



**UNIVERSIDAD  
DE ANTIOQUIA**

**IMPLEMENTACIÓN DE UN SISTEMA PARA LA  
GESTIÓN, CONTROL Y VIGILANCIA DEL  
TRANSPORTE ESCOLAR EN LA CIUDAD (TEO  
KIDS – FASE 1)**

Autor(es)

Richard Stiveen Morales Cardona

Universidad de Antioquia

Departamento de Ingeniería de Sistemas

Medellín, Colombia

2020



IMPLEMENTACIÓN DE UN SISTEMA PARA LA GESTIÓN, CONTROL Y  
VIGILANCIA DEL TRANSPORTE ESCOLAR EN LA CIUDAD (TEO KIDS – FASE 1)

**Richard Stiveen Morales Cardona**

Tesis o trabajo de investigación presentada(o) como requisito parcial para optar al título de:

**Pregrado en Ingeniería de Sistemas**

Asesores (a):

Sandra Patricia Zabala Orrego, Ingeniera Informática

Universidad de Antioquia  
Departamento de Ingeniería de Sistemas  
Medellín, Colombia  
2020.

## Tabla de contenido

Resumen .....	3
Introducción.....	3
Objetivos.....	3
Objetivo general.....	3
Objetivos Específicos .....	3
Marco teórico.....	4
Metodología.....	5
Resultados y análisis.....	5
Figura 1. Diagrama del modelo de base de datos generado con SQLServer Management 17 .....	6
Figura 2. Arquitectura general del sistema .....	7
Figura 3. Arquitectura de 3 capas del API.....	8
Figura 4. TEODriver instalado en Android .....	8
Figura 5. Arquitectura MVVM.....	8
Figura 6. Icono de la aplicación .....	8
Figura 7. Pantalla de ingreso.....	9
Figura 8. Pantalla principal .....	10
Figura 9. Pantalla de creación de rutas .....	10
Figura 10. Pantalla de lista de rutas .....	11
Figura 11. Pantalla de lista de TEOS .....	11
Figura 12. Pantalla de asignación de TEOS a TEO Driver .....	12
Figura 13. Pantalla de lista de vehículos.....	12
Figura 14. Pantalla de lista de vehículos.....	13
Figura 15. Pantalla de lista de notificaciones .....	13
Figura 16. Pantalla de generación de notificaciones.....	14
Conclusiones.....	14
Referencias .....	15

## **Resumen**

Por medio de la metodología ágil SCRUM se pudo implementar un sistema para gestionar, controlar y vigilar el transporte escolar en la ciudad, inicialmente del lado del conductor. Dicho sistema está integrado por una aplicación móvil implementada con la plataforma de desarrollo Xamarin, una API Web usando el framework ASP.Net Core y una base de datos utilizando el motor de base de datos SQL Server.

## **Introducción**

NEWSOFT S.A.S. es una empresa con más de 20 años de experiencia en el sector del desarrollo y consultoría de sistemas informáticos en innumerables empresas del país, con proyectos de talla nacional e internacional.

Uno de los enfoques de la empresa, además de los servicios que ya presta, es invertir y promover en proyectos propios que no solo impactan empresas, sino también a la sociedad directamente y es aquí justamente donde nace el proyecto TEO KIDS.

Dicho proyecto surge desde la necesidad de los padres de la ciudad de tener vigilancia y control sobre las rutas escolares de sus hijos y asimismo de los conductores de dichas rutas de tener la información centralizada y un sistema de gestión y comunicación efectivo para mantener informados a los padres sobre el estado de las rutas.

La solución propuesta, llamada “Teo Kids”, cubre dicha necesidad por medio de la implementación un sistema móvil para la gestión logística de las rutas escolares, todo esto por medio del desarrollo de un sistema íntegro haciendo uso de tecnologías reconocidas en la industria como SQL Server, Asp.Net Core y Xamarin, y el marco de trabajo para desarrollo ágil de software SCRUM.

## **Objetivos**

### **Objetivo general**

Implementar un sistema para gestionar, controlar y vigilar el transporte escolar en la ciudad. (Para esta fase del proyecto solo del lado del conductor, llamado TEO Driver).

### **Objetivos Específicos**

- Evaluar y levantar los requerimientos necesarios para implementar esta fase del proyecto con el product owner.
- Realizar las historias de usuario asociadas a los requerimientos de esta fase del proyecto.
- Analizar el modelo de base de datos conveniente para el proyecto.

- Implementar el modelo de base de datos en la tecnología SQL Server, conveniente para el proyecto.
- Implementar una API (Application Programming Interface) en la tecnología Asp.Net Core para esta fase del proyecto.
- Implementar una aplicación móvil multiplataforma funcional para el TEO Driver (conductor), en la tecnología Xamarin basados en el alcance de esta fase del proyecto.

## **Marco teórico**

A la hora de contratar servicios de transporte escolares encontramos algunas falencias por la falta de un sistema de gestión efectivo, mediante el cual padres y transportadores estén sincronizados en los aspectos que surgen de la contratación de dicho servicio. Por lo general en la prestación de dicho servicio se presentan algunos inconvenientes o riesgos como los siguientes: los conductores de rutas escolares manejan a memoria sin tener rutas registradas en algún sistema; los transportadores llevan en la memoria el registro de los escolares que transportan; los padres no tienen fácil acceso a la información referente a las rutas de sus hijos; no hay una forma efectiva de notificar a todos los padres sobre alguna novedad que pueda suceder en la vía, por ejemplo un trancón que retrasará la llegada del escolar; los padres no conocen las rutas que usa el transportador para llevar a sus hijos a su colegio. Por lo anterior, se evidencia una oportunidad de mejora en la prestación de dicho servicio por medio de smartphone y el uso del internet, herramientas tecnológicas a la mano en la mayoría de la población actual.

Partiendo de esto, se desarrolló una aplicación móvil que permite a padres y transportadores estar informados acerca de todo lo que implica el transporte de los escolares. Dicho proyecto se llevará a cabo por fases, siendo la que se describe en este documento la fase 1, que es el desarrollo de una solución funcional para la gestión del proceso de transporte por parte de los transportadores (llamados en la aplicación TEO Drivers).

Para la implementación de la solución móvil y de la API se utilizó el entorno de desarrollo integrado (IDE) Microsoft Visual Studio 2017. Para la aplicación móvil se utilizó la plataforma de desarrollo Xamarin. Dicha plataforma permite desarrollar en lenguaje C# todo el sistema. La gran ventaja de esta tecnología es que podemos construir aplicaciones nativas para iOS, Android y Windows, con un único código base compartido .NET. Además del código compartido podemos desarrollar nativamente para cada plataforma en específico. Por otro lado, para el desarrollo de la web API se usó el framework ASP.Net Core que cuenta con una amplia gama de librerías y herramientas para el desarrollo de aplicaciones de servidor seguras. Por último para la implementación de la base de datos que sustentará al sistema se usó SQL Server que es un potente sistema de gestión de bases de datos relacionales que nos brinda todas las herramientas necesarias.

## Metodología

La implementación de la solución se llevó a cabo haciendo uso del marco de trabajo para desarrollo ágil de software llamado SCRUM. El rol de “product owner” lo desempeñó el señor Jhon Fredy García Martínez, gerente de la empresa.

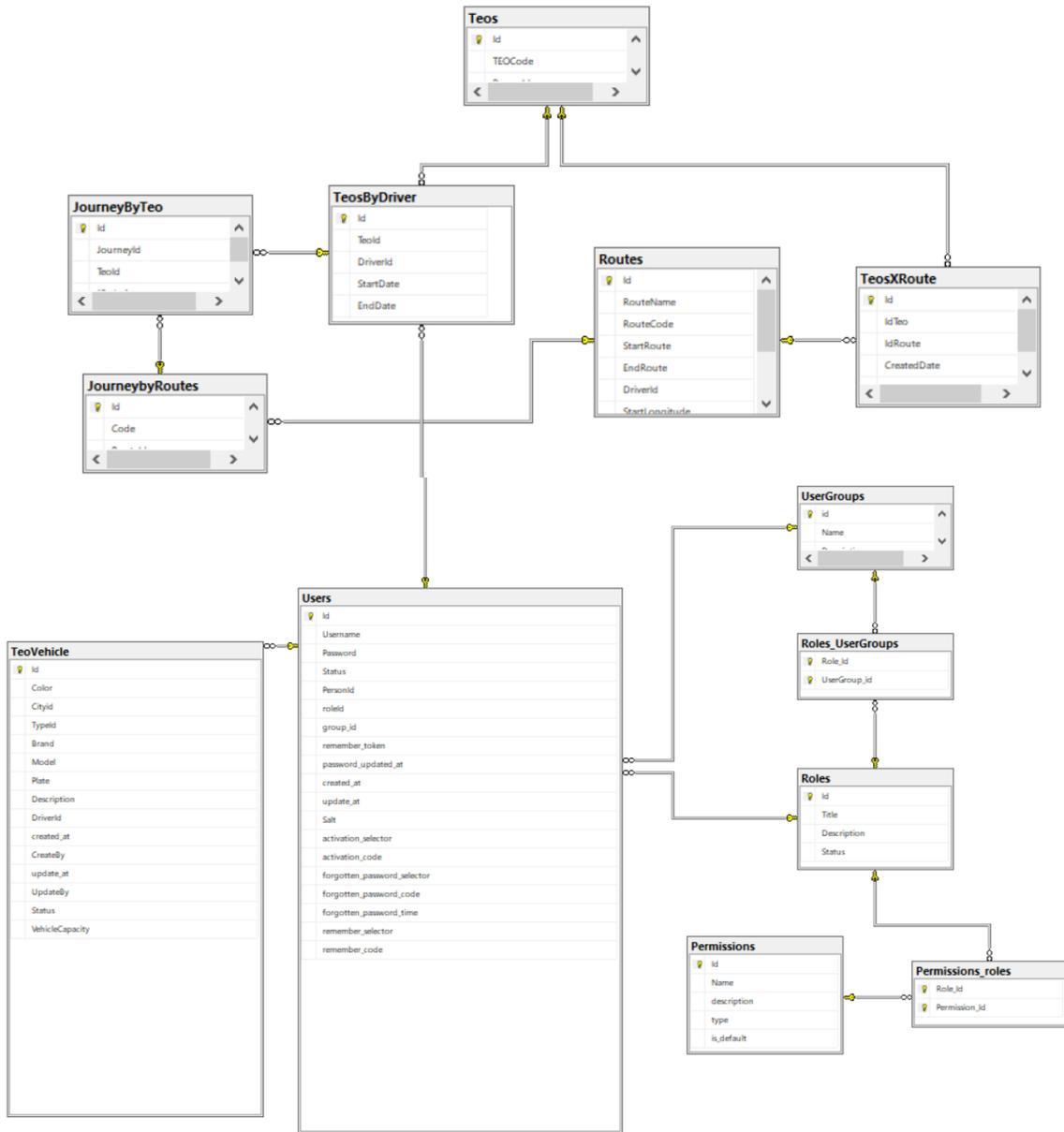
Dicha metodología estipula una serie de iteraciones llamadas “Sprints”, los cuales se dieron en lapsos de 2 semanas.

Además al final de cada sprint se realizó una reunión de seguimiento para validar el trabajo realizado en el lapso que duró la iteración.

Sprint	Inicio	Final	Septiembre				Octubre				Noviembre				Diciembre			
Sprint 1	02/09/2019	15/09/2019	■	■														
Sprint 2	16/09/2019	29/09/2019			■	■												
Sprint 3	30/09/2019	13/10/2019					■	■										
Sprint 4	14/10/2019	27/10/2019						■	■									
Sprint 5	28/10/2019	10/11/2019								■	■							
Sprint 6	11/11/2019	24/11/2019										■	■					
Sprint 7	25/11/2019	08/12/2019												■	■			
Sprint 8	09/12/2019	22/12/2019														■	■	

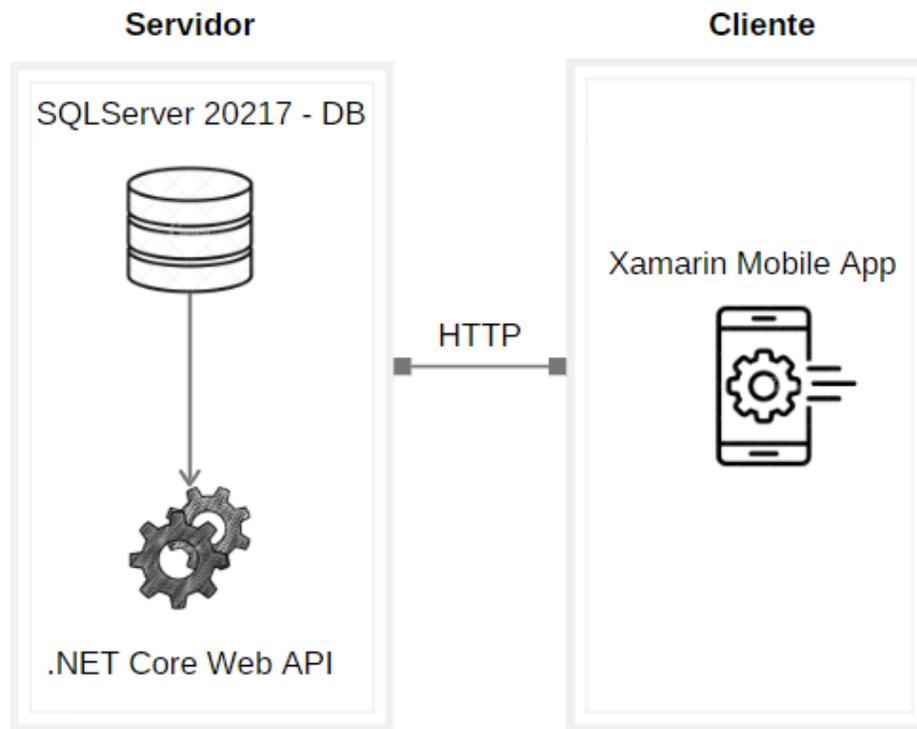
## Resultados y análisis

El modelo de base de datos obtenido después de su implementación fue el presentado en la Figura 1.



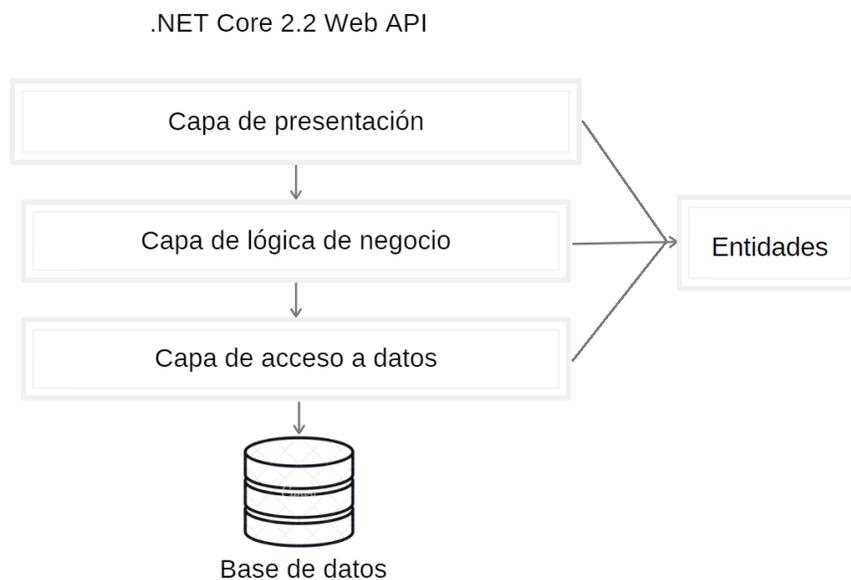
**Figura 1. Diagrama del modelo de base de datos generado con SQLServer Management 17**

Adicionalmente se planteó la arquitectura general del sistema la cual sigue un modelo cliente-servidor como se puede ver en la Figura 2.



**Figura 2. Arquitectura general del sistema**

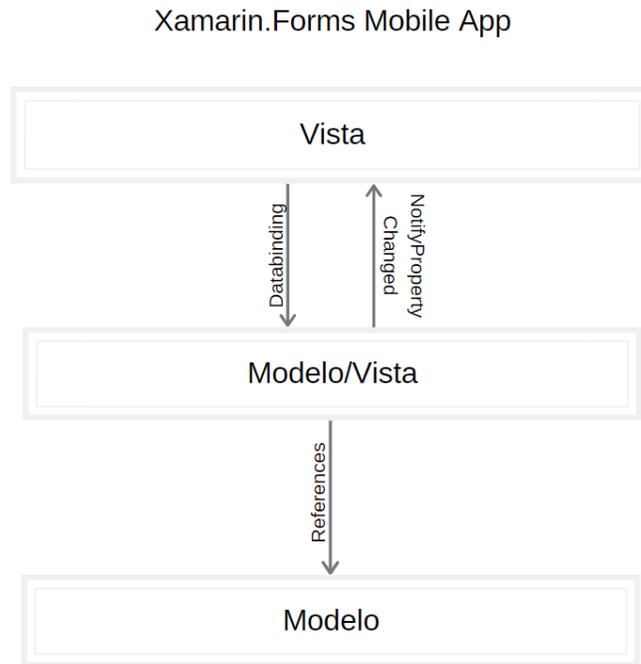
Por otro lado, se formuló la arquitectura de cada una de las partes: servidor y cliente. Para el servidor se propuso una arquitectura en 3 capas (ver Figura 3.) las cuales fueron: presentación; lógica de negocio y acceso a datos. Se agregó también una capa transversal a todas que contiene las entidades manejadas en el sistema.



### Figura 3. Arquitectura de 3 capas del API

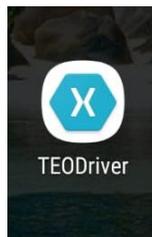
El cliente (la aplicación móvil) fue trabajado bajo el patrón arquitectónico MVVM (Modelo-Vista-Modelo de vista), que es el propuesto para aplicaciones Xamarin.

### Figura 4. TEODriver instalado en Android



### Figura 5. Arquitectura MVVM

La aplicación se instala en el S.O. móvil Android como TEODriver. A continuación se muestra una pantalla de la aplicación en el menú. Figura 6.



### Figura 6. Icono de la aplicación

A continuación se dispondrán algunas pantallas de la aplicación móvil y su respectiva explicación (el código fuente no se pone aquí por temas de confidencialidad)

## Pantalla de ingreso

En esta pantalla se debe ingresar un nombre usuario registrado en la plataforma y su respectiva contraseña, para luego presionar el botón “Ingresar”. En caso tal de que el usuario no tenga cuenta puede registrarse presionando el botón “Registrar”.

The image shows a mobile application interface for user login and registration. It is divided into two main sections: 'INICIAR SESIÓN?' (Login) and 'Registro' (Registration).

**INICIAR SESIÓN? (Login):**

- Fields: 'Usuario o Correo' and 'Password'.
- Buttons: '[ LOGIN ]' and '[ REGISTER ]'.

**Registro (Registration):**

- Header: 'Fomulario de registro'.
- Disclaimer: '\* Todos los campos son obligatorios'.
- Section: 'Registro' with subtext: 'Guardaremos tus datos de una manera segura para brindar siempre una mejor experiencia'.
- Section: 'Datos personales' with fields:
  - Nombres: 'Escribe tu nombre(s)'
  - Apellidos: 'Escribe tus apellidos'
  - Tipo Documento: (empty field)
  - Número Documento: 'Escribe número documento'
  - Número Contacto: (empty field)
- Section: 'Correo electrónico:' with field: 'Escribe tu correo'.
- Section: 'Usuario:' with field: 'Escribe tu usuario'.
- Section: 'Contraseña:' with subtext 'Debe tener al menos 8 caracteres con números y letras' and field: 'Escribe tu contraseña'.
- Section: 'Repetir Contraseña:' with field: 'Repite tu contraseña'.
- Footer: 'Para conocer a acerca de nuestros terminos y condiciones has click [aquí](#)' and a red 'REGISTRATE' button.

**Figura 7. Pantalla de ingreso**

## Pantalla principal

Esta es la pantalla de navegación principal. Desde aquí se puede navegar a los distintos módulos implementados hasta el momento. Además, se encuentra en la parte superior derecha la opción de “Salir”.



**Figura 8. Pantalla principal**

Aquí se pueden crear nuevas rutas asociadas al TEO Driver.

**Crea tu nueva ruta**

Aquí podrás establecer una ruta definiendo el punto donde deseas empezar y donde deseas finalizar junto con los teos que deseas que estén asociados a esta ruta

**Punto de inicio:**  
Punto de inicio

**Punto de llegada:**  
Punto de llegada

**Región:**  
Nombre Región

**País:**  
Nombre País

**Ciudad:**  
Nombre de Ciudad

**Nombre de tu ruta:**  
Escribe el nombre de la ruta

**GUARDAR RUTA**

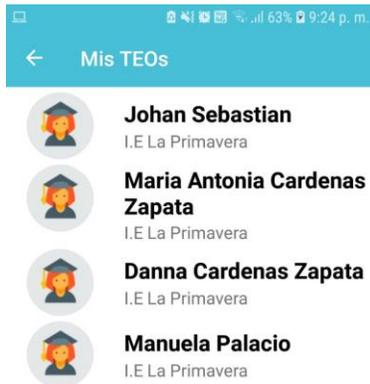
**Figura 9. Pantalla de creación de rutas**

Aquí se listan las rutas creadas por un TEO Driver específico.



**Figura 10. Pantalla de lista de rutas**

Aquí se listan los TEOS asociados, por código, a un TEO Driver específico.



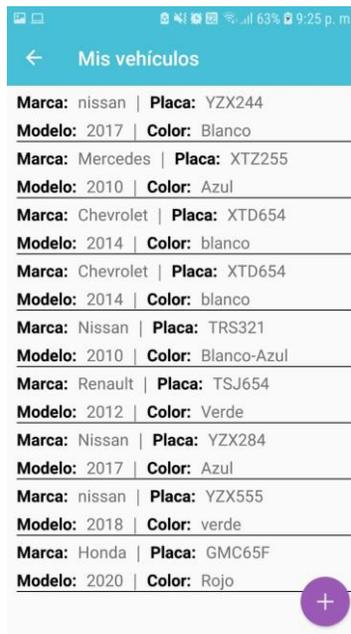
**Figura 11. Pantalla de lista de TEOS**

Aquí se pueden asignar por medio de un código, un TEO (escolar) a un TEO Driver.



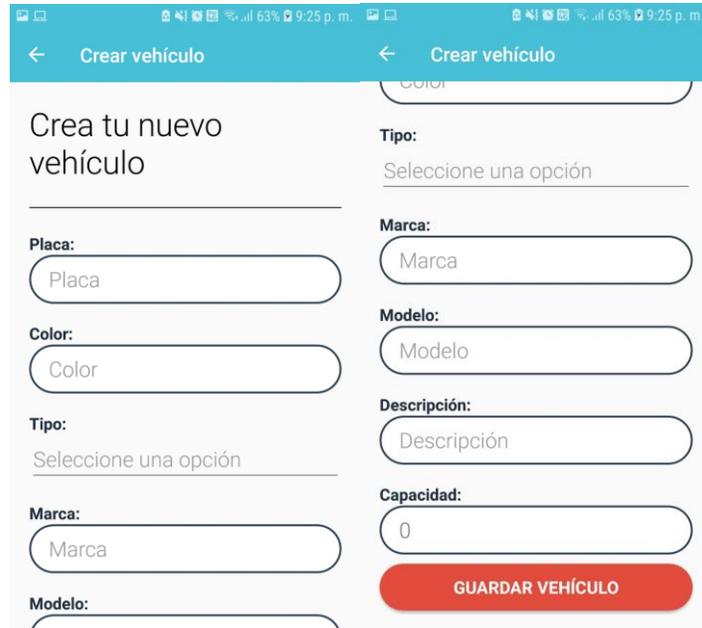
**Figura 12. Pantalla de asignación de TEOS a TEO Driver**

Aquí se listan los vehículos creadas por un TEO Driver específico.



**Figura 13. Pantalla de lista de vehículos**

Aquí se pueden crear nuevos vehículos asociados a un TEO Driver.



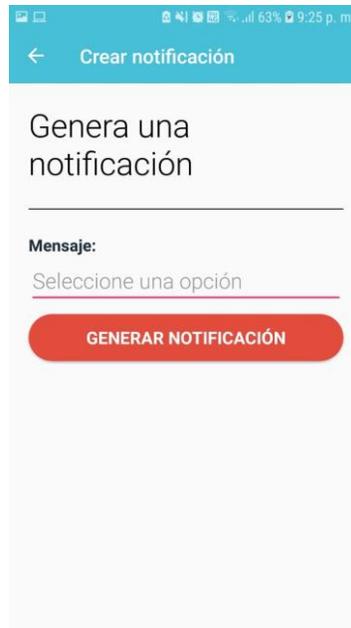
**Figura 14. Pantalla de lista de vehículos**

Aquí se listan las notificaciones generadas por un TEO Driver específico.



**Figura 15. Pantalla de lista de notificaciones**

Aquí un TEO Driver puede generar notificaciones de alerta o informativas que en una fase posterior del proyecto podrán ser visualizadas por los TEO Family.



**Figura 16. Pantalla de generación de notificaciones**

## **Conclusiones**

- La metodología SCRUM implementada para el desarrollo del proyecto fue ideal debido a que fue un proyecto cuyas reglas de negocio cambiaron eventualmente por lo que a la hora de realizar ajustes se ha podido notificar y ejecutar rápidamente.
- El proceso de práctica empresarial permite afianzar los conocimientos adquiridos durante la carrera.
- Tener un método ayuda a no perder el rumbo y optimizar el tiempo.

## Referencias

- [1] Microsoft (2019) ASP.NET Documentation. Recuperado de: <https://docs.microsoft.com/en-us/aspnet/core/?view=aspnetcore-2.2>
- [2] Microsoft (2019) Xamarin Documentation. Recuperado de: <https://docs.microsoft.com/en-us/xamarin/>
- [3] Microsoft (2019) SQL Server Documentation. Recuperado de: <https://docs.microsoft.com/en-us/sql/sql-server/sql-server-technical-documentation?view=sql-server-2017>
- [4] Sinnaps (2019) Metodología SCRUM. Recuperado de: <https://www.sinnaps.com/blog-gestion-proyectos/metodologia-scrum>