

**SISTEMA DE INFORMACIÓN DE REGISTRO, CONTROL Y GESTIÓN DEL
TRANSPORTE ESCOLAR UTILIZANDO LA TECNOLOGÍA ARDUINO GPRS
CON PLATAFORMA WEB EN JAVA.**

SANTIAGO DELGADO MEJIA

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FISICOMECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA**

2018

**SISTEMA DE INFORMACIÓN DE REGISTRO, CONTROL Y GESTIÓN DEL
TRANSPORTE ESCOLAR UTILIZANDO LA TECNOLOGÍA ARDUINO GPRS
CON PLATAFORMA WEB EN JAVA.**

SANTIAGO DELGADO MEJIA

**Trabajo de Grado para optar al título de
Ingeniero de Sistemas**

Director:

FERNANDO ANTONIO ROJAS MORALES

M.Sc. Ingeniero de Sistemas

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FISICOMECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA**

2018

DEDICATORIA

Dedico este trabajo de grado a mi madre María Eugenia por siempre estar ahí cuando la necesité, apoyándome de corazón en todo lo que necesité y brindándome su amor incondicional.

A mi padre José Antonio por ser siempre un motivador para que nunca desfalleciera en el logro de mis objetivos y sueños como lo es graduarme profesionalmente.

A mi hermana Nathalia por toda la paciencia y consideración que tuvo conmigo en momentos difíciles y sobre todo por el apoyo incondicional que siempre me brinda en todos los objetivos que me propongo.

A mi hermano Cristian Fernando por estar pendiente de cada cosa que necesité y por apoyarme en cada decisión de mi vida.

Santiago Delgado Mejia

TABLA DE CONTENIDO

	Pág.
INTRODUCCIÓN	15
1. DESCRIPCIÓN DEL PROYECTO	17
1.1. PLANTEAMIENTO Y JUSTIFICACIÓN DEL PROBLEMA	17
1.2. OBJETIVOS	18
1.2.1. Objetivo general	18
1.2.2. Objetivos específicos	18
2. MARCO TEÓRICO	19
2.1. PLATAFORMA Y CONCEPTOS CLAVES PARA ARDUINO	19
2.1.1. Arduino	19
2.1.2. Microcontrolador	19
2.1.3. Memoria ROM	19
2.1.4. Memoria RAM	19
2.1.5. Arduino IDE (Integrated development environment)	20
2.1.6. Tecnología RFID	20
2.2. ARQUITECTURA DE SOFTWARE	20
2.2.1. Modelo vista controlador (MVC)	20
2.3. CONCEPTOS CLAVES Y LENGUAJES PARA EL DESARROLLO WEB ...	21
2.3.1. JEE (Java Enterprise Edition)	21
2.3.2. Servlets	22
2.3.3. EJB (Enterprise Java Bean)	22
2.3.4. Persistencia	22
2.3.5. JPA (Java Persistence API)	22
2.3.6. POJO (Plain Old Java Object)	23
2.3.7. ORM (Object Relational Mapping)	23
2.4. FRAMEWORKS	23
2.4.1. JSF (Java Server Faces)	24
2.4.2. Primefaces	24
2.4.3. JDBC (Java Database Connectivity)	24
2.4.4. Hibernate	24
3. ESTADO DEL ARTE	25
3.1. ONTRACK SCHOOL	25
3.2. TU RUTA ESCOLAR	25
4. METODOLOGÍA	26
4.1. PRIMERA ETAPA (ANÁLISIS)	27
4.2. SEGUNDA ETAPA (DISEÑO RÁPIDO)	28
4.3. TERCERA ETAPA (ESTUDIO DE CONOCIMIENTOS PREVIOS)	28
4.4. CUARTA ETAPA (CONSTRUCCIÓN DEL PROTOTIPO)	28
4.5. QUINTA ETAPA (EVALUACIÓN DEL PROTOTIPO)	28

4.6. SEXTA ETAPA (REFINAMIENTO DEL PROTOTIPO).....	29
4.7. SÉPTIMA ETAPA (PRODUCTO FINAL)	29
5. DESARROLLO DEL PROYECTO.....	30
5.1. CARACTERÍSTICAS DE LA APLICACIÓN	30
5.2. REQUISITOS	31
5.2.1. Requisitos funcionales	31
5.2.2. Requisitos no funcionales	36
5.3. DIAGRAMA ENTIDAD RELACIÓN	37
5.4. CASOS DE USO	38
5.5. ARQUITECTURA DE OBJETOS.....	42
5.5.1. Diagrama de clases	44
5.5.2. Diccionario de clases	44
5.6. PLANEACIÓN Y ELECCIÓN DE FRAMEWORKS	50
5.7. PROTOTIPOS	52
5.7.1. Primer prototipo.....	52
5.7.2. Segundo prototipo.....	55
5.7.3. Tercer prototipo.....	62
5.7.4. Prototipo final o producto final.....	64
6. PRUEBAS DE FUNCIONALIDAD.....	83
6.1. PRUEBAS DEL PRIMER PROTOTIPO	83
6.2. PRUEBAS DEL SEGUNDO PROTOTIPO	83
6.3. PRUEBAS DEL TERCER PROTOTIPO.....	84
6.4. PRUEBAS DEL PRODUCTO FINAL.....	85
7. CONCLUSIONES	87
8. RECOMENDACIONES	88
BIBLIOGRAFÍA.....	90
ANEXOS.....	94

LISTA DE TABLAS

Tabla 1. Características Usuario no registrado	30
Tabla 2. Características Usuario. Administrador	30
Tabla 3. Características Usuario. Acudiente	30
Tabla 4. Características Usuario. Conductor	31
Tabla 5. Requisito Funcional 01. Validar Usuario	32
Tabla 6. Requisito Funcional 02. Validar información según el usuario	32
Tabla 7. Requisito Funcional 03. Registrar usuario	32
Tabla 8. Requisito Funcional 04. Consultar estado del estudiante	33
Tabla 9. Requisito Funcional 05. Crear existencias	33
Tabla 10. Requisito Funcional 06. Modificar y eliminar registros	33
Tabla 11. Requisito Funcional 07. Gestionar información de estudiantes	34
Tabla 12. Requisito Funcional 08. Información pertinente a conductores.....	34
Tabla 13. Requisito Funcional 09. Información de estudiantes inscritos a algún conductor.....	34
Tabla 14. Requisito Funcional 10. Filtrar información	35
Tabla 15. Requisito Funcional 11. Información editable del conductor	35
Tabla 16. Requisito Funcional 12. Información editable del acudiente	35
Tabla 17. Requisito no funcional 01. Seguridad de inicio	36
Tabla 18. Requisito no funcional 02. Disponibilidad de información	36
Tabla 19. Requisito no funcional 03. Obtención de estado de los estudiantes	37
Tabla 20. Casos de uso del usuario administrador	39
Tabla 21. Casos de uso del usuario acudiente	39
Tabla 22. Casos de uso del usuario conductor	40
Tabla 23. Atributos Usuario.....	45
Tabla 24. Métodos Usuario	45
Tabla 25. Atributos Administrador.....	46
Tabla 26. Atributos Acudiente	46
Tabla 27. Métodos Acudiente	46
Tabla 28. Atributos Conductores.....	47
Tabla 29. Métodos Conductor	47
Tabla 30. Atributos Rol	48
Tabla 31. Atributos Estudiante	48
Tabla 32. Atributos Bus.....	49
Tabla 33. Atributos Ruta	49
Tabla 34. Atributos GestionarTransporte	50
Tabla 35. Características de frameworks escogidos.....	51
Tabla 36. Pruebas primer prototipo.....	83
Tabla 37. Pruebas segundo prototipo	83
Tabla 38. Pruebas tercer prototipo.....	84

Tabla 39. Pruebas producto final85

LISTA DE FIGURAS

Figura 1. POJO	23
Figura 2. Diagrama Entidad-Relación	38
Figura 3. Diagrama de casos de uso del administrador.....	40
Figura 4. Diagrama de casos de uso del conductor	41
Figura 5. Diagrama de casos de uso del acudiente	41
Figura 6. Diagrama de clases	44
Figura 7. Modelo entidad relación. Primer prototipo	53
Figura 8. Inicio de sesión. Primer prototipo.....	53
Figura 9. Ventana Administrador Primer Prototipo.....	54
Figura 10. Ventana Acudiente. Primer prototipo	54
Figura 11. Ventana Conductor. Primer prototipo.....	55
Figura 12. Registro de usuarios. Segundo prototipo.....	56
Figura 13. Crear Usuario. Segundo prototipo	57
Figura 14. Ventana estudiantes del Administrador. Segundo prototipo	57
Figura 15. Modificar usuario. Segundo prototipo	58
Figura 16. Eliminar usuario. Segundo prototipo	59
Figura 17. Conductores desde ventana Acudiente. Segundo prototipo	60
Figura 18. Estudiantes de un conductor. Segundo prototipo	61
Figura 19. Estudiantes desde acudiente. Segundo prototipo.....	62
Figura 20. Inicio de sesión. Tercer prototipo	63
Figura 21. Hijos de acudiente. Tercer prototipo	63
Figura 22. Perfil del conductor. Tercer prototipo	64
Figura 23. Inicio de sesión. Producto Final	65
Figura 24. Registrarse. Producto Final	65
Figura 25. Seguridad de acceso. Producto final	66
Figura 26. Ventana principal administrador. Producto final.....	66
Figura 27. Usuario administrador. Producto final	67
Figura 28. Acudiente Administrador. Producto final	67
Figura 29. Conductores Administrador. Producto final.....	68
Figura 30. Seleccionar usuario. Producto final	69
Figura 31. Crear usuario. Producto final	69
Figura 32. Modificar usuario. Producto final.....	70
Figura 33. Eliminar Usuario. Producto final.....	70
Figura 34. Estudiantes de Administrador. Producto final	71
Figura 35. Registrar acudiente. Producto final	72
Figura 36. Buses de Administrador. Producto final	72
Figura 37. Rutas de Administrador. Producto final.....	73
Figura 38. Registrar conductor. Producto final.....	74
Figura 39. Historial de estudiantes. Administrador. Producto final.....	74

Figura 40. Ventana Principal Acudiente. Producto final	75
Figura 41. Hijos de Acudientes. Producto final	75
Figura 42. Conductores de Acudiente. Producto final.....	76
Figura 43. Modificar hijos de Acudientes. Producto final	76
Figura 44. Historial hijos. Producto final.....	77
Figura 45. Ventana principal Conductor. Producto final	77
Figura 46. Estudiantes de conductores. Producto final.....	78
Figura 47. Perfil Conductor. Producto final	78
Figura 48. Buses de Conductor. Producto final.....	79
Figura 49. Rutas de conductor. Producto final	79
Figura 50. Modificar buses de Conductor. Producto Final	80
Figura 51. Modificar Rutas de conductor. Producto Final	80
Figura 52. Crear bus Conductores. Producto Final.....	81
Figura 53. Crear ruta de Conductor. Producto Final	81
Figura 54. Historial de estudiantes de Conductor. Producto Final	82
FIGURA 55. Conexiones Arduino	94
FIGURA 56. Interacción RFID.....	95
FIGURA 57. Banda de frecuencias.....	96
FIGURA 58. RFID-RC522 conexión con Arduino uno	97
FIGURA 59. Jumper TX(D7) y RX(D8)	98
FIGURA 60. Diagrama de conexión entre Arduino uno y SIM900	99

LISTA DE ANEXOS

Anexo A. Conexiones Arduino	94
Anexo B. Lector RFID-RC522.....	95
Anexo C. SIM 900 GPRS/GSM.....	98

RESUMEN

TÍTULO: SISTEMA DE INFORMACIÓN DE REGISTRO, CONTROL Y GESTIÓN DEL TRANSPORTE ESCOLAR UTILIZANDO LA TECNOLOGÍA ARDUINO GPRS CON PLATAFORMA WEB EN JAVA*

AUTOR: SANTIAGO DELGADO MEJIA**

PALABRAS CLAVE: FRAMEWORKS, SISTEMA DE INFORMACIÓN, DESARROLLO WEB, ARDUINO, RFID, BASE DE DATOS.

DESCRIPCIÓN:

El servicio de transporte escolar en Colombia es un servicio bastante usado por los padres de familia que tienen niños pequeños o menores de edad; ya que es un problema de tiempo llevar a los hijos a sus respectivos colegios y de manera inversa, llevarlos del colegio a sus casas. Hay veces que algunos padres de familia pueden llevar o acompañar a los niños a coger la buseta que los va a llevar al colegio, pero muchas otras veces el padre no puede realizar este acompañamiento. Debido a esto se genera una deficiencia en el sistema del transporte escolar, ya que la mayoría de las veces los padres no se enteran si sus hijos llegaron a tiempo y bien al colegio o si por el contrario llegaron bien a sus casas.

Con el objetivo de mejorar o hacer desaparecer esta deficiencia, se decidió realizar este proyecto de aplicación web combinada con la tecnología Arduino que se encarga de registrar automáticamente mediante una lectura de un tag RFID cada vez que los niños se suban al bus escolar y cada vez que se bajen, con el fin de obtener el estado actual del estudiante y de esta manera mantener informado todo el tiempo al padre de familia, al colegio y hasta a los conductores de en donde se encuentra cada niño. Esta información se encuentra contenida en una aplicación web que va a poder ser consultada por las entidades o personas mencionadas anteriormente, cada vez que lo requieran.

* Trabajo de grado

** Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingeniería de Sistemas e Informática. Director: MSc. Fernando Antonio Rojas Morales

ABSTRACT

TITLE: SCHOOL TRANSPORTATION, CONTROL AND MANAGEMENT INFORMATION SYSTEM USING ARDUINO GPRS TECHNOLOGY WITH WEB PLATFORM IN JAVA.*

AUTHOR: SANTIAGO DELGADO MEJIA**

KEYWORDS: FRAMEWORKS, INFORMATION SYSTEM, WEB DEVELOPMENT, ARDUINO, RFID, DATABASE.

DESCRIPTION:

The school transport service in Colombia is a service that is widely used by parents who have small children or minors, since carrying out children to their schools and their homes is a problem of time. There are times that some parents can take or accompany the children to take the bus that will take them to school, but many times the father cannot carry out this accompaniment. Due to this, a deficiency in the school transport system is generated, since most of the time the parents do not find out if their children arrived on time and well at school or if, on the contrary, they arrived safely at home.

With the aim of improving or eliminating this deficiency, it was decided to carry out this web application project combined with the Arduino technology that is responsible for automatically registering by reading an RFID tag every time children get on the school bus and every time to get off, in order to obtain the current status of the student and thus keep informed at all times to the parent, the school and even the drivers of where each child is. This information is contained in a web application that can be consulted by the entities or persons mentioned above, whenever they require it.

* Bachelor Thesis

** Faculty of Physicomechanical Engineering. School of Systems Engineering and Informatics. Director: MSc. Fernando Antonio Rojas Morales.

INTRODUCCIÓN

Gracias a los colegios o escuelas que hay en el lugar en donde cada uno vive, miles de niños son educados correctamente esperando que estos crezcan con valores y conocimientos básicos, los cuales aporten a que estos niños sean ciudadanos de bien en un futuro cercano y que puedan en base a esos conocimientos primarios que obtuvieron en estos colegios, poder desarrollarse laboralmente en el mundo en el que viven. Un tema a tener en cuenta en la educación de estos niños por parte de los padres de familia es la rutina diaria del desplazamiento de estos niños de la casa a su colegio y viceversa; muchos de los padres de estos niños no tienen la posibilidad de ir a llevar a sus hijos al colegio y luego recogerlos, para llevarlos a su casa, por diversos asuntos.

En este punto es donde toma real importancia el tema del transporte de sus hijos, por lo cual nació el servicio del transporte escolar, que se encarga de llevar a sus niños desde cada una de sus casas a sus colegios, y de manera contraria, del colegio a cada una de sus casas de manera segura.

Sin embargo, en la actualidad se han presentado en el país una serie de episodios que involucran a los colegios y al transporte escolar en situaciones bochornosas y hasta peligrosas para el cuidado de los niños que transportan y para la tranquilidad de los padres de familia de estos.

Desafortunadamente se han visto noticias en las que no deja bien parado al sector de transporte escolar y en la que pone en duda a los padres de familia de la seguridad de este servicio, ya que por reiteradas ocasiones se han visto casos en los que el niño se queda dentro de la buseta y el conductor no se percata de esto, situación en la que se inmediatamente se produce angustia por parte de los padres o acudientes del menor al no saber en dónde se encuentra su hijo.

Por esta problemática, se decidió desarrollar una aplicación web que permita en todo momento a los padres de familia y al colegio saber en dónde se encuentran

sus hijos mediante un registro automático de cada vez que los niños se suban y se bajen de la buseta. De esta manera podrán saber si a su hijo ya los recogió el transporte escolar, si ya los dejó en el colegio, o si ya los dejó en sus casas. Además de esta información también podrán tener acceso a información pertinente, como los datos personales del conductor del transporte de sus hijos, para comunicarse con ellos si en debido caso es necesario.

1. DESCRIPCIÓN DEL PROYECTO

1.1. PLANTEAMIENTO Y JUSTIFICACIÓN DEL PROBLEMA

El transporte escolar en el mundo es un servicio que definitivamente ha ayudado en gran medida a los padres de familia que no tienen tiempo para llevar a sus hijos al colegio; este servicio que inicio como una ayuda para las familias, se ha convertido en un servicio esencial y de gran importancia para que los niños puedan ir de forma segura a sus colegios y regresar igualmente a casa sin que los padres tengan que emplear tiempo valioso en este proceso.

Principalmente el transporte escolar está dirigido para niños desde pre jardín hasta bachillerato, pero para los que mayor beneficio trae es para los niños de pre jardín hasta primaria, ya que, en este período, los niños están muy pequeños y se hace complicado enviarlos en transporte público por su seguridad. Sin embargo, últimamente se han visto noticias de niños que se quedan dentro del transporte escolar y no llegan a casa por descuido del transportador, incluso se han visto casos en los que los niños se montan en otro transporte escolar y el conductor ni se da cuenta; todos estos casos invitan a una mejora en el sistema del transporte aprovechando la tecnología con el objetivo principal de reducir riesgos y así gozar de un servicio totalmente seguro y eficaz para los niños.

El nuevo sistema de transporte escolar consiste en que cada niño quede registrado tecnológicamente cada vez que ingrese o salga del transporte, mediante una tarjeta Arduino que lea y envíe la información a una base de datos compartida en la web de que la persona ingresó o salió; para así llevar un registro y un control de que efectivamente el niño se montó en el transporte o en el caso contrario se dejó en la casa o colegio. Este sistema soluciona de raíz el problema del descuido del transportador, haciendo que sea mucho más seguro mandar a los hijos al colegio sin temer que se pierdan o que se olviden del niño.

Este registro puede ser consultado por los padres o colegio de manera inmediata en la web, haciendo que los padres estén constantemente informados de si su hijo llegó al colegio o fue dejado en su casa.

1.2. OBJETIVOS

1.2.1. Objetivo general. Desarrollar una aplicación para gestionar el registro y control de usuarios de transporte escolar en tiempo real, utilizando tecnología Web y Arduino.

1.2.2. Objetivos específicos

- Determinar las características esperadas en una aplicación de registro y control en línea de usuarios de transporte escolar.
- Diseñar una arquitectura de objetos que den soporte a la aplicación, de acuerdo con las características determinadas.
- Ponderar opciones de implementación mediante frameworks que permitan realizar un desarrollo ágil del aplicativo web.
- Implementar la persistencia de los objetos de la aplicación mediante el mapeo objeto-relacional (ORM).
- Estudiar y aplicar la tecnología Arduino para el realizar el registro automático de entrada y salida del transporte escolar.
- Evaluar mediante pruebas la funcionalidad del aplicativo web con el fin de comprobar si se cumple el objetivo de mejorar el sistema de transporte escolar.

2. MARCO TEÓRICO

2.1. PLATAFORMA Y CONCEPTOS CLAVES PARA ARDUINO

2.1.1. Arduino. Arduino es una placa electrónica o microcontrolador de código abierto que permite programar sobre este módulo de una manera fácil y eficiente mediante un IDE de tal modo que se pueda interactuar con el mundo físico por medio de sensores, dispositivos y otras placas electrónicas compatibles.

2.1.2. Microcontrolador. Un microcontrolador es un circuito integrado que es programable. Lo componen una unidad central de proceso (CPU), las memorias ROM y RAM, líneas de entrada y líneas de salida o periféricos.¹

2.1.3. Memoria ROM. La memoria ROM (Read Only Memory) es una memoria de lectura en la que se guardan los programas que se ejecutan en un ordenador o microcontrolador. No necesita de corriente eléctrica y su principal característica es que no se borra lo que hay dentro de ella cuando se apaga el sistema²

2.1.4. Memoria RAM. La memoria RAM (Random Access Memory) es una memoria volátil, es decir los datos que se guardan en esta no se guardan de manera permanente, si no que se pierde la información cuando no es alimentada por una fuente de energía. En esta memoria se almacenan las instrucciones que se dan desde la CPU (Unidad Central de Procesamiento).³

¹ SHERLIN.XBOT.ES. 1. ¿Qué es un microcontrolador? [en línea]. [Consultado: 22 de Mayo de 2017]. Disponible en: <http://sherlin.xbot.es/microcontroladores/introduccion-a-los-microcontroladores/que-es-un-microcontrolador>

² VIALFA, Carlos. Memoria de solo lectura (ROM). [en línea]. [Consultado: 22 de Mayo de 2017]. Disponible en: <http://es.ccm.net/contents/399-memoria-de-solo-lectura-rom>

³ ABC TECNOLOGÍA. ¿Qué es y por qué es importante la memoria RAM?. [en línea]. [Consultado: 19 de Mayo de 2017]. Disponible en: <http://www.abc.es/tecnologia/consultorio/20150129/abci-memoria-RAM-que-es-para-que-sirve-201501281425.html>

2.1.5. Arduino IDE (Integrated development environment). Como su propia definición lo dice es un entorno de desarrollo integrado que contiene un editor de código, un compilador y un depurador en el que podemos escribir código o programar para que la placa física realice lo que queramos por medio de sensores, leds y diferentes dispositivos o placas que le conectemos.⁴

2.1.6. Tecnología RFID. La tecnología RFID (Radio Frequency Identification) es una tecnología que permite la identificación mediante un lector de una etiqueta o tag único que dispone una persona o elemento. Sus usos varían desde identificación de productos en un supermercado a identificación de personas y animales.⁵

2.2. ARQUITECTURA DE SOFTWARE

2.2.1. Modelo vista controlador (MVC). La arquitectura de software de modelo vista controlador tiene la intención de separar los datos y la lógica de negocio, es decir separa en tres módulos los datos, la vista o interfaz de usuario y por último un módulo desde donde se controla los datos que son mostrados en la vista. Esto lo proponen los gurús de la ingeniería del software para hacer más robusto el software y más ordenado de manera que al hacerle mantenimiento sea más fácil y se pueda reutilizar código.⁶

⁴ APRENDIENDO ARDUINO. Qué es Arduino. [en línea]. [Consultado: 17 de Mayo de 2017]. Disponible en: <https://aprendiendoarduino.wordpress.com/2016/09/25/que-es-arduino/>

⁵ BY. ¿Qué es RFID?. [en línea]. [Consultado: 19 de Mayo de 2017]. Disponible en: <https://www.by.com.es/blog/que-es-rfid>

⁶ ALVAREZ, Miguel Angel. Qué es MVC. [en línea]. [Consultado:22 de Mayo de 2017]. Disponible en: <https://desarrolloweb.com/articulos/que-es-mvc.html>

2.2.1.1. Modelo. En este módulo se presentan los datos de la aplicación, así como su lógica de negocio o reglas del juego por llamarlo de una manera coloquial y también puede haber mecanismos de persistencia. Por tanto, contiene métodos o mecanismos para acceder a la información y para actualizarla.

2.2.1.2. Vista. Es la capa o módulo que se le presenta al usuario, es decir en esta capa está diseñada toda la interfaz gráfica que es capaz de ver el usuario que utiliza el software.

2.2.1.3. Controlador. Es la capa que comunica a la capa de modelo y a la capa de vista, haciéndola la responsable de gestionar o controlar el flujo de información que se presentan entre estas.

2.3. CONCEPTOS CLAVES Y LENGUAJES PARA EL DESARROLLO WEB

2.3.1. JEE (Java Enterprise Edition). Es un conjunto de tecnologías para el desarrollo de aplicaciones empresariales o software que está presente en servidores. Se caracterizan por ser aplicaciones portables, robustas, escalables y seguras. JEE proporciona una arquitectura multicapa, es decir, que puede estar constituida por aplicaciones de escritorio o aplicaciones web en navegadores. Algunas de las tecnologías que se presentan son Servlets, JSP, JSF, EJB, JMS o Web Services. El objetivo de estas tecnologías es reducir el tiempo de desarrollo y la complejidad de la programación.⁷

⁷ ESPECIALISTA UNIVERSITARIO EN JAVA ENTERPRISE EDITION. ¿Qué es Java Enterprise?. [en línea]. [Consultado: 22 de Mayo de 2017]. Disponible en: <http://www.jtech.ua.es/j2ee/2006-2007/jee.html>

2.3.2. Servlets. Los servlets son módulos que se escriben en lenguaje JAVA que sirven para que el usuario pueda interactuar con la aplicación, de tal manera que el usuario le pida una petición desde un navegador web a la aplicación y el servlet sea el encargado de procesar esta petición y devolver la respuesta al usuario mediante el navegador web.⁸

2.3.3. EJB (Enterprise Java Bean). Es un componente de negocio de JEE que da acceso a los servicios de un contenedor EJB (manejo de transacciones, persistencia, seguridad, entre otras), lo que permite que el programador no se preocupe de todos aspectos que pueden suceder en la aplicación.⁹

2.3.4. Persistencia. Como su propia palabra lo dice, la persistencia es el principio de mantenerse en el tiempo, es decir, cuando se dice que los datos son persistentes, se quiere decir que no desaparecen y que por el contrario están ahí para cuando se les necesite.¹⁰

2.3.5. JPA (Java Persistence API). JPA o el API de persistencia de JAVA es un conjunto de métodos y clases que se encargan de que la base de datos relacional se correlacione con la programación orientada a objetos de forma persistente. Esta correlación es llamada ORM (Object Relational Mapping) la cual genera anotaciones sobre entidades o clases POJOS.¹¹

⁸ BARRIOS, Juan Manuel. Java Servlets. [en línea]. [Consultado: 22 de Mayo de 2017]. Disponible en: <https://users.dcc.uchile.cl/~jbarrios/servlets/general.html>

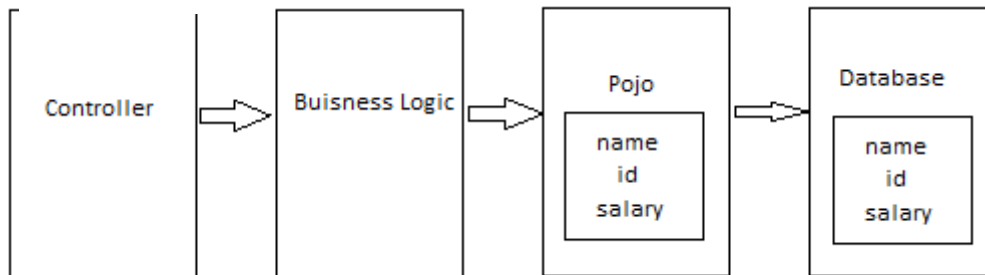
⁹ PESQUERA, Carlos. ¿QUÉ ES UN POJO, EJB Y UN BEAN?. [en línea]. [Consultado: 22 de Mayo de 2017]. Disponible en: <http://carlospesquera.com/que-es-un-pojo-ejb-y-un-bean>

¹⁰ LIBROS WEB. 11.7. Persistencia de datos. [en línea]. [Consultado: 22 de Mayo de 2017]. Disponible en: http://librosweb.es/libro/algoritmos_python/capitulo_11/persistencia_de_datos.html

¹¹ ORACLE JUNIORS. ¿Qué es JPA (Java Persistence API)?. [en línea]. [Consultado: 22 de Mayo de 2017]. Disponible en: <http://oraclejuniors.blogspot.com.co/2014/11/que-es-jpa-java-persistence-api.html>

2.3.6. POJO (Plain Old Java Object). Un POJO en resumidas cuentas es una clase con atributos y métodos que nos encapsulan esos atributos de tal manera que podamos acceder a estos. Un POJO no puede extender, ni implementar otras clases.¹²

Figura 1. POJO



Fuente: GeeksforGeeks. POJO vs Java Beans. [en línea]. [Consultado: 22 de Mayo de 2017]. Disponible en: <https://www.geeksforgeeks.org/pojo-vs-java-beans>

2.3.7. ORM (Object Relational Mapping). ORM o Mapeo de objeto relacional es el modelo de la programación para convertir la base de datos relacional en entidades, las cuales le permiten al programador olvidarse de la base de datos y centrarse en la programación orientada a objetos; ya que las tablas de la base de datos se convierten en objetos, el acceso y modificación de estas tablas, se pueden hacer a nivel JAVA o de un lenguaje de alto nivel orientado a objetos.¹³

2.4. FRAMEWORKS

Los frameworks son herramientas o software en general que se le da al programador para que lo utilice en cada uno de sus proyectos, permitiendo reutilización de código y permitiendo hacer las cosas de una manera más simple.¹⁴

¹² PESQUERA, Carlos. ¿QUÉ ES UN POJO, EJB Y UN BEAN?. [en línea]. [Consultado: 22 de Mayo de 2017]. Disponible en: <http://carlospesquera.com/que-es-un-pojo-ejb-y-un-bean>

¹³ TUPROGRAMACION.COM. ¿Qué es un ORM?. [en línea]. [Consultado: 25 de Mayo de 2017]. Disponible en: <http://www.tuprogramacion.com/glosario/que-es-un-orm/>

¹⁴ GUTIERREZ, Javier J. ¿Qué es un framework web?. [en línea]. [Consultado: 25 de Mayo de 2017]. Disponible en: http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf

2.4.1. JSF (Java Server Faces). Java Server Faces es un framework del lado del servidor que utiliza la arquitectura de software MVC para aplicaciones JAVA que permite al programador simplificar el código para realizar una interfaz de usuario en una aplicación web. JSF proporciona componentes en forma de etiquetas en páginas XHTML.¹⁵

2.4.2. Primefaces. Es una librería de componentes de código abierto que complementa a JSF. Dentro de sus componentes, están: DataTable, Charts, HtmlEditor, InputText, entre otros.¹⁶

2.4.3. JDBC (Java Database Connectivity). JDBC o Conectividad con base de datos JAVA, es un API que, por medio de clases e interfaces escritas en JAVA, permite la conexión a bases de datos relacionales independientemente del lenguaje de la base de datos.¹⁷

2.4.4. Hibernate. Hibernate es un framework para JAVA que simplifica el mapeo de objeto relacional, es decir, convierte las tablas de una base de datos relacional en POJOS sin necesidad de escribir código. Este proceso lo hace mediante archivos XML.¹⁸

¹⁵ UNIVERSIDAD DE ALICANTE. Introducción a JavaServer Faces. [en línea]. [Consultado: 25 de Mayo de 2017]. Disponible en: <http://www.jtech.ua.es/j2ee/publico/jsf-2012-13/sesion01-apuntes.html>

¹⁶ CALENDAMIA. PrimeFaces: framework sobre JSF 2.0. Primeros pasos. [en línea]. [Consultado: 25 de Mayo de 2017]. Disponible en: <https://www.genbetadev.com/frameworks/primefaces-framework-sobre-jsf-2-0-primeros-pasos>

¹⁷ YERKO MUÑOZ, Felipe Fernández. JDBC. [en línea]. [Consultado: 25 de Mayo de 2017]. Disponible en: <https://users.dcc.uchile.cl/~lmateu/CC60H/Trabajos/jfermand>

¹⁸ VERGARA JURADO, Alida. JPA vs Hibernate ¿Cual es la diferencia?. [en línea]. [Consultado: 22 de Mayo de 2017]. Disponible en: <https://www.facilcloud.com/noticias/jpa-vs-hibernate>

3. ESTADO DEL ARTE

3.1. ONTRACK SCHOOL

OnTrack School es una plataforma que permite tener un control total por parte del transportador, colegio y padres de familia de las rutas y ubicación del transporte y estudiantes a todo momento.

Las funciones de la plataforma para el transportador, es saber exactamente en qué sitios debe recoger estudiantes mediante un mapa con el que cuentan, además pueden enviar notificaciones a los padres de familia de que van a llegar o se van a demorar.

Los padres de familia y el colegio pueden seguir la ruta que recorre el transporte escolar donde van los estudiantes, además reciben notificaciones de si el estudiante se dejó en el colegio o en las respectivas casas.

On track School cuenta con una aplicación móvil y una página web para gestionar y observar todo lo relacionado con los transportes escolares.¹⁹

3.2. TU RUTA ESCOLAR

Es una aplicación móvil que permite a los padres y colegios monitorear desde un computador, un. a Tablet o celular el recorrido del bus escolar que lleva a los niños. Además de poder ver en donde va el transporte, también pueden ver y escuchar lo que pasa al interior del transporte.²⁰

¹⁹ ONTRACK. ONTRACK SCHOOL. [en línea]. [Consultado:01 Junio de 2017]. Disponible en: <https://ontrack.global/services/school>

²⁰ TU RUTA ESCOLAR. Tu Ruta Escolar. [en línea]. [Consultado: 01 de Junio de 2017]. Disponible en: <http://www.turutaescolar.com/portal/index.php/features>

4. METODOLOGÍA

Una metodología es un modelo o estructura de trabajo que se plantea para desarrollar algún sistema o proyecto. Ahora bien, una metodología de desarrollo de software consiste en aplicar herramientas, métodos, modelos, estructuras o técnicas para desarrollar un software; una metodología de desarrollo de software puede constar de un ciclo de trabajo o de etapas bien definidas.

Existen varios tipos de metodologías que se pueden escoger dependiendo del desarrollo de software o proyecto que se quiera realizar, pero todas estas están basadas principalmente en los siguientes enfoques de desarrollo:²¹

- Modelo en cascada: Framework lineal.
- Prototipado: Framework iterativo.
- Incremental: Combinación de framework lineal e iterativo.
- Espiral: Combinación de framework lineal e iterativo.
- RAD: Rapid Application Development, framework iterativo.

De acuerdo con esto, para este proyecto se ha decidido usar una combinación de la metodología de prototipado con la metodología incremental ayudando así a un desarrollo ágil por medio de prototipos y de ciclos que permite verificar y cerciorarse de que el producto final satisface los requerimientos del proyecto.

El modelado de prototipos tiene las siguientes cinco etapas:

- Recolección y refinamiento de requisitos

²¹ UNIVERSITAT DE BARCELONA. ¿Qué son las metodologías de desarrollo de software?. [en línea]. [Consultado: 01 de Junio de 2017]. Disponible en: <https://www.obs-edu.com/int/blog-project-management/metodologia-agile/que-son-las-metodologias-de-desarrollo-de-software>

- Diseño rápido
- Construcción del prototipo
- Evaluación del prototipo
- Refinamiento del prototipo
- Producto final

Por otro lado, la metodología incremental consiste en ir entregando incrementos de un producto, es decir, cada incremento es una pequeña versión de alguna funcionalidad del producto. Al igual que el modelado de prototipos tiene unas etapas. Estas etapas son:

- Análisis
- Diseño
- Código
- Prueba

Como anteriormente se dijo que se iba a utilizar una combinación de las dos metodologías, a continuación, vamos a definir exactamente cómo se va a combinar estas dos.

4.1. PRIMERA ETAPA (ANÁLISIS)

En esta etapa se analiza los objetivos que tiene el proyecto, con el fin de tener claro lo que se desea realizar, para de esta manera levantar todos los requisitos que son necesarios para desarrollar el software. Se debe comprender el ámbito de la información del software, así como la función, el rendimiento y las interfaces requeridas.

4.2. SEGUNDA ETAPA (DISEÑO RÁPIDO)

En esta fase se descompone y se organiza el sistema en elementos que puedan elaborarse por componentes o por separado, lo que permita una simple identificación y realización de cada uno. Básicamente el software se divide en la estructura de los datos, la arquitectura del software, el detalle procedimental y la caracterización de la interfaz; permitiendo tener claro cada uno de los componentes del software y tecnologías a usar para el desarrollo de este.

4.3. TERCERA ETAPA (ESTUDIO DE CONOCIMIENTOS PREVIOS)

Esta etapa es muy importante dentro del desarrollo del proyecto, ya que es la parte donde se obtienen conocimientos previos de los lenguajes y frameworks que se va a usar para la etapa de codificación. Esta fase es necesaria ya que no se tienen conocimientos de ningún framework, del desarrollo de aplicaciones web con java, de la programación con Arduino, del concepto persistencia con la base de datos, entre otras cosas.

4.4. CUARTA ETAPA (CONSTRUCCIÓN DEL PROTOTIPO)

Es la fase del proyecto donde se implementa el código fuente. Dependiendo del lenguaje de programación y su versión se crean las bibliotecas y componentes reutilizables dentro del mismo proyecto para hacer que la programación sea un proceso mucho más rápido y ágil.

4.5. QUINTA ETAPA (EVALUACIÓN DEL PROTOTIPO)

En esta etapa se verifica que todo el software esté funcionando de manera correcta, y que a su vez el software desarrollado, cumpla con los objetivos planteados al principio del desarrollo del prototipo.

4.6. SEXTA ETAPA (REFINAMIENTO DEL PROTOTIPO)

Esta es el punto de arrancada del siguiente incremento o prototipo refinado, es decir, en esta fase observamos que nos hace falta para terminar el proyecto por completo y que necesidades no se han terminado de cubrir; según esto volvemos a la primera etapa para recorrer nuevamente todas las fases de la metodología y así tener al final un proyecto totalmente terminado y satisfactorio para el cliente.

4.7. SÉPTIMA ETAPA (PRODUCTO FINAL)

Después de haber hecho varios incrementos asegurándose de que el producto sea lo que el cliente estaba esperando y de haber alcanzado los objetivos propuestos al inicio del desarrollo del proyecto, esta etapa finalmente es la de la entrega del software o producto que se había requerido.

5. DESARROLLO DEL PROYECTO

5.1. CARACTERÍSTICAS DE LA APLICACIÓN

Tabla 1. Características Usuario no registrado

Usuario	No registrado
Características	
El usuario no registrado lo único que puede hacer si desea acceder a información, es registrarse.	

Tabla 2. Características Usuario. Administrador

Usuario	Registrado	Rol	Administrador
Características			
Usuario que pertenece al colegio y tiene el cargo de administrador de la plataforma de registro y control de estudiantes en transporte escolar. Dentro de las funciones a realizar, se encuentran:			
<ul style="list-style-type: none">• Consultar información referente a usuarios, estudiantes, acudientes y conductores.• Modificar información de usuarios, estudiantes, conductores y acudientes.• Eliminar información de usuarios, estudiantes, conductores y acudientes.• Crear existencias de usuarios, estudiantes, conductores y acudientes.			

Tabla 3. Características Usuario. Acudiente

Usuario	Registrado	Rol	Acudiente
Características			
Usuario que tiene a un estudiante o a varios matriculados en el colegio. Además de tener estudiantes en el colegio, también utiliza el servicio de transporte escolar			

que brinda colegio. Algunas de las funciones que se le permiten realizar dentro de la aplicación son:

- Consultar información pertinente a su hijo/as que tenga registrado en la aplicación
- Modificar información pertinente al acudiente y al hijo/as
- Consultar un listado de los conductores que se encuentran inscritos en el colegio con sus rutas y toda la información pertinente a estos

Tabla 4. Características Usuario. Conductor

Usuario	Registrado	Rol	Conductor
Características			
<p>Usuario que presta el servicio de transporte escolar en el colegio. Algunas de las funciones que puede realizar dentro de la aplicación son:</p> <ul style="list-style-type: none"> • Consultar un listado de los estudiantes que tiene a su cargo en el transporte escolar con información pertinente a ellos y a su acudiente. • Consultar y modificar información de él mismo, así como de su bus escolar y sus rutas 			

5.2. REQUISITOS

5.2.1. Requisitos funcionales. Los requisitos funcionales se refieren a todas las características y restricciones que pueda tener el sistema y que a su vez satisfaga las necesidades del cliente. A continuación, se mostrarán los requisitos funcionales que tiene el sistema.

Tabla 5. Requisito Funcional 01. Validar Usuario

Nombre	Validar usuario		
ID	RF 01	Prioridad	Alta
Usuarios	Usuario registrado		
Descripción			
La aplicación permite hacer una validación de tres tipos de usuarios: Administrador, acudiente y conductor			

Tabla 6. Requisito Funcional 02. Validar información según el usuario

Nombre	Validar información según tipo de usuario		
ID	RF 02	Prioridad	Alta
Usuarios	Usuario registrado		
Descripción			
Permite según el tipo de usuario que ingrese, hacer una clasificación de la información pertinente a cada uno			

Tabla 7. Requisito Funcional 03. Registrar usuario

Nombre	Registrar usuario		
ID	RF 03	Prioridad	Alta
Usuarios	Usuario no registrado		
Descripción			
Permite el registro de personas que no cuentan con un usuario en la aplicación			

Tabla 8. Requisito Funcional 04. Consultar estado del estudiante

Nombre	Consultar estado del estudiante		
ID	RF 04	Prioridad	Alta
Usuarios	Usuario registrado		
Descripción			
Permite consultar la información de los cuatro posibles estados (Del bus al colegio, en el colegio, del bus a la casa o en la casa) del niño, por parte de los tres tipos de usuario.			

Tabla 9. Requisito Funcional 05. Crear existencias

Nombre	Crear existencias		
ID	RF 05	Prioridad	Media
Usuarios	Administrador		
Descripción			
Permite que el administrador cree existencias de estudiantes, acudientes y conductores			

Tabla 10. Requisito Funcional 06. Modificar y eliminar registros

Nombre	Modificar y eliminar registros		
ID	RF 06	Prioridad	Media
Usuarios	Administrador		
Descripción			
Así de la misma manera que el administrador puede crear existencias, también puede modificar y eliminar acudientes, conductores y estudiantes			

Tabla 11. Requisito Funcional 07. Gestionar información de estudiantes

Nombre	Gestionar información de estudiantes		
ID	RF 07	Prioridad	Media
Usuarios	Acudiente		
Descripción			
Permite a los acudientes poder consultar el estado del hijo o los hijos que tenga inscritos en el colegio, así como poder editar información pertinente a estos.			

Tabla 12. Requisito Funcional 08. Información pertinente a conductores

Nombre	Información de Conductores		
ID	RF 08	Prioridad	Media
Usuarios	Acudiente		
Descripción			
Permite a los acudientes consultar un listado de todos los conductores con su información pertinente como número de celular, ruta, bus entre otras cosas.			

Tabla 13. Requisito Funcional 09. Información de estudiantes inscritos a algún conductor

Nombre	Información de estudiantes inscritos a conductores		
ID	RF 09	Prioridad	Media
Usuarios	Conductor		
Descripción			
El conductor va a tener un listado de los estudiantes inscritos a su transporte			

Tabla 14. Requisito Funcional 10. Filtrar información

Nombre	Filtrar información		
ID	RF 10	Prioridad	Baja
Usuarios	Usuarios registrados		
Descripción			
Se permite a los usuarios hacer uso de un filtro para algunos campos de cierta información			

Tabla 15. Requisito Funcional 11. Información editable del conductor

Nombre	Información editable del conductor		
ID	RF 11	Prioridad	Baja
Usuarios	Conductor		
Descripción			
Los conductores podrán editar información pertinente a ellos y a sus buses escolares y rutas.			

Tabla 16. Requisito Funcional 12. Información editable del acudiente

Nombre	Información editable del acudiente		
ID	RF 12	Prioridad	Baja
Usuarios	Acudiente		
Descripción			
Los padres de familia solo podrán editar información pertinente a los hijos			

5.2.2. Requisitos no funcionales. Los requisitos no funcionales son todas las características que son visibles para el usuario, pero que además no tiene nada que ver con las funciones que tenga que tener el software. Dentro de los aspectos o características que pueden ser no requisitos funcionales están: La apariencia de la interfaz de usuario, temas de tiempo de respuesta de la aplicación, temas de seguridad y confiabilidad, temas de rendimiento, entre otros aspectos. A continuación, se muestran los requisitos no funcionales de la aplicación²²

Tabla 17. Requisito no funcional 01. Seguridad de inicio

Nombre	Seguridad de inicio		
ID	RF 01	Prioridad	Alta
Usuarios	Usuario no registrado		
Descripción			
Existe una validación de seguridad de tal modo que se tenga prohibido el ingreso de una persona ajena a la información, y que no se tenga permitido el ingreso a una página por medio de la URL sin haberse logueado.			

Tabla 18. Requisito no funcional 02. Disponibilidad de información

Nombre	Disponibilidad de información		
ID	RF 02	Prioridad	Alta
Usuarios	Usuario registrado		
Descripción			
En todo momento el aplicativo web debe estar disponible con la información a consultar por parte de los usuarios.			

²² SYNERGIX. Tipos de requisitos: Funcional vs. No Funcional. [en línea]. [Consultado: 05 de Junio de 2017]. Disponible en: <https://synergix.wordpress.com/2008/07/07/requisito-funcional-y-no-funcional>

Tabla 19. Requisito no funcional 03. Obtención de estado de los estudiantes

Nombre	Obtención de estado de los estudiantes		
ID	RF 03	Prioridad	Alta
Usuarios	Usuario registrado		
Descripción			
El aplicativo web va a tener una base de datos, la cual va a interactuar con Arduino para recibir la información del estado del estudiante.			
.			

5.3. DIAGRAMA ENTIDAD RELACIÓN

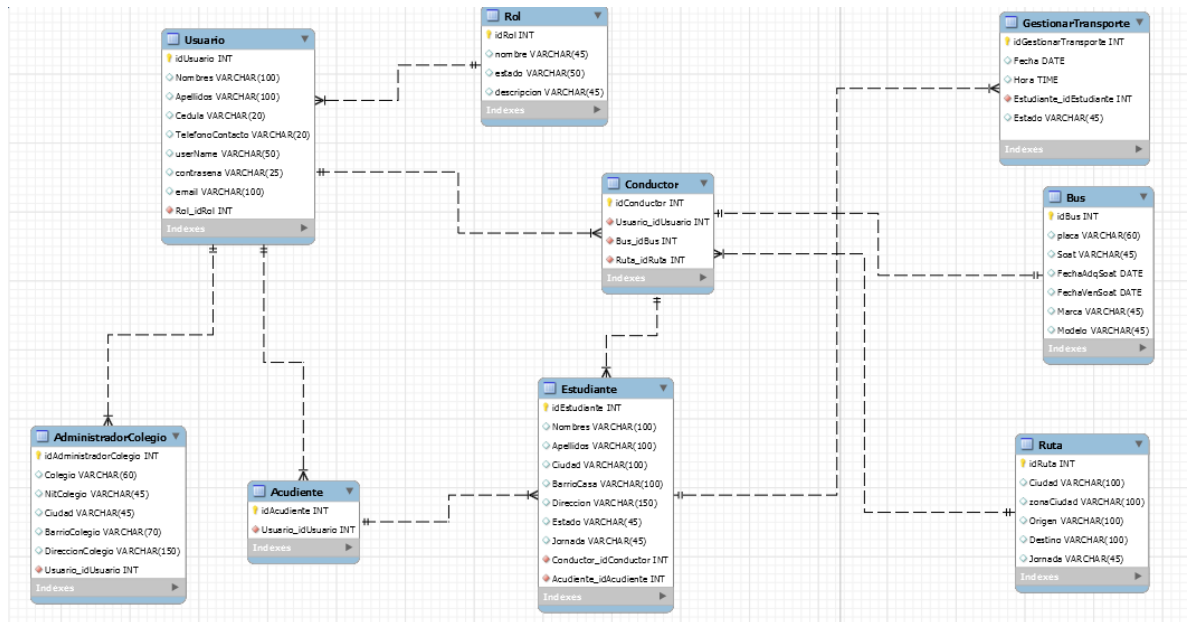
El diagrama entidad-relación (ERD) es una representación gráfica de lo que es nuestro sistema de información, es decir nos permite visualmente observar cómo se van a comportar y a relacionar nuestras clases por medio de entidades (tablas) y relaciones (flechas con cardinalidad) entre estas. Las entidades en nuestro caso son los tipos de usuario, y otros objetos de la vida real que hacen parte del sistema, como lo son la ruta, los estudiantes, el bus, entre otros.

En este ERD se observa como nuestras clases se relacionan entre sí con diferentes cardinalidades (Uno a uno, uno a muchos, muchos a muchos). Además de observar cómo se relacionan, también se observa las claves principales (primary key) y las claves foráneas (foreign key) de cada clase. Cada clase como se observa en el diagrama cuenta con varias filas que son llamadas atributos de la clase, los cuales representan las características que deben tener las clases.

Construir un Diagrama entidad-relación antes de realizar una aplicación cobra una gran importancia, ya que nos va a permitir organizar nuestras ideas de cómo va a ser nuestro sistema y como se van a relacionar los objetos de este mediante un

modelo lógico; permitiendo de esta manera ahorrar tiempo y esfuerzos al evitar cometer errores de tipo estructural.²³

Figura 2. Diagrama Entidad-Relación



Fuente: Autor

5.4. CASOS DE USO

Los casos de uso son una manera de identificar las funcionalidades que debe tener el sistema, es decir, mediante los casos de uso describimos lo que debe hacer el aplicativo. En un diagrama de casos de uso se le denominan actores, a la representación de usuarios o entidades que participan en el sistema.²⁴

²³ LUCIDCHART. Qué es un diagrama entidad-relación. [en línea]. [Consultado: 06 de Junio de 2017]. Disponible en: <https://www.lucidchart.com/pages/es/qu%C3%A9-es-un-diagrama-entidad-relaci%C3%B3n>

²⁴ IBM. Definición de casos de uso. [en línea]. [Consultado: 06 de Junio de 2017]. Disponible en: https://www.ibm.com/support/knowledgecenter/es/SSWSR9_11.0.0/com.ibm.pim.dev.doc/pim_tsk_arc_definingusecases.html

A continuación, se muestran todas las funcionalidades del sistema clasificadas por todos los tipos de usuario existentes:

Tabla 20. Casos de uso del usuario administrador

Tipo de Usuario: Administrador
<ul style="list-style-type: none">• Crear conductor• Consultar conductor• Modificar conductor• Eliminar conductor• Crear Bus• Crear Ruta• Modificar Ruta• Modificar Bus• Eliminar Ruta• Eliminar Bus• Crear estudiante• Consultar estudiante• Consultar estado de estudiante• Modificar estudiante• Eliminar estudiante• Crear acudiente• Modificar acudiente• Eliminar acudiente

Tabla 21. Casos de uso del usuario acudiente

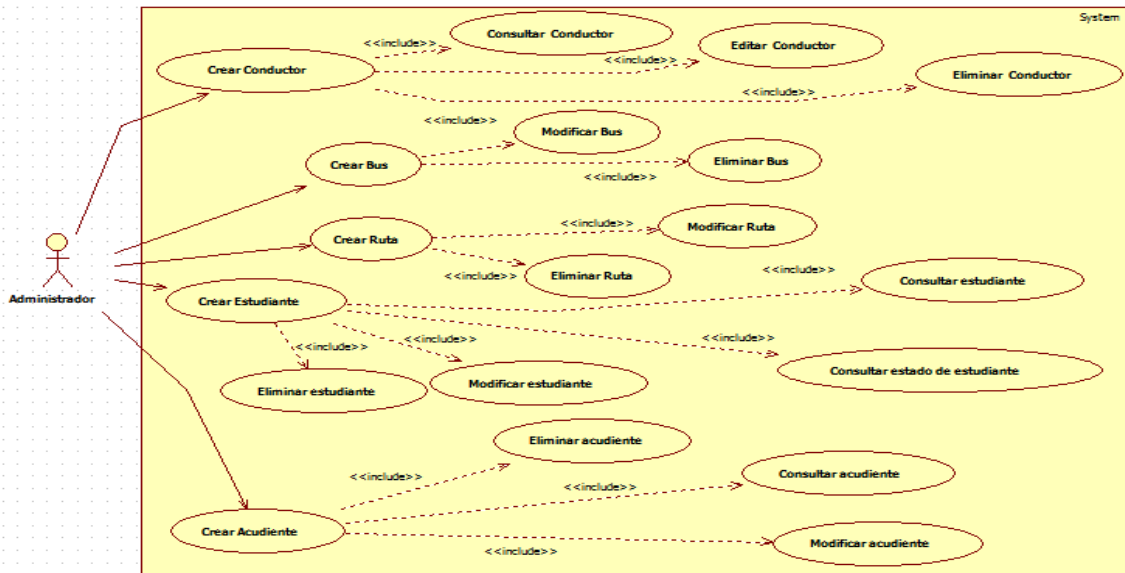
Tipo de Usuario: Acudiente
<ul style="list-style-type: none">• Consultar hijos• Consultar estado de hijos• Editar hijos• Consultar conductores

Tabla 22. Casos de uso del usuario conductor

Tipo de Usuario: Conductor
<ul style="list-style-type: none"> • Consultar estudiantes a cargo • Consultar estado de estudiantes a cargo • Consultar perfil • Modificar perfil • Consultar buses a cargo • Consultar rutas a cargo • Editar buses a cargo • Editar rutas a cargo

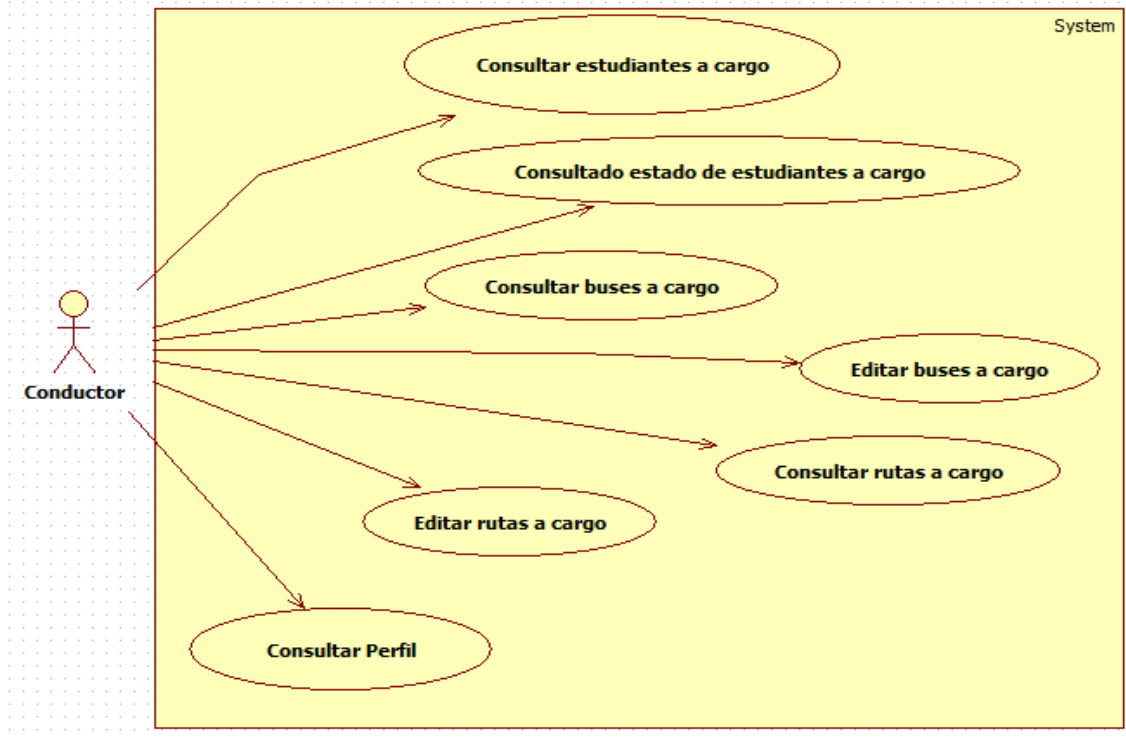
Ya habiendo identificado las funciones que debe realizar cada usuario de la aplicación, posteriormente se puede crear los diagramas de casos de uso. En los diagramas que se presentan a continuación se puede identificar los actores, los casos de uso y además la relación dependencia (Include); la cual nos indica que el caso de cambio o modificación del caso de uso desde el cual parte la relación cambia o modifica el caso de uso al cual llega la relación mencionada. Los diagramas de caso de uso de la aplicación son los siguientes:

Figura 3. Diagrama de casos de uso del administrador



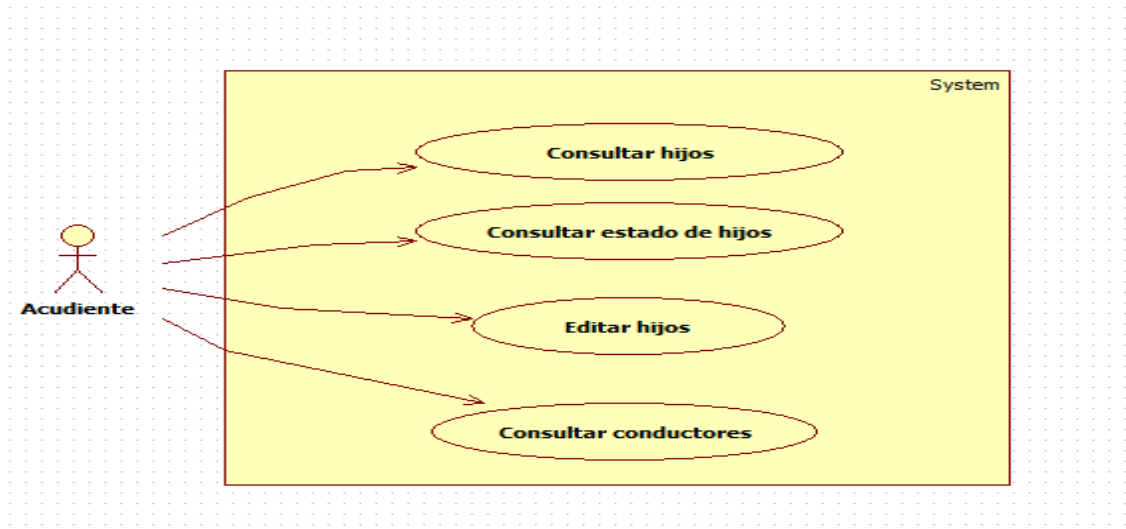
Fuente: Autor

Figura 4. Diagrama de casos de uso del conductor



Fuente: Autor

Figura 5. Diagrama de casos de uso del acudiente



Fuente: Autor

5.5. ARQUITECTURA DE OBJETOS

Antes de definir lo que es una arquitectura de objetos, se debe definir lo que es una clase. Una clase es una unidad en la que se encuentra toda la información que puede tener un objeto o entidad del sistema con el que se va a trabajar. Además, hay que decir que una clase se divide en los siguientes tres campos:

- **Nombre:** Es el nombre que se le va a atribuir a la entidad u objeto del sistema que se quiere representar.
- **Atributos:** Los atributos son todas las características o campos que posee una clase o entidad.
- **Métodos:** Los métodos de una clase son todas las operaciones que puede realizar este objeto representado, es decir, los métodos es la forma en que interactúa una clase con su entorno.

Ya habiendo definido el concepto de una clase, se puede hablar de lo que es un diagrama de clases. Un diagrama de clases no es otra cosa más que la representación gráfica de las relaciones que existen entre las clases que pertenecen a un sistema.

Un diagrama de clases puede tener varios tipos de relaciones, entre las cuales se encuentran:

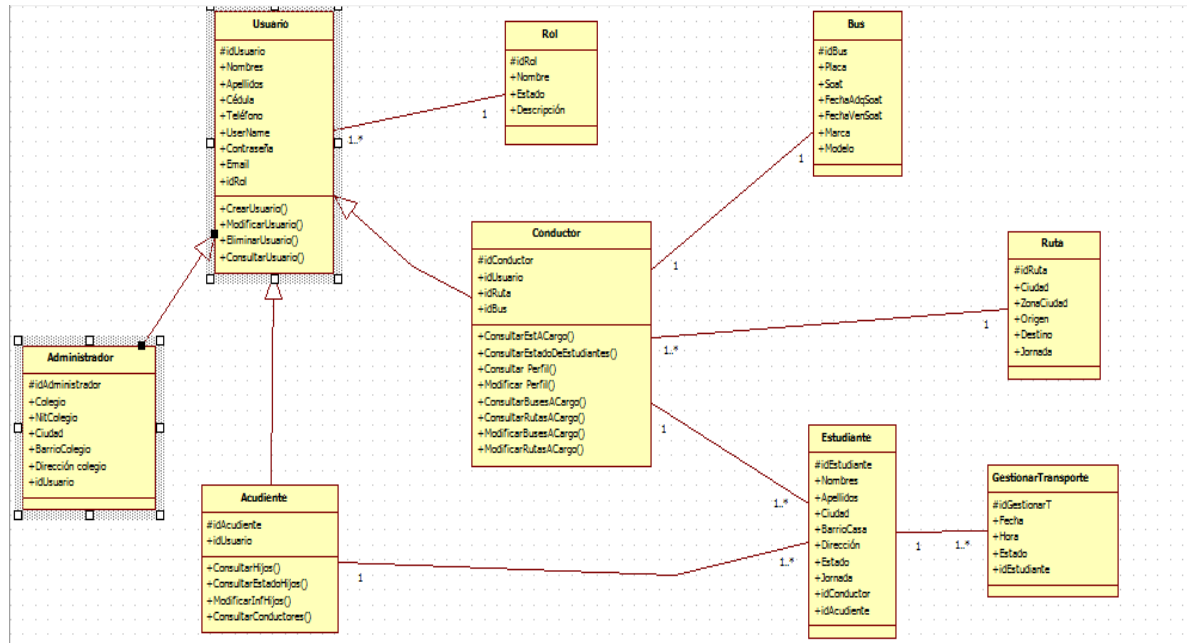
- **Herencia:** La herencia o generalización, es una relación que indica que las clases que heredan de otra clase llamada superclase o clase padre heredan o por decirlo de otro modo, pueden acceder a todos los atributos y métodos que tiene la clase padre. Se representa con una flecha vacía.

- Agregación: La agregación es un tipo de relación que indica que una clase es parte de otra clase, es decir, a una clase que es el 'todo', se agrega otra clase que depende de esta; pero esto no quiere decir que si la clase que es el 'todo' deja de existir, la clase que se agrega también deja de existir. Se representa con un rombo vacío.
- Composición: La composición es un tipo de agregación, pero más fuerte que al desaparecer la clase contenedor o la clase todo, desaparece la clase contenida o las clases que se relacionaban por composición con la clase contenedor.
- Asociación: Es un tipo de relación simple que permite asociar una clase con otra.

Ya habiendo definido lo que es una clase y un diagrama de clases, se puede abarcar en el término arquitectura de objetos. Una arquitectura de objetos consta de un diagrama de clases y su respectivo diccionario de clases, en el cual se especificar los atributos de cada clase, el tipo de datos de los atributos y de igual manera los métodos de estas clases con sus parámetros y valores permitidos y restringidos.

5.5.1. Diagrama de clases

Figura 6. Diagrama de clases



Fuente: Autor

5.5.2. Diccionario de clases

USUARIO.

Con la creación de esta clase se puede acceder a la aplicación web. Dependiendo del tipo de rol que tenga, se muestra la información pertinente a cada tipo de usuario. Esta clase consta de los siguientes atributos y métodos:

Tabla 23. Atributos Usuario

ATRIBUTOS	TIPO	VALORES PERMITIDOS
idUsuario	INT	Desde 1 en adelante
Nombres	String	Ningún tipo de restricción
Apellidos	String	Ningún tipo de restricción
Cédula	String	Ningún tipo de restricción
Teléfono	String	Ningún tipo de restricción
UserName	String	Ningún tipo de restricción
Contraseña	String	Ningún tipo de restricción
Email	String	Debe llevar el símbolo @

Tabla 24. Métodos Usuario

MÉTODOS	TIPO	PARAMETROS	VALORES DEVUELTOS
CrearUsuario()	Usuario	Todos los campos que tiene la clase Usuario	Devuelve un usuario tipo Usuario
ModificarUsuario()	Usuario	idUsuario	Devuelve un usuario tipo Usuario
EliminarUsuario()	Usuario	idUsuario	No devuelve nada
ConsultarUsuario()	Usuario	idUsuario	Devuelve un usuario tipo Usuario

ADMINISTRADOR.

La clase administrador representa al tipo de usuario administrador de un colegio, es decir, la persona que tiene todos los permisos de una aplicación. Esta clase hereda todos los métodos y atributos de la clase Usuario, por lo que no tiene métodos propios. Los atributos de la clase administrador son los siguientes:

Tabla 25. Atributos Administrador

ATRIBUTOS	TIPO	VALORES PERMITIDOS
idAdministrador	INT	Desde 1 en adelante
Colegio	String	Ningún tipo de restricción
NitColegio	String	Ningún tipo de restricción
Ciudad	String	Ningún tipo de restricción
BarrioColegio	String	Ningún tipo de restricción
Dirección Colegio	String	Ningún tipo de restricción

ACUDIENTE.

Es un tipo de usuario. Representa a todos los usuarios que son padres de familia de los estudiantes que están matriculado en el colegio y a su vez tienen un transporte escolar. Los atributos y métodos del acudiente son los siguientes:

Tabla 26. Atributos Acudiente

ATRIBUTOS	TIPO	VALORES PERMITIDOS
idAcudiente	INT	Desde 1 en adelante

Tabla 27. Métodos Acudiente

MÉTODOS	TIPO	PARAMETROS	VALORES DEVUELTOS
ConsultarHijos()	Estudiante	idEstudiante	Devuelve un Estudiante
ConsultarEstadoHijos()	Estudiante	idUsuario	Devuelve un Estudiante
ModificarInfHijos()	Estudiante	idUsuario	Devuelve un Estudiante
ConsultarConductores()	Usuario	idUsuario	Devuelve un usuario tipo Usuario

CONDUCTORES.

Es un tipo de usuario que representa a los conductores de los transportes escolares que prestan servicio al colegio. Los atributos y métodos de la clase son los siguientes:

Tabla 28. Atributos Conductores

ATRIBUTOS	TIPO	VALORES PERMITIDOS
idConductor	INT	Desde 1 en adelante

Tabla 29. Métodos Conductor

MÉTODOS	TIPO	PARAMETROS	VALORES DEVUELTOS
ConsultarEstACargo	Estudiante	idEstudiante	Devuelve un Estudiante
ConsultarEstadoEstudiantes()	String	idEstudiante	Devuelve un String
ConsultarPerfil()	Conductor	idConductor	Devuelve un Conductor
ModificarPerfil()	Conductor	idConductor	Devuelve un Conductor
ConsultarBusesACargo()	Bus	idBus	Devuelve un Bus
ConsultarRutasACargo()	Ruta	idRuta	Devuelve un Bus
ModificarBusesACargo()	Bus	idBus	Devuelve un Bus
ModificarRutasACargo()	Ruta	idRuta	Devuelve un Bus

ROL.

La clase Rol es la clase que determina el tipo de rol que tiene el usuario; estos roles son administrador, acudiente y conductor. Los atributos de la clase rol son:

Tabla 30. Atributos Rol

ATRIBUTOS	TIPO	VALORES PERMITIDOS
idRol	INT	Desde 1 en adelante
Nombre	String	Solo es permitido los roles de administrador, conductor y acudiente
Estado	String	Activo, Desactivo
Descripción	String	Todos los valores son permitidos

ESTUDIANTE.

Es la clase que representa a todos los estudiantes que se encuentran matriculados en el colegio y que además tienen un transporte escolar activo. Los atributos de la entidad estudiante son los siguientes:

Tabla 31. Atributos Estudiante

ATRIBUTOS	TIPO	VALORES PERMITIDOS
idEstudiante	INT	Desde 1 en adelante
Nombres	String	Todos los valores son permitidos
Apellidos	String	Todos los valores son permitidos
Ciudad	String	Todos los valores son permitidos
BarrioCasa	String	Todos los valores son permitidos
Dirección	String	Todos los valores son permitidos
Estado	String	Hay cuatro tipos de estado: En la casa, en el colegio, del bus al colegio, del bus a la casa
Jornada	String	Mañana, tarde

BUS.

Representa a todos los buses o transportes que están prestando servicio de transporte escolar en el colegio. Sus atributos son:

Tabla 32. Atributos Bus

ATRIBUTOS	TIPO	VALORES PERMITIDOS
idBus	INT	Desde 1 en adelante
Placa	String	6 caracteres, donde los primeros tres son letras y los otros son números
Soat	String	Activo, Vencido
FechaAdqSoat	Date	Representado por el siguiente formato: YYYY-MM-DD
FechaVecSoat	Date	Representado por el siguiente formato: YYYY-MM-DD
Marca	String	Todos los valores son permitidos
Modelo	Int	Valor entero de cuatro dígitos

RUTA.

Representa a todas las rutas, las cuales los conductores recorren día a día para recoger a los estudiantes de cada transporte y llevarlos de la casa al colegio y del colegio a la casa. Sus atributos son:

Tabla 33. Atributos Ruta

ATRIBUTOS	TIPO	VALORES PERMITIDOS
idRuta	INT	Desde 1 en adelante
Ciudad	String	Todos los valores son permitidos
ZonaCiudad	String	Todos los valores son permitidos
Origen	String	Todos los valores son permitidos
Destino	String	Todos los valores son permitidos
Jornada	String	Mañana-Tarde

GESTIONARTRANSPORTE.

Representa el registro de cada vez que un estudiante pase la tarjeta por el sensor o equipo instalado en los buses para detectar al estudiante. Sus atributos son:

Tabla 34. Atributos GestionarTransporte

ATRIBUTOS	TIPO	VALORES PERMITIDOS
idGestionarT	INT	Desde 1 en adelante
Fecha	Date	Representado por el siguiente formato: YYYY-MM-DD
Hora	Time	Representado por el siguiente formato: HH-MM-SS
Estado	String	Hay cuatro tipos de estado: En la casa, en el colegio, del bus al colegio, del bus a la casa

5.6. PLANEACIÓN Y ELECCIÓN DE FRAMEWORKS

Para la planeación del desarrollo de la aplicación web se tuvo en cuenta la metodología a usar y el enfoque de programación. Además de esto también se miró como punto de referencia el objetivo de aprender nuevas herramientas que faciliten el diseño e implementación de la aplicación. Todas estas consideraciones llevaron a mirar los frameworks que existían en el mercado para facilitar el desarrollo; se tuvo en cuenta el fácil aprendizaje de la herramienta, la suficiente documentación existente y el contar con el apoyo de una persona con experiencia en el framework.

Para la elección de los framework de desarrollo web se decidió usar Java Server Faces (JSF) en combinación con Primefaces, Hibernate y el servidor de aplicaciones GlassFish. A continuación, se detalla aspectos importantes por los cuales se escogieron estos frameworks.

Tabla 35. Características de frameworks escogidos

Creación rápida de prototipos	Los mayores beneficios para la productividad con JSF son los asistentes disponibles de los IDE que generan la mayor parte del código
Complejidad	Existen implementaciones de código abierto de la especificación JSF que permiten el uso de contenedores que no son de Java EE como Tomcat o GlassFish, lo que reduce enormemente la complejidad de tener que ejecutar un servidor de aplicaciones Java Enterprise completo
Facilidad de uso	JSF intenta proporcionar un framework fácil de usar para crear componentes web reutilizables en sus aplicaciones Java sin una gran curva de aprendizaje. JSF 2.2 fue ratificado y publicado recientemente como parte de la especificación Java EE 7. JSF es muy fácil de poner en marcha y a menudo, no requiere descargas o configuraciones adicionales, ya que el código necesario se incluye en cualquier servidor de aplicaciones compatible con Java EE.
Documentación y comunidad	A JSF lo soporta Oracle, lo cual hace que se encuentra mucha información respecto al uso de componentes con este framework, además de soporte y documentación de tecnologías alternas que se pueden complementar con JSF como lo son MyFaces, Primefaces y IceFaces. Además de esto se encuentran tutoriales e información adicionales en Youtube
Ecosistema	El ecosistema de JSF incluye bibliotecas de componentes como Richfaces, OmniFaces, Primefaces, IceFaces, ADF Faces. Sin embargo, el valor real forma parte de la especificación Java EE. JSF se puede agregar fácilmente a los entornos existentes de Java EE e inmediatamente recibe todos los beneficios inherentes al ecosistema Java EE
Rendimiento y escalabilidad	Las aplicaciones JSF pueden funcionar, pero realmente las ganancias de rendimiento provienen de los servidores de aplicaciones de Java EE. JSF en sí mismo no proporciona soporte explícito para llamadas asíncronas y se basa en la especificación Java EE para proporcionar anotaciones @Asynchronous y @Schedule en los EJB de lógica empresarial.
Mantenimiento	JSF se aprovecha del patrón de arquitectura Modelo-Vista-Controlador(MVC) que nos permite separar los datos y la lógica de negocio de la interfaz de usuario; lo cual hace que haya una gran flexibilidad y reutilización de código en el software

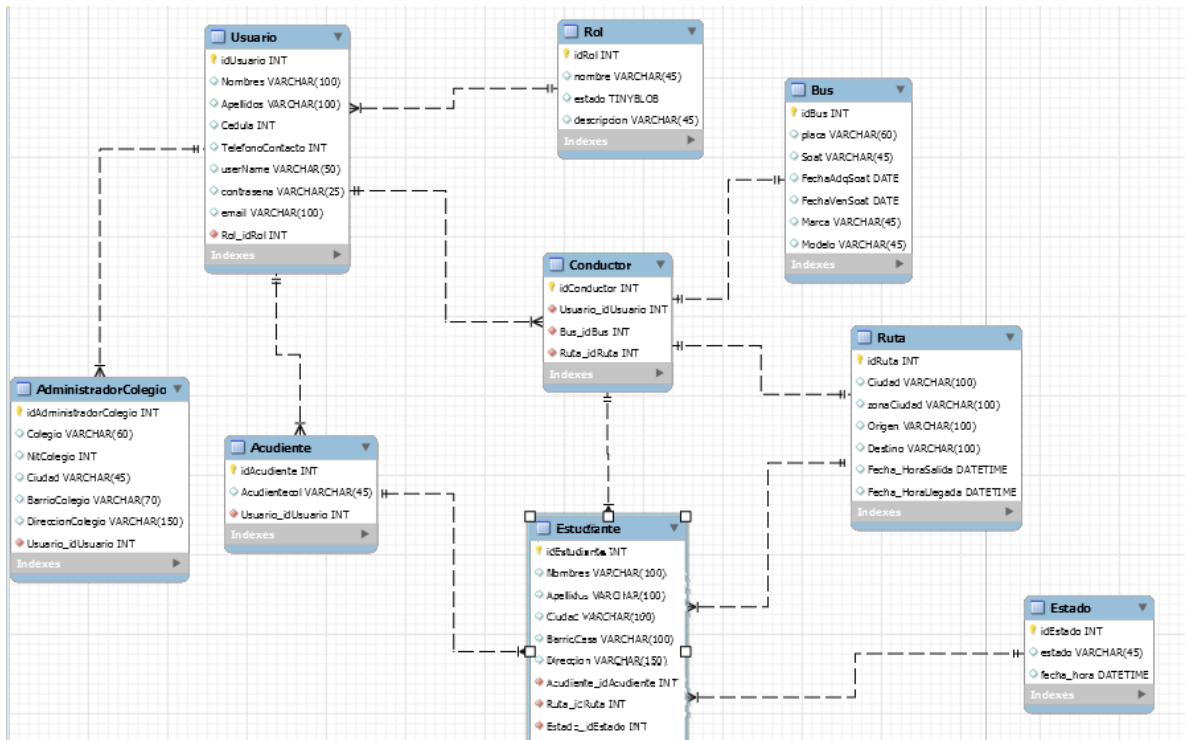
Compatibilidad	También se escogió Primefaces ya que es una librería de componentes que se complementa a la perfección con JSF y además sus componentes web son de excelente diseño visual muy agradable a la vista. Además de esto, Primefaces es de código abierto y cuenta con suficiente documentación y demos que hacen muy fácil su entendimiento y aplicación
----------------	--

5.7. PROTOTIPOS

5.7.1. Primer prototipo. En la elaboración del primer prototipo se tuvo en cuenta los requerimientos 01 y 02 de contar con tres tipos de usuario (Administrador, Acudiente y Conductor), además de que para cada tipo de usuario se clasificara el contenido a visualizar y se restringiera funciones. Para el cumplimiento de este primer prototipo se hizo un estudio exhaustivo de las tecnologías Java Server Faces, Primefaces y el framework de Hibernate que como ya se había mencionado, permite hacer el mapeo de la base de datos relacional al proyecto (ORM), además de obtener la persistencia de los datos y la facilidad de poder trabajar con el paradigma orientado a objetos sin necesidad de tener que mirar la base de datos relacional.

Para este primer prototipo se diseñó el siguiente modelo entidad relación:

Figura 7. Modelo entidad relación. Primer prototipo



Fuente: Autor

Las vistas o interfaces de usuario para el cumplimiento de estos requerimientos fueron los siguiente:

Login o Inicio de sesión

Figura 8. Inicio de sesión. Primer prototipo

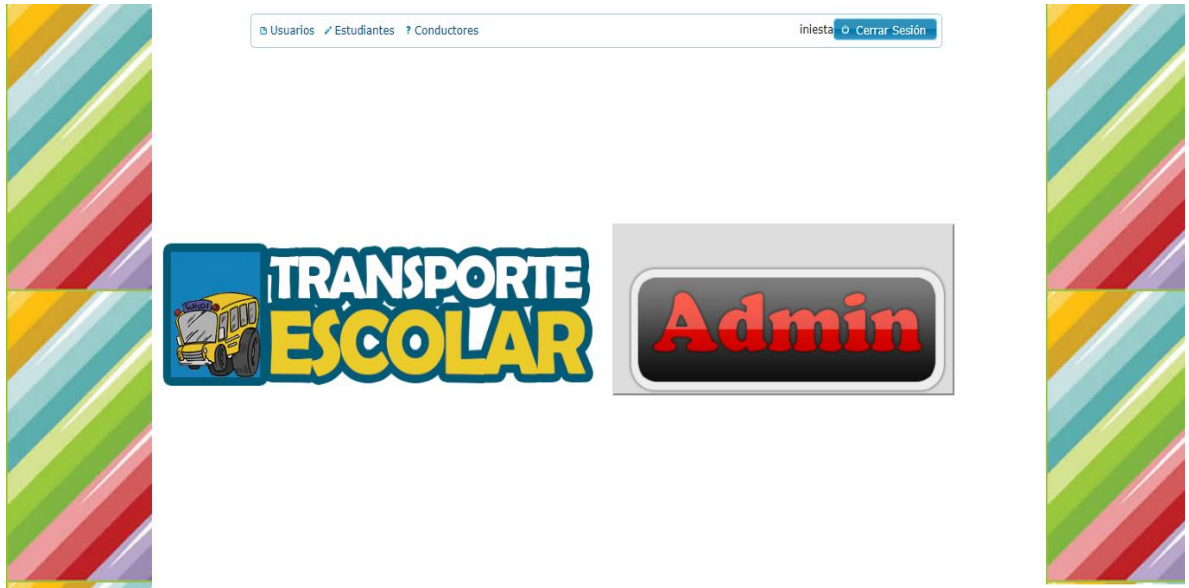
Ingresar Usuario y clave

Usuario:

Clave:

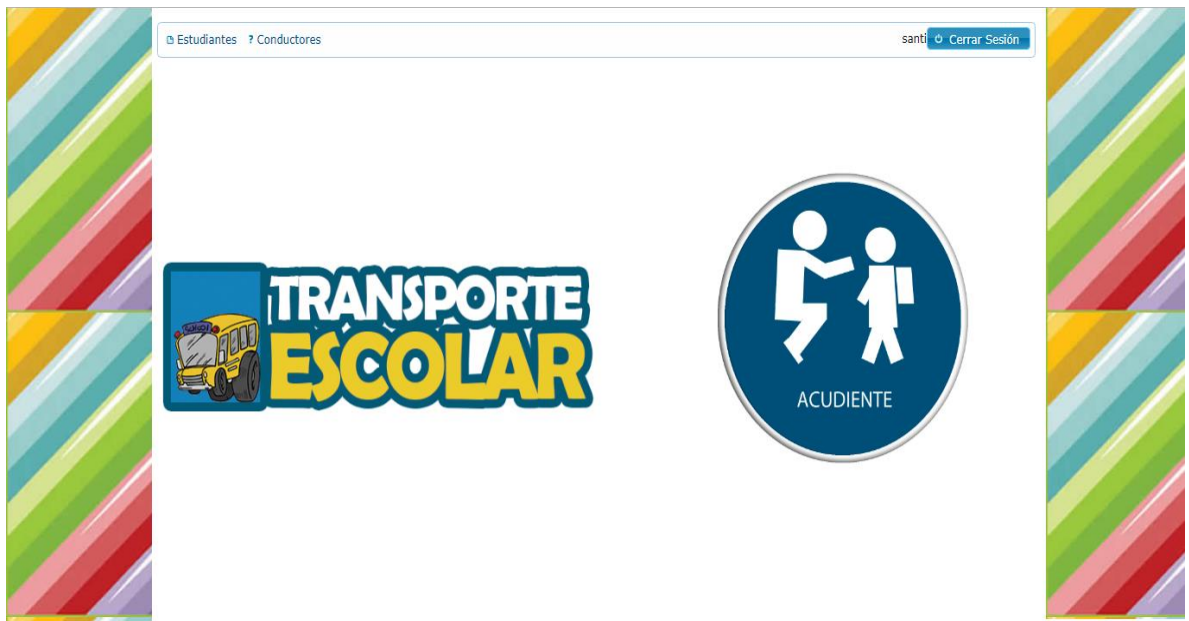
Ventana principal del usuario administrador

Figura 9. Ventana Administrador Primer Prototipo



Ventana principal del usuario Acudiente

Figura 10. Ventana Acudiente. Primer prototipo



Ventana principal del usuario Conductor

Figura 11. Ventana Conductor. Primer prototipo



5.7.2. Segundo prototipo. En este prototipo se buscó cumplir con el requisito 03, 05, 06, 08, 09 y 12

Permitir el registro de personas que no cuentan con usuario en la aplicación.

Figura 12. Registro de usuarios. Segundo prototipo

The image shows a user registration form with two main sections. The top section, titled "Datos Personales", contains input fields for "Id" (with the value "10"), "Nombres", "Apellidos", "Cédula", "Teléfono", and "E-mail". A "Volver" button is located in the top right corner of this section. The bottom section, titled "Datos de la Cuenta", contains a dropdown menu for "Tipo de Usuario" (set to "-- Seleccione --"), input fields for "Usuario", "Clave", and "Repetir Clave", and a "Registrar" button with a checkmark icon.

Permitir que el administrador cree existencias de administradores, acudientes y conductores.

Mediante esta ventana al darle en el botón crear, se abre el formulario que se observa en la imagen, donde permite crear un nuevo usuario, ingresando el usuario, los nombres, apellidos, email, contraseña y por último el Rol o tipo de usuario que desea crear.

Figura 13. Crear Usuario. Segundo prototipo

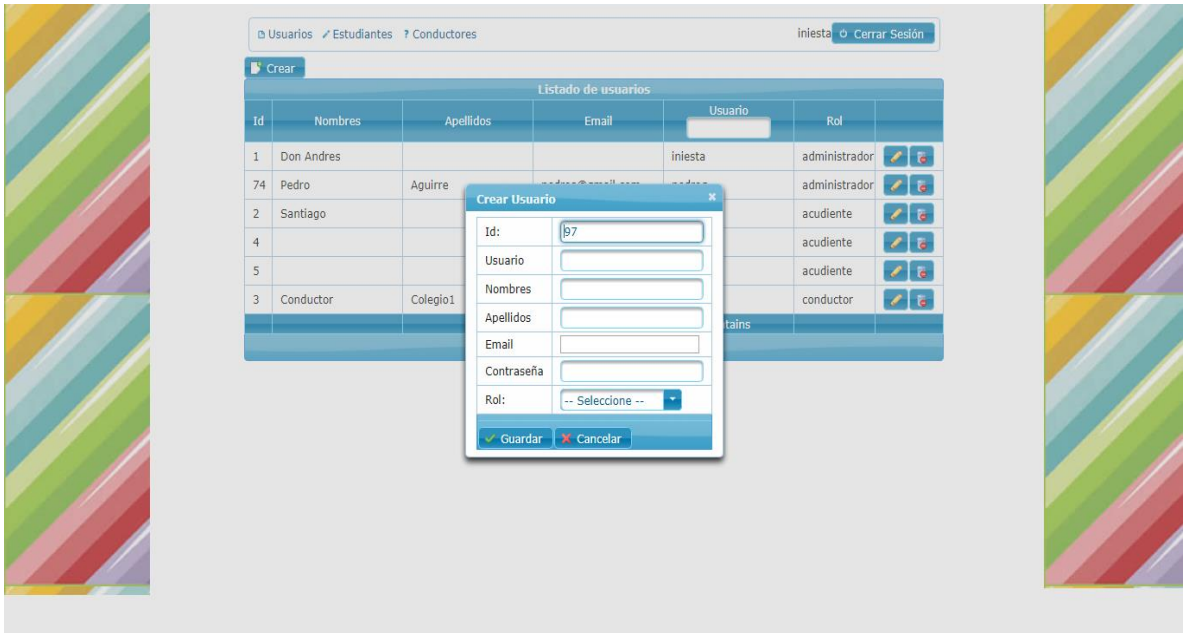


Tabla de estudiantes del usuario administrador

En esta ventana de igual forma que para crear un tipo de usuario, al darle en el botón crear, se abre un formulario donde al ingresar los datos pertinentes de un estudiante nuevo, permite crear una nueva existencia de estudiantes

Figura 14. Ventana estudiantes del Administrador. Segundo prototipo

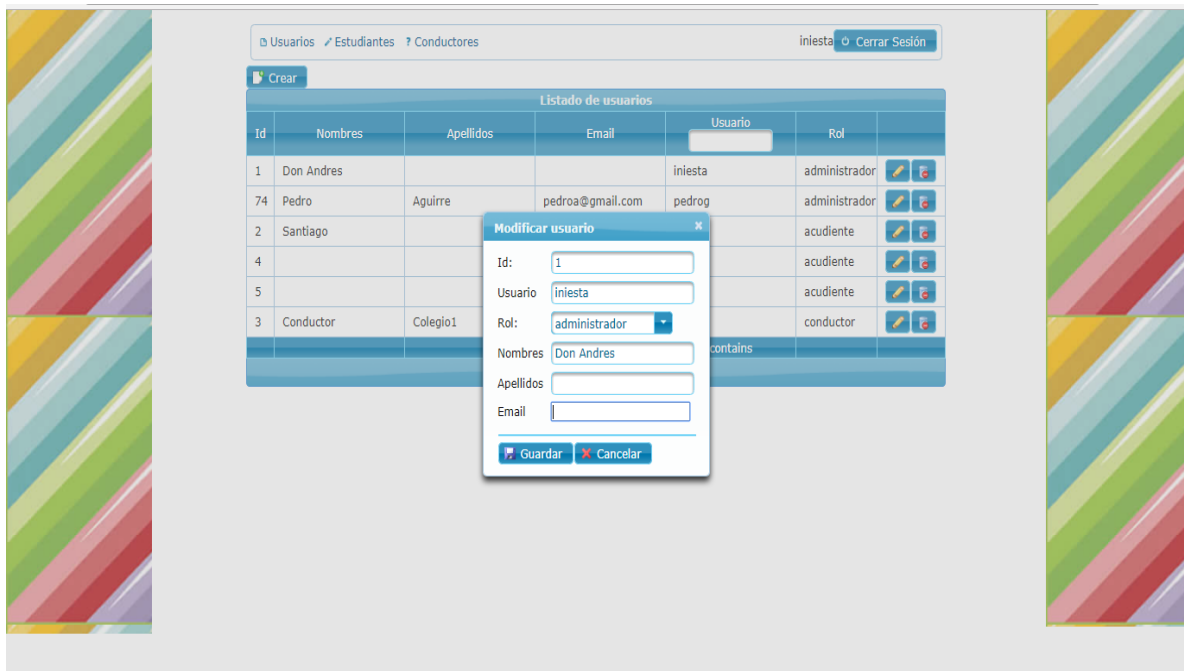


Modificar a los usuarios administrador, acudiente o conductor

En la ventana principal del usuario administrador se puede observar que se despliega una tabla con los registros de todos los usuarios existentes en la base de datos, de igual manera se observa que cada registro se ubica en una fila diferente y que al final de cada fila, hay un icono de un lápiz y al lado otro icono de una canasta de basura. El lápiz nos simboliza la función editar, de manera que la canasta nos representa la función eliminar.

Ahora bien, al darle click sobre el icono del lápiz sobre una fila, se abre una ventana o formulario que permite modificar el usuario de la fila donde se oprimió sobre el icono del lápiz.

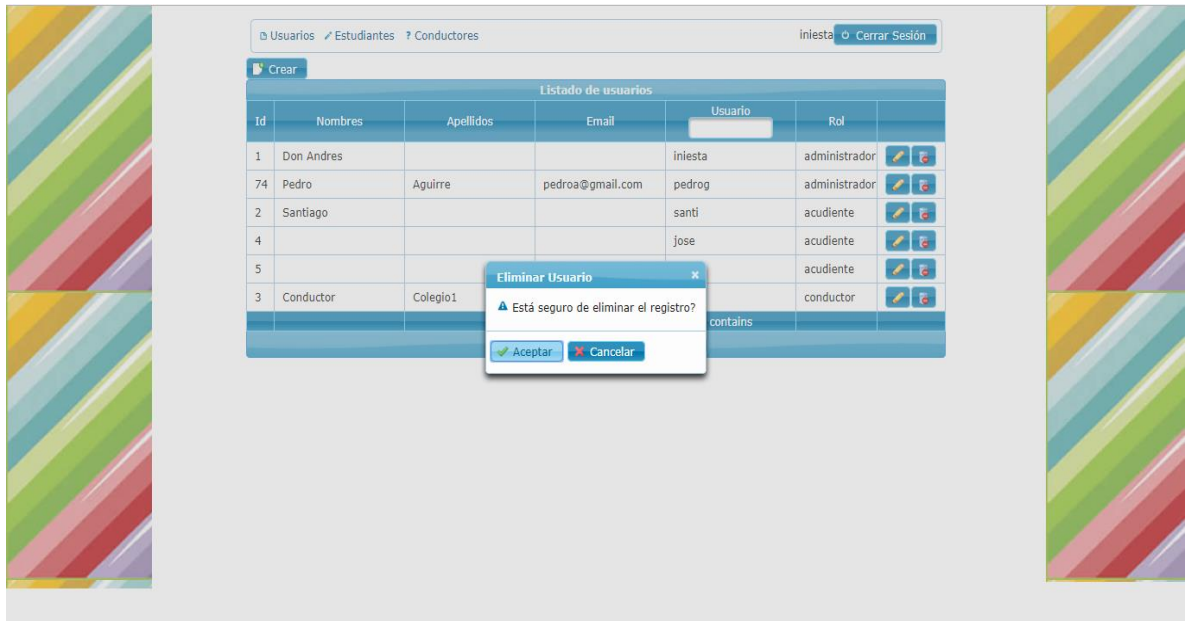
Figura 15. Modificar usuario. Segundo prototipo



Eliminar usuario administrador, acudiente o conductor

Siguiendo la misma metodología que para modificar un usuario, se da click sobre el icono de la canasta en la fila que pertenezca al usuario que se quiera eliminar; luego se abre un dialogo preguntando si está seguro de eliminar el registro, con el botón aceptar y el botón cancelar. Por consiguiente, al darle click en el botón aceptar, se elimina el usuario.

Figura 16. Eliminar usuario. Segundo prototipo



Listado conductores desde el usuario acudiente

Desde el usuario acudiente, se puede observar los conductores inscritos al colegio donde estudia el hijo o hijos del acudiente que entró al sistema mediante el botón Conductores que se encuentra en la parte superior sobre el menú.

Figura 17. Conductores desde ventana Acudiente. Segundo prototipo

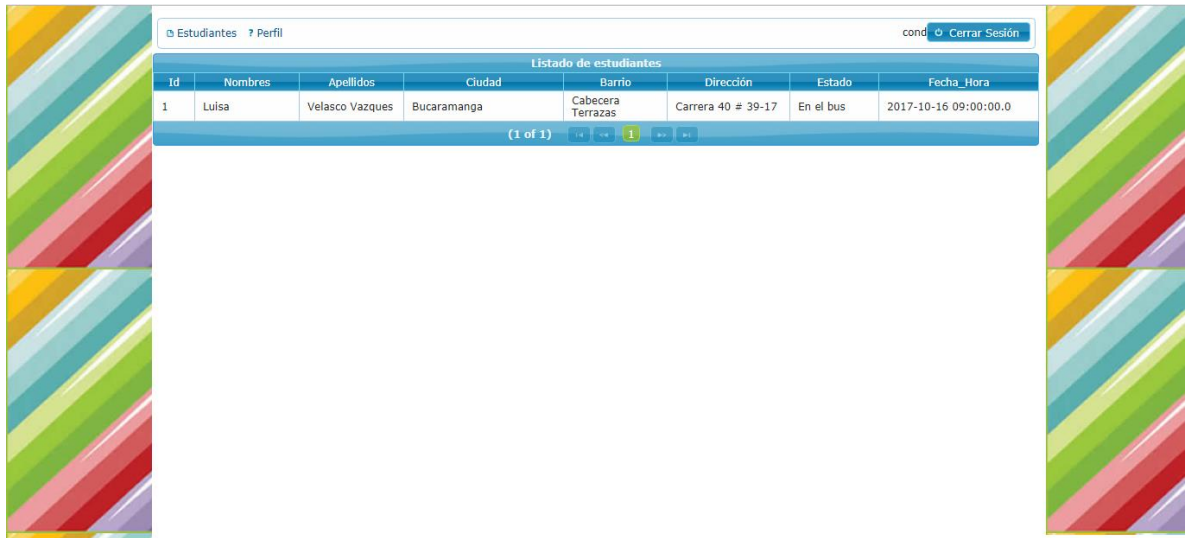
Listado de Conductores										
Id	Nombres	Apellidos	Telefono	E-mail	Bus	Rutas	Origen	Destino	FechaHoraSalida	FechaHoraLegada
1	Conductor	Colegio1	3156760914	condu@gmail.com	XMD-345	Cafaveral			2017-10-11 07:00:00.0	2017-10-11 08:00:00.0
2	Martin		3154327840		SAT-554	Centro			2017-10-12 05:00:00.0	2017-10-12 06:00:00.0
3					ZAS-093	Cabecera			2017-10-18 17:00:00.0	2017-10-18 18:00:00.0

contains (1 of 1)

Listado de estudiantes que pertenecen al transporte de un usuario conductor

Desde el usuario conductor al oprimir el botón estudiantes que se encuentra en la parte superior de la ventana, la aplicación abre una pantalla o ventana donde muestra una tabla con todos los estudiantes que pertenecen a un conductor o un transporte que es manejado por un usuario conductor, además de los datos pertinentes de cada uno de estos.

Figura 18. Estudiantes de un conductor. Segundo prototipo



Estudiantes		Perfil		cond					Cerrar Sesión
Listado de estudiantes									
Id	Nombres	Apellidos	Ciudad	Barrio	Dirección	Estado	Fecha_Hora		
1	Luisa	Velasco Vazques	Bucaramanga	Cabecera Terrazas	Carrera 40 # 39-17	En el bus	2017-10-16 09:00:00.0		

Ventana de hijos de un acudiente

Siguiendo las formas de proceder anteriores, desde el usuario acudiente y el botón estudiantes, se puede modificar la información pertinente a los hijos de este usuario acudiente que ingresó mediante el símbolo lápiz que se encuentra al final de cada registro, fila o estudiante.

Figura 19. Estudiantes desde acudiente. Segundo prototipo

Estudiantes													
Listado de estudiantes													
Id	Nombre	Apellido	Ciudad	Barrio	Dirección	Estado	Fecha_Hora	Nombres Conductor	Apellidos Conductor	Teléfono Conductor	Cédula Conductor	E-mail Conductor	Ruta
1	Luisa	Velasco Vazquez	Bucaramanga	Cabecera Terraza	Carrera 40 # 39-17	En el bus	2017-10-16 09:00:00.0	Conductor	Colegio1	3156760914	14325676	condu@gmail.com	Cabecera
4	Carlos	Beltran				En el bus	2017-10-16 09:00:00.0	Martin		3154327840	10945726		Centro

5.7.3. Tercer prototipo. En este prototipo se buscó mejorar la vista de inicio de sesión, optimizar la vista o ventana de estudiantes de un determinado acudiente y se añadió una ventana del perfil de un conductor.

Cuando se dice que se buscó optimizar la ventana de estudiantes de un acudiente, se hace referencia a que como un padre de familia o acudiente no va a tener una gran cantidad de hijos inscritos en un colegio, se buscó representar la información de esos hijos o estudiantes de una manera que fuera más visible, de tal manera que se eliminó la tabla que se presentó en el prototipo número 2 y se reemplazó por un formulario que permite ir pasando de registro, según la cantidad de hijos que tenga este padre de familia o acudiente.

Inicio de sesión

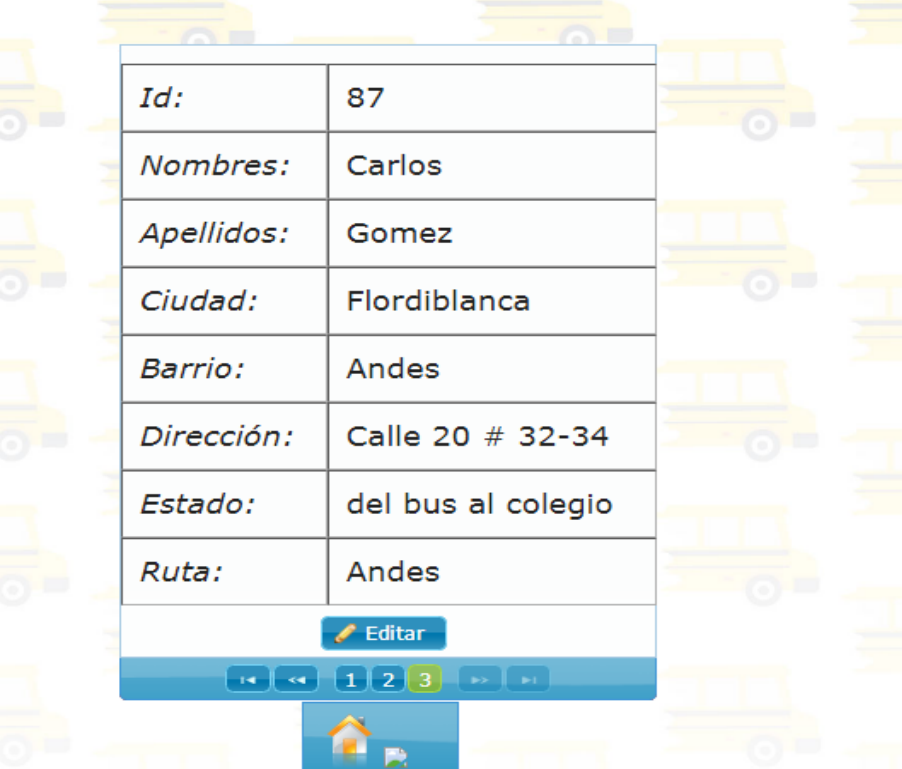
Figura 20. Inicio de sesión. Tercer prototipo



A login form titled "Ingresar Usuario y clave" (Enter user and password). It features two input fields: "Usuario:" (User) and "Clave:" (Password). Below the fields is a blue button with a person icon and the text "Login".

Vista de hijos de un acudiente

Figura 21. Hijos de acudiente. Tercer prototipo

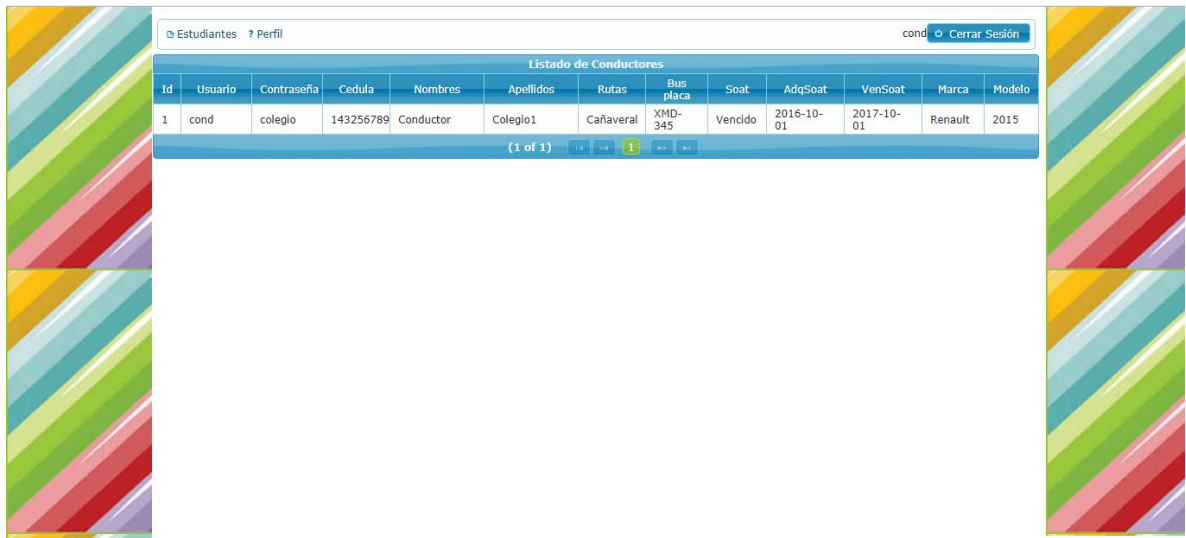


A table displaying child information for a parent. The table has two columns: labels and values. Below the table is an "Editar" (Edit) button with a pencil icon. At the bottom, there is a navigation bar with a home icon and a list of three items, with the third item highlighted.

<i>Id:</i>	87
<i>Nombres:</i>	Carlos
<i>Apellidos:</i>	Gomez
<i>Ciudad:</i>	Flordiblanca
<i>Barrio:</i>	Andes
<i>Dirección:</i>	Calle 20 # 32-34
<i>Estado:</i>	del bus al colegio
<i>Ruta:</i>	Andes

Perfil de un conductor

Figura 22. Perfil del conductor. Tercer prototipo

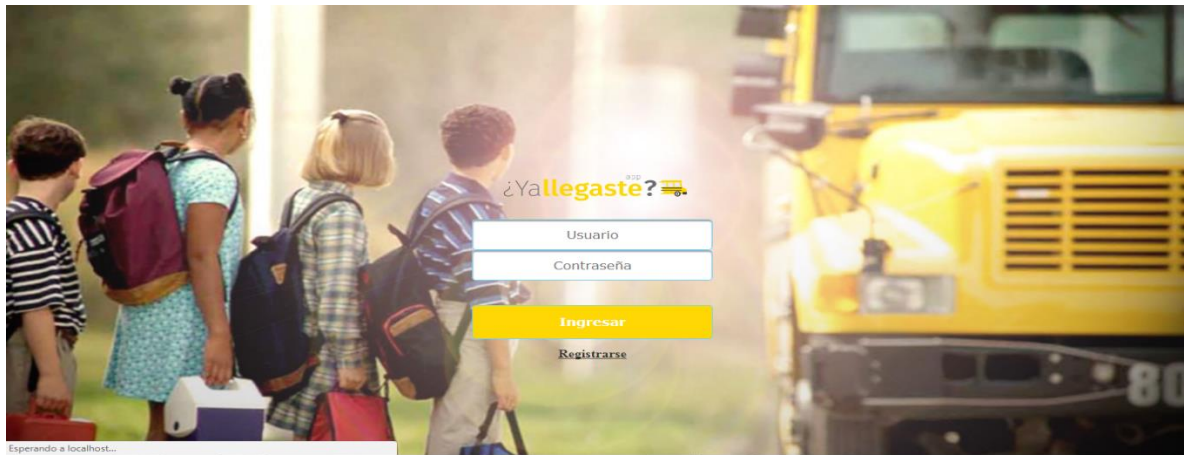


Estudiantes ? Perfil													cond Cerrar Sesión	
Listado de Conductores														
Id	Usuario	Contraseña	Cedula	Nombres	Apellidos	Rutas	Bus placa	Soat	AdqSoat	VenSoat	Marca	Modelo		
1	cond	colegio	143256789	Conductor	Colegio1	Cañaveral	XMD-345	Vencido	2016-10-01	2017-10-01	Renault	2015		

5.7.4. Prototipo final o producto final. En este producto final se alcanzó el objetivo de cumplir con todos los requisitos funcionales que se habían propuesto anteriormente, adicionalmente se mejoró la interfaz de usuario de todas las vistas según el usuario y por último se tuvo en cuenta la forma de presentación de cierta información. A continuación, se van a presentar la mayoría de las pantallas de la aplicación dependiendo del usuario y de las funciones a realizar.

Inicio de Sesión

Figura 23. Inicio de sesión. Producto Final



Registrarse

Figura 24. Registrarse. Producto Final

The registration form is set against a background of a repeating school bus pattern. It is divided into two main sections: 'Datos Personales' and 'Datos de la Cuenta'. The 'Datos Personales' section includes input fields for 'Id' (containing the number '2'), 'Nombres', 'Apellidos', 'Cédula', 'Teléfono', and 'E-mail'. The 'Datos de la Cuenta' section features a dropdown menu for 'Tipo de Usuario' (set to 'Seleccione --'), and input fields for 'Usuario', 'Clave', and 'Repetir Clave'. A blue 'Registrar' button with a checkmark is located at the bottom left of this section. A 'Volver' button is positioned in the top right corner of the form area.

Seguridad de no poder ingresar a una página si no se ha logueado de manera correcta

Figura 25. Seguridad de acceso. Producto final

Discutpa, Acceso denegado

Tus credenciales no pueden ser verificadas o no estás autorizado para ver esta página

To Home Page: Este vínculo está desactivado porque no se ha podido asignar un caso de navegación.

5.7.4.1. Usuario administrador


Ventana Principal

Figura 26. Ventana principal administrador. Producto final



Usuarios

Figura 27. Usuario administrador. Producto final


¿Ya llegaste? app  INIESTA [Cerrar Sesión](#)

[Crear](#)

Listado de usuarios


Id	Nombres	Apellidos	Cedula	Telefono	Email	Usuario	Rol
1	Andres	Iniesta Lujano	10987654325	3176793120	iniesta@barcelona.com	iniesta	Administrador
2	Santiago	Delgado Mejia	1098739576	3176709541	santiago.sdm@gmail.com	santi	Acudiente
4	Maria Eugenia	Mejia Pinto	37832295	3158112528	asesoriasmariae@hotmail.com	maria	Acudiente
2870	Jose Antonio	Delgado Castro	364897652	3164326578	segurosjad@hotmail.com	josea	Acudiente
3	Martin	Sanchez	10954326572	3165430943	martinsanchez@hotmail.com	martin	Conductor
5	Leonidas	Valverde	376453290	3214569084	leonidas@hotmail.com	cond	Conductor
17	Gilberto	Galarza	1097452345	3124569879	ggalarza@hotmail.com	ggalarza	Conductor

(1 of 1) [Editar](#) [Eliminar](#)



Acudientes

Figura 28. Acudiente Administrador. Producto final


¿Ya llegaste? app  INIESTA [Cerrar Sesión](#)

[Registrar Acudiente](#)

Listado de Acudientes

Id	Nombres	Apellidos	Cedula	Telefono	Email	Usuario
1	Santiago	Delgado Mejia	1098739576	3176709541	santiago.sdm@gmail.com	santi
2	Maria Eugenia	Mejia Pinto	37832295	3158112528	asesoriasmariae@hotmail.com	maria
3	Jose Antonio	Delgado Castro	364897652	3164326578	segurosjad@hotmail.com	josea

(1 of 1) [Eliminar](#)



Conductores

Figura 29. Conductores Administrador. Producto final

¿Ya llegaste? app INIESTA [Cerrar Sesión](#)

[Registrar Conductor](#)

Listado de Conductores

Id	Nombres	Apellidos	Cedula	Telefono	Email	Usuario	Placa Bus	Marca	Modelo	Soat	Fecha.Adq	Fecha.Ven	Ciudad Ruta	Zona	Origen	Destino
1	Martin	Sanchez	10954326572	3165430943	martinsanchez@hotmail.com	martin	XME-090	Renault	2016	Activo	2017-06-06	2018-06-06	Floridablanca	Cañaveral - Lagos 1	Cañaveral	Lagos 1
2	Leonidas	Valverde	376453290	3214569084	leonidas@hotmail.com	cond	ZAE-136	Chevrolet	2016	Activo	2017-10-18	2018-10-18	Floridablanca	Andes-Limoncito	Andes	Limoncito
3	Gilberto	Galarza	1097452345	3124569879	ggalarza@hotmail.com	ggalarza	WXV-488	Mazda	2017	Activo	2018-01-02	2019-01-02	Floridablanca	Cañaveral - Lagos 1	Cañaveral	Lagos 1

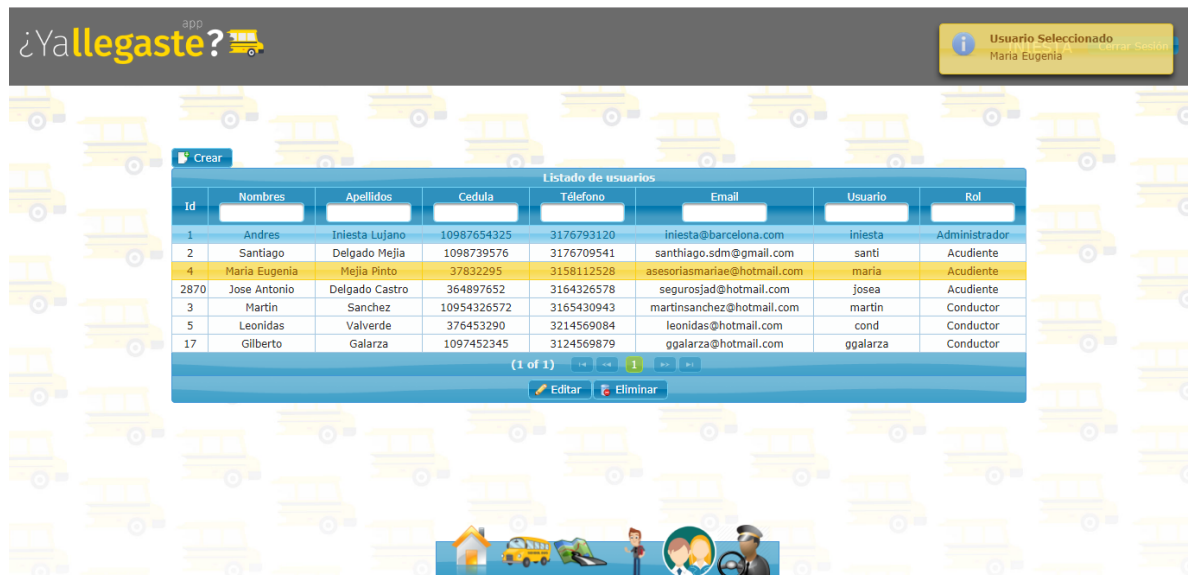
(1 of 1) [Eliminar](#)

Función seleccionar usuario

Vale la pena aclarar, que como se observó, en este producto final desapareció los iconos de editar y eliminar al final de cada fila, ya que se determinó que estaba ocupando espacio innecesario; en vez de eso se colocó debajo de la tabla un botón editar y otro botón eliminar, que al seleccionar un usuario se activan y se puede hacer uso de estas funciones.

Además de esto también cabe resaltar que se creó un menú dinámico al final de cada ventana, donde se pueden hacer a las diferentes ventanas y por consiguiente a información diferente y útil.

Figura 30. Seleccionar usuario. Producto final



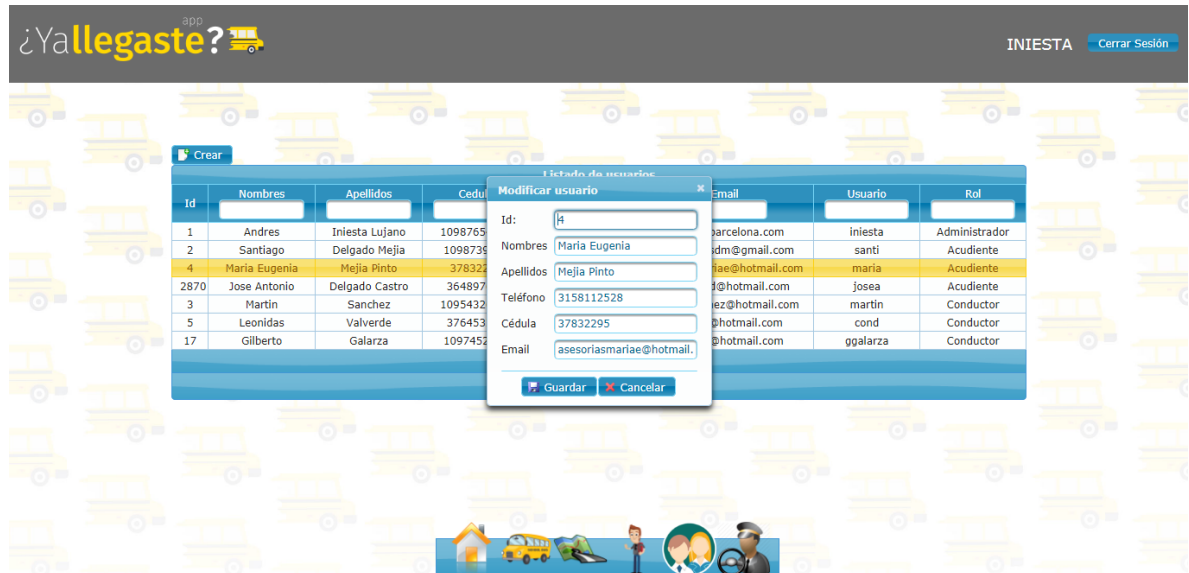
Crear usuario

Figura 31. Crear usuario. Producto final



Función Modificar Usuario

Figura 32. Modificar usuario. Producto final



Función Eliminar Usuario

Figura 33. Eliminar Usuario. Producto final



Estudiantes

Figura 34. Estudiantes de Administrador. Producto final

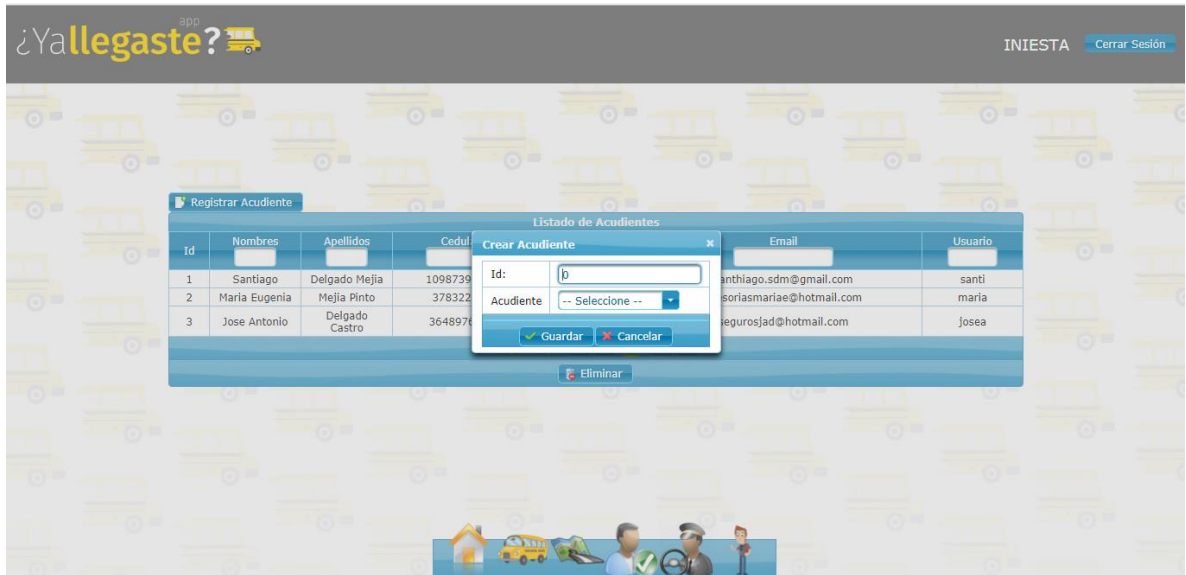
¿Yallegaste?																		
INIESTA Cerrar Sesión																		
Crear																		
Listado de estudiantes																		
Id	Nombres	Apellidos	Ciudad	Barrio	Dirección	Jornada	Estado	Nombres Acudiente	Apellidos	Celular	Email	Nombres Conductor	Apellidos	Celular	Bus	Ruta	Origen	Destino
15	Jose David	Rueda	Floridablanca	Cañaveral	Plazuela Cañaveral T.A Apto. 502	Mañana	En la casa	Santiago	Delgado Mejía	3176709541	santhiago.sdm@gmail.com	Martin	Sanchez	3165430942	XME-090	Cañaveral - Lagos 1	Cañaveral	Lagos 1
87	Carlos Alfredo	Gomez	Floridablanca	Andes	Calle 60 #22-89	Mañana	En la casa	Santiago	Delgado Mejía	3176709541	santhiago.sdm@gmail.com	Leonidas	Valverde	3214569084	ZAE-136	Andes-Limoncito	Andes	Limoncito

(1 of 1) [Editar](#) [Eliminar](#)

Registrar Acudiente

Luego de que se haya creado un usuario acudiente, el administrador debe registrarlo como tal en la ventana de acudientes mediante el botón registrar acudiente. Cuando se oprime este botón, se abre un formulario con dos campos; el primer campo nos pide el ID que va a ser el número siguiente a el último ID de acudiente que se registró; ahora bien, en el segundo campo se muestra una lista en donde están todos los usuarios de tipo acudiente, es el deber del administrador seleccionar los usuario acudientes que no se hayan registrado como tal.

Figura 35. Registrar acudiente. Producto final




Buses

Figura 36. Buses de Administrador. Producto final



Rutas


Figura 37. Rutas de Administrador. Producto final

¿Yallegaste? APP  INIESTA [Cerrar Sesión](#)

[Crear](#)

Listado de Rutas					
Id	Ciudad	Zona Ciudad	Origen	Destino	Jornada
1	Floridablanca	Cañaveral - Lagos 1	Cañaveral	Lagos 1	Mañana
2	Floridablanca	Andes-Limoncito	Andes	Limoncito	Mañana
3	Bucaramanga	Terrazas	Terrazas-Cra 39	Cra 33 # 27-10	Mañana

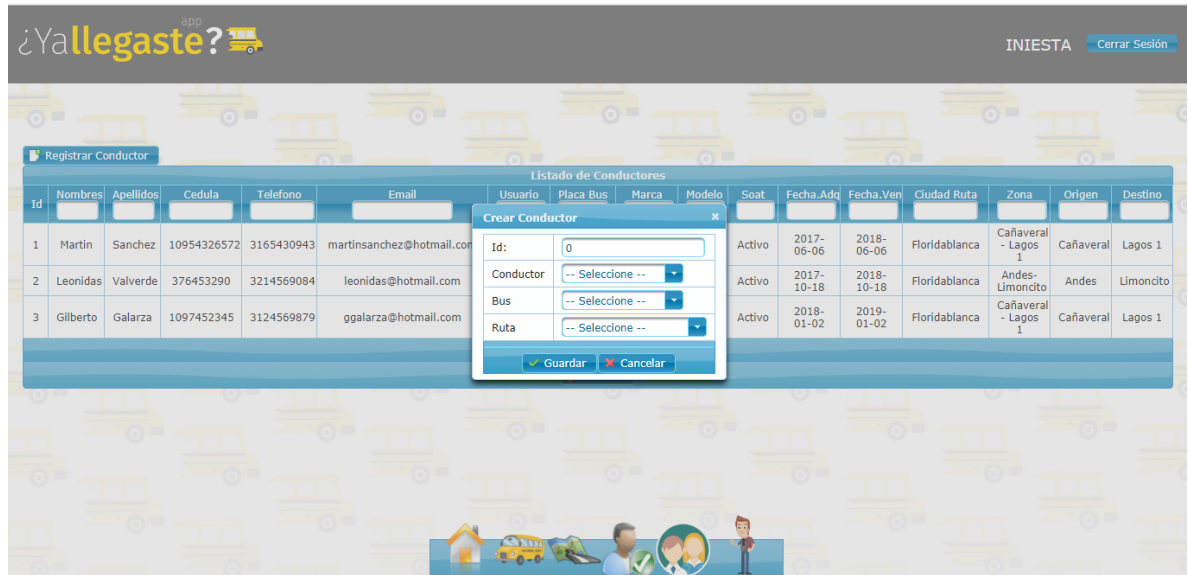
(1 of 1) [1](#) [Editar](#) [Eliminar](#)



Registrar Conductor.

Igualmente que en la función de registrar acudiente, el administrador debe registrar los acudiente que aún no lo están; de nuevo pide llenar varios campos en donde está el conductor que se quiere registrar con su respectivo bus y su respectiva ruta, en donde igualmente van a salir unas listas en donde están los buses y las rutas creadas del sistema.

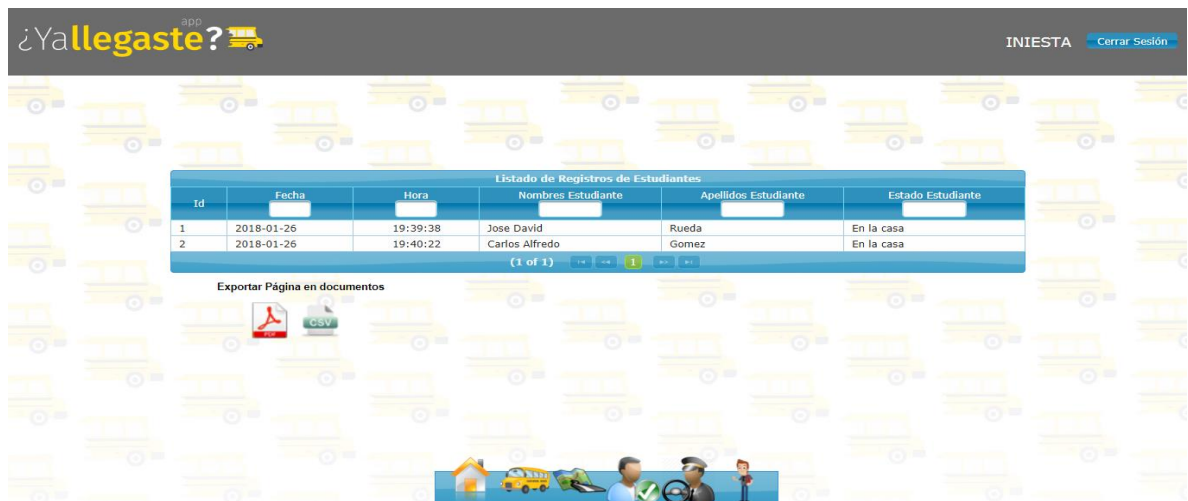
Figura 38. Registrar conductor. Producto final



Historial Estudiantes

Aquí con esta función, nos referimos a todo el historial de todos los estudiantes que están registrados en el sistema; es decir, aquí se van a visualizar todos los registros de cada vez que un estudiante se suba o se baje de un bus escolar, con sus respectivas fechas, horas y estado de los estudiantes. Además hay dos funciones adicionales, como exportar el documento en PDF y en Excel, inclusive filtrando los campos que se necesiten o deseen

Figura 39. Historial de estudiantes. Administrador. Producto final



5.7.4.2. Usuario Acudiente

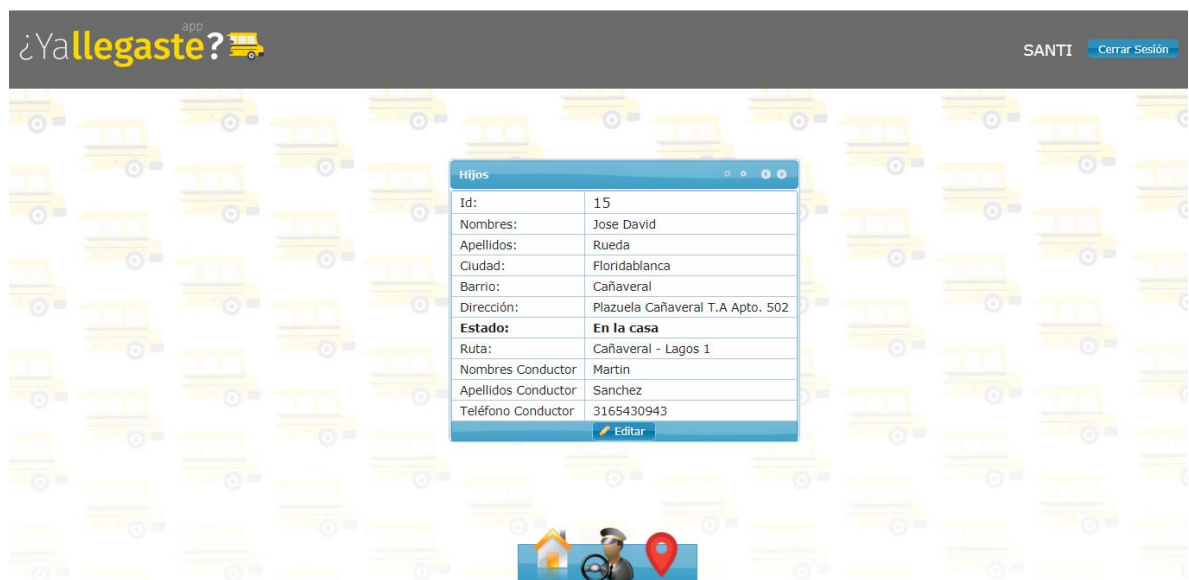
Ventana Principal

Figura 40. Ventana Principal Acudiente. Producto final



Estudiantes

Figura 41. Hijos de Acudientes. Producto final



Conductores

Figura 42. Conductores de Acudiente. Producto final

Listado de Conductores										
Id	Nombres	Apellidos	Teléfono	E-mail	Placa Bus	Soat	Ciudad Ruta	Zona	Origen	Destino
1	Martin	Sanchez	3165430943	martinsanchez@hotmail.com	XME-090	Activo	Floridablanca	Cañaveral - Lagos 1	Cañaveral	Lagos 1
2	Leonidas	Valverde	3214569084	leonidas@hotmail.com	ZAE-136	Activo	Floridablanca	Andes-Limoncito	Andes	Limoncito
3	Gilberto	Galarza	3124569879	ggalarza@hotmail.com	WXV-488	Activo	Floridablanca	Cañaveral - Lagos 1	Cañaveral	Lagos 1

Modificar Estudiantes

Figura 43. Modificar hijos de Acudientes. Producto final

Hijo

Modificar Hijo

Id: 15

Nombres: Jose David

Apellidos: Rueda

Ciudad: Floridablanca

Barrio: Cañaveral

Dirección: Plazuela Cañaveral T.A Ap

Conductores: martin

Guardar Cancelar

Historial hijos

En esta ventana podemos observar todo el historial o todos los registros que puedan tener los hijos de acudiente y además se pueden filtrar por el campo que se desee, facilitando lo que se quiera observar e inclusive analizando que días faltó un estudiante o hijo al transporte escolar.

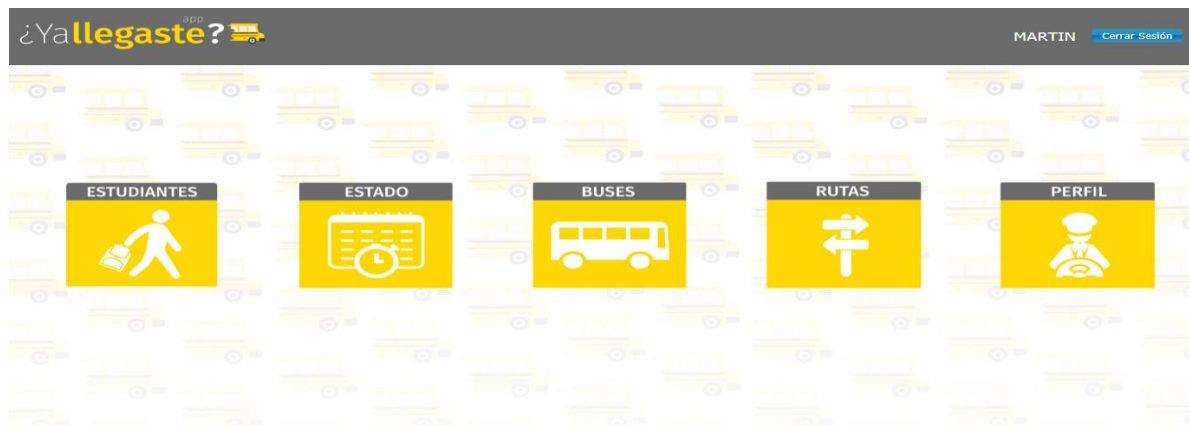
Figura 44. Historial hijos. Producto final

Listado de Registros de Hijos					
Id	Fecha	Hora	Nombres Estudiante	Apellidos Estudiante	Estado Estudiante
1	2018-01-26	19:39:38	Jose David	Rueda	En la casa
2	2018-01-26	19:40:22	Carlos Alfredo	Gomez	En la casa

5.7.4.3. Usuario conductor

Ventana Principal

Figura 45. Ventana principal Conductor. Producto final



Estudiantes

Figura 46. Estudiantes de conductores. Producto final

Id	Nombres	Apellidos	Ciudad	Barrio	Dirección	Estado	Nombres Acudiente	Apellidos	Teléfono	Email
15	Jose David	Rueda	Floridablanca	Cañaveral	Plazuela Cañaveral T.A Apto. 502	En la casa	Santiago	Delgado Mejia	3176709541	santhiago.sdm@gmail.com

(1 of 1)

Navigation icons: Home, School Bus, Location, Profile.

Perfil

Figura 47. Perfil Conductor. Producto final

Datos Personales	
Id	3
Nombres	Martin
Apellidos	Sanchez
Email	martinsanchez@hotmail.com
Telefono	3165430943
Cedula	10954326572
Editar	

Navigation icons: Home, School Bus, Location, Profile.

Buses

En esta ventana, se visualiza solo los buses que tiene adjudicado el usuario conductor que ingresó a la aplicación

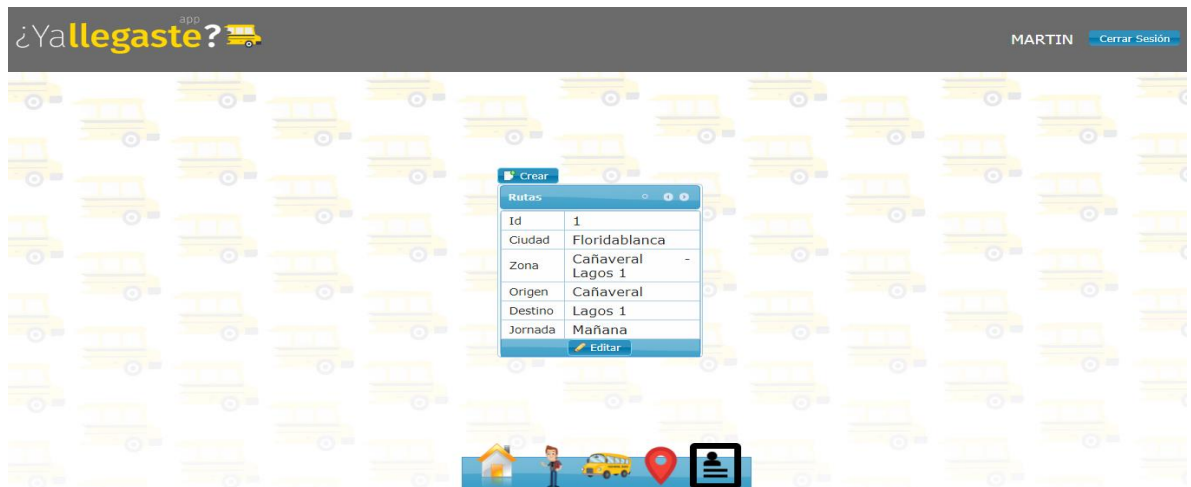
Figura 48. Buses de Conductor. Producto final



Rutas

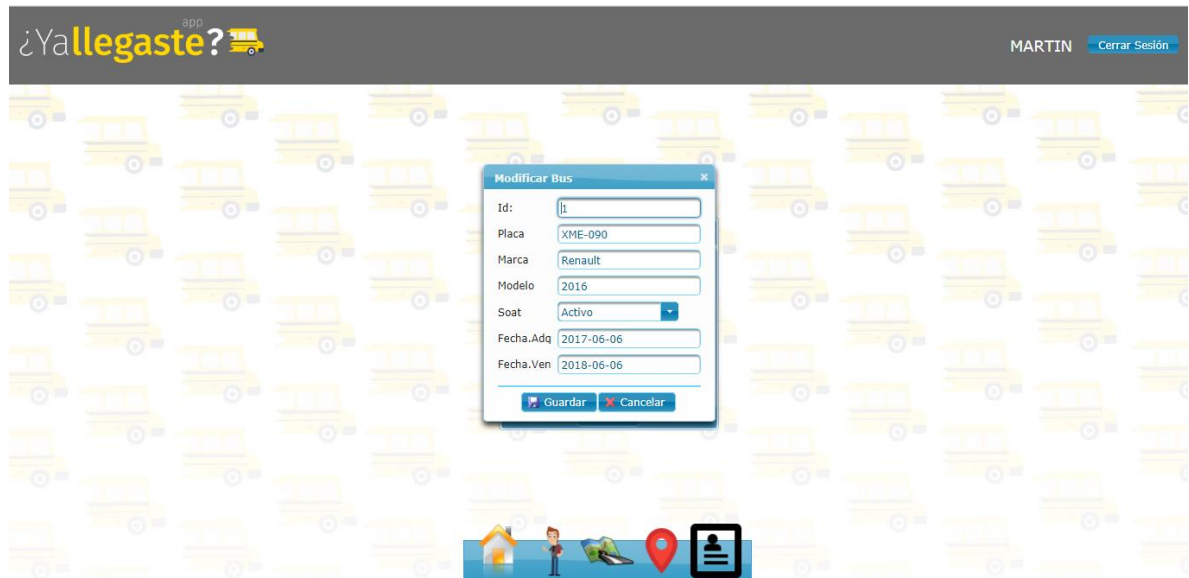
En esta ventana, se visualiza solo las rutas que tiene adjudicado el usuario conductor que ingresó a la aplicación

Figura 49. Rutas de conductor. Producto final



Modificar Buses

Figura 50. Modificar buses de Conductor. Producto Final



Modificar Rutas

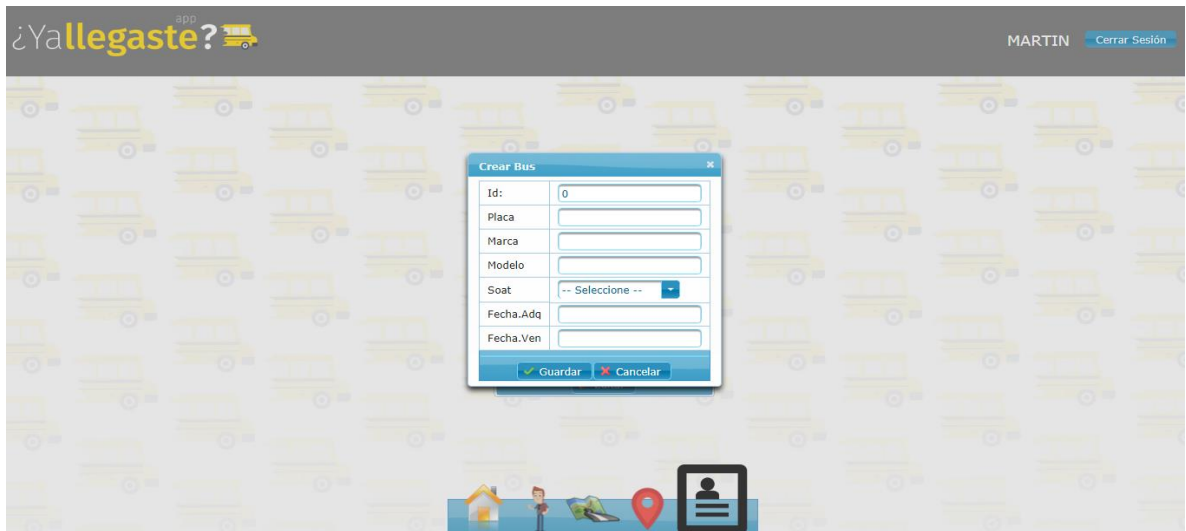
Figura 51. Modificar Rutas de conductor. Producto Final



Crear Bus

En esta ventana el conductor puede crear un bus, pero no va a quedar registrado a su cargo; para que quede registrado a su cargo, el administrador debe registrar el bus a cargo de él

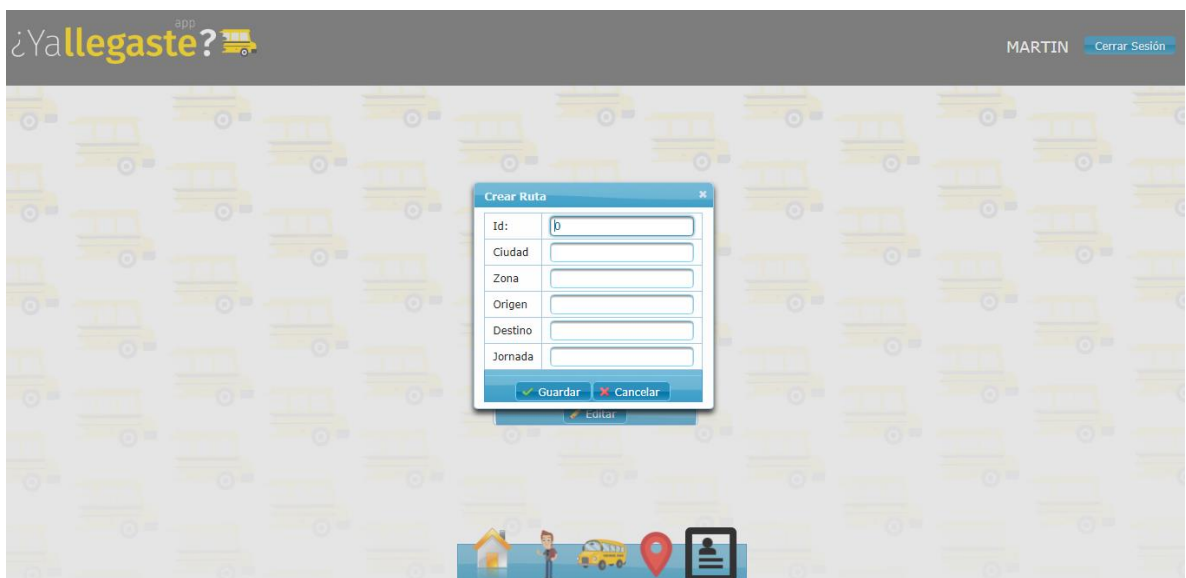
Figura 52. Crear bus Conductores. Producto Final



Crear Ruta

De igual manera que la función crear bus, está la función crear ruta

Figura 53. Crear ruta de Conductor. Producto Final



Historial de estudiantes

Se observa solo un registro de un estudiante porque solo hay uno, sin embargo a medida que se empiecen a agregar registros de los estudiantes, van a salir en esta pantalla todos los registros de los estudiantes que este conductor tiene a su cargo

Figura 54. Historial de estudiantes de Conductor. Producto Final



The screenshot shows the '¿Ya llegaste?' app interface. At the top left is the logo '¿Ya llegaste? app' with a bus icon. At the top right, the user name 'MARTIN' and a 'Cerrar Sesión' button are visible. The main content area features a table titled 'Listado de Registros de Estudiantes' with the following data:

Id	Fecha	Hora	Nombres Estudiante	Apellidos Estudiante	Estado Estudiante
1	2018-01-26	19:39:38	Jose David	Rueda	En la casa

Below the table, there is a pagination control showing '(1 of 1)' and navigation arrows. At the bottom of the screen, there is a navigation bar with icons for a home, a person, a bus, and a document.

6. PRUEBAS DE FUNCIONALIDAD

Se implementaron pruebas de funcionalidad para cada prototipo que se desarrolló, dependiendo de lo que se quería con cada uno de estos, se ejecutaron estas pruebas para verificar si se habían cumplido con los requisitos; las pruebas fueron las siguientes:

6.1. PRUEBAS DEL PRIMER PROTOTIPO

Tabla 36. Pruebas primer prototipo

CASOS DE PRUEBA	RESULTADOS
Inicio de sesión de usuarios	Exitoso
Clasificación de información según usuario que inicie	Exitoso
Mostrar información de la base de datos para cada usuario	Exitoso

6.2. PRUEBAS DEL SEGUNDO PROTOTIPO

Tabla 37. Pruebas segundo prototipo

CASOS DE PRUEBA	RESULTADOS
Creación de página de registro para personas sin usuario	Exitoso
Creación de usuarios por parte del administrador	Exitoso
Modificación de usuarios por parte del administrador	Exitoso
Eliminación de usuarios por parte del administrador	Exitoso

Creación de estudiantes por parte del administrador	Exitoso
Modificación de estudiantes por parte del administrador	Exitoso
Eliminación de estudiantes por parte del administrador	Exitoso
Mostrar conductores desde usuario acudiente	Exitoso
Mostrar estudiantes o hijos de un acudiente	Exitoso
Mostrar estudiantes que pertenecen a una ruta o conductor	Exitoso

6.3. PRUEBAS DEL TERCER PROTOTIPO

Tabla 38. Pruebas tercer prototipo

CASOS DE PRUEBA	RESULTADOS
Mejorar interfaz de usuario de inicio de sesión de usuarios	No Exitoso
Representar de una manera más visible y óptima del espacio los hijos de un acudiente	Exitoso
Mostrar Perfil del conductor	Exitoso

6.4. PRUEBAS DEL PRODUCTO FINAL

Tabla 39. Pruebas producto final

CASOS DE PRUEBA	RESULTADOS
Mejorar diseño de inicio de sesión	Exitoso
Quitar iconos de editar y eliminar por cada registro y en lugar colocar dos botones de editar y eliminar	Exitoso
Seleccionar registro y activar botón de editar y eliminar	Exitoso
Mejorar diseño de las ventanas de usuario, acudientes, estudiantes, conductores	Exitoso
Crear páginas para poder crear instancias de buses y rutas por parte del usuario	Exitoso
Verificar la seguridad de no poder entrar a páginas mediante la URL	Exitoso
Permitir al usuario administrador registrar Conductores y Acudientes con sus respectivos campos	Exitoso
Visualizar el estado actual de los estudiantes desde cualquier usuario	Exitoso
Visualizar historial de registros de los estudiantes desde cualquier usuario	Exitoso
Permitir crear buses y rutas al conductor, pero sin inscribirlas	Exitoso
Mejorar diseño de la página de perfil del conductor	Exitoso

Modificar perfil de un conductor desde la sesión del conductor	Exitoso
Permitir generar archivos PDF del historial de registros de estudiantes desde administrador	Exitoso
Permitir generar archivos Excel del historial de registros de estudiantes desde administrador	Exitoso
Colocar filtros a los campos de los usuarios y estudiantes	Exitoso

7. CONCLUSIONES

- ✓ Se creó una aplicación web combinada con la tecnología Arduino que permite facilitar y automatizar el registro de estudiantes en el sector del transporte escolar mediante un chip RFID y un lector RFID, la cual ayuda a padres de familia, colegio y conductores a conocer en todo momento si su hijo, estudiante o pasajero ya llegó al lugar destinado o no ha llegado.
- ✓ Se comprobó que el uso de frameworks ayudan a los programadores de aplicaciones y de software a realizar de manera más óptima los proyectos que se plantean y sobre todo facilitan el trabajo de crear componentes web.
- ✓ Se comprobó que existen muchas tecnologías que ayudan a la realización de proyectos de aplicaciones y que además aprenderlas, requiere una gran dedicación de tiempo y esmero para lograr el aprendizaje de estas tecnologías.
- ✓ A nivel personal me doy cuenta de que a pesar de haber adquirido conocimientos de ingeniería sistemas durante toda la carrera con la ayuda de los profesores que hay en la universidad, son muchos los campos y tecnologías que existen por explorar y aprender, para tratar cada vez ser un mejor profesional y desempeñarme mejor en lo que es la ingeniería de sistemas.
- ✓ Por último, me queda como reflexión que no solo es importante aprender una tecnología o un lenguaje de programación, si no que para la realización de un proyecto es aún más importante aplicar los conceptos y prácticas que se vieron durante la carrera, como lo son: Manejo y aplicación de bases de datos, la ingeniería del software con todas sus etapas de análisis y diseño, realizar buenas prácticas de programación, entre muchos más conceptos.

8. RECOMENDACIONES

- ✓ Con el fin de cada vez hacer una aplicación más completa, se propone para versiones posteriores realizar un módulo o implementar funciones que permitan observar estadísticamente datos como la hora de inicio de las rutas, la hora de llegada de las rutas al colegio, los días que faltaron los estudiantes al colegio según el registro del bus escolar, por períodos de tiempo establecidos como puede ser, por meses, por semanas, por semestres, etc.

- ✓ Se recomienda tener en cuenta para próximas versiones, la inclusión del usuario auxiliar del bus escolar, con el fin de que la persona que va en la parte de atrás del bus acompañando y vigilando la seguridad de los niños en el transporte, tenga acceso a la aplicación y a toda la información pertinente del sistema de transporte escolar.

- ✓ Se recomienda escalar la aplicación web a una aplicación móvil, de modo que los usuarios tengan la facilidad de disfrutar de todas las funciones de la aplicación web mediante el celular de cada persona, y así hacer más completo el mejoramiento del sistema de registro de estudiantes en transportes escolares.

- ✓ Además de escalar la aplicación web a aplicación móvil también se recomienda que aprovechando la tecnología Arduino con la placa SIM 900, se envíe un mensaje de texto al celular registrado del padre de familia cada vez que su hijo pase la tarjeta por el sensor, avisándoles del estado de cada uno de sus hijos. De esta forma se ahorran ir a la aplicación cada vez que quieran saber el estado de los hijos.

- ✓ También se recomienda para próximas versiones implementar un rastreo del bus escolar donde va cada estudiante por medio de GPS, de tal forma que los padres de familia conozcan en todo momento por dónde va el bus escolar que transporta a sus hijos

- ✓ Por último, aprovechando que exista la aplicación móvil, se recomienda implementar alguna manera de que el conductor se pueda comunicar por medio de mensajes automáticos con el padre de familia de cada estudiante por si se llega a retrasar la ruta o se presenta algún tipo de inconveniente.

BIBLIOGRAFÍA

ABC. ¿Qué es y por qué es importante la memoria RAM? [en línea]. [Consultado: 19 de Mayo de 2017]. Disponible en: <http://www.abc.es/tecnologia/consultorio/20150129/abci-memoria-RAM-que-es-para-que-sirve-201501281425.html>

ALVAREZ, Miguel Angel. Qué es MVC. [en línea]. [Consultado: 22 de Mayo de 2017]. Disponible en: <https://desarrolloweb.com/articulos/que-es-mvc.html>

APRENDIENDO ARDUINO. Qué es Arduino. [en línea]. [Consultado: 17 de Mayo de 2017]. Disponible en: <https://aprendiendoarduino.wordpress.com/2016/09/25/que-es-arduino/>

ARDUINO. ¿QUÉ ES ARDUINO?. [en línea]. [Consultado: 17 de Mayo de 2017]. Disponible en: <http://arduino.cl/que-es-arduino>

BY. ¿Qué es RFID?. [en línea]. [Consultado: 19 de Mayo de 2017]. Disponible en: <https://www.by.com.es/blog/que-es-rfid/>

CALENDAMAIA. PrimeFaces: framework sobre JSF 2.0. Primeros pasos. [en línea]. [Consultado: 25 de Mayo de 2017]. Disponible en: <https://www.genbetadev.com/frameworks/primefaces-framework-sobre-jsf-2-0-primeros-pasos>

GEARY, David; HORSTMANN, Cay. CORE JAVASERVER Faces. 3 ed. Santa Clara: Prentice Hall,2010

GUTIÉRREZ, Javier J. ¿Qué es un framework web?. [en línea]. [Consultado: 25 de Mayo de 2017]. Disponible en: http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf

JAVA ENTERPRISE UNIVERSIDAD DE ALICANTE. ¿Qué es Java Enterprise?. [en línea]. [Consultado: 22 de Mayo de 2017]. Disponible en: <http://www.ittech.ua.es/j2ee/2006-2007/jee.html>

IBM Knowledge Center. Objetos POJO (Plain old Java object). [en línea]. [Consultado: 22 de Mayo de 2017]. Disponible en: https://www.ibm.com/support/knowledgecenter/es/SS8PJ7_8.5.1/org.eclipse.jst.ejb.doc.user/topics/cpojosandee5.html

LIBROSWEB. 11.7. Persistencia de datos. [en línea]. [Consultado: 22 de Mayo de 2017]. Disponible en: http://librosweb.es/libro/algoritmos_python/capitulo_11/persistencia_de_datos.html

MEGINO BARQUINERO, José María. Agregación vs Composición en diagramas de clases. UML. [en línea]. [Consultado: 26 de Mayo de 2017]. Disponible en: <https://www.seas.es/blog/informatica/agregacion-vs-composicion-en-diagramas-de-clases-uml/>

ORACLE JUNIORS. ¿Qué es JPA (JAVA PERSISTENCE API)?. [en línea]. [Consultado: 22 de Mayo de 2017]. Disponible en: <http://oraclejuniors.blogspot.com.co/2014/11/que-es-jpa-java-persistence-api.html>

PESQUERA, Carlos. ¿QUÉ ES UN POJO, EJB Y UN BEAN?. [en línea]. [Consultado: 22 de Mayo de 2017]. Disponible en: <http://carlospesquera.com/que-es-un-pojo-ejb-y-un-bean/>

SHERLIN.XBOT.ES. 1. ¿Qué es un microcontrolador?. [en línea]. [Consultado: 17 de Mayo de 2017]. Disponible en: <http://sherlin.xbot.es/microcontroladores/introduccion-a-los-que-es-un-microcontrolador>

SZNAJDLEDER, Pablo Augusto. Java a fondo: ESTUDIO DEL LENGUAJE Y DESARROLLO DE APLICACIONES: INCLUYE INTRODUCCIÓN A HIBERNATE Y SPRING. 2 ed. Mexico: Alfaomega, 2013.

TUPROGRAMACION.COM. ¿QUÉ ES UN ORM?. [en línea]. [Consultado: 25 de Mayo de 2017]. Disponible en: <http://www.tuprogramacion.com/glosario/que-es-un-orm/>

Universidad de Alicante. Introducción a JavaServer Faces. [en línea]. [Consultado: 25 de Mayo de 2017]. Disponible en: <http://www.jtech.ua.es/j2ee/publico/jsf-2012-13/sesion01-apuntes.html>

UNIVERSITAT DE BARCELONA. ¿Qué son las metodologías de desarrollo de software?. [en línea]. [Consultado: 25 de Mayo de 2017]. Disponible en: <https://www.obs-edu.com/int/blog-project-management/metodologia-agile/que-son-las-metodologias-de-desarrollo-de-software>

VERGARA JURADO, Alida. JPA vs Hibernate. ¿Cuál es la diferencia?. [en línea]. [Consultado: 22 de Mayo de 2017]. Disponible en: <https://www.facilcloud.com/noticias/jpa-vs-hibernate/>

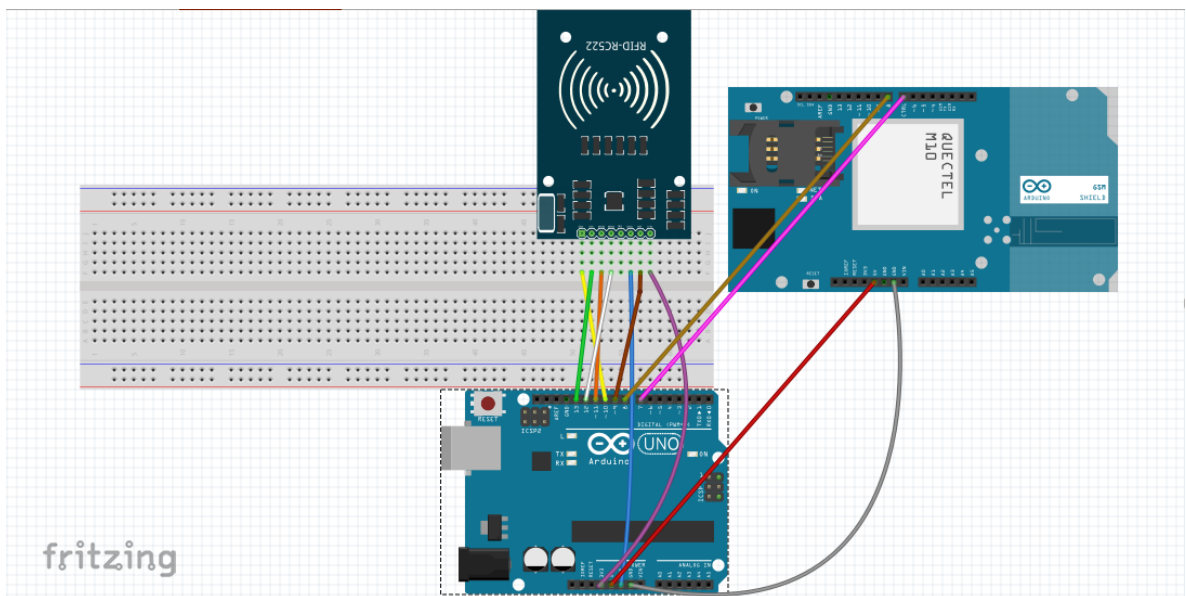
YERKO MUÑOZ, Felipe Fernández. Qué es JDBC?. [en línea]. [Consultado: 25 de Mayo de 2017]. Disponible en: <https://users.dcc.uchile.cl/~Imateu/CC60H/Trabajos/jfernand>

ANEXOS

Anexo A. Conexiones Arduino

El Arduino uno se conecta por medio de las conexiones que se ven en el siguiente esquema entre la placa RFID-RC522 y la placa SIM900 GPRS/GSM. Además, para energizar la sim card que se inserta en la SIM900 se necesitar alimentar la placa de Arduino uno con una alimentación externa de 12 v y 2 A.

FIGURA 55. Conexiones Arduino



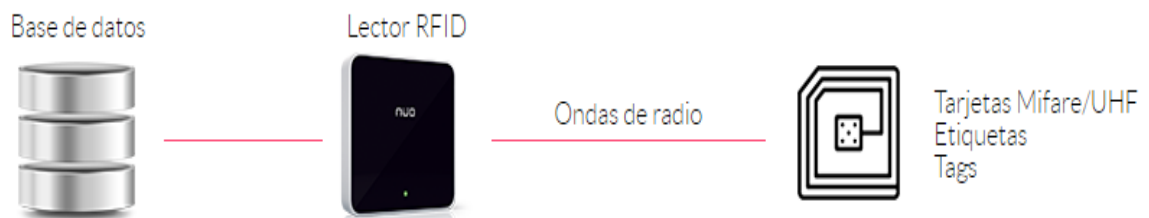
Fuente: Autor

Anexo B. Lector RFID-RC522

RFID significa Radio Frequency Identification, es decir, Identificación por radiofrecuencia. El objetivo de la tecnología RFID es identificar mediante un lector a distancia, un tag que puede estar insertado desde en una tarjeta hasta en un llavero o en todos los materiales que se nos ocurra.²⁵

En el siguiente esquema se observa como es la interacción y lo que sucede cuando se quiere leer un tag RFID mediante un lector RFID. Se observa que el lector siempre hace peticiones mediante ondas de radio a las tarjetas RFID, las cuales responden al lector mediante las mismas ondas de radio, pasando la información que contienen al lector, y de ahí enviando esa información a la base de datos asociada.

FIGURA 56. Interacción RFID



Fuente: BY. ¿Qué es RFID?. [en línea]. [Consultado: 22 de Febrero de 2018]. Disponible en: <https://www.by.com.es/blog/que-es-rfid/>

El código de identificación que contiene un tag RFID es único. Además, esta tecnología funciona mediante antenas que contienen cada dispositivo al interactuar

²⁵ BY. ¿Qué es RFID?. [en línea]. [Consultado: 22 de Febrero de 2018]. Disponible en: <https://www.by.com.es/blog/que-es-rfid/>

mediante radiofrecuencia. Las formas y características de las antenas dependen de la frecuencia en las que funcionen.

Bandas de frecuencia utilizadas en RFID:

FIGURA 57. Banda de frecuencias

Banda de frecuencias	Descripción	Rango
125 kHz	LF (Baja Frecuencia)	Hasta 50 cm.
13,56 MHz	HF (Alta Frecuencia)	De 8 cm.
400 MHz - 1.000 MHz	UHF (Ultra Alta Frecuencia)	De 3 a 10 m.
2,45 GHz - 5,4 GHz	Microondas	Más de 10 m.

Fuente: Fuente: BY. ¿Qué es RFID?. [en línea]. [Consultado: 22 de Febrero de 2018]. Disponible en: <https://www.by.com.es/blog/que-es-rfid/>

El módulo lector RFID- RC522 que utilizamos en este proyecto se alimenta con 3.3V y puede ser controlado a través del protocolo SPI o mediante un puerto UART. El RC522 utiliza un sistema de modulación y demodulación avanzado para todo tipo de dispositivos pasivos de 13.56 MHz.

La tarjeta que contiene el chip RFID tiene 64 bloques de memoria (0-63) donde se puede hacer lectura y/o escritura. Cada bloque de memoria tiene la capacidad de almacenar hasta 16 Bytes. Respecto al número de identificación consiste en 6 valores hexadecimales que también se pueden mostrar en forma decimal.²⁶

²⁶ Orlando. Módulo RFID-RC522 RF con Arduino UNO SPI. [en línea]. [Consultado: 22 de Febrero de 2018]. Disponible en: <https://hetpro-store.com/TUTORIALES/modulo-lector-rfid-rc522-rf-con-arduino/>

Cabe aclarar que para este proyecto se utilizó chips RFID pasivos por lo cual es necesario acercar las tarjetas RFID al lector a una distancia no mayor a 60mm.

En la siguiente figura se muestra las conexiones del Arduino uno a la placa RFID-RC522

FIGURA 58. RFID-RC522 conexión con Arduino uno

ARDUINO UNO	RFID RC552
DIGITAL PIN #10	SDA
DIGITAL PIN #13	SCK
DIGITAL PIN #11	MOSI
DIGITAL PIN #12	MISO
N/A	IRQ
POWER GND	GND
DIGITAL PIN #5	RST
POWER 3.3 V	3.3 V

Fuente: Fuente: Orlando. Módulo RFID-RC522 RF con Arduino UNO SPI. [en línea]. [Consultado: 22 de Febrero de 2018]. Disponible en: <https://www.by.com.es/blog/que-es-rfid/>

Anexo C. SIM 900 GPRS/GSM

La placa SIM900 GPRS/GSM es una tarjeta compacta de comunicación inalámbrica que, además, es compatible con todos los modelos de Arduino uno. Esta placa está basada en el módulo SIM900 GSM 4. El GPRS está configurado y es controlada por vía UART usando comandos AT.²⁷ La placa es Quad-Band 850/900/1800/1900 MHz.

Para la transmisión de datos entre la SIM900 y Arduino uno se utilizan los pines 7 y 8, por lo cual hay que asegurarse de que los jumpers de la tarjeta GSM estén posicionados en D7 y D8, lo que permite activar los pines mencionados anteriormente.

FIGURA 59. Jumper TX(D7) y RX(D8)

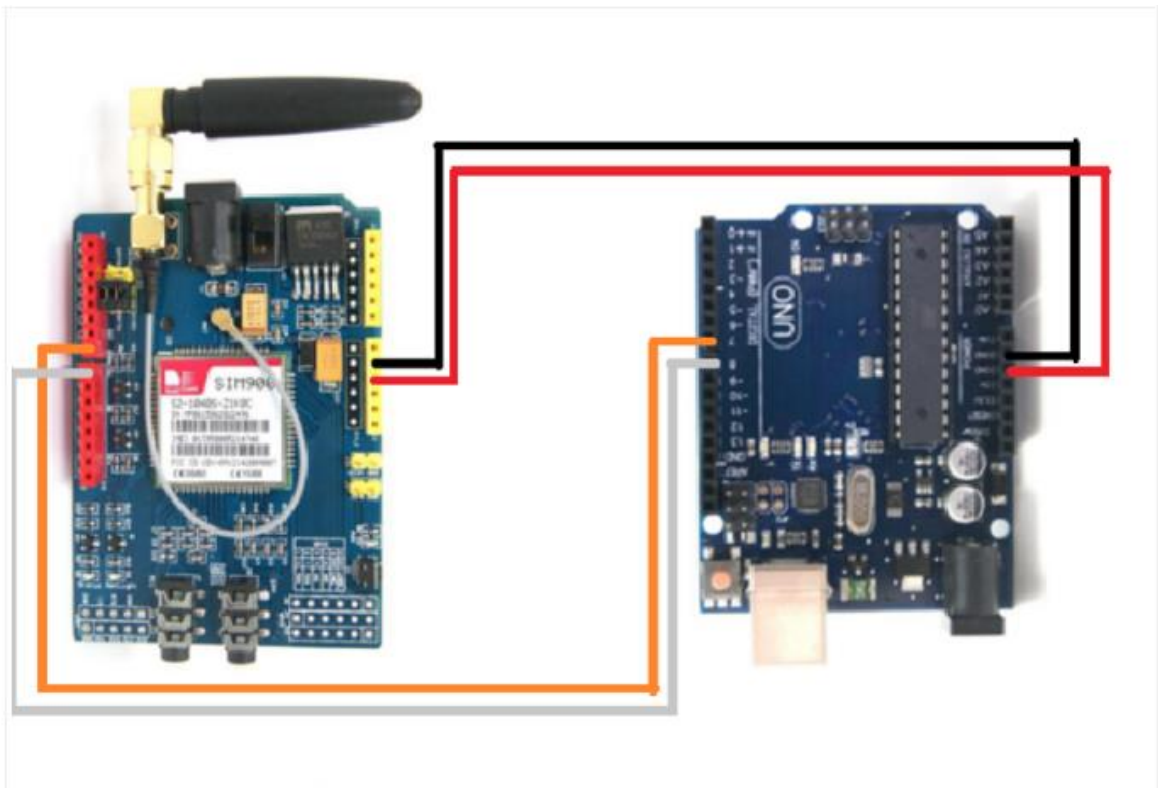


Fuente: LARA, Eduardo. SIM900 GSM GPRS Shield con Arduino UNO. [en línea]. [Consultado: 22 de Febrero de 2018]. Disponible en: <https://hetpro-store.com/TUTORIALES/sim900-gsm-shieldarduino/>

²⁷ LARA, Eduardo. SIM900 GSM GPRS Shield con Arduino UNO. [en línea]. [Consultado: 22 de Febrero de 2018]. Disponible en: <https://hetpro-store.com/TUTORIALES/sim900-gsm-shieldarduino/>

Por último, en la siguiente imagen se muestra las conexiones que se deben hacer para que interactúe la SIM900 con Arduino Uno de forma correcta. Hay que recalcar que detrás de la SIM900 se debe ingresar una Sim Card con datos móviles que permitan enviar los datos a la base de datos que se tenga en el servidor y además, esta Sim Card se debe energizar mediante una alimentación externa que se conecta al Arduino uno de 12V y 2A

FIGURA 60. Diagrama de conexión entre Arduino uno y SIM900



Fuente: LARA, Eduardo. SIM900 GSM GPRS Shield con Arduino UNO. [en línea]. [Consultado 22 de Febrero de 2018]. Disponible en: <https://hetpro-store.com/TUTORIALES/sim900-gsm-shieldarduino/>