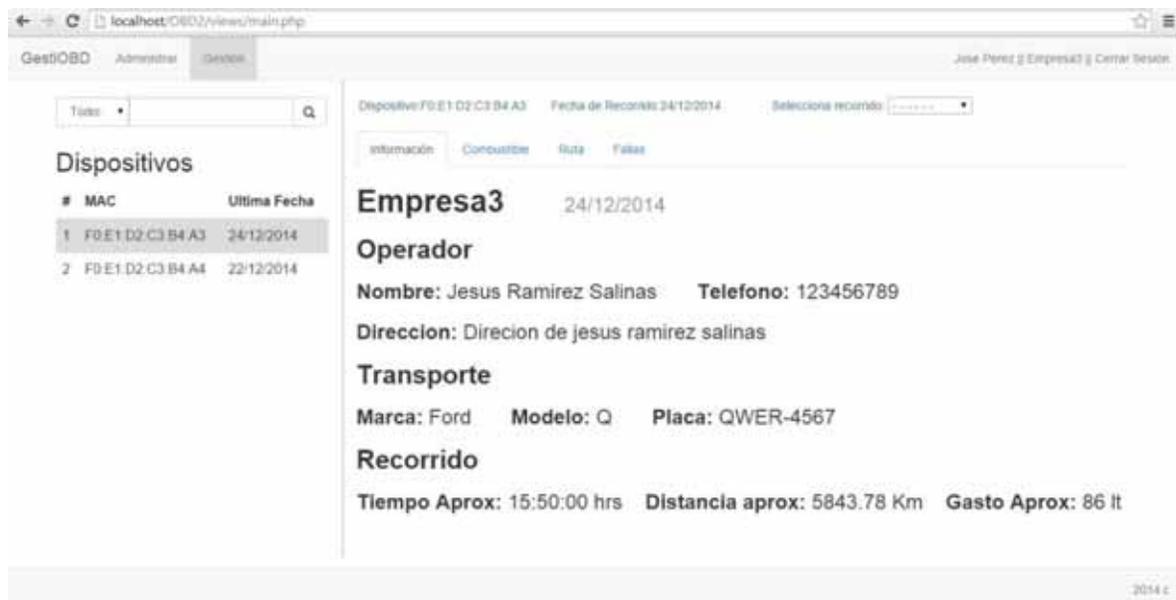


Fig.5.3.3a. Vista del módulo gestión en gestión.php

Como puede observarse en la figura de arriba; de la misma manera que el módulo de administrar, este está dividido en dos partes; la primera muestra una lista de los dispositivos registrados en la empresa; esto se realiza con el método *listrec(id usuario)* que al momento de seleccionarlo se refresca la segunda parte de la pantalla referente a los datos del recorrido; el dar click por medio de JavaScript se envía el id del recorrido (en este caso del último) y con AJAX se refresca solo la parte de la derecha de la pantalla.

La segunda parte está desarrollada en el script *contenido.php* en el que la información es organizada en pestañas; el cual recibe por medio del método GET el id del recorrido y del dispositivo para la obtención de toda la información de la base de datos. Con el método *getListFech(id del recorrido)* se genera una lista en la cual por medio de JavaScript, al seleccionar una opción se refresca de nuevo por medio de AJAX la información del contenido.

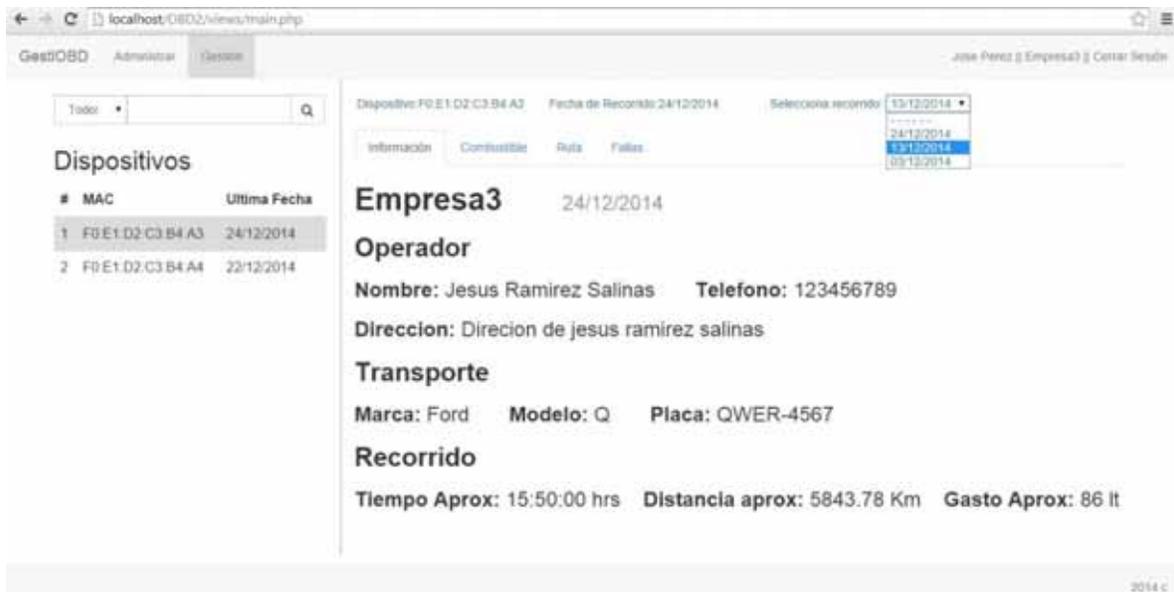


Fig.5.3.3b. Vista del módulo gestion con lista de fechas en gestion.php

En el script *contenido.php* estan definidas as pestañas; con el método *getInfo(id del dispositivo)* se obtienen todos los datos para la pestaña de información como se muestra en la figura anterior. Para la pestaña de combustible se utiliza la librería *jpgraph* en la cual se genera la gráfica referente a las lecturas del dispositivo sobre el consumo de combustible; para que se cree la imagen debe de establecerse una etiqueta de imagen:

```

```

Como puede observarse se establece la fuente con el nombre *grafica.php?idrec* el cual recibe el id del recorrido; esto puede observarse detalladamente en la seccion de entregables; en este, unicamente se obtienen los valores con el método *getCombustible(id del recorrido)* y se guardan en un arreglo para poder crear la imagen de la gráfica. Con los métodos *get\_gasto(id del recorrido)* y *rendimiento(id del recorrido)* de la clase *recorrido.php* se obtienen respectivamente el combustible inicial,final y el gasto; asi com el rendimiento en Km/l. como se muestra en la siguiente figura:



Fig. 5.3.3c. Vista de la pestaña combustible en gestion.php

Para la pestaña de Ruta en contenido.php se creó un iframe en donde se mostrará a información llamado *mapa.php*; en este script con ayuda de la API de google maps que utiliza javascript y AJAX, es posible crear la ruta recorrida por el vehículo. Con el método *getcoordenadas(id del recorrido)* se reciben los datos referentes a la latitud y longitud de cada lectura echa por el dispositivo durante el recorrido; con ellos se puede obtener la coordenada de inicio y la de fin con ayudas de funciones propias del lenguaje php. Para poder crear una linea se necesitan crear puntos los cuales se vana unir con una linea; estos puentos se generaron con un algoritmo que basicamente lo que hace es concatenar cada lectura de latitud con un de longitud e imprimir la instrucción para crear un punto; al mismo tiempo que eso se lleva a cabo, tambien se va generando el valor de la distancia para cada iteracion del numero de coordenadas; con el método *Distance(latitud coordenada actual, longitud coordenada actual, latitud coordenada anterior, longitud coordenada anterior, unidades en este caso:kilometros "K")* se obtiene la distacia de un punto a otro y se van sumando hasta que concluye el ciclo, obteniendo la distancia total del recorrido; tambien en este script con el método *getTiempo(id rec)* se determina el horario de salida, llegada y el tiempo transcurrido: como se puede observar en la siguiente figura:

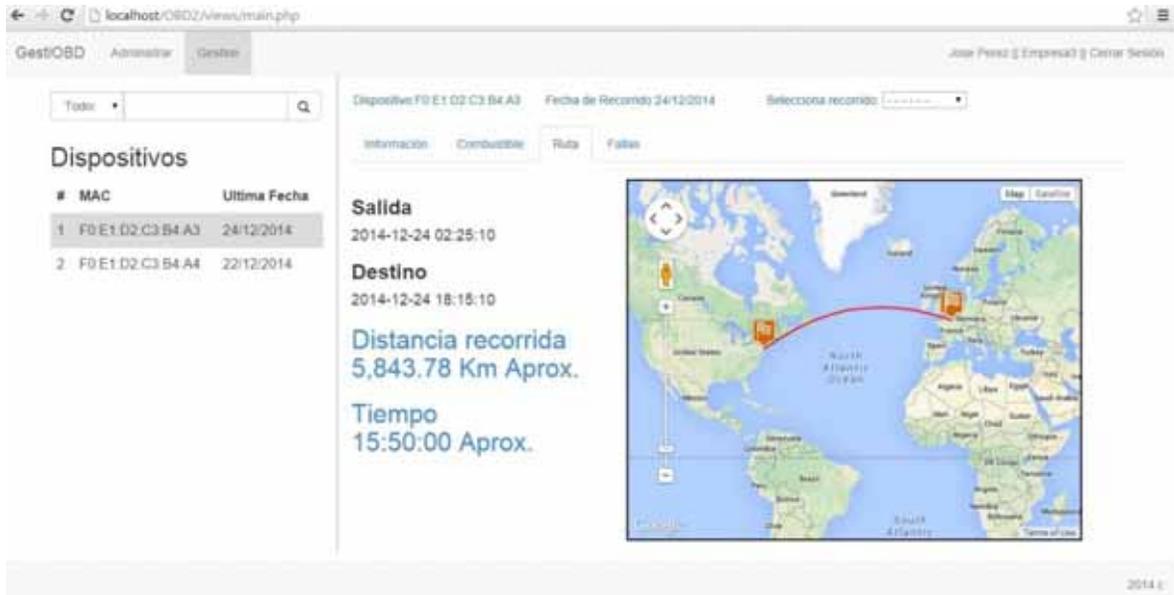


Fig.5.3.3d. Vista de la pestaña ruta en gestion.php

Por último en *contenido.php* se tiene la pestaña referente a las fallas; en esta parte solo se utiliza el método *getDTC(id del recorrido)* el cual obtiene el código de error así como la localización y las causas del mismo; del catálogo dtc de la base de datos.



Fig.5.3.3e. Vista de la pestaña fallas en gestion.php

### 5.3.4 Información adicional

Para cerrar sesión por medio de javascript se debe de confirmar la acción; al ser aceptada se redirige a login.php; esto se hace en el script controlador *salir.php*.

La clase *myconn.php* es fundamental en la aplicación ya que ésta está presente en todos los scripts que necesitan realizar operaciones en la base de datos; aquí se realiza la conexión y se establecen los parámetros para la misma como lo es el servidor, usuario, password y base de datos a utilizar; esta clase abstrae todo lo relacionado con la base de datos; y una de las ventajas es que con solo modificar los parámetros se puede conectar a otra base de datos sin tener que tocar ningún script de la aplicación; así como, si se desea migrar a otro manejador solo se requeriría adaptar esta clase y la aplicación no tendría ningún cambio. En esta clase se realizan las operaciones de selección, inserción, actualización y eliminación de información en la base de datos. Otra clase utilizada para el desarrollo de la aplicación es la clase *misc* la cual se encuentra dentro de *myconn.php*; de ésta solo se utiliza el método *imprime* (lo que se desea imprimir) ya que en php la función *print* si llegara a haber un error retorna un valor y sería más fácil encontrarlo.

Para el uso de AJAX ya se tiene un script llamado *ajax.js* el cual contiene una función para hacer más simple el uso de esta técnica, esta función llamada *MostrarConsulta* (dirección donde se va a hacer la manipulación de los datos enviados por el método GET, id del objeto que se va a ver afectado) la cual trabaja de manera asincrónica; se encuentra en este script la función *MostrarConsulta2* (dirección donde se va a hacer la manipulación de los datos enviados por el método GET, id del objeto que se va a ver afectado) que trabaja de manera síncrona; la cual no fue utilizada durante la implementación de la aplicación.

API Google Maps.

Para utilizar de manera eficiente la API se debe de tener una cuenta de google; una vez iniciada la sesión en google se debe acceder a la consola de desarrollo de google <https://console.developers.google.com/project> ; seleccionar el proyecto, después seleccionar API de la opción APIs y Autenticación; buscar Google Maps JavaScript API v3 y activar. Una vez activada seleccionar la opción *credenciales de APIs y Autenticación* y se podrá observar la key para poder tener información del uso que se le da a la API.

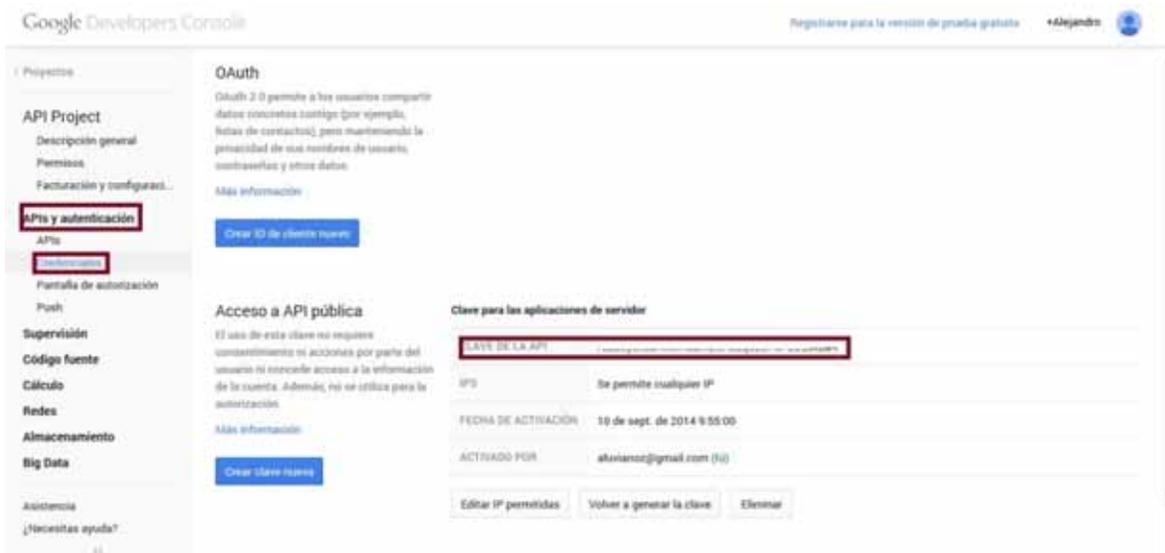


Fig.5.3.4 Obtención de la clave de API.

## 6. Conclusión

Una aplicación de esta naturaleza requiere seguir una metodología para poder ser implementada adecuadamente; en este caso se siguió el modelo incremental puesto que a pesar de tener claros los requerimientos del sistema, al estar buscando información sobre aspectos técnicos y teóricos se encuentran “funcionalidades extras” que tal vez pudiera contener la aplicación y pudiera generar retraso en los tiempos de entrega; a pesar de ello esto implica una mejor implementación de la aplicación.

Durante la implementación es donde realmente se observa el alcance que tuvo la etapa de análisis y desarrollo; en este caso estas etapas fueron bien implementadas; a pesar de eso se debieron ir adaptando partes de código que no se tenían contempladas (uso de clases externas, usar un iframe para la ruta, etc)

La aplicación tiene una interfaz intuitiva y de fácil manejo; gracias a la librería bootstrap la cual ayuda demasiado al desarrollo de la misma; ya que es relativamente fácil su implementación; gracias a la documentación en línea de los desarrolladores de la librería.

Se observó la gran importancia del lenguaje Javascript en el desarrollo de la aplicación; ya que este lenguaje permite una interfaz más funcional y atractiva para el usuario final; considero que en una aplicación web lo esencial es HTML, CSS y javascript ya que es lo que se ejecuta del lado del cliente y hay que tener cuidado en la forma de adaptar todos estos lenguajes al análisis y diseño del sistema; en este caso en particular se dejó un poco de lado esa parte; habría que poner más énfasis en todas las acciones de los objetos de la página y la interacción entre ellos.

Para la realización de la aplicación se optó por seguir la tendencia que se tiene ahora en la WEB lo cual implica dejar de lado las redirecciones de página en cada enlace, módulo u opción de la aplicación; lo que era tedioso al momento de estar utilizando las aplicaciones; de aquí la importancia de AJAX lo cual ayuda a no tener que estar refrescando la pantalla en cada petición que se uiera del cliente; si no únicamente refrescar el contenido deseado; esto ayuda a que sea mejor la usabilidad del sistema.

Cabe destacar que la aplicación desarrollada en este proyecto aún tiene más funcionalidades a implementar; que se realizarán en posteriores iteraciones de la misma; por motivos de tiempo de entrega no se pudieron considerar esos cambios; de ahí la importancia de poner más énfasis en el modelo de desarrollo y la aplicación del mismo para considerar todos los aspectos que pudieran presentarse.

## 7. Bibliografía

- Larman, Craig. *UML y patrones: Una introducción al análisis y diseño orientado a objetos y al proceso unificado*. Segunda edición. Madrid: Pearson Educación, 2003.
- Pressman, Roger. *Ingeniería del Software un enfoque práctico*. Quinta edición. Madrid: Mc Graw Hill, 2002.
- Abraham Silberschatz, Henry Korth, S. Sudarshan. *Fundamentos de bases de datos*. Cuarta edición. Madrid: Mc Graw Hill, 2002.
- Lopez Quijado, Jose. *Domine PHP y MySQL: Programación dinámica en el lado del servidor*, Segunda edición. Editorial Alfaomega, 2007.
- *Desarrollo iterativo y creciente*. Colaboradores de Wikipedia. 21 de Noviembre de 2014.  
[http://es.wikipedia.org/w/index.php?title=Desarrollo\\_iterativo\\_y\\_creciente&oldid=78272176](http://es.wikipedia.org/w/index.php?title=Desarrollo_iterativo_y_creciente&oldid=78272176) [consulta:02/01/2015]
- *OBD (ON BOARD DIAGNOSTIC)*. Aficionados a la mecánica. 2014.  
<http://www.aficionadosalamecanica.com/obd2.htm> [consulta:04/01/2015]
- *Versión 3 del API de JavaScript de Google Maps*. Enero 31, 2013.  
<https://developers.google.com/maps/documentation/javascript/tutorial?hl=es> [consulta:28/11/2014]
- *Using PHP/MySQL with Google Maps*, Ben Appleton, Google Geo Team With contributions from Lary Stucker, Maps API Developer April 2007.  
[https://developers.google.com/maps/articles/phpsqlajax\\_v3?hl=es](https://developers.google.com/maps/articles/phpsqlajax_v3?hl=es) [consulta:30/11/2014]
- CSS. @mdo and @fat, v3.3.1. <http://getbootstrap.com/css/> [consulta:28/10/2014]
- Components. @mdo and @fat, v3.3.1. <http://getbootstrap.com/components/> [consulta:28/10/2014]
- Javascript. @mdo and @fat, v3.3.1. <http://getbootstrap.com/javascript/> [consulta:28/10/2014]
- *Introducción a JavaScript*. Javier Eguiluz. <http://librosweb.es/javascript/> [consulta:04/11/2014]

- *Introducción a CSS*. Javier Eguiluz. <http://librosweb.es/javascript/>  
[consulta:10/11/2014]
- *Collection of HowTo's*. Asial Corporation. <http://jpgraph.net/doc/howto.php>  
[consulta: 26/12/2014]

## 8. Entregables

El código fuente de la aplicación; como se mencionó anteriormente sigue el Modelo-Vista-Controlador, el cual está organizado de la siguiente manera:



El código fuente se encuentra en el CD de entrega del proyecto con el nombre **OBD2.zip** organizado de la forma mostrada anteriormente; también se incluye el script de la base de datos en MySQL en formato zip con nombre **appobd2script.zip**.

Cabe mencionar que las carpetas css, js y fonts son de la librería bootstrap v3; la carpeta jpgraph es de la librería jpgraph v3, las cuales están presentes en el archivo comprimido.