

UNIVERSIDAD NACIONAL PEDRO RUIZ GALLO
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
ESCUELA PROFESIONAL DE INGENIERÍA EN
COMPUTACIÓN E INFORMÁTICA



TESIS

**Desarrollo de un sistema informático en tiempo real de geolocalización,
control y monitoreo para el servicio de transporte de estudiantes en la
ciudad de Chiclayo**

**Presentada para optar el Título Profesional de
Ingeniero en Computación e Informática**

Investigadores:

Bonilla Coronado, Bernardo Amado

Diaz Nuñez, Hans Pritton

Asesor:

Mg. Ing. Valdivia Salazar, Carlos Alberto

Lambayeque, 2020

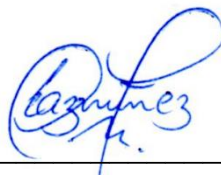
**Desarrollo de un sistema informático en tiempo real de geolocalización,
control y monitoreo para el servicio de transporte de estudiantes en la
ciudad de Chiclayo**

**Presentada para optar el Título Profesional de
Ingeniero en Computación e Informática**

Presentada por:

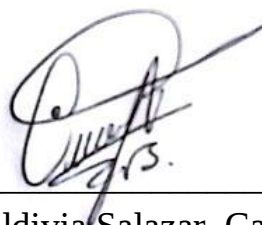


Bach. Bonilla Coronado, Bernardo Amado
Autor



Bach. Diaz Nuñez, Hans Pritton
Autor

Asesorado por:



Mg. Ing. Valdivia Salazar, Carlos Alberto
Asesor

Lambayeque, 2020

**Desarrollo de un sistema informático en tiempo real de geolocalización,
control y monitoreo para el servicio de transporte de estudiantes en la
ciudad de Chiclayo**

**Presentada para optar el título profesional de
Ingeniero en Computación e Informática**

Aprobada por:



Dr. Ing. Armando Moreno Heredia
Presidente



Dr. Ing. Gilberto Carrión Barco
Secretario



Mg. Ing. Denny John Fuentes Adrianzén
Vocal

Lambayeque, 2020



ACTA DE SUSTENTACIÓN N°021-2020-D/FACFyM

(Sustentación Autorizada por Resolución N° 246 - 2020-D/FACFyM)

En la ciudad de Lambayeque, siendo las 11:00 a.m. del día 05 de marzo del 2020 se reunieron en Sala de Videoteca del laboratorio de Física de la FACFyM los miembros del Jurado designados mediante Resolución N° 1609-2018-D/FACFyM, los docentes:

Dr. Ing. Armando José Moreno Heredia	Presidente
Dr. Ing. Gilberto Carrión Barco	Secretario
Mg. Ing. Denny John Fuentes Adrianzén	Vocal

Para recibir la tesis titulada:

"Desarrollo de un Sistema Informático en Tiempo Real de Geocalización, control y Monitoreo para el Servicio de Transporte de Estudiantes en la ciudad de Chiclayo"

desarrollada por los Bachilleres en Computación e Informática, **Bonilla Coronado Bernardo Amado y Díaz Nuñez Hans Pritton.**

Después de escuchar la exposición y las respuestas a las preguntas formuladas por los miembros del Jurado, se acordó APROBAR el trabajo por UNANIMIDAD con el calificativo de MUY BUENO.

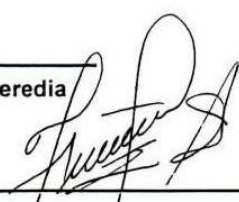
En consecuencia, los Bachilleres en referencia quedan aptos para recibir el Título Profesional de **Ingeniero en Computación e Informática**, de acuerdo a la Ley Universitaria, el Estatuto y Reglamento de la Universidad Nacional Pedro Ruiz Gallo de Lambayeque.

Observaciones:

Para constancia del hecho firman.


Dr. Ing. Armando José Moreno Heredia
Presidente


Dr. Ing. Gilberto Carrión Barco
Secretario


Mg. Ing. Denny John Fuentes Adrianzén
Vocal



CONSTANCIA DE VERIFICACIÓN DE ORIGINALIDAD

Yo, M. Sc. Carlos Alberto Valdivia Salazar _____ usuario revisor de:

Tesis ☒

Trabajo de Suficiencia Profesional ☐

Trabajo Académico ☐

Titulado Desarrollo de un sistema informático en tiempo real de geolocalización, control y monitoreo para el servicio de transporte de estudiantes en la ciudad de Chiclayo _____

Cuyo(s) autor(es) es(son):

Bach. Hans Pritton Díaz Nuñez DNI° 46993425,

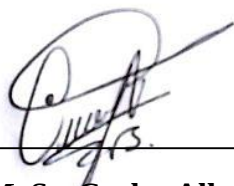
Bach. Bernardo Amado Bonilla Coronado DNI° 7 1 8 7 4 2 9 1 ,

declaro que la evaluación realizada por el Programa informático, ha arrojado un porcentaje de similitud 17____%, verificables en el Resumen del Reporte Automatizado de similitudes que se acompaña.

El(La/Los/Las) suscrito(a/s/as) analizó y concluyó que cada una de las coincidencias detectadas dentro del porcentaje de similitud permitido no constituyen plagio y que el documento cumple con la integridad científica y con las normas para el uso de citas y referencias establecidas en los protocolos respectivos.

Se cumple con adjuntar el Recibo Digital a efectos de la trazabilidad respectiva del proceso.

Lambayeque, 15____de enero____del 2026____



Nombres y Apellidos: M. Sc. Carlos Alberto Valdivia Salazar

DNI°: 16621586

ASESOR

Defina la modalidad con [X]

Adjuntar

- *Reporte Automatizado de similitudes*
- *Recibo Digital*

Informe_Final_Tesis

INFORME DE ORIGINALIDAD

17 %	15 %	8 %	7 %
INDICE DE SIMILITUD	FUENTES DE INTERNET	PUBLICACIONES	TRABAJOS DEL ESTUDIANTE

FUENTES PRIMARIAS

1	ribuni.uni.edu.ni Fuente de Internet	4 %
2	repositorio.unesum.edu.ec Fuente de Internet	2 %
3	repositorio.ucv.edu.pe Fuente de Internet	1 %
4	Submitted to University of Westminster Trabajo del estudiante	1 %
5	dspace.cordillera.edu.ec Fuente de Internet	1 %
6	repositorio.upea.bo Fuente de Internet	1 %
7	repositorio.utc.edu.ec Fuente de Internet	1 %
8	randomtable.github.io Fuente de Internet	1 %
9	id.scribd.com Fuente de Internet	1 %



M. SC. Carlos Alberto Valdivia Salazar
DNI N° 16621586
Asesor

10	Submitted to University of Northampton Trabajo del estudiante	<1 %
11	repository.bsi.ac.id Fuente de Internet	<1 %
12	Submitted to University of Hertfordshire Trabajo del estudiante	<1 %
13	Submitted to City University Trabajo del estudiante	<1 %
14	repositorio.usanpedro.edu.pe Fuente de Internet	<1 %
15	eprints.akakom.ac.id Fuente de Internet	<1 %
16	repository.uksw.edu Fuente de Internet	<1 %
17	Paredes, Luis Hernando Lozano. "Governing Spaces of their Own: Platform Drivers in Bogotá, Colombia", University of Technology Sydney (Australia), 2024 Publicación	<1 %
18	Christian Andrés Artieda Guachamin, Juan Sebastián Reza Morales, Henry Daniel Rojas Cisneros. "Sistema móvil para la oferta de recorridos con rutas fijas", REVISTA ODIGOS, 2021 Publicación	<1 %



M. SC. Carlos Alberto Valdivia Salazar
DNI N° 16621586
Asesor

19	Submitted to Quezon City University Trabajo del estudiante	<1 %
20	www.omsemut.com Fuente de Internet	<1 %
21	Submitted to University of Central Lancashire Trabajo del estudiante	<1 %
22	Submitted to Universidad Cesar Vallejo Trabajo del estudiante	<1 %
23	Submitted to Xavier University Trabajo del estudiante	<1 %
24	e-campus.iainbukittinggi.ac.id Fuente de Internet	<1 %
25	gitlab.cosanpa.pa.gov.br Fuente de Internet	<1 %
26	repo.uinbukittinggi.ac.id Fuente de Internet	<1 %
27	Luis Enrique Prado Naranjo, Juan Miguel Sánchez Toapanta, Rita Amalia Sánchez Toapanta. "Gestión en el proceso de búsqueda de eventos en un rango variable de 50 km de distancia", REVISTA ODIGOS, 2022 Publicación	<1 %
28	Submitted to Universidad Nacional Abierta y a Distancia, UNAD, UNAD Trabajo del estudiante	<1 %


 M. SC. Carlos Alberto Valdivia Salazar
 DNI N° 16621586
 Asesor

29 Chapilliquen Rodriguez, Monica Gladys. <1 %
"Competencias digitales en estudiantes con
diferentes estilos de aprendizaje, del septimo
ciclo de educacion secundaria, desarrolladas
a traves de la red social educativa Edmodo en
una institucion educativa publica de la Unidad
de Gestion Educativa Local N° 03, el ano
2015.", Pontificia Universidad Catolica del
Peru - CENTRUM Catolica (Peru), 2021
Publicación

30 "Enfoque de optimización para la transición <1 %
desde jornadas parciales a jornadas
completas en un sistema escolar", Pontificia
Universidad Catolica de Chile, 2018
Publicación

31 Submitted to University of Greenwich <1 %
Trabajo del estudiante

32 de Poma, Amparo Argüelles Romero. "Estado <1 %
Situacional del Area de Formación para el
Trabajo del Programa Educativo de la I.E.P.G.
Cristiana Dirigido a Adolescentes en Estado
de Vulnerabilidad del Distrito de Chorrillos",
Pontificia Universidad Catolica del Peru
(Peru), 2022
Publicación

33 Kaya, Özkan. "Development of a GSM and <1 %
GPS Based Vehicle Tracking-Routing System",
Dokuz Eylul Universitesi (Turkey), 2024
M. SC. Carlos Alberto Valdivia Salazar
DNI N° 16621586
Asesor



Recibo digital

Este recibo confirma que su trabajo ha sido recibido por **Turnitin**. A continuación podrá ver la información del recibo con respecto a su entrega.

La primera página de tus entregas se muestra abajo.

Autor de la entrega: Bonilla_coronado Diaz_nuñez
Título del ejercicio: Quick Submit
Título de la entrega: Informe_Final_Tesis
Nombre del archivo: TESIS_AMADO_HANS_2020.docx
Tamaño del archivo: 8.77M
Total páginas: 220
Total de palabras: 30,882
Total de caracteres: 202,668
Fecha de entrega: 01-abr.-2024 05:31p. m. (UTC-0500)
Identificador de la entrega: 2337269721

UNIVERSIDAD NACIONAL PEDRO RUIZ GALLO
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
ESCUELA PROFESIONAL DE INGENIERÍA EN
COMPUTACIÓN E INFORMÁTICA



TESIS

"Desarrollo de un Sistema Informático en Tiempo Real de
Gestión, Control y Monitoreo para el Servicio de Transporte de
Estudiantes en la Ciudad de Chiclayo"

Investigadores:
Ruth Bonilla Coronado, Bernardo Ananda
Ruth Diaz Nuñez, Hans Prión

Asesor:
Mg. Ing. Valdivia Salazar, Carlos Alberto

Lambayeque, 2020

M. Sc. Carlos Alberto Valdivia Salazar
DNI N° 16621586
Asesor

DECLARACIÓN JURADA DE ORIGINALIDAD

Nosotros, **Bernardo Amado Bonilla Coronado** y **Hans Pritton Diaz Nuñez**, Investigadores Principales y **Mg. Ing. Carlos Alberto Valdivia Salazar**, Asesor del Trabajo de Investigación **“Desarrollo de un sistema informático en tiempo real de geolocalización, control y monitoreo para el servicio de transporte de estudiantes en la ciudad de Chiclayo”**, declaramos bajo juramento que este trabajo no ha sido plagiado, ni contiene datos falsos. En caso se demostrara lo contrario, asumimos responsablemente la anulación de este informe y por ende el proceso administrativo a que hubiera lugar. Que puede conducir a la anulación del título o grado emitido como consecuencia de este informe.

Lambayeque, enero de 2020.

Nombre de Investigadores:

Bernardo Amado Bonilla Coronado

Hans Pritton Diaz Nuñez

Nombre de Asesor:

Mg. Ing. Carlos Alberto Valdivia Salazar

DEDICATORIA

A Dios, por permitirme haber cumplido este proyecto de tesis, a mis padres por siempre estar presentes, a mis hermanos, mi abuela Rosa que es un gran motivo para toda la familia, mis abuelos y mi hermana que no me acompañan físicamente pero se que espiritualmente siempre van estar para mi, mi tía Marisol y mi Hermana Geral que ahora nos a dado una alegría a toda la familia con la llegada de Joaquin.

Autor: Bernardo Amado Bonilla Coronado

A Dios, por ser el que me guió a cumplir mi principal objetivo de culminar mi proyecto de tesis, a mis padres por haberme apoyado en todas instancias de mi vida, y todas mis amistades que siempre han confiando en mi.

Autor: Hans Pritton Diaz Nuñez

AGRADECIMIENTO

En primera instancia, agradecer a Dios Todopoderoso, por permitirnos haber vivido y disfrutado de la vida universitaria; con la disciplina y ambición de haber terminado nuestra carrera universitaria y por guiar e iluminar nuestros caminos a cumplir con los objetivos trazados.

A nuestro docente, asesor y sobre todo un gran amigo, Ing. Carlos Alberto Valdivia Salazar por su gran apoyo, colaboración, capacidad, experiencia y conocimientos durante el proceso y desarrollo del presente trabajo de investigación.

A los integrantes del jurado, que nos dieron el apoyo y las sugerencias necesarias para mejorar este proyecto.

A nuestros docentes de la Escuela Profesional de Ingeniería en Computación e Informática, que con su conocimiento, contribuyeron con nuestra formación académica.

A nuestra alma mater, la “Universidad Nacional Pedro Ruiz Gallo”, y en especial a la “Escuela Profesional de Ingeniería en Computación e Informática”, que sigue formando profesionales competentes con el desarrollo de la humanidad.

Los Autores.

INDICE GENERAL

RESUMEN	24
ABSTRACT.....	25
INTRODUCCIÓN	26
Capítulo I: Diseño Teórico	27
1.1. Problemática de la investigación	27
1.1.1. Planteamiento del problema.....	27
1.2. Formulación del problema de investigación	31
1.3. Objetivos	31
1.3.1. Objetivo general.....	31
1.3.2. Objetivos específicos	31
1.4. Marco metodológico	31
1.4.1. Tipo de investigación.....	31
1.4.2. Hipótesis	32
1.4.3. Definición y operacionalización de variables	33
1.5. Marco teórico.....	35
1.5.1. Antecedentes.....	35
1.6. Bases teóricas.....	38
1.6.1. Conceptos generales	38
1.6.2. Metodología de desarrollo	40
1.6.3. Sistema de gestión de base de datos	41
1.6.4. Lenguaje de programación.....	46
1.6.5. Arquitectura de software.....	48
1.6.6. Entorno de desarrollo integrado.....	49
1.6.7. Servidor web	50
1.6.8. Tecnologías web	54

1.6.9. Arduino	58
1.7. Diseño de arquitectura	60
Capítulo II: Métodos y Materiales	61
2.1. Procedimiento para el desarrollo de la investigación	61
2.2. Población y muestra.....	62
2.3. Técnicas e instrumentos de la recolección de datos	62
2.3.1. Técnicas	62
2.3.2. Instrumentos.....	63
Capítulo III: Resultados y Discusión	64
3.1. Identificación de la situación actual de las unidades escolares.....	64
3.2. Configuración inicial para el desarrollo del sistema informático	65
3.2.1. Diseño de base de datos	65
3.2.2. Entorno de desarrollo integrado.....	66
3.2.3. Firebase	69
3.2.4. Servidor Web	73
3.2.5. Plataforma Arduino.....	76
3.3. Planificación y diseño	82
3.3.1. Establecimiento de historias de usuario	82
3.3.2. Iteración 1	83
3.3.3. Iteración 2	101
3.3.4. Iteración 3	116
3.3.5. Iteración 4	134
3.4. Diagrama de componentes.....	145
3.5. Diagrama de despliegue.....	146
3.6. Diagrama de modelo relacional	147
3.6.1. Diseño lógico de base de datos	147
3.6.2. Script de la base de datos.....	148

3.7. Implementación de funcionalidades	153
3.7.1. Mantenimiento de apoderados	153
3.7.2. Generación de contratos.....	185
3.7.3. Reporte de asignaciones.....	201
Capítulo IV: Costos y Beneficios	208
4.1. Análisis de costos.....	208
4.1.1. Costo de software.....	208
4.1.2. Costo de hardware	208
4.1.3. Costo de implementación módulo GPS	209
4.1.4. Costo de personal.....	209
4.1.5. Costo de servicios	210
4.1.6. Resumen de costos.....	210
4.2. Análisis de beneficios	211
4.2.1. Beneficios tangibles.....	211
4.2.2. Beneficios intangibles.....	211
4.3. Cálculo de VAN y TIR	211
4.3.1. Valor actual neto (VAN).....	211
4.3.2. Tasa interno de retorno (TIR)	212
4.3.3. Período de recuperación de inversión (PRI)	213
Capítulo V: Conclusiones	214
Capítulo VI: Recomendaciones	216
BIBLIOGRAFÍA REFERENCIADA.....	217
ANEXOS	219

INDICE DE TABLAS

Tabla 1 Vehículos autorizados para servicio de movilidad escolar Lima Metropolitana	30
Tabla 2 Operacionalización de variable dependiente	33
Tabla 3 Operacionalización de variable independiente	34
Tabla 4 Establecimiento de historias de usuario.....	82
Tabla 5 Historia N° 01: Mantenimiento de colegios	84
Tabla 6 Tarea N° 01: Diseño de interfaz de mantenimiento de colegios	84
Tabla 7 Tarea N° 02: Gestión de mantenimiento de colegios	85
Tabla 8 Historia N° 02: Mantenimiento de apoderados	85
Tabla 9 Tarea N° 03: Diseño de interfaz de mantenimiento de apoderados.....	86
Tabla 10 Tarea N° 04: Gestión de mantenimiento de apoderados	86
Tabla 11 Historia N° 03: Mantenimiento de escolares	87
Tabla 12 Tarea N° 05: Diseño de interfaz de mantenimiento de escolares	87
Tabla 13 Tarea N° 06: Gestión para el mantenimiento de escolares	88
Tabla 14 Historia N° 04: Mantenimiento de transportistas	88
Tabla 15 Tarea N° 07: Diseño de interfaz de mantenimiento de transportistas	89
Tabla 16 Tarea N° 08: Gestión para el mantenimiento de transportistas	89
Tabla 17 Historia N° 05: Mantenimiento de unidades escolares.....	90
Tabla 18 Tarea N° 09: Diseño de interfaz de mantenimiento de unidades escolares.....	90
Tabla 19 Tarea N° 10: Gestión para el mantenimiento de unidades escolares.....	91
Tabla 20 Historia N° 06: Mantenimiento de choferes	91
Tabla 21 Tarea N° 11: Diseño de interfaz de mantenimiento de choferes	92
Tabla 22 Tarea N° 12: Gestión para el mantenimiento de choferes	92
Tabla 23 Historia N° 07: Mantenimiento de rutas	93
Tabla 24 Tarea N° 13: Diseño de interfaz de mantenimiento de rutas	93
Tabla 25 Tarea N° 14: Gestión para el mantenimiento de rutas	94
Tabla 26 Historia N° 08: Mantenimiento de asignaciones	94
Tabla 27 Tarea N° 15: Diseño de interfaz de mantenimiento de asignaciones	95
Tabla 28 Tarea N° 16: Gestión para el mantenimiento de asignaciones	95
Tabla 29 Historia N° 09: Evaluar conformidad de unidades escolares	102
Tabla 30 Tarea N° 17: Diseño de interfaz de evaluación de conformidad de unidades escolares.....	103

Tabla 31 Tarea N° 18: Gestión de evaluación de conformidad de unidades escolares	103
Tabla 32 Historia N° 10: Evaluar conformidad de choferes.....	104
Tabla 33 Tarea N° 19: Diseño de interfaz de evaluación de conformidad de choferes	104
Tabla 34 Tarea N° 20: Gestión de evaluación de conformidad de unidades escolares	105
Tabla 35 Historia N° 11: Generación de contratos	105
Tabla 36 Tarea N° 21: Diseño de interfaz de generación de contratos.....	106
Tabla 37 Tarea N° 22: Gestión de generación de contratos	106
Tabla 38 Historia N° 12: Ordenar domicilios para el horario de ingreso escolar.....	107
Tabla 39 Tarea N° 23: Diseño de interfaz de ordenar domicilios para el horario de ingreso escolar	107
Tabla 40 Tarea N° 24: Gestión de ordenar domicilios para el horario de ingreso escolar	108
Tabla 41 Historia N° 13: Ordenar domicilios para el horario de salida escolar	108
Tabla 42 Tarea N° 25: Diseño de interfaz de ordenar domicilios para el horario de salida escolar	109
Tabla 43 Tarea N° 26: Gestión de ordenar domicilios para el horario de salida escolar	109
Tabla 44 Historia N° 14: Monitorear unidades escolares desde el sistema web	110
Tabla 45 Tarea N° 27: Diseño de interfaz de monitorear unidades escolares desde el sistema web.....	110
Tabla 46 Tarea N° 28: Visualización de monitoreo de unidades escolares desde el sistema web.....	111
Tabla 47 Historia N° 15: Ingresar al sistema web	111
Tabla 48 Tarea N° 29: Diseño de interfaz de ingresar al sistema web	112
Tabla 49 Tarea N° 30: Verificar autenticación de usuario para ingresar al sistema web	112
Tabla 50 Historia N° 16: Ingresar al aplicativo móvil.....	118
Tabla 51 Tarea N° 31: Diseño de interfaz de ingresar al aplicativo móvil.....	118
Tabla 52 Tarea N° 32: Verificar autenticación de usuario para ingresar al aplicativo móvil	119
Tabla 53 Historia N° 17: Configurar domicilio	119
Tabla 54 Tarea N° 33: Diseño de interfaz de configurar domicilio.....	120
Tabla 55 Tarea N° 34: Gestión de configurar domicilio	120
Tabla 56 Historia N° 18: Reportar inasistencia	121

Tabla 57 Tarea N° 35: Diseño de interfaz de registrar inasistencia.....	121
Tabla 58 Tarea N° 30: Gestión de registrar inasistencia	122
Tabla 59 Historia N° 19: Historial de recorridos	122
Tabla 60 Tarea N° 37: Diseño de interfaz de historial de recorridos	123
Tabla 61 Tarea N° 38: Gestión de historial de recorridos	123
Tabla 62 Historia N° 20: Reseteo de contraseñas desde el aplicativo móvil	124
Tabla 63 Tarea N° 39: Diseño de interfaz de resetear contraseñas desde el aplicativo móvil	124
Tabla 64 Tarea N° 40: Gestión de resetear contraseñas desde el aplicativo móvil	125
Tabla 65 Historia N° 21: Crear recorrido	125
Tabla 66 Tarea N° 41: Diseño de interfaz de crear recorrido	126
Tabla 67 Tarea N° 42: Gestión de crear recorrido	126
Tabla 68 Historia N° 22: Notificar incidencias	127
Tabla 69 Tarea N° 43: Diseño de interfaz de notificar incidencias	127
Tabla 70 Tarea N° 44: Gestión de notificar incidencias	128
Tabla 71 Historia N° 23: Marcar recogida.....	128
Tabla 72 Tarea N° 45: Diseño de interfaz de marcar recogida.....	129
Tabla 73 Tarea N° 46: Gestión de marcar recogida.....	129
Tabla 74 Historia N° 24: Reporte de rutas.....	135
Tabla 75 Tarea N° 47: Diseño de interfaz de reporte de rutas.....	135
Tabla 76 Tarea N° 48: Gestión de reportes de rutas	136
Tabla 77 Historia N° 25: Reporte de asignaciones	136
Tabla 78 Tarea N° 49: Diseño de interfaz de reporte de asignaciones	137
Tabla 79 Tarea N° 50: Gestión de reportes de asignaciones	137
Tabla 80 Historia N° 26: Reporte de contratos.....	138
Tabla 81 Tarea N° 51: Diseño de interfaz de reportes de contratos	138
Tabla 82 Tarea N° 52: Gestión de reportes de contratos	139
Tabla 83 Historia N° 27: Reporte de asistencia	139
Tabla 84 Tarea N° 53: Diseño de interfaz de reportes de asistencia	140
Tabla 85 Tarea N° 54: Gestión de reportes de asignaciones	140
Tabla 86 Historia N° 28: Reseteo de contraseñas desde el sistema web.....	141
Tabla 87 Tarea N° 55: Diseño de interfaz de resetear contraseñas desde el sistema web	141
Tabla 88 Tarea N° 56: Gestión de resetear contraseñas desde el sistema web.....	142

Tabla 89 Costo de software	208
Tabla 90 Costo de hardware	208
Tabla 91 Costo de implementación módulo de GPS con Arduino	209
Tabla 92 Costo de personal.....	209
Tabla 93 Costo de servicio.....	210
Tabla 94 Resumen de costos.....	210
Tabla 95 Beneficios tangibles.....	211

INDICE DE FIGURAS

Figura 1. Diseño de arquitectura.....	60
Figura 2. Instalación de ERwin Data Modeler versión 7.3.8.....	65
Figura 3. Instalación de DBeaver versión 6.3.1.....	66
Figura 4. Instalación Sublime Text versión 3.....	67
Figura 5. Estructura de proyecto web.....	67
Figura 6. Instalación de Android Studio versión 3.5.1.....	69
Figura 7. Plataforma Firebase de Google.....	70
Figura 8. Pantalla de bienvenida de la consola de Firebase.....	70
Figura 9. Creación de nombre del proyecto moviescool.....	71
Figura 10. Proyecto moviescool listo para configurar con Firebase Realtime Database.....	71
Figura 11. Crear y habilitar reglas de seguridad de Realtime Database.....	72
Figura 12. Creación de Realtime Database moviescool por defecto en null.....	72
Figura 13. Realtime Database en formato JSON configurado desde el módulo GPS con Arduino.....	73
Figura 14. Panel de administración desde el dominio starttecperu.com.....	74
Figura 15. Creación de base de datos starttec_moviescool desde el panel starttecperu.com.....	74
Figura 16. Creación de usuario starttec_moviescool desde el panel starttecperu.com.....	75
Figura 17. Usuario starttec_moviescool añadido a base de datos starttec_moviescool.....	75
Figura 18. Panel phpMyAdmin desde el dominio starttecperu.com.....	76
Figura 19. Diseño de circuitería del módulo de geolocalización con Arduino.....	76
Figura 20. Módulo administrador: Mantenimiento de colegios.....	96
Figura 21. Módulo colegio: Mantenimiento de apoderados.....	96
Figura 22. Módulo colegio: Mantenimiento de escolares.....	97
Figura 23. Módulo colegio: Mantenimiento de transportistas.....	97
Figura 24. Módulo colegio: Mantenimiento de unidades escolares.....	98
Figura 25. Módulo colegio: Mantenimiento de choferes.....	98
Figura 26. Módulo transportista: Mantenimiento de unidades escolares.....	99
Figura 27. Módulo transportista: Mantenimiento de choferes.....	99
Figura 28. Módulo transportista: Mantenimiento de rutas.....	100
Figura 29. Módulo transportista: Mantenimiento de asignaciones.....	100

Figura 30. Diagrama de base de datos: Iteración 1	101
Figura 31. Módulo colegio: Interfaz de evaluar conformidad de unidades escolares ..	113
Figura 32. Módulo colegio: Interfaz de evaluar conformidad de choferes.....	113
Figura 33. Módulo transportista: Interfaz de generación de contratos	114
Figura 34. Módulo transportista: Interfaz de ordenar domicilios para el horario de ingreso escolar	114
Figura 35. Módulo transportista: Interfaz de ordenar domicilios para el horario de salida escolar	115
Figura 36. Módulo administrador - Módulo colegio - Módulo transportista: Interfaz de monitoreo de unidades escolares desde el sistema web.....	115
Figura 37. Módulo inicio de sesión: Interfaz de ingresar al sistema web.....	116
Figura 38. Diagrama de base de datos: Iteración 2	116
Figura 39. Módulo inicio de sesión: Interfaz de ingresar al aplicativo móvil	130
Figura 40. Módulo apoderado: Interfaz de configurar domicilio	130
Figura 41. Módulo apoderado: Interfaz de registrar inasistencia	131
Figura 42. Módulo apoderado: Interfaz de historial de recorridos	131
Figura 43. Módulo apoderado y chofer: Interfaz de cambiar clave desde el aplicativo móvil	132
Figura 44. Módulo chofer: Interfaz de crear recorrido	132
Figura 45. Módulo chofer: Interfaz de notificar incidencias	133
Figura 46. Módulo chofer: Interfaz de marcar recogida	133
Figura 47. Diagrama de base de datos: Iteración 3.....	134
Figura 48. Módulo colegio: Interfaz de reporte de rutas	142
Figura 49. Módulo colegio: Interfaz de reporte de asignaciones.....	143
Figura 50. Módulo colegio: Interfaz de reporte de contratos	143
Figura 51. Diagrama de base de datos: Iteración 4.....	144
Figura 52. Diagrama de componentes	145
Figura 53. Diagrama de despliegue	146
Figura 54. Diseño de modelo relacional de base de datos	147
Figura 55. Tablas: ESCOLARES, APODERADOS Y USUARIOS.....	153
Figura 56. Módulo colegio: Interfaz de listado de apoderados.....	157
Figura 57. Módulo colegio: Interfaz de registrar apoderado	157
Figura 58. Módulo colegio: Interfaz de editar apoderado.....	158
Figura 59. Módulo colegio: Interfaz de ver información del apoderado	158

Figura 60. Módulo colegio: Interfaz de registrar escolar.....	159
Figura 61. Tablas: CONTRATOS, ESCOLARES, ASIGNACIONES	185
Figura 62. Módulo transportista: Interfaz de listado de contratos	186
Figura 63. Módulo transportista: Interfaz de registrar contrato	187
Figura 64. Módulo transportista: Interfaz de cargar listado de escolares	187
Figura 65. Módulo transportista: Interfaz de cargar listado de asignaciones	188
Figura 66. Tablas: ASIGNACIONES, UNIDADESESCOLARES, CHOFERES, RUTAS	201
Figura 67. Módulo colegio: Interfaz de listado de asignaciones	203
Figura 68. Módulo colegio: Interfaz de reporte de asignaciones PDF	203

RESUMEN

El presente proyecto de investigación, está enfocado en el desarrollo de un sistema informático de geolocalización para los procesos de control y monitoreo del servicio de transporte de estudiantes; lo cual permita mejorar la comunicación entre las instituciones educativas, padres de familia y choferes o transportistas en la ciudad de Chiclayo.

Para lograr tal fin, se utilizó la metodología XP (Xtreme Programming); el cual permite agilizar el desarrollo del proyecto en aspecto de planificación y diseño.

La plataforma Arduino, ha servido para armar un módulo de geolocalización, el cual permite obtener las coordenadas GPS y además de poder conectarnos a internet haciendo uso de una tarjeta SIM utilizando la tecnología GSM.

El desarrollo del sistema web, permite a las instituciones educativas y transportistas tener control de forma óptima de la información de los padres de familia, escolares, choferes, unidades escolares, contratos, monitoreo en tiempo real, entre otros, de la misma manera el desarrollo del aplicativo móvil; permite a los padres de familia y choferes, visualizar en tiempo real el recorrido de las unidades escolares, crear recorridos, ver y notificar incidencias.

Finalmente, a través del módulo GSM de la tarjeta SIM; arduino enviará las coordenadas GPS hacia la base de datos que está alojada en el servicio de Firebase Realtime Database. Este servicio será consumido tanto el sistema web como el aplicativo móvil para el monitoreo en tiempo real.

Palabras clave: Plataforma Arduino, Geolocalización, Tecnología GSM, Firebase Realtime Database.

ABSTRACT

The present research project is focused on the development of a geolocation computer system for the control and monitoring processes of the student transport service; which allows improving communication between educational institutions, parents and drivers or transporters in the city of Chiclayo.

To achieve this goal, the XP (Xtreme Programming) methodology was used; which allows to speed up the development of the project in planning and design aspect.

The Arduino platform has been used to build a geolocation module, which allows you to obtain GPS coordinates and also to connect to the Internet using a SIM card using GSM technology.

The development of the web system, allows educational institutions and carriers to have optimal control of the information of parents, schoolchildren, drivers, school units, contracts, real-time monitoring, among others, in the same way the development of the mobile application; It allows parents and drivers to visualize in real time the route of the school units, create routes, view and report incidents.

Finally, through the GSM module of the SIM card; arduino will send the GPS coordinates to the database that is hosted in the Firebase Realtime Database service. This service will be consumed both the web system and the mobile application for real-time monitoring.

Keywords: Arduino Platform, Geolocation, GSM Technology, Firebase Realtime Database.

INTRODUCCIÓN

El proyecto de investigación denominado “Desarrollo de un Sistema Informático en Tiempo Real de Geolocalización, Control y Monitoreo para el Servicio de Transporte de Estudiantes en la Ciudad de Chiclayo”, se ha realizado debido a la falta comunicación que existe entre las instituciones educativas, padres de familia y choferes o transportistas que brindan el servicio de transporte de estudiantes. Así mismo, el desarrollo del sistema informático permite mejorar el proceso de control y monitoreo de las unidades escolares.

La investigación se ha agrupado en los siguientes capítulos:

Capítulo I: Describe el planteamiento del problema que acontece el servicio de movilidad escolar, se especifica el objetivo general y los objetivos específico que se deben alcanzar en el proyecto, los antecedentes sobre sistemas de monitoreo en tiempo real, además el diseño de arquitectura a proponer para el desarrollo, definicion y operacionalizacion de variables.

Capítulo II: Describe los métodos y materiales a utilizar, la cual está compuesta por el procedimiento para el desarrollo de la investigación, población y muestra, técnicas e instrumentos de la recolección de datos.

Capítulo III: Identifica la problemática que presenta el servicio de transporte de estudiantes, se detallan las configuraciones iniciales que se han realizado para el desarrollo del sistema informático teniendo en cuenta la metodología XP y las implementaciones de las funcionalidades de las historias de usuario de mantenimiento de apoderados, generacion de contratos y reporte de asignaciones.

Capitulo IV y V: Describe las conclusiones y recomendaciones a las que se ha llegado con el proyecto desarrollado, cumpliendo con cada uno de los objetivos planteados.

CAPÍTULO I: DISEÑO TEÓRICO

1.1. Problemática de la investigación

1.1.1. Planteamiento del problema

El primer trimestre de cada año, con la aproximación de la temporada escolar, los padres de familia alistan el retorno de sus hijos al colegio. Para muchos, se han puesto en la obligación de analizar no sólo los grandes gastos que exige la temporada escolar como la matricula del colegio, la calidad de enseñanza del docente, la lista de útiles escolares, los uniformes; sino también la seguridad para trasladarlos a su institución educativa. El no disponer del tiempo suficiente para llevar a sus hijos al centro educativo, les genera gran preocupación porque tendrán que confiar en la recomendación de algún contacto para contratar el servicio de movilidad escolar, ante esto, los padres de familia están en la búsqueda de un servicio de movilidad escolar que les garantice seguridad e integridad en el traslado de sus hijos y les permita ahorrar tiempo para otras actividades, además les ayude a prevenir accidentes de tránsito.

Según Giordanelli (2016), especialista en seguridad vial y director de Tracklink Octo (sistema de rastreo y monitoreo vehicular), nos brinda 6 recomendaciones que se deben tener en cuenta previamente antes de solicitar una movilidad escolar:

- **Autorización:** Verifique que el conductor de la movilidad escolar cuente con la autorización expedida por la Gerencia de Transporte de la Municipalidad de Lima. Para ello puede visitar la página web de la Gerencia de Transporte Urbano (para averiguar si las unidades están debidamente registradas en la Municipalidad de Lima) y el portal del MTC (para conocer el récord de papeletas que tiene el conductor ingresando la placa del vehículo).

- **Seguro vehicular:** Sea el SOAT o AFOCAT, la movilidad escolar, por exigencia de la ley, debe contar con el seguro contra accidentes de tránsito. Recordemos que, en caso de sufrir algún inconveniente vehicular, el seguro debe activarse inmediatamente.
- **Verificar aspectos por el interior y exterior de la unidad:** Revisar que el piso interior de la movilidad escolar esté recubierto con material antideslizante, que tenga los asientos fijos y no plegables, frenos, cinturones de seguridad en los asientos del piloto, copiloto y posteriores, extintor, botiquín, asideros en los asientos, y en las puertas del vehículo.
- **Espacio destinado a equipaje:** Es primordial que el vehículo no permita que ninguna maleta pueda colocarse en los pasillos de la movilidad escolar. Para ello, debemos supervisar que la movilidad cuente con una maletera que permita guardar todo el equipaje que lleven los estudiantes. Ello con la finalidad de evitar posibles accidentes al bajar, subir del vehículo o cuando se tenga que frenar intempestivamente.
- **Cinturones de seguridad:** El objetivo es que todos los pasajeros puedan ingresar cómodamente y utilizar los cinturones de seguridad, con ello evitará exponerse al riesgo de muerte, dado que, al usarlo reducimos las víctimas mortales hasta en 75%.
- **Dispositivos tecnológicos:** En el país existen diferentes dispositivos tecnológicos para la seguridad de los pasajeros de la movilidad escolar, como por ejemplo alguno que pueda brindar ayuda especializada inmediata en caso de accidentes de tránsito o, inclusive, otros que puedan evaluar el estilo de manejo del conductor de la movilidad de tal manera que los padres puedan estar tranquilos que sus hijos van al colegio en buenas manos.

Según García (2013), Gerente de Desarrollo Vial y Transportes de la Municipalidad de Chiclayo, nos informa que un total de 12 vehículos fueron sancionados por no contar con autorización para prestar el servicio, carecer de SOAT, llevar exceso de pasajeros, entre otras faltas. Asimismo, detalló que a los conductores se les impuso una multa que oscila entre los S/. 280 y S/. 1400, dependiendo del tipo de infracción, y además se dispuso el traslado e internamiento en el depósito municipal de los vehículos.

Según Valencia (2014), Gerente de Transporte y Desarrollo Vial de la Municipalidad de Chiclayo; nos informa que en lo que va del año un total de 30 movilidades escolares han sido sancionadas por operar de manera informal y transportar a estudiantes sin las mínimas condiciones de seguridad. Asimismo, detalló que a todas estas unidades se le aplicó una multa del 40% de una Unidad Impositiva Tributaria (UIT) y además se internó las unidades en el depósito municipal. Otras unidades fueron sólo amonestadas por la policía de tránsito por no contar con SOAT y cinturones de seguridad en todos los asientos para los menores. Recalcó que la mayor parte de unidades que infringen la normativa son unidades tipo combis y miniván que son acondicionadas para brindar este servicio.

Según Landa (2017), Gerente de Tránsito de Huancayo, nos menciona que muchos vehículos se dedican al rubro de manera informal poniendo en riesgo la integridad de los alumnos, debido a la falta de medidas de seguridad de los conductores; como llevar más pasajeros que los asientos que figuran en la tarjeta de propiedad, incluso se han detectado que se transportan a los menores en las maleteras de autos y camionetas. Así mismo nos advirtió que las unidades (autos, minivanes y camionetas rurales), que presten este servicio deben contar con el distintivo con letras negras y fondo amarillo en el tapasol del (parte frontal) y en las partes laterales. En cuanto a las medidas de seguridad, todos los asientos de la unidad deben contar con cinturón de seguridad, botiquín para primeros auxilios, extintor, herramientas mecánicas y rueda de repuesto. Además, algunos han decidido instalar cámaras

en el interior del vehículo para certificar un servicio de calidad a los alumnos que movilizan a diario.

Según el Organismo Peruano de Consumidores y Usuarios (2019), nos informa que pidió a la Municipalidad Metropolitana de Lima publicar el padrón de transporte escolar autorizado que presta servicio de movilidad a los estudiantes escolares, así los padres de familia accederán a información veraz y oportuna que les permitirá contratar unidades vehiculares seguras para sus menores hijos. Asimismo, el Organismo Peruano de Consumidores y Usuario (OPECU) presentó un cuadro estadístico del transporte escolar autorizado para prestar el servicio de movilidad desde el año 2011 al 2018, siendo de 2742 vehículos formales en el 2011, de 3365 en el 2012, de 2985 en el 2013, de 3165 en el 2014, de 1400 en el 2015, de 3372 en el 2016, de 3625 en el 2017, de 3749 en el 2018 y de 4346 en el año 2019.

Tabla 1

Vehículos autorizados para servicio de movilidad escolar Lima Metropolitana

AÑO	VEHÍCULOS AUTORIZADOS	FECHA FUENTE
2011	2742	21ENE2011 (*)
2012	3365	24ENE2012 (*)
2013	2985	15ENE2013 (*)
2014	3165	22FEB2014
2015	1400	19FEB2015
2016	3372	25FEB2016
2017	3625	11FEB2017
2018	3749	11FEB2018
2019	4346	10FEB2019

Fuente: Elaborado por OPECU 2019

1.2. Formulación del problema de investigación

¿De qué manera influirá el desarrollo de un sistema informático de geolocalización para los procesos de control y monitoreo de las unidades escolares en la mejora de comunicación entre los centros educativos, padres de familia y transportistas que brindan el servicio de transporte de estudiantes en la ciudad de Chiclayo?

1.3. Objetivos

1.3.1. Objetivo general

- Desarrollar un sistema informático en tiempo real de geolocalización, control y monitoreo para el servicio de transporte de estudiantes en la ciudad de Chiclayo.

1.3.2. Objetivos específicos

- Identificar y evaluar las condiciones actuales de cómo se lleva a cabo el proceso de servicio de transporte de estudiantes.
- Definir una arquitectura de la aplicación enfocada a una solución web y móvil que permita alojar la información de manera centralizada y que pueda ser consultada en tiempo real desde cualquier lugar.
- Diseñar una base de datos para llevar el control de incidencias.
- Implementar el módulo de control de rutas del servicio escolar.
- Implementar el módulo de notificaciones.
- Implementar el aplicativo de control y monitoreo para el servicio de transporte escolar.

1.4. Marco metodológico

1.4.1. Tipo de investigación

Investigación tecnológica formal.

1.4.2. Hipótesis

El desarrollo de un sistema informático permitirá mejorar la comunicación entre centros educativos, padres de familia y empresas de transportes, además permitirá agilizar los procesos de geolocalización, control y monitoreo en tiempo real a los vehículos que brindan el servicio de transporte de estudiantes en la ciudad de Chiclayo.

1.4.3. Definición y operacionalización de variables

Tabla 2

Operacionalización de variable dependiente

Variable Dependiente	Descripción	Dimensión	Indicador	Instrumentos
Comunicación entre los centros educativos, padres de familia y empresas de transportes del servicio público escolar	Esta comunicación se dará mediante el aplicativo que informará a estos actores del sistema sobre todo lo que sucede con los estudiantes y ver si hay algún imprevisto y demás acciones hacia ellos	Lugar de partida	Posición exacta en la cual parte el estudiante al colegio	Encuesta
		Localización de unidad móvil	Posición exacta de donde se encuentra la unidad vehicular	
		Alertas de incidencias que se muestran en el camino	Mediante algún tipo de mensaje enviado hacia los interesados	
		Tiempo estimado de llegada a destino	Minutos calculados entre el punto de partida hacia el de llegada	
		Lugar de llegada	Posición exacta en la cual fue dejado el estudiante	

Fuente: Elaboración propia

Tabla 3

Operacionalización de variable independiente

Variable Independiente	Descripción	Dimensión	Indicador	Instrumentos
Sistema informático para localizar, controlar y monitorear la unidad escolar en la ciudad de Chiclayo	Sistema elaborado con el fin de llevar un control en el transporte escolar que ayudará tanto a transportistas, colegios y padres de familia	Elaborar el diseño del sistema informático para localizar, controlar y monitorear la unidad escolar en la ciudad de Chiclayo	Elección del sistema para la obtención del conjunto de variables significativas para la resolución del problema Desarrollar la arquitectura adecuada del sistema a desarrollar	Encuesta, Guías de Observación

Fuente: Elaboración propia

1.5. Marco teórico

1.5.1. Antecedentes

Para respaldar el siguiente estudio se consultaron varias investigaciones sobre el Desarrollo de un Sistema Informático en Tiempo Real de Geolocalización, Control y Monitoreo para el Servicio de Transporte de Estudiantes en diferentes ciudades.

1.5.1.1. Internacionales

Yago, (2015) en su trabajo de fin de grado *Aplicación Web y Móvil para el Seguimiento de Autobuses*, tuvo como objetivo de su proyecto la creación y despliegue de una plataforma web y móvil para el seguimiento de autobuses, mediante el uso del framework Cakephp, la cual permite gestionar un servicio de geolocalización de flotas por GPS. Esta plataforma se centra en ofrecer, mediante Push Notifications o email, un servicio de alertas sobre la proximidad de buses escolares a una zona definida, es decir, a las paradas dentro de una ruta. Este servicio está pensado para informar a los padres de la llegada de sus hijos a su parada habitual antes de que ésta tenga lugar. La plataforma se compone de varios módulos PHP ya que por una parte ofrecemos la posibilidad de realizar labores de administración; p.ej., creación de usuarios, asignación de dispositivos y por otra parte un frontend a los usuarios registrados en nuestro sistema. También se ha desarrollado una aplicación móvil, utilizando para ello el framework Phonegap, tanto para recibir las notificaciones mencionadas anteriormente como para ofrecer de una forma rápida información en tiempo real sobre la localización del dispositivo GPS. Cabe destacar que se ha utilizado un template, basado en Bootstrap 3 y que incluye directrices en AngularJS, para dotar de diseño a la interfaz web.

Américo (2011), en su trabajo de licenciamiento en sistemas “Monitoreo satelital de vehículos mediante una aplicación web”, tuvo como objetivo de su trabajo propone la creación de un prototipo de sistema web de monitoreo satelital de vehículos para analizar datos provistos por dispositivos especiales de posicionamiento que se instalan en ellos. Se

hace una introducción básica a GPS y Sistemas de Información Geográfica (SIG), explicando su funcionamiento y áreas en las que se utilizan. Se describen algunas de las diferentes aplicaciones posibles para SIG y la importancia de la información geográfica, su correcta administración y las herramientas disponibles para su análisis. Por último, se describe el prototipo de sistema web, su funcionamiento y su modo de uso.

1.5.1.2. Nacionales

Bocanegra (2012), en su trabajo de tesis “Desarrollo de una aplicación web para el monitoreo de vehículos con dispositivos GPS que comercializa una empresa de Telecomunicaciones”, tuvo como objetivo de su proyecto de investigación aplicada presentar la metodología de desarrollo comprendida en la realización de una solución de monitoreo para vehículos con dispositivos GPS, el software propuesto forma parte del servicio de monitoreo vehicular que ofrece una empresa de telecomunicaciones. Se describe y especifica la arquitectura del sistema a través de casos de uso en las diversas disciplinas del proceso de desarrollo de software. Si bien existen diversas soluciones de monitoreo vehicular que son usadas en distintos centros de control, la solución propuesta agrupa las funcionalidades representativas de las mejores soluciones de monitoreo del mercado y las presenta de manera objetiva a través de una propuesta completa y puntual donde el usuario final pueda dar uso de la solución a través de un navegador web, así mismo. se muestra el diseño de la solución utilizando las disciplinas y patrones que exige la ingeniería de software.

Meza & Leño (2017), en su trabajo de tesis “Sistema de monitoreo de una red de buses de Transporte público e información para usuarios Empleando transceptores GPS/GSM”, tuvo como objetivo desarrollar de un sistema de monitoreo de una red de buses de transporte público integral, mediante el empleo de tecnologías modernas de comunicación y posicionamiento, con un sistema de procesamiento de información que permita al

administrador del sistema la regulación de los flujos de tránsito, controlar las unidades de transporte, disminuir el riesgo de accidentes y ofrecer una mejor calidad de vida a los ciudadanos. El sistema de monitoreo se desarrolla principalmente para ser implementado en la flota de las empresas de transporte público, teniendo como finalidad apoyar a la creación de un sistema inteligente de transporte para nuestra capital. El empleo de transceptores GPS/GSM nos permitirá obtener datos de posición (latitud, longitud), fecha, hora y velocidad de desplazamiento del móvil monitoreado. Esta información será enviada a una central en la que será procesada junto a la data recopilada de otras fuentes.

Una vez procesada la información, el administrador podrá tomar acciones correctivas necesarias, como aumentar o disminuir la frecuencia de los buses, controlar el exceso de velocidad, informar a los choferes de potenciales peligros y rutas alternativas, proveer la información de tiempo y distancia de llegada a los usuarios a través de mensajes de texto o aplicación para Smartphone. Se muestra las potencialidades del sistema mediante el desarrollo de un sistema de visualización que provea información de horas de arribo a los usuarios. Este sistema de monitoreo puede ser integrado a los sistemas actuales como el Metropolitano y el Metro de Lima, unificando así los mecanismos de transporte y lograr el tan ansiado Sistema Integrado de Transporte.

Bermúdez & Chávez (2015), en su proyecto de tesis “Diseño de un sistema para la mejora en el control de las unidades de las Empresas de transporte urbano en la ciudad de Trujillo”, tuvo como objetivo el diseño de un sistema de control de llegada a cada punto de marcaje, por medio del uso de dispositivos GPS logrando el marcaje de manera automática al llegar a cada punto de marcaje para empresas de transporte urbano de la ciudad de Trujillo, este sistema será capaz de verificar la llegada a cada punto de marcaje y enviar un mensaje con los datos de la correcta llegada desde el vehículo hacia la central de la empresa para el ser procesada, y verificar la correcta llegada a los puntos de marcaje de cada

uno de los vehículos de dicha empresa. Si bien existen diversas soluciones de control de llegada para vehículos que son usadas en algunas empresas, la solución propuesta agrupa las funcionalidades representativas de las mejores soluciones de control, y las presenta de manera completa y puntual en el diseño de un sistema donde el usuario final podrá dar uso de la solución de manera accesible.

1.6. Bases teóricas

1.6.1. Conceptos generales

1.6.1.1. Geolocalización

La geolocalización es una tecnología que utiliza datos obtenidos de la computadora o dispositivo móvil de un individuo para identificar o describir su ubicación física real. Un sistema de geolocalización es una solución de la tecnología de la información que determina la ubicación de un objeto en un entorno físico (geoespacial) o virtual (Internet). A menudo, el objeto es una persona que quiere utilizar un servicio basado en la ubicación, mientras mantiene su privacidad.

Utilizando un dispositivo conectado a Internet, con la geolocalización es posible obtener distintos tipos de información en tiempo real y localizarlos en el mapa con una gran precisión en un momento determinado en el tiempo. Los datos de geolocalización se pueden recopilar de muchas formas:

- Mediante la navegación web a través de direcciones IP.
- Teléfonos móviles.
- Dispositivos GPS.
- Identificación de radiofrecuencia (RFID).
- Transacciones con tarjeta de crédito o débito.

- Etiquetas en fotografías y mensajes (tales como las geoetiquetas o verificaciones mediante aplicaciones como Foursquare) en sitios de redes sociales como Facebook y Twitter.

La tecnología de geolocalización se ha convertido en una base para los servicios de posicionamiento por ubicación y las aplicaciones sensibles a la ubicación que se ejecutan en los teléfonos inteligentes.

Los datos de geolocalización tienen una variedad de usos, cada uno de los cuales se puede personalizar para determinadas aplicaciones, entornos o empresas. Actualmente estos usos incluyen la ubicación y/o personalización de los contenidos distribuidos, la aplicación de restricciones de acceso y distribución de acuerdo a la ubicación geográfica, prevención de fraudes y análisis de tráfico de redes. (ISACA, 2017).

1.6.1.2. Sistemas de localización en tiempo real

Los sistemas de localización en tiempo real (RTLS) e Indoor Positioning Solutions (IPS) ofrecen a los administradores de red la capacidad de identificar y rastrear la ubicación física de los dispositivos en tiempo real para la gestión de inventarios, el seguimiento de la ubicación de una persona y la optimización de flujos de operativos. RTLS utiliza la intensidad de la señal Wi-Fi de un dispositivo para obtener su ubicación aproximada, mediante la triangulación a través de múltiples puntos de acceso (APs). RTLS puede leer la señal Wi-Fi de una función de adaptadores Wi-Fi (por ejemplo, en iPads, iPhones, ordenadores portátiles, etc) o de variables externas conectadas a un dispositivo (por ejemplo, un proyector).

Con RTLS de Xirrus, se puede personalizar la información para ser entregados en base a la ubicación específica del usuario. El sistema de administración está compuesto por varios componentes en función de lo que requiere el proyecto. Por ejemplo, un sistema puede agregar los datos de localización para proporcionar la ubicación de un usuario en particular.

Otro sistema podría ser utilizado para proporcionar ofertas específicas para ese usuario en función de dónde se encuentren. Para soluciones RTLS, diferentes piezas se pueden mezclar y combinar para obtener la mejor solución para cada necesidad particular.

El componente principal del sistema RTLS es la infraestructura inalámbrica. La infraestructura inalámbrica recoge la señal de los dispositivos de cliente, que transmite la información al software de sistema de localización.

Los RTLS pueden proporcionar muchos beneficios a las organizaciones, entre ellas:

- Ahorrar dinero
- Aumento de la productividad
- Mejora de la seguridad
- Protección de los activos
- Ofrecer mejores servicios

1.6.2. Metodología de desarrollo

1.6.2.1. Extreme programming (XP)

Es una metodología ágil y flexible utilizada para la gestión de proyectos. Extreme Programming se centra en potenciar las relaciones interpersonales del equipo de desarrollo como clave del éxito mediante el trabajo en equipo, el aprendizaje continuo y el buen clima de trabajo. Esta metodología pone el énfasis en la retroalimentación continua entre cliente y el equipo de desarrollo y es idónea para proyectos con requisitos imprecisos y muy cambiantes.

Principales características:

- Se considera al equipo de proyecto como el principal factor de éxito del proyecto
- Software que funciona por encima de una buena documentación.
- Interacción constante entre el cliente y el equipo de desarrollo.
- Planificación flexible y abierta.

- Rápida respuesta a cambios.

Roles

- Cliente: responsable de definir y conducir el proyecto, así como sus objetivos.
- Programadores: estiman tiempos de desarrollo de cada actividad y programan el proyecto.
- Tester: Encargado de Pruebas.
- Tracker: Encargado de Seguimiento.
- Coach: Entrenador. Su papel es guiar y orientar al equipo.
- Big Boss: Gestor del proyecto, gerente del proyecto, debe tener una idea general del proyecto y estar familiarizado con su estado.

1.6.3. Sistema de gestión de base de datos

1.6.3.1. MySQL

Es un motor de base de datos relacional (RDBMS) de código abierto, originalmente creado por la empresa MySQL AB y actualmente de propiedad de Oracle Corporation.

Como administrador de bases de datos relacional, MySQL almacena los datos en forma de tablas estructuradas, con campos, índices, llaves foráneas e integridad referencial en la actualización o borrado en cascada. Usa el lenguaje SQL para las diferentes transacciones de datos.

Se puede instalar prácticamente en cualquier sistema operativo: Unix, Linux, Windows, MacOS; solo o acompañado de otros programas del entorno de programación como el servidor web Apache, PHP y PhpMyAdmin; en las distribuciones WAMP, LAMP o XAMPP.

Algunas características:

- Autenticación de usuarios con permisos específicos para ciertas bases de datos, atadas a las direcciones IP de origen.

- Gestión de memoria y cache para una cantidad determinada de consultas o transacciones simultáneas.
- Conectores para integración en ambientes PHP, Perl, Python, ODBC.
- Replicación transaccional en línea de la base de datos, a otra base paralela.
- Monitoreo de usuarios, transacciones, uso de memoria y de procesos.
- Uso de triggers o disparadores para actuar sobre creación, edición o borrado de registros.
- Uso automático de valores autonuméricos en las tablas.
- Se pueden crear vistas, procedimientos almacenados y funciones.
- Posee funciones de chequeo del desempeño de consultas y sus índices, cómo también de los parámetros de memoria, caché, transacciones, etc. Podemos encontrar un consejero de desempeño en Phpmyadmin.
- Se puede administrar desde consola, phpmyadmin, o desde programas externos como MySQLfront o Sequel (para MacOS) mediante conexión por el puerto TCP 3306.
- Puede manejar millones de registros en una sola tabla.
- Gestiona el bloqueo de tablas para evitar conflictos de transacciones simultáneas.
- Se pueden importar o exportar datos o la estructura misma de la base, sus tablas, índices, etc. (Ramírez Navia, 2018).

1.6.3.2. BDeaver

Es un cliente SQL y una herramienta de administración de bases de datos. Se trata de una aplicación de escritorio escrita en Java y basada en la plataforma Eclipse. Si lo que queremos es trabajar con bases de datos relacionales, va a utilizar la API JDBC para interactuar con bases de datos a través de un controlador JDBC. Para otras bases de datos no SQL utiliza controladores de bases de datos patentados. También nos va a proporcionar un editor que

admite la finalización del código y el resaltado de sintaxis. También dispondremos de una arquitectura de complementos basada en la arquitectura de complementos de Eclipse, que nos va a permitir a los usuarios modificar gran parte del comportamiento de la aplicación para conseguir funciones o características específicas de la base de datos.

Este software está destinado a desarrolladores y soporta no solo Microsoft SQL. También nos va a permitir trabajar con muchas otras bases de datos relacionales populares como MySQL, PostgreSQL, SQLite, Oracle, DB2, MariaDB, Sybase, Teradata, Netezza, etc. Al mismo tiempo admite algunas bases de datos NoSQL como MongoDB, Cassandra, Redis, Apache Hive, etc. (Amoedo, 2019).

1.6.3.3. *Firebase*

Se trata de una plataforma móvil creada por Google, cuya principal función es desarrollar y facilitar la creación de apps de elevada calidad de una forma rápida, con el fin de que se pueda aumentar la base de usuarios y ganar más dinero. La plataforma está subida en la nube y está disponible para diferentes plataformas como iOS, Android y web. Contiene diversas funciones para que cualquier desarrollador pueda combinar y adaptar la plataforma a medida de sus necesidades.

Principales características de Firebase:

- **Desarrollo:** Firebase permite la creación de mejores apps, minimizando el tiempo de optimización y desarrollo, mediante diferentes funciones, entre las que destacan la detección de errores y de testeo, que supone poder dar un salto de calidad a la app. Poder almacenar todo en la nube, testear la app o poder configurarla de manera remota, son características destacables de la plataforma.
- **Analítica:** Tener un control máximo del rendimiento de la app mediante métricas analíticas, todo desde un único panel y de forma gratuita, es una de las ventajas

que ofrece Firebase respecto a la analítica web. Los datos analíticos que facilita Firebase, facilita la toma de decisiones basadas y fundamentadas en datos reales.

- Poder de crecimiento: Permite gestionar de manera fácil todos los usuarios de las aplicaciones, con el añadido de que se pueden captar nuevos usuarios, mediante invitaciones o notificaciones.
- Monetización: Mediante AdMob, Firebase permite que puedas ganar dinero.
- Rapidez: Implementar Firebase puede ser fácil y rápido, gracias a su API que es muy intuitiva, sostenida en un solo SDK. Con Firebase puedes centrar tus esfuerzos en resolver los problemas de tus clientes y así poder evitar la pérdida de tiempo en la creación de una infraestructura compleja.
- Agilidad: Firebase ofrece apps multiplataforma con una APIs integradas a SDK individuales para iOS, Android y JavaScript, de tal forma que se puede gestionar diferentes apps sin necesidad de salir de la propia plataforma. (Pérez Cardona, 2016).

Los servicios que nos ofrece Firebase son las siguientes:

- Base de datos Realtime: La base de datos de firebase almacena y sincroniza los datos con nuestra base de datos, todo esto se aloja en la nube (internet) estos datos que están en la nube son almacenadas en JSON y se pueden agregar reglas para permitir requests con token o solo desde una URL y los datos de la base se sincronizan con todos los clientes en tiempo real esto ayuda mucho cuando la app no tiene conexión a Internet. La aplicación de firebase responde, aunque no tenga internet esto es gracias al SDK (kit de desarrollo de software) de firebase base de datos Realtime hace que nuestros datos persistan en el disco, cuando la conexión regresa el dispositivo lo reconoce y lo guarda en el servido actual.

- **Autenticación:** Es un servicio que nos simplifica el inicio de sesión y la gestión de la misma en nuestra aplicación. Si la usamos en aplicaciones web o Android Studio es extremadamente fácil de configurar, con un simple método ya podremos guardar un usuario con su contraseña y correo. Firebase guarda datos de conexión en la nube de forma segura así para que las aplicaciones inicie sesión con la cuenta guardada
- **Almacenamiento:** El servicio de almacenamiento de firebase se llama Cloud Storage es un servicio que almacena objetos potentes los SDK (kit de desarrollo de software) de Firebase para Cloud Storage te dan seguridad de Google a la hora de descargar o subir un archivo esto sin importar la calidad de red que tengas, es muy bueno para aplicaciones que requieran guardar archivos del usuario. También nos sirve si queremos subir estáticos ya que existe un botón desde la interfaz o podemos programar algo. En la aplicación que estoy desarrollando actualmente me sirve para subir fotos de restaurantes ya acceder a ellas en cualquier momento.
- **Hosting:** Con una colección de estáticos podemos subir una aplicación y esta automáticamente contará con SSL y HTTP2. Sin necesidad de que configures nada para que el contenido sea de forma segura. Si tenemos una app con Angular o Firebase podemos hacer un build desde nuestros ordenadores y subir estos archivos generados y nos funcionarán sin problemas. Si necesitamos un Backend tendríamos que subirlo a otro lugar o hacer uso de las Cloud Functions. Cuando subes un archivo se almacena en la cache del SSD esto es muy bueno si el contenido se tiene que entregar el contenido se entregara rápidamente. (Peña, 2018).

1.6.4. Lenguaje de programación

1.6.4.1. PHP

Es un lenguaje de scripting de código abierto, del lado del servidor, con programación HTML integrada que se utiliza para crear páginas web dinámicas. Las ventajas de PHP son su flexibilidad y su alta compatibilidad con otras bases de datos. Además, PHP es considerado como un lenguaje fácil de aprender.

Las funciones de PHP se relacionan con los scripts del lado del servidor. PHP puede realizar cualquier tarea que cualquier programa CGI (Common Gateway Interface) puede hacer y maneja el intercambio de datos entre el servidor y el software. Por lo tanto, PHP puede recopilar datos o crear webs de contenido dinámico.

Tres funciones principales distinguen en particular el PHP:

- Programación del lado del servidor: esto requiere tres componentes, que son un analizador PHP, un navegador web y un servidor web. Este último está conectado con una instalación PHP. El navegador web se utiliza para acceder al programa PHP. El navegador muestra páginas web que se almacenan en el servidor y se generan con PHP. La programación del lado del servidor es ideal para webs de prueba antes de su publicación en la web.
- Programación a través de la línea de comandos: Los scripts PHP pueden ser creados sin un navegador o servidor. En este caso, sólo necesitarás un analizador PHP. Estos scripts son adecuados para las tareas regulares que se llevan a cabo en una web.
- Escribir aplicaciones de escritorio: esta función no se utiliza con tanta frecuencia como la programación de páginas web. Sin embargo, el PHP es adecuado para el desarrollo de aplicaciones de escritorio debido a su complejidad, que también puede ser utilizado a través de múltiples plataformas.

El lenguaje de scripting PHP puede ser usado con todos los sistemas operativos populares. Además, PHP es compatible con la mayoría de los servidores web como Apache. Las ventajas para el usuario son:

- PHP puede ser usado con cualquier sistema operativo virtual y cualquier tipo de servidor.
- Las aplicaciones se pueden programar orientadas al procedimiento o al proyecto.
- HTML, imágenes, PDF, o archivos flash pueden ser generados dinámicamente con PHP.
- PHP soporta una gran cantidad de bases de datos diferentes, por lo que este lenguaje de scripting es ideal para la creación de webs basados en bases de datos.
- PHP ofrece soporte completo para la comunicación del servidor con otros protocolos.

1.6.4.2. Java

Es un lenguaje de programación de propósito general, concurrente, orientado a objetos que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Su intención es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo (conocido en inglés como WORA, o "write once, run anywhere"), lo que quiere decir que el código que es ejecutado en una plataforma no tiene que ser recompilado para correr en otra. Su sintaxis deriva en gran medida de C y C++, pero tiene menos utilidades de bajo nivel que cualquiera de ellos. Las aplicaciones de Java son generalmente compiladas a bytecode (clase Java) que puede ejecutarse en cualquier máquina virtual Java (JVM) sin importar la arquitectura de la computadora subyacente. (ICTEA, 2019).

Existen múltiples razones por las que es bueno aprender a programar en Java:

- Está dentro de los lenguajes más usados en la actualidad y corre en casi todas las plataformas que hay en el mercado.
- Si quieres comenzar a desarrollar en Android, Java es una base importante que necesitas y debes de aprender.
- Existe gran soporte, documentación y comunidades de Java a las cuales podrás acudir si necesitas ayuda para entender mejor el lenguaje.
- Java también cuenta con una serie de librerías (nativas y de terceros) que amplían sus funcionalidades, desde manipular archivos de Office hasta reconocer huellas digitales y mucho más.
- Java no es un lenguaje complicado como se podría pensar, ya que es un tipo de programación orientada a objetos, comprendiendo aspectos básicos de este tipo de programación el aprendizaje de Java será de manera intuitiva.
- Java está diseñado para crear software altamente robusto y fiable, para ello proporciona numerosas comprobaciones durante la compilación y en tiempo de ejecución, cuenta con un recolector de basura que elimina la necesidad de liberación de memoria explícita. (Guevara Benites, s.f.).

1.6.5. Arquitectura de software

1.6.5.1. Patrón MVC (*Modelo-Vista-Controlador*)

Es un patrón de diseño arquitectónico de software, que sirve para clasificar la información, la lógica del sistema y la interfaz que se le presenta al usuario. En este tipo de arquitectura existe un sistema central o controlador que gestiona las entradas y la salida del sistema, uno o varios modelos que se encargan de buscar los datos e información necesaria y una interfaz que muestra los resultados al usuario final. Es muy usado en el desarrollo web porque al tener que interactuar varios lenguajes para crear un sitio es muy fácil generar

confusión entre cada componente si estos no son separados de la forma adecuada. Este patrón permite modificar cada uno de sus componentes sin necesidad de afectar a los demás.

- **Modelo:** Este componente se encarga de manipular, gestionar y actualizar los datos. Si se utiliza una base de datos aquí es donde se realizan las consultas, búsquedas, filtros y actualizaciones.
- **Vista:** Este componente se encarga de mostrarle al usuario final las pantallas, ventanas, páginas y formularios; el resultado de una solicitud. Desde la perspectiva del programador este componente es el que se encarga del frontend; la programación de la interfaz de usuario si se trata de una aplicación de escritorio, o bien, la visualización de las páginas web (CSS, HTML, HTML5 y JavaScript).
- **Controlador.** Este componente se encarga de gestionar las instrucciones que se reciben, atenderlas y procesarlas. Por medio de él se comunican el modelo y la vista: solicitando los datos necesarios; manipulándolos para obtener los resultados; y entregándolos a la vista para que pueda mostrarlos. (García, 2017).

1.6.6. Entorno de desarrollo integrado

1.6.6.1. Sublime Text

Es actualmente el editor de código más omnipresente que existe. No solo por su utilidad indiscutida en una operativa de desarrollo concreta (siempre es posible encontrar editores o entornos con características especializadas para una tecnología en concreto), sino por su presencia generalizada en el desarrollo web; con independencia del lenguaje de programación, sistema operativo o perfil frontend/backend del usuario. Es un entorno ágil de utilizar, agradable, y con muchas buenas características que otros entornos intentan copiar continuamente. Todo ello respaldado por un sistema de plugins que garantiza la continua adición de funciones, sólo limitada por la imaginación de los que lo desarrollan. (Herrera, 2016).

1.6.6.2. Android Studio

Es el entorno de desarrollo integrado (IDE) oficial para el desarrollo de apps para Android, basado en IntelliJ IDEA. Además del potente editor de códigos y las herramientas para desarrolladores de IntelliJ, Android Studio ofrece incluso más funciones que aumentan tu productividad cuando desarrollas apps para Android, como las siguientes:

- Un sistema de compilación flexible basado en Gradle.
- Un emulador rápido y cargado de funciones.
- Un entorno unificado donde puedes desarrollar para todos los dispositivos Android.
- Aplicación de cambios para insertar cambios de códigos y recursos a la aplicación en ejecución sin reiniciar la aplicación.
- Integración con GitHub y plantillas de código para ayudarte a compilar funciones de apps comunes y también importar código de ejemplo.
- Variedad de marcos de trabajo y herramientas de prueba.
- Herramientas de Lint para identificar problemas de rendimiento, usabilidad y compatibilidad de la versión, entre otros.
- Compatibilidad con C++ y NDK.
- Compatibilidad integrada para Google Cloud Platform, que facilita la integración con Google Cloud Messaging y App Engine. (Developers Android, s.f.).

1.6.7. Servidor web

1.6.7.1. WAMP Server

Una de las mejores opciones es trabajar con servidores WAMP, que es el acrónimo que se emplea para describir cualquier sistema que emplee Windows (como sistema operativo), Apache (como servidor web), MySQL (como base de datos) y PHP o Perl o Python (como

lenguajes de programación). Si se emplea la misma infraestructura en el sistema operativo Linux la solución se denominará LAMP, y si es en Macintosh estaremos ante un MAMP.

En esencia se trata de programas que contienen el software necesario para poder montar un servidor web en cualquier ordenador. La mayoría de aplicaciones web que existen en la actualidad trabajan con Apache, MySQL y PHP, por lo que este tipo de soluciones facilitan mucho el proceso de instalación.

La utilidad que tienen las aplicaciones WAMP se centra en:

- **Facilidad de uso.** La mayoría de soluciones vienen configuradas de tal manera que es muy sencillo arrancarlas y empezar a trabajar. Además, la mayoría de ellas se gestionan por medio de una interfaz mucho más intuitiva que un fichero editable .ini.
- **Configuración.** El proceso de configurar, tanto cada una de las aplicaciones como el paquete WAMP es automático. De hecho, en la mayoría de los casos se configura automáticamente teniendo en cuenta la unidad donde se ha almacenado la carpeta que contiene el software. Y, como es evidente, también es posible editar dicha configuración de forma manual.
- **Portabilidad.** El hecho de que todas las aplicaciones que se empleen en un sitio web se almacenen en una misma carpeta facilita mucho el trabajo. Especialmente si debemos llevar los ficheros de un lado a otro, o si tenemos que cambiar varias veces de ordenador para poder trabajar. De hecho, es posible, incluso, realizar la instalación sobre un pendrive y trabajar de manera más autónoma.
- **Velocidad.** El trabajo con el sitio web en modo local es mucho más rápido que cuando se realiza en modo online: la base de datos está en local, aumentando la velocidad de acceso a los datos; los ficheros están en la misma máquina que hace la petición, por lo que la demora es mucho menor... Además, no es necesaria una

conexión a Internet para poder trabajar, ya que siempre estarás conectado a tu localhost.

- No es necesario contar con una IP estática para poder trabajar, ya que se puede cargar el sitio web desde `http://localhost`. (Senso, 2019).

1.6.7.2. *Hosting web*

Es un servicio en línea que te permite publicar un sitio o aplicación web en Internet. Cuando te registras en un servicio de hosting, básicamente alquilas un espacio en un servidor donde puedes almacenar todos los archivos y datos necesarios para que tu sitio web funcione correctamente.

Cuando decides crear un nuevo sitio web, tienes que encontrar una empresa de hosting que te proporcione espacio en un servidor. Tu proveedor de hosting almacena todos tus archivos, medios y bases de datos en el servidor. Cada vez que alguien escribe tu nombre de dominio en la barra de direcciones de su navegador, tu servidor transfiere todos los archivos necesarios para atender la solicitud.

Debes elegir el plan de hosting que mejor se adapte a tus necesidades y comprarlo. De hecho, el hosting web funciona de manera similar al alquiler de viviendas, tienes que pagar el alquiler regularmente para poder mantener el servidor funcionando continuamente.

Tu proveedor de hosting es el responsable de mantener el servidor en funcionamiento, protegerlo de ataques maliciosos y transferir tu contenido (texto, imágenes, archivos) desde el servidor a los navegadores de tus visitantes.

Además de proporcionar un espacio de servidor para tu sitio web, los proveedores de hosting también pueden ofrecer otros servicios relacionados con la administración del sitio web, tales como:

- Certificados SSL (para sitios seguros utilizando el protocolo `https://`)
- Hosting de correo electrónico.

- Creadores de páginas web.
- Herramientas para desarrolladores.
- Servicio de atención al cliente (generalmente con chat en vivo)
- Copias de seguridad de sitios web automáticas.
- Instalaciones de software con un solo clic (por ejemplo, para WordPress o Drupal). (Gustavo B., ¿Qué es un hosting?, 2019)

1.6.7.3. Dominio web

Es esencialmente el equivalente a una dirección física. De la misma manera que un satélite necesita una dirección o un código postal para dar direcciones, un navegador web necesita un nombre de dominio para dirigirte a un sitio web.

Un dominio de internet se forma a partir de dos elementos principales. Por ejemplo, el nombre de dominio Facebook.com consiste del nombre del sitio web (Facebook) y la extensión del dominio (.com). Cuando una empresa (o una persona) compra un dominio web, puede especificar a qué servidor apunta el nombre de dominio.

Los registros de nombres de dominios son supervisados por una organización llamada ICANN (Corporación de Internet para Nombres y Números Asignados, por sus siglas en inglés). ICANN especifica qué extensiones de dominios están disponibles y mantiene una base de datos centralizada de dónde apuntan los dominios web.

Todos los sitios web que visitas constan de dos elementos principales: un dominio web y un servidor web.

Un servidor web es una máquina física que aloja los archivos y las bases de datos que conforman tu sitio web y los envía a las personas a través de Internet cuando visitan tu sitio desde su computadora.

El dominio web o dominio de internet es lo que las personas escriben para acceder a tu sitio, y apunta el navegador web hacia el servidor que almacena esos recursos. Sin un

dominio web, las personas tendrían que recordar la dirección IP específica de tu servidor, y eso simplemente no va a suceder. (Gustavo B., ¿Qué es un dominio web?, 2019)

1.6.8. Tecnologías web

1.6.8.1. HTML5

Es el último estándar para que los navegadores puedan mostrar e interactuar con páginas web. Aprobado en 2014, es la primera actualización de HTML en 14 años.

El propósito de HTML5 es principalmente hacer que sea más fácil para los desarrolladores web y los creadores de navegadores a seguir estándares basados en el consenso que hacen el cumplimiento más eficiente y potenciador. También está diseñado para proporcionar mejores experiencias de usuarios, más rápidas y más consistentes para los visitantes de escritorio y móviles.

Estas son algunas mejoras clave en HTML5:

- Hay una estructura de elementos más sencilla y más directa a las páginas, que facilita la creación, el ajuste y la depuración, así como la creación de servicios automatizados que le ayudan a encontrar recursos importantes en la web.
- Proporciona elementos estándar para objetos multimedia comunes que anteriormente requerían plugins molestos para audio, video, etc. Estos complementos necesitan ser actualizados regularmente, es decir, descargas repetidas para administrar la seguridad.
- Hay integración nativa con interfaces para aprovechar las modernas necesidades de la Web y de los dispositivos móviles. Uno de mis ejemplos favoritos de esto es la geolocalización, que le permite determinar las coordenadas GPS de un visitante web a través de su navegador. Esta función estaba previamente restringida a aplicaciones de teléfono equipadas con GPS. (Reifman, 2016).

1.6.8.2. CSS3

El CSS (Cascading Style Sheets), en español Hojas de Estilo en Cascada, fue desarrollado por W3C en 1996.

Se le denomina Hojas de Estilos en Cascada porque las características se aplican de arriba a abajo mediante reglas que poseen un esquema prioritario.

Esta especificación es un lenguaje de diseño gráfico que se escribe dentro del código HTML del sitio web y, permite crear páginas de una manera más exacta y aplicarles estilos (colores, márgenes, formas, tipos de letras, etc.) por lo que se tiene mayor control de los resultados finales.

La tecnología CSS está diseñada para marcar la separación del contenido de las páginas web y la forma de presentación de estas. Lo que genera múltiples beneficios, como:

- Presentar el documento final en diferentes estilos (pantalla, voz, impresión).
- Tener un sitio web responsivo.
- Evitar hacer archivos demasiado pesados.
- Definir el estilo visual de todo un sitio web. Así, si cambiamos una página, cambiarán todas automáticamente.
- Trabajar con estándares y separar (hasta cierto punto) la estructura de la presentación logrando un trabajo más definido.
- Provee más flexibilidad y control en las especificaciones del sitio web.
- Simplifica la creación de la página.

Las ventajas de utilizar las Hojas de Estilo en Cascada son numerosas, algunas ya te las mencionamos anteriormente y otras te las diremos a continuación:

- La posibilidad de hacer modificaciones visuales en un solo lugar (sin tener que editar todos los documentos HTML por separado).
- La facilidad de organizar y hacer cambios.

- El permitir que las páginas descarguen con mayor rapidez.
- Mediante la separación se mejora la accesibilidad y escaneabilidad del contenido.
- La simplicidad para crear versiones dependiendo de los dispositivos (tablets, teléfonos inteligentes o dispositivos móviles, entre otros), etc.

1.6.8.3. JavaScript

JavaScript es uno de los lenguajes de programación más populares para el desarrollo web, ejecutado del lado del cliente (esta expresión significa que los scripts o programas son ejecutados en el navegador del usuario), y funciona como una extensión del código HTML.

Por ser un lenguaje interpretado, no necesita de compilación y, por lo tanto, no utiliza los recursos del servidor. JavaScript es un lenguaje que soporta la programación orientada a objetos. Aunque no posee varias características de esta orientación nativamente, es posible realizar algunas emulaciones.

Es muy importante recordar que JavaScript no es lo mismo que Java, su sintaxis es diferente y necesita un compilador para que el navegador pueda interpretarlo. En cambio, JavaScript tiene su propio script interpretado por él mismo.

El script del desarrollador será enviado junto al HTML para el navegador, que, al leerlo, hará la diferenciación del código HTML por medio de la tag <script>. (Pelhon, 2019).

1.6.8.4. Bootstrap 4

Es un framework de plantillas web prediseñadas en la versión 4, es una librería desarrollada por Twitter que simplifica el proceso de maquetación de páginas web según estándares del diseño web. Provee un conjunto de librerías CSS y JavaScript con plantillas listas a utilizar para crear interfaces de usuario rápidas agradables y atractivas.

Este Bootstrap templates provee plantillas CSS prediseñadas que podemos personalizar a nuestro gusto.

Entre los componentes o plantillas que provee podemos mencionar: grid, cards, fonts, navbar, form, modal, image gallery, etc. Básicamente son bloques de código HTML5, CSS y JavaScript que solo se añaden a una página web en edición según las necesidades.

Dichos componentes son adaptables o responsivos para todos los tamaños o resoluciones de pantalla y compatibles con la mayoría de navegadores comúnmente utilizados. (RBjuarez, 2019).

1.6.8.5. AJAX

Es el acrónimo de Asincrónico JavaScript y XML, es básicamente un conjunto de tecnologías de desarrollo web que tiene como foco la construcción de aplicaciones más interactivas, teniendo como uno de los principales objetivos la respuesta rápida de páginas web a través del intercambio de pequeñas informaciones entre el servidor web, además de ahorrar la necesidad de la página de ser recargada entera cada vez que se realiza una nueva acción.

Los recursos AJAX, hablando de forma bien generalista, se pueden resumir en tres palabras: usabilidad, velocidad e interactividad. Es un paquete de tecnologías que posibilitan hacer la relación entre interfaz web y usuario más productiva y rica.

Conceptualmente hablando, utilizar AJAX remite a conocer bien JS, trabajando con, DOM, XML y CSS.

- Aplicación web común: La solicitud HTTP se envía desde el explorador al servidor; este servidor recibe la solicitud y busca los datos; después de la búsqueda el servidor retorna con los datos solicitados; por fin el navegador, recibiendo los datos, recarga la página mostrando la nueva información (resalvo que durante todo este proceso el usuario no puede hacer nada a no ser esperar los resultados de su solicitud ser cargados, ese proceso además de ser tedioso, acaba requiriendo del servidor muchos recursos).

- Aplicación AJAX: A través del navegador, se genera una llamada en JS y posteriormente activa el XMLHttpRequest; en el fondo de este proceso, el navegador crea una solicitud HTTP para el servidor; entonces el servidor recibe la solicitud, hace la búsqueda y devuelve los datos al navegador; por fin el navegador recibe entonces los datos enviados por el servidor y los muestra directamente en la página (sin necesitar que la página sea recargada). (Silva, 2019).

1.6.9. Arduino

Es una plataforma de creación de electrónica de código abierto, la cual está basada en hardware y software libre, flexible y fácil de utilizar para los creadores y desarrolladores. Esta plataforma permite crear diferentes tipos de microordenadores de una sola placa a los que la comunidad de creadores puede darles diferentes tipos de uso. El Arduino es una placa basada en un microcontrolador ATMELE. Los microcontroladores son circuitos integrados en los que se pueden grabar instrucciones, las cuales las escribes con el lenguaje de programación que puedes utilizar en el entorno Arduino IDE. Estas instrucciones permiten crear programas que interactúan con los circuitos de la placa.

El microcontrolador de Arduino posee lo que se llama una interfaz de entrada, que es una conexión en la que podemos conectar en la placa diferentes tipos de periféricos. La información de estos periféricos que conectes se trasladará al microcontrolador, el cual se encargará de procesar los datos que le lleguen a través de ellos. El tipo de periféricos que puedas utilizar para enviar datos al microcontrolador depende en gran medida de qué uso le estés pensando dar. Pueden ser cámaras para obtener imágenes, teclados para introducir datos, o diferentes tipos de sensores. También cuenta con una interfaz de salida, que es la que se encarga de llevar la información que se ha procesado en el Arduino a otros periféricos.

Estos periféricos pueden ser pantallas o altavoces en los que reproducir los datos procesados, pero también pueden ser otras placas o controladores.

Arduino es un proyecto y no un modelo concreto de placa, lo que quiere decir que compartiendo su diseño básico te puedes encontrar con diferentes tipos de placas. Las hay de varias formas, tamaños y colores para a las necesidades del proyecto en el que estés trabajando, las hay sencillas o con características mejoradas, Arduinos orientados al Internet de las Cosas o la impresión 3D y, por supuesto, dependiendo de estas características te encontrarás con todo tipo de precios. Además, las placas Arduino también cuentan con otro tipo de componentes llamados Escudos (Shields) o mochilas. Se trata de una especie de placas que se conectan a la placa principal para añadirle una infinidad de funciones, como GPS, relojes en tiempo real, conectividad por radio, pantallas táctiles LCD, placas de desarrollo, y un larguísimo etcétera de elementos. (Yubal FM, 2018).

1.7. Diseño de arquitectura

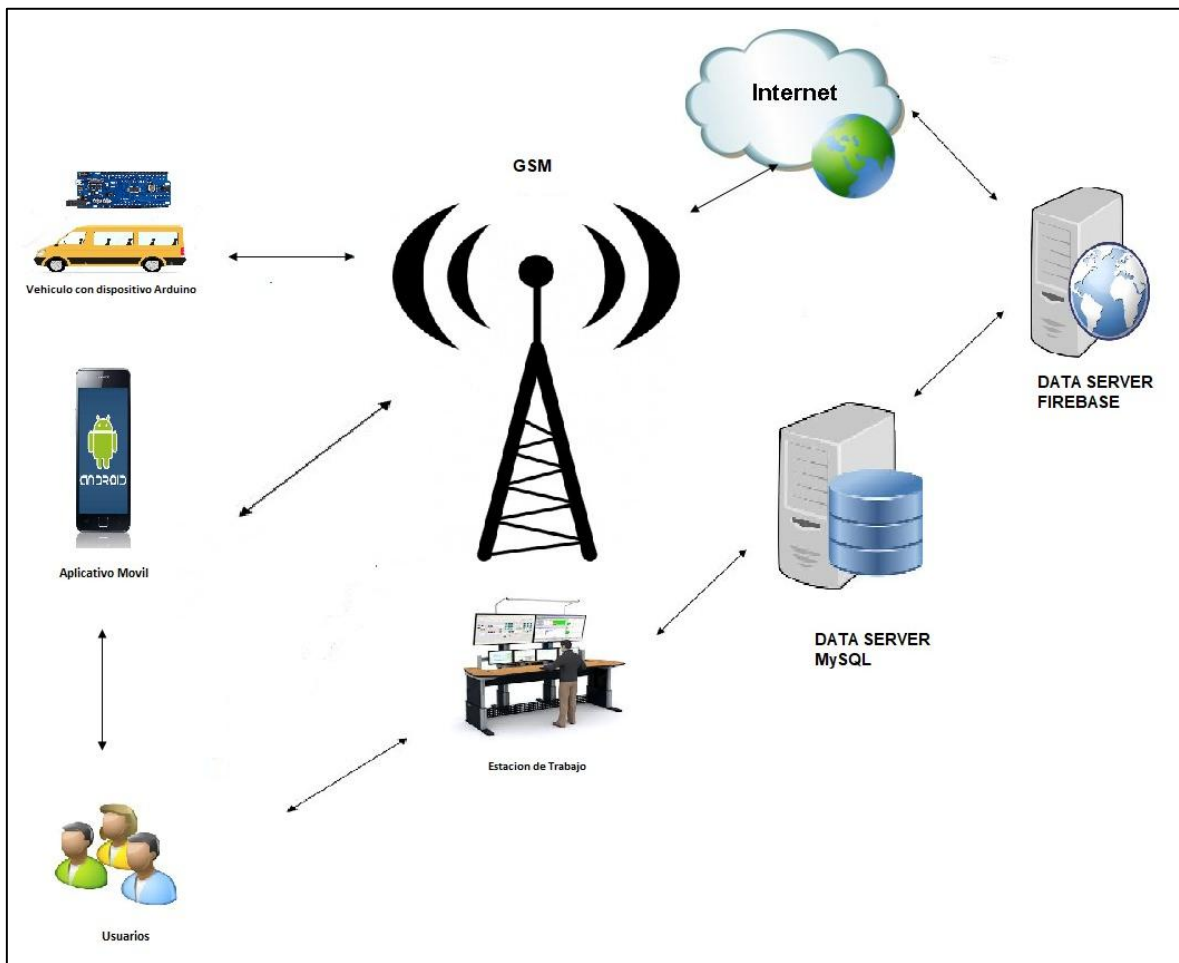


Figura 1. Diseño de arquitectura

Fuente: Elaboración propia

CAPÍTULO II: MÉTODOS Y MATERIALES

2.1. Procedimiento para el desarrollo de la investigación

La presente investigación, se desarrollará con los siguientes procedimientos:

- Se realizará un cuestionario a los padres de familia, el cual permitirá identificar los principales riesgos que existen a diario en el servicio de transporte de estudiantes en la ciudad de Chiclayo.
- Se realizará visitas al exterior de las instituciones educativas para entrevistar a los choferes o transportistas, lo cual permitirá conocer más a fondo el proceso que conlleva a contratar una movilidad escolar.
- Se utilizará la plataforma Arduino para armar el módulo de geolocalización. Los principales componentes electrónicos que se utilizarán son: Arduino Nano, es la placa donde se programará la logica para el funcionamiento del sistema informático. U-blox NEO-6M, es un GPS satelital que permitirá captar las coordenadas de los satélites para luego enviarlas a Arduino. SIM8001, es un módulo GSM que permitirá conectarnos a internet mediante una tarjeta SIM utilizando la tecnología GSM
- Se creará una cuenta de correo electrónico Gmail, que proporcionará acceso para usar la plataforma Firebase de Google, el cual ofrece el servicio de base de datos en tiempo real (Realtime Database). A través de la consola de Firebase, se creará el proyecto denominado moviescool, el cual permitirá crear y habilitar las reglas de seguridad de la base de datos en tiempo real.

- Se creará un sistema web que permitirá tener control de la información de los procesos del servicio de transporte escolar.
- Se creará un aplicativo móvil que se encargará de realizar el seguimiento en tiempo real de las unidades escolares a través del módulo de geolocalización con Arduino.

2.2. Población y muestra

Para el presente estudio se consideró en la población a instituciones educativas particulares de la ciudad de Chiclayo, choferes o transportistas que realizan el servicio de movilidad escolar y padres de familia que contratan el servicio de movilidad escolar (02 instituciones educativas, 05 choferes, 13 padres de familia)

2.3. Técnicas e instrumentos de la recolección de datos

2.3.1. Técnicas

2.3.1.1. Observación

Se utilizó la observación para estudiar el estado actual del servicio de transporte escolar, se permitió obtener un informe de estudio de campo, que sirvió de ayuda para el diseño propuesto; además, de ser de mucha utilidad para un análisis posterior.

2.3.1.2. Entrevista

Se realizó una serie de visitas al exterior de las instituciones educativas privadas y se aplicó la entrevista a los choferes o transportistas que ofrecen el servicio de movilidad escolar en la ciudad de Chiclayo, además permitió conocer de manera directa los problemas que acontecen en el servicio de transporte escolar, en donde se obtuvo conocimiento de los procesos internos entre los padres de familia y choferes o transportistas; relacionados a los contratos o acuerdos, horarios de recogidas del escolar en casa para el traslado al colegio y viceversa, entre otros; con el propósito de identificar los requisitos necesarios para el desarrollo del sistema informático.

2.3.2. Instrumentos

2.3.2.1. Cuestionario

El instrumento utilizado fue el cuestionario, que es un instrumentos de investigación que posee un alto grado objetivo, además de que es un medio útil para recoger información en tiempo relativamente breve. (Detalle del cuestionario en el ANEXO 1).

Las preguntas fueron hechas de acuerdo al ajuste de los propósitos y fueron validadas según por la mayoría de resultados satisfactorios.

CAPÍTULO III: RESULTADOS Y DISCUSIÓN

3.1. Identificación de la situación actual de las unidades escolares

Para identificar la problemática que presenta el servicio de transporte de estudiantes, se elaboró un cuestionario haciendo uso de la herramienta que nos proporciona Google Forms (Formulario de Google); donde se obtuvo información a través de un formulario de un total de 14 preguntas obligatorias, lo cual se diseñó específicamente para los padres de familia que contratan el servicio de movilidad escolar. El cuestionario se envió a un total de 15 padres de familia a través de un enlace a sus correos electrónicos asegurando la confidencialidad de los encuestados (ANEXO 1).

El primer resultado de la encuesta, indica que sólo el 66.7% de los encuestados utiliza el servicio de movilidad escolar por falta de tiempo, mientras que 16.7% de los encuestados lo utiliza por la lejanía entre la casa y el centro educativo; además por el riesgo de llegar tarde.

El tercer resultado de la encuesta, indica que sólo el 83.3% de los encuestados, se les avisa por medio del claxon o bocina del vehículo que la movilidad escolar llegó a la casa para llevar a su hijo al colegio, mientras que un 16.7% de los encuestados se les avisa por medio de una llamada telefónica.

El cuarto resultado de la encuesta, indica que sólo un 66.7% de los encuestados informa a la movilidad escolar por medio de una llamada telefónica que su hijo no podrá ir al colegio por alguna situación de enfermedad u otra índole, mientras que un 33.3% de los encuestados espera la llegada de la movilidad escolar para informar sobre tal situación.

Así mismo nos indica que un 75% de los encuestados desconoce sobre el estado actual en que se encuentra su hijo.

En general, los resultados obtenidos con la encuesta, indica que existe problemas en cómo se lleva el servicio de las movilidades escolares, reflejándose que se desconoce lo que sucede a lo largo del recorrido hacia el colegio; e incluso si existen demoras en las llegadas, así mismo se identifica que el problema es la falta de comunicación entre los padres de familia, colegios y transportista; el cual no permite saber con exactitud si ocurrieron incidencias durante el recorrido tanto en el horario de ingreso como de salida del escolar, entre otros.

3.2. Configuración inicial para el desarrollo del sistema informático

Para el desarrollo del sistema informático, se realizaron diversas configuraciones y descargas de software, para lograr un entorno de desarrollo más adecuado; y nos ayude a lograr el objetivo planteado en el desarrollo.

3.2.1. Diseño de base de datos

3.2.1.1. *Diseño lógico*

Para diseñar el modelo lógico de la base de datos, se usó ERwin Data Modeler y se instaló la versión 7.3.8.

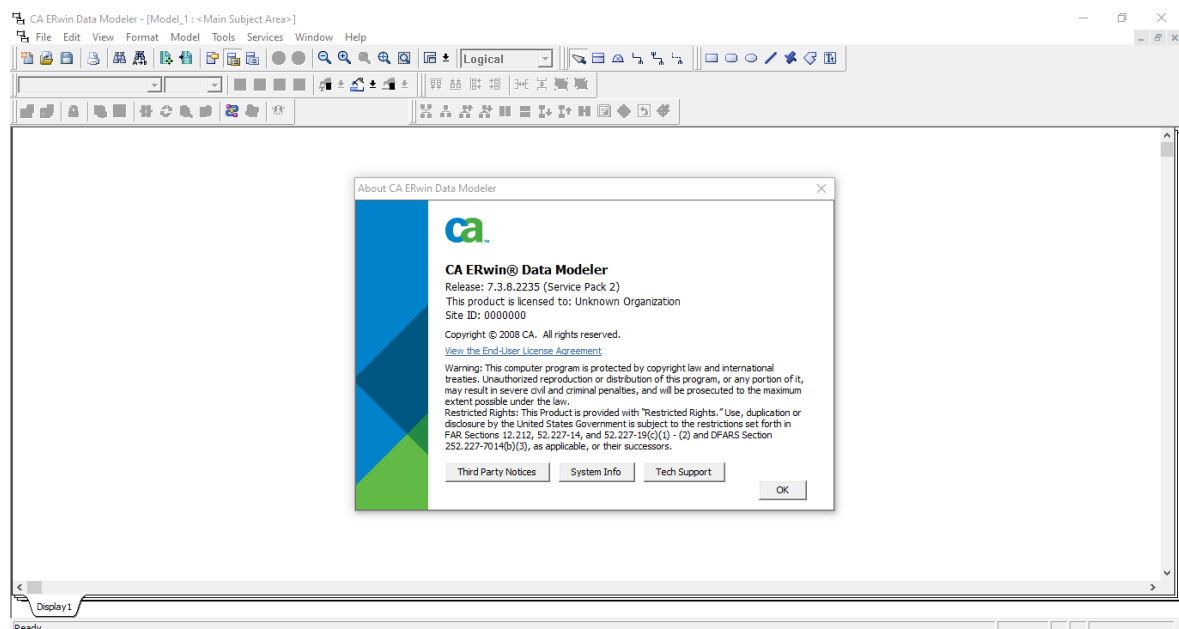


Figura 2. Instalación de ERwin Data Modeler versión 7.3.8

Fuente: Elaboración propia

3.2.1.2. Gestor de base de datos

Para la administración de la base de datos, se usó DBeaver, que es una herramienta que nos permite gestionar múltiples bases de datos y se instaló la versión 6.3.1.

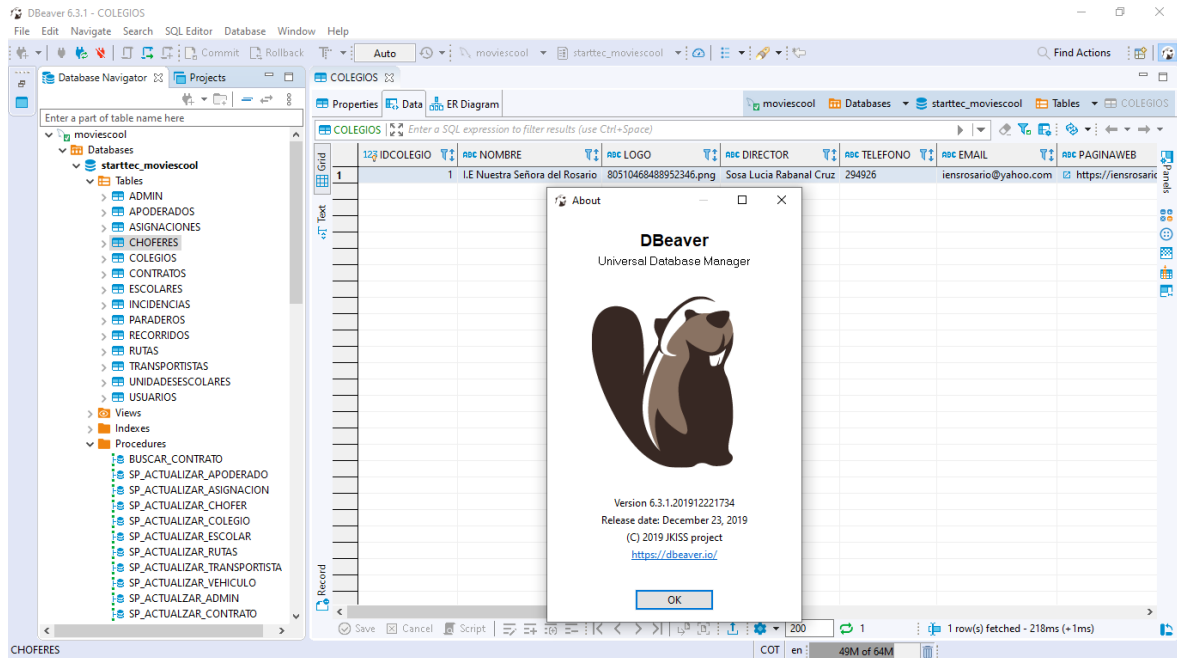


Figura 3. Instalación de DBeaver versión 6.3.1

Fuente: Elaboración propia

3.2.2. Entorno de desarrollo integrado

3.2.2.1. Sublime Text

Para la programación del sistema web, se instaló Sublime Text 3, que es un editor de código que soporta múltiples lenguajes de programación entre ellos PHP y de marcados como HTML5.

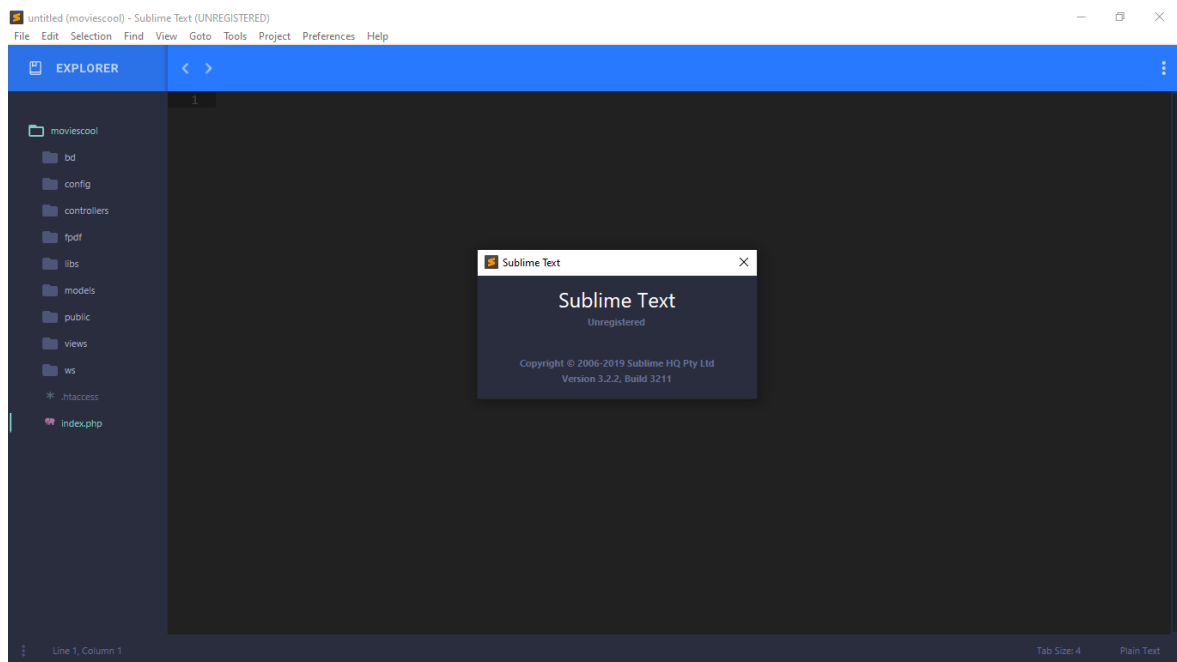


Figura 4. Instalación Sublime Text versión 3

Fuente: Elaboración propia

3.2.2.1.1. Estructura de proyecto

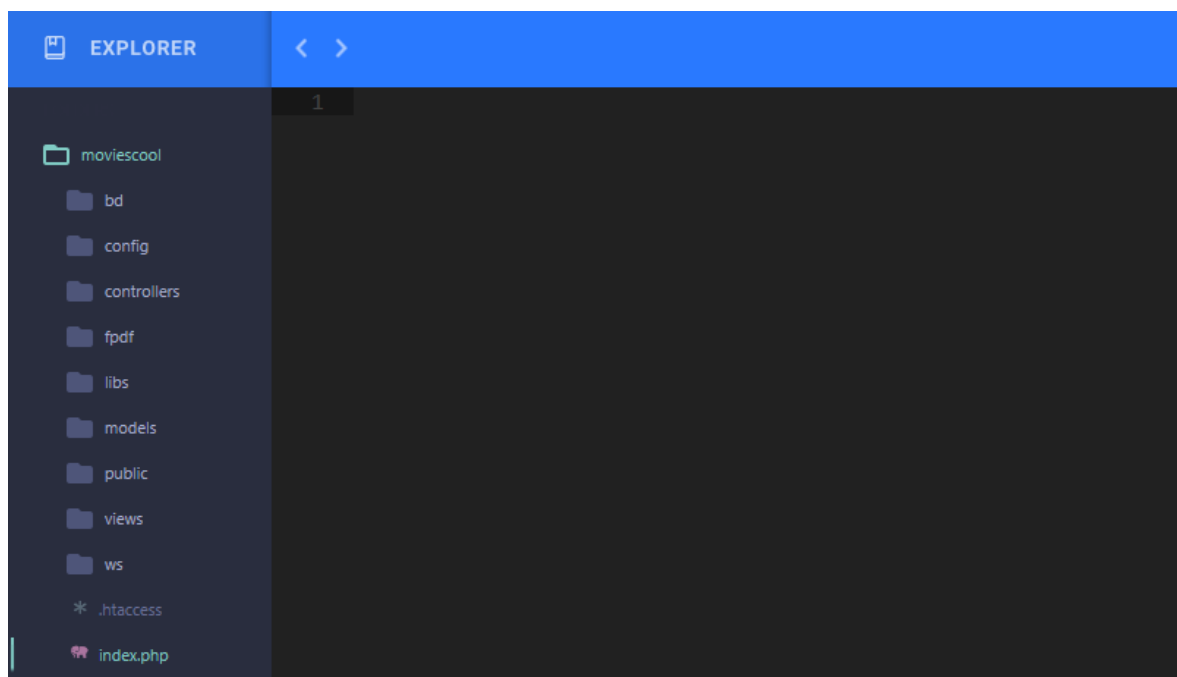


Figura 5. Estructura de proyecto web

Fuente: Elaboración propia

- **moviescool:** Es la carpeta raíz del proyecto.
- **bd:** Es la carpeta que contiene el script para crear la base de datos, tablas y procedimientos almacenados.
- **config:** Es la carpeta que contiene el archivo de configuración de la base de datos y el archivo de configuración para acceder al proyecto web.
- **controllers:** Es la carpeta que contiene las clases controladores. Es la capa que une la vista y el modelo. El usuario solicitará información por medio de la vista y esta hará la petición al controlador. Posteriormente, este, realizará la petición al modelo.
- **fpdf:** Es la carpeta que contiene la librería fpdf para poder generar archivos en formato .pdf.
- **libs:** Es la carpeta que contiene las clases que servirán de apoyo para desarrollar nuestro sistema en php nativo.
- **models:** Es la carpeta que contiene las clases modelos. Gestiona todo lo relacionado con la información y la iteración con los datos de nuestra aplicación. Toda la petición de acceso a los datos pasará por esta capa.
- **public:** Esta carpeta contiene todos los archivos css, js, img, bootstrap.
- **views:** Es la carpeta que contiene las vistas del proyecto. Es la capa que nos mostrará la información formateada. También desde dónde el usuario puede solicitar más información.
- **ws:** Es la carpeta que contiene archivos web service para realizar nuestras peticiones desde el aplicativo móvil.
- **.htaccess:** Es un archivo de configuración que permite configurar el comportamiento del servidor ante errores de conexión y optimizar cargas de las páginas.

- **index.php:** Es el archivo que sirve de controlador frontal donde se cargará todo desde la aplicación.

3.2.2.2. *Android Studio*

Para la programación del aplicativo móvil, se instaló Android Studio en su versión 3.5.1, que es el entorno de desarrollo oficial para Android.

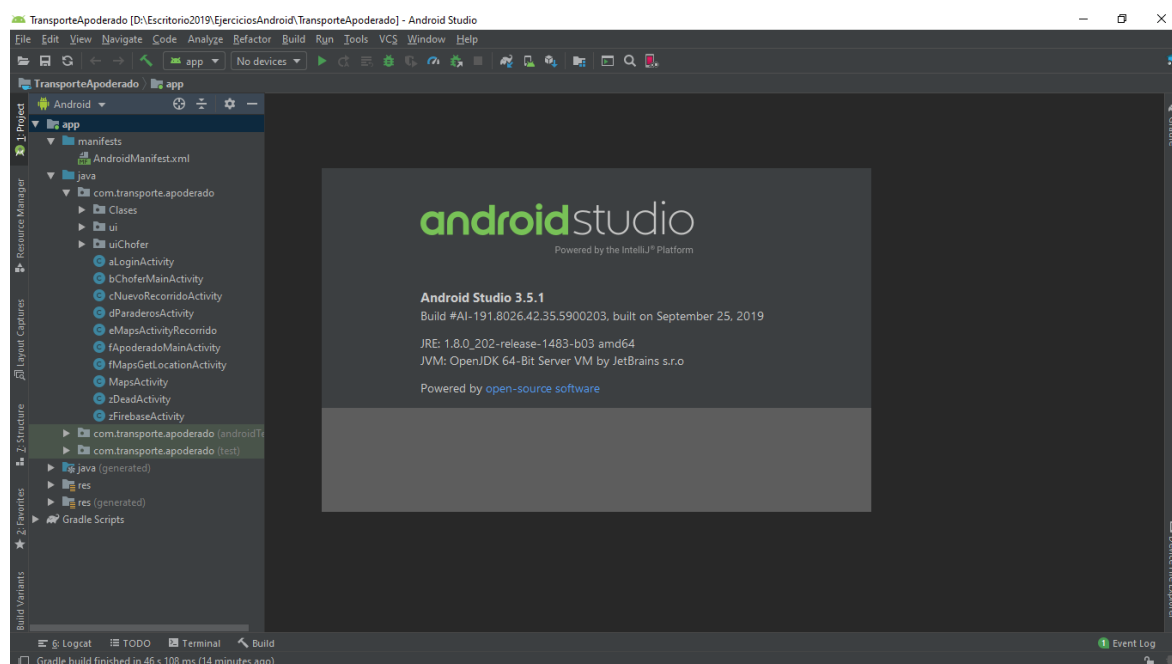


Figura 6. Instalación de Android Studio versión 3.5.1

Fuente: Elaboración propia

3.2.3. **Firestore**

3.2.3.1. *Realtime Database*

Para el monitoreo en tiempo real del sistema informático, se utilizó la plataforma Firebase de Google; que está orientada a crear aplicaciones de alta calidad, el cual ofrece el servicio de base de datos en la nube (Realtime Database). Los datos se almacenan en formato JSON y se sincronizan en tiempo real con cada cliente conectado. La comunicación es bidireccional, es decir que cuando se modifican datos en la base de datos; estos cambios se van reflejados directamente en la web y viceversa.

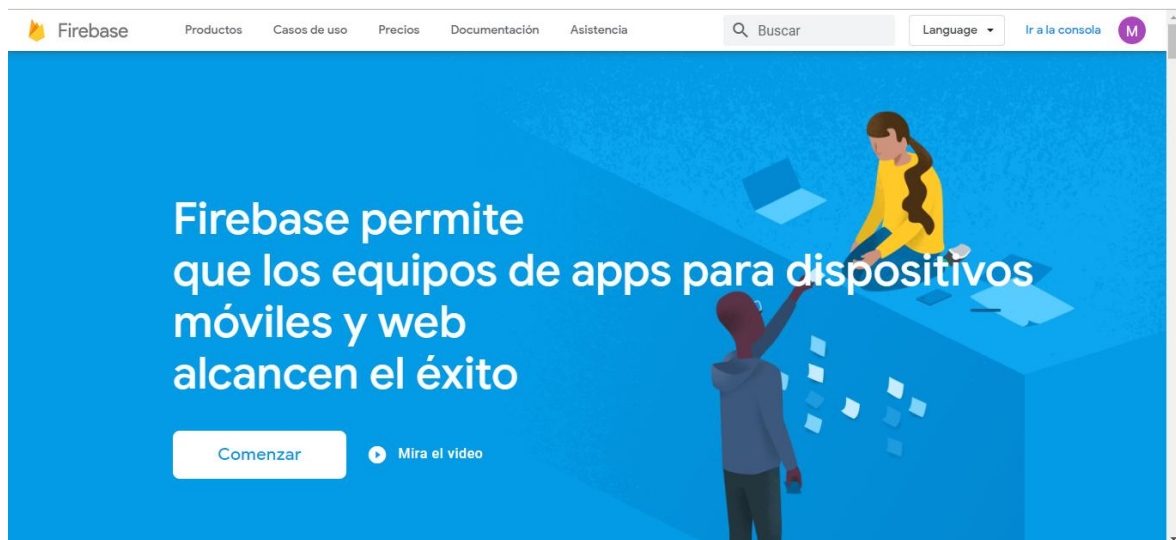


Figura 7. Plataforma Firebase de Google

Fuente: Elaboración propia

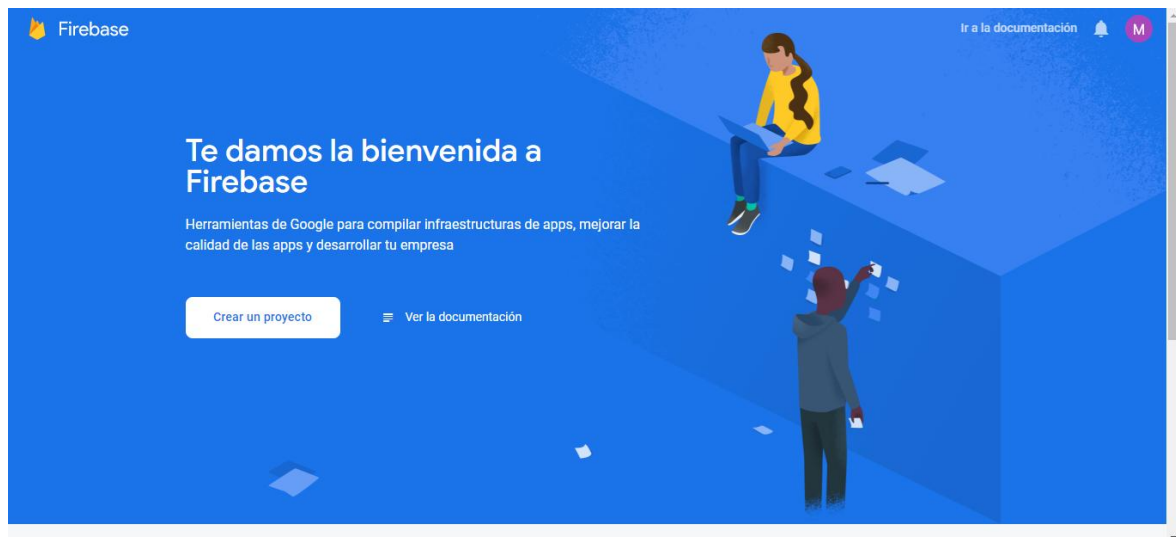


Figura 8. Pantalla de bienvenida de la consola de Firebase

Fuente: Elaboración propia

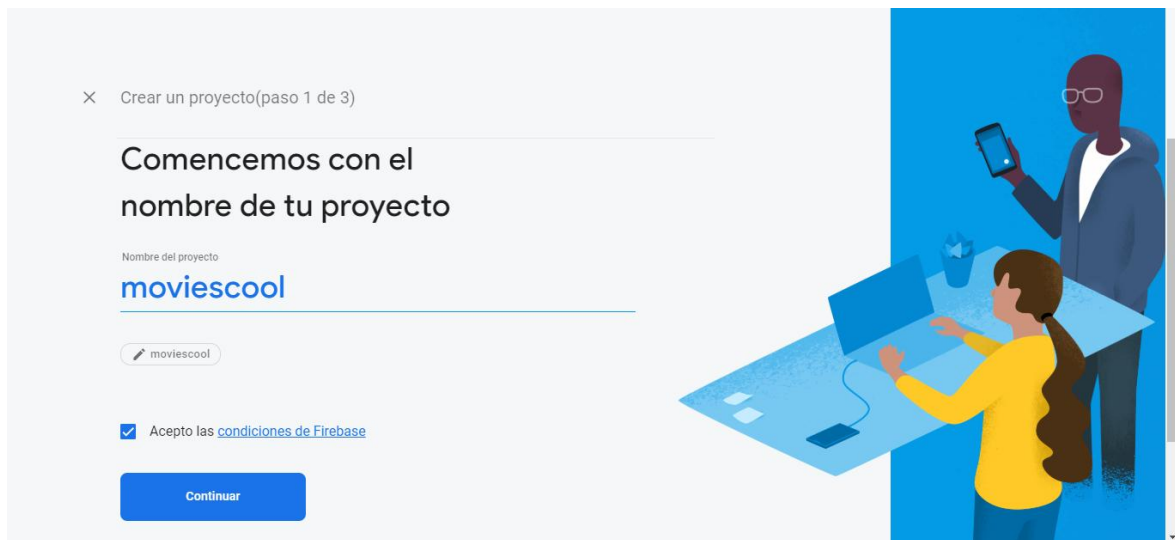


Figura 9. Creación de nombre del proyecto moviescool

Fuente: Elaboración propia

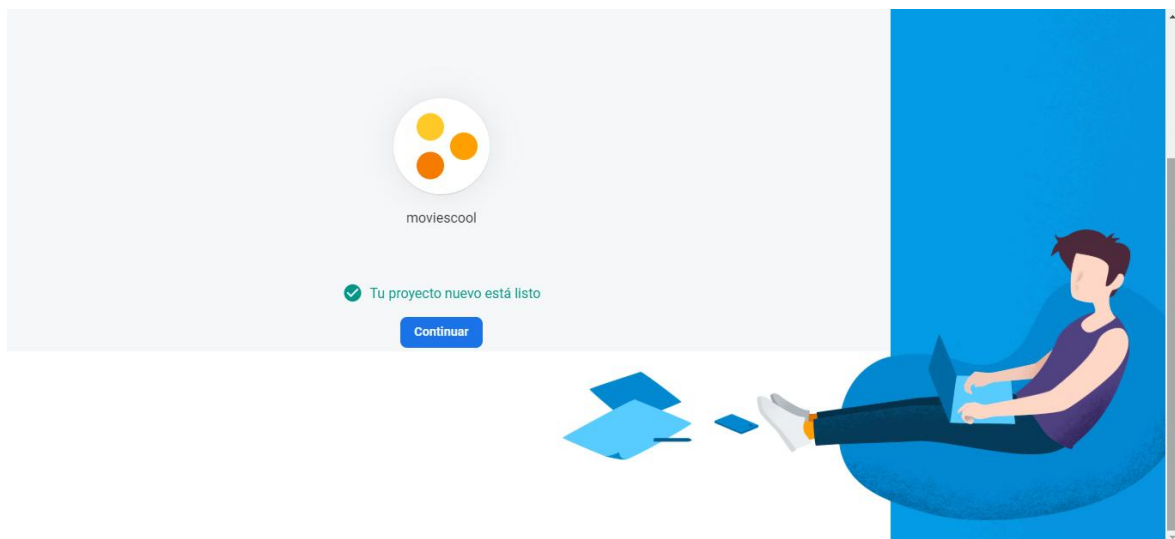


Figura 10. Proyecto moviescool listo para configurar con Firebase Realtime Database

Fuente: Elaboración propia

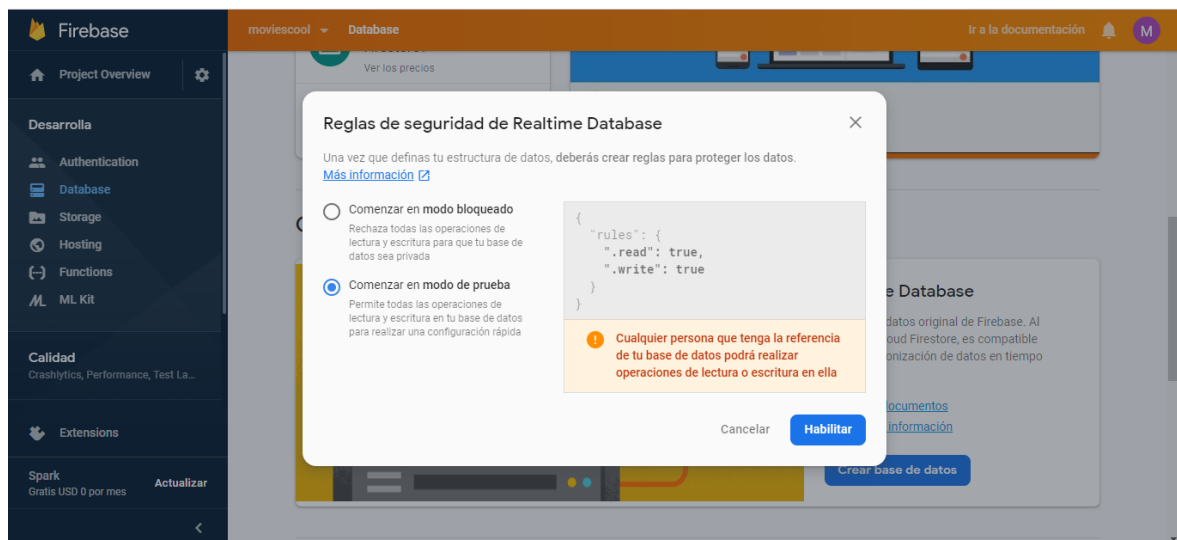


Figura 11. Crear y habilitar reglas de seguridad de Realtime Database

Fuente: Elaboración propia

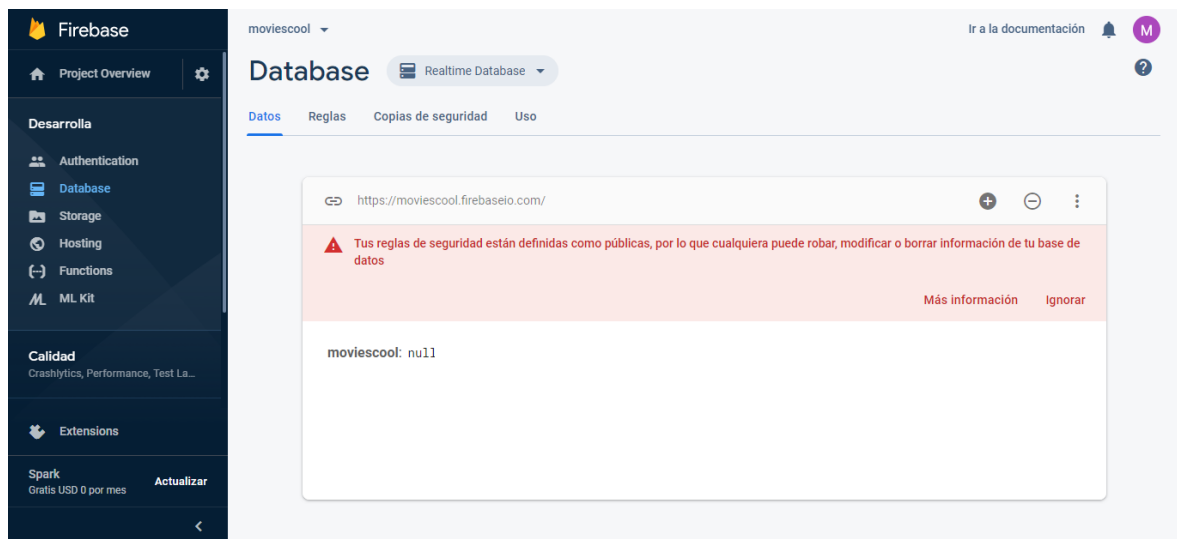


Figura 12. Creación de Realtime Database moviescool por defecto en null

Fuente: Elaboración propia

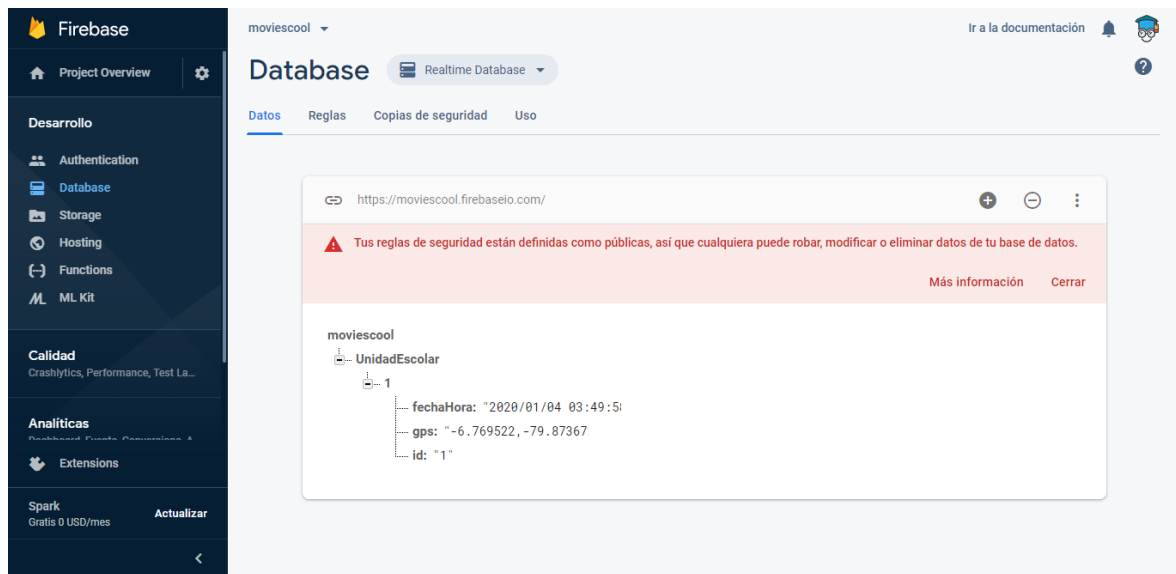


Figura 13. Realtime Database en formato JSON configurado desde el módulo GPS con Arduino

Fuente: Elaboración propia

3.2.4. Servidor Web

3.2.4.1. WampServer

Antes de subir nuestra aplicación a un servidor de producción, nuestro sistema web estará alojado en un servidor de desarrollo lo cual no estará conectado a internet.

Se instaló WampServer 3, que es un sistema que nos permite ejecutar de forma local nuestras páginas web basadas en PHP además tiene una gran variedad de herramientas para poder administrarlo, teniendo el servidor web Apache, gestor de base de datos MySQL y PHP para la programación.

3.2.4.2. Hosting web

Concluidos nuestros cambios y modificaciones en el servidor de desarrollo de nuestro sistema web, subiremos nuestra aplicación a un servidor de producción.

Nuestro servidor web tiene un directorio para la publicación de las páginas web. Ese sería el lugar donde hay que subir los archivos .php.

Nuestro alojamiento se ha contratado para el dominio <https://www.starttecperu.com/>, deberíamos subir los archivo al directorio de publicación y accederíamos al archivo escribiendo: <https://www.starttecperu.com/rest/moviescool/>.

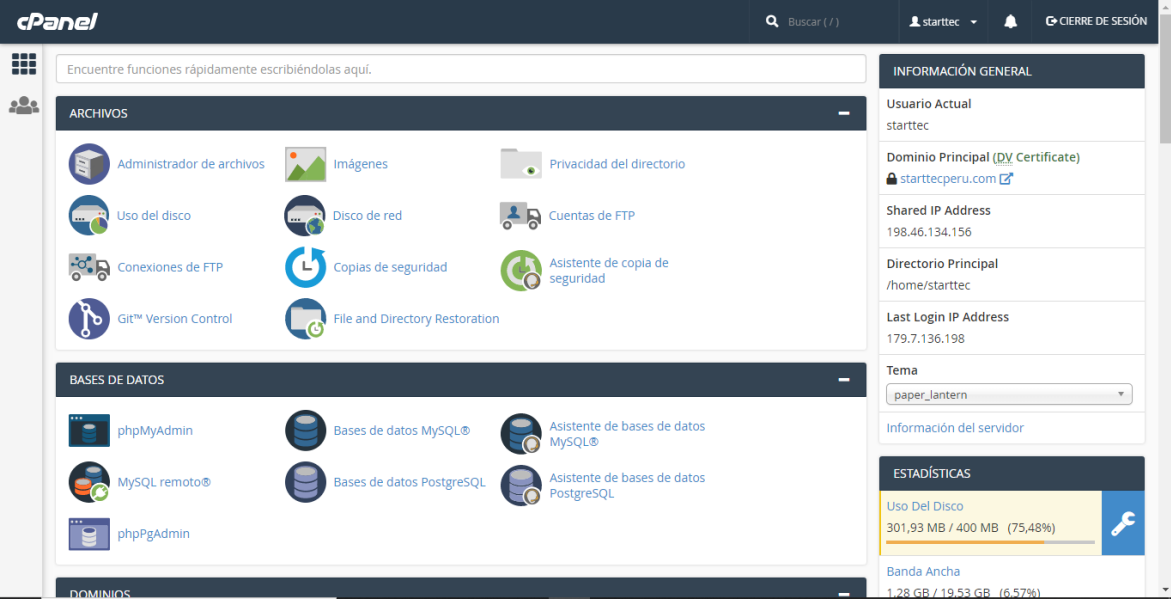


Figura 14. Panel de administración desde el dominio starttecperu.com

Fuente: Elaboración propia

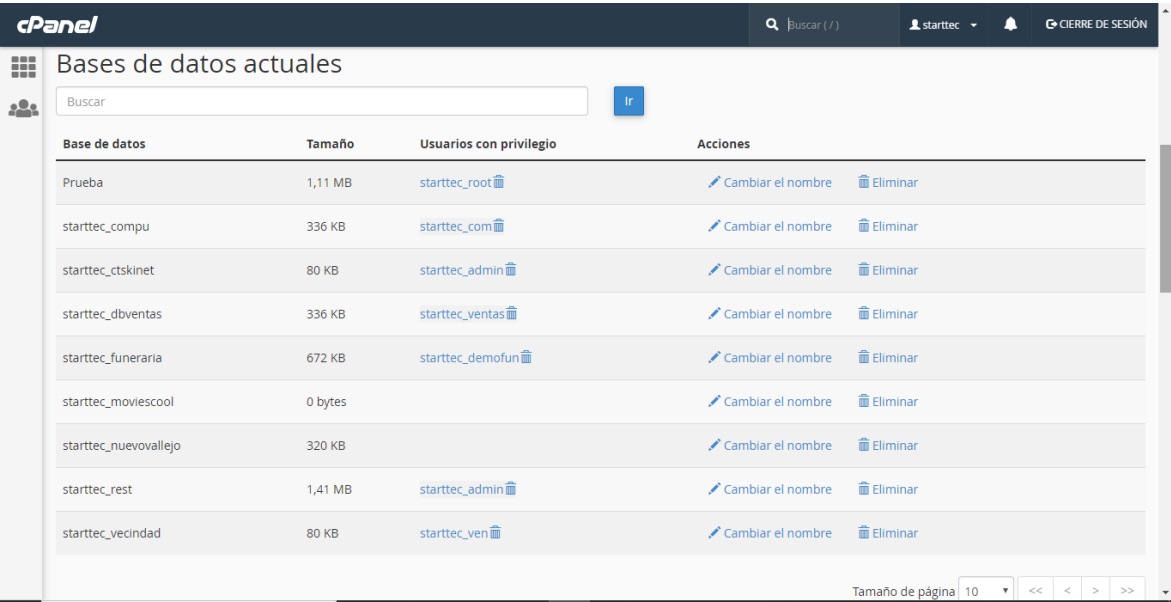


Figura 15. Creación de base de datos starttec_moviescool desde el panel starttecperu.com

Fuente: Elaboración propia

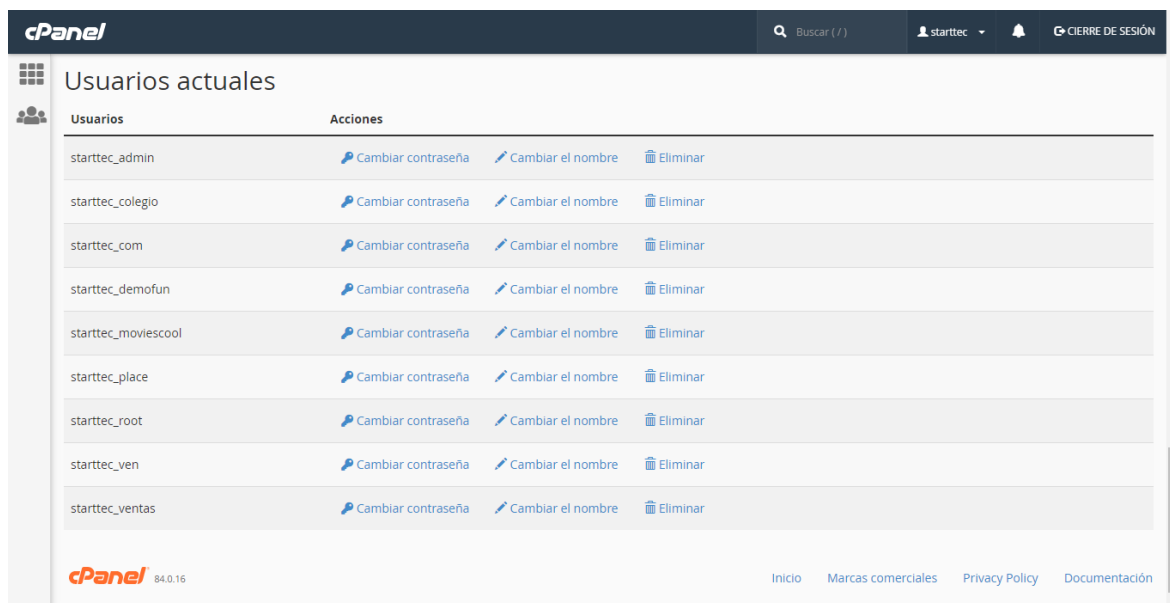


Figura 16. Creación de usuario starttec_moviescool desde el panel starttecperu.com

Fuente: Elaboración propia

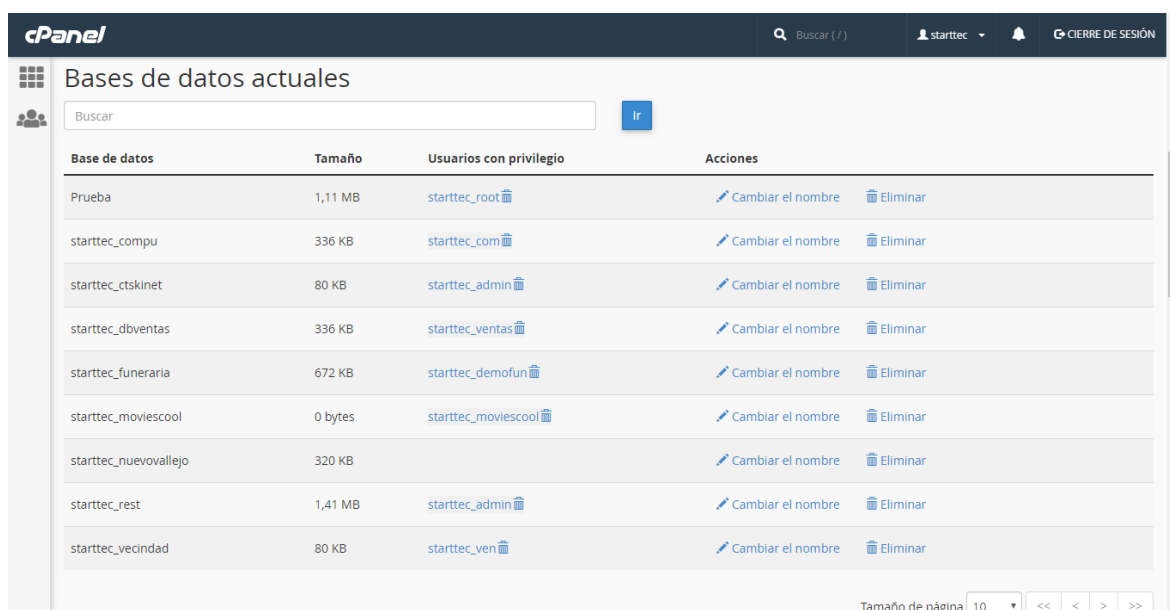


Figura 17. Usuario starttec_moviescool añadido a base de datos starttec_moviescool

Fuente: Elaboración propia

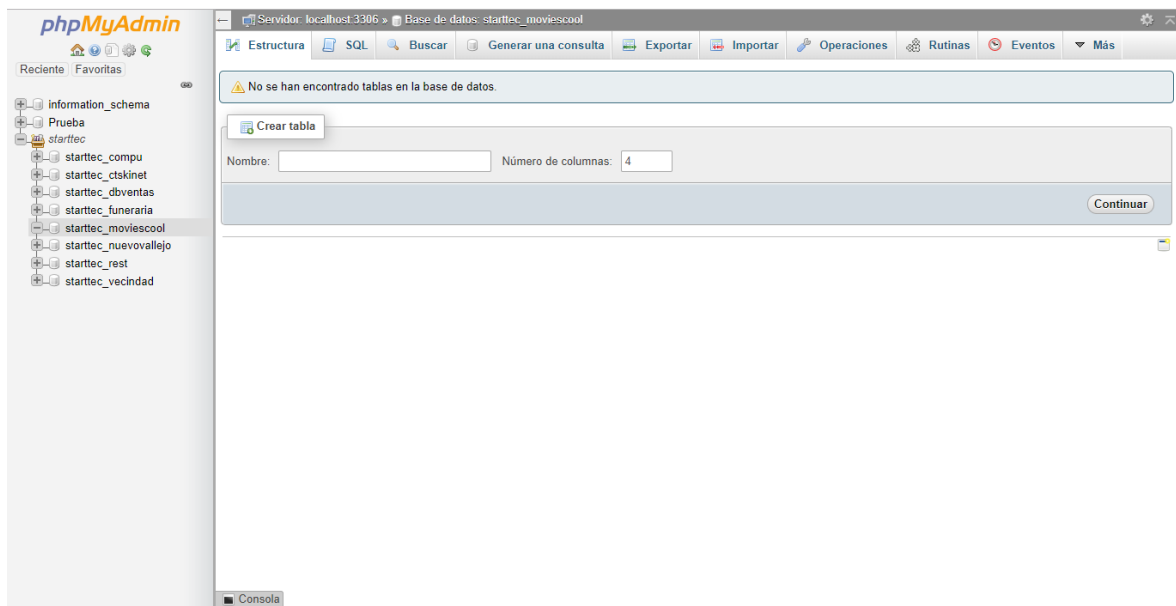


Figura 18. Panel phpMyAdmin desde el dominio starttecperu.com

Fuente: Elaboración propia

3.2.5. Plataforma Arduino

3.2.5.1. Esquema de circuitería de geolocalización

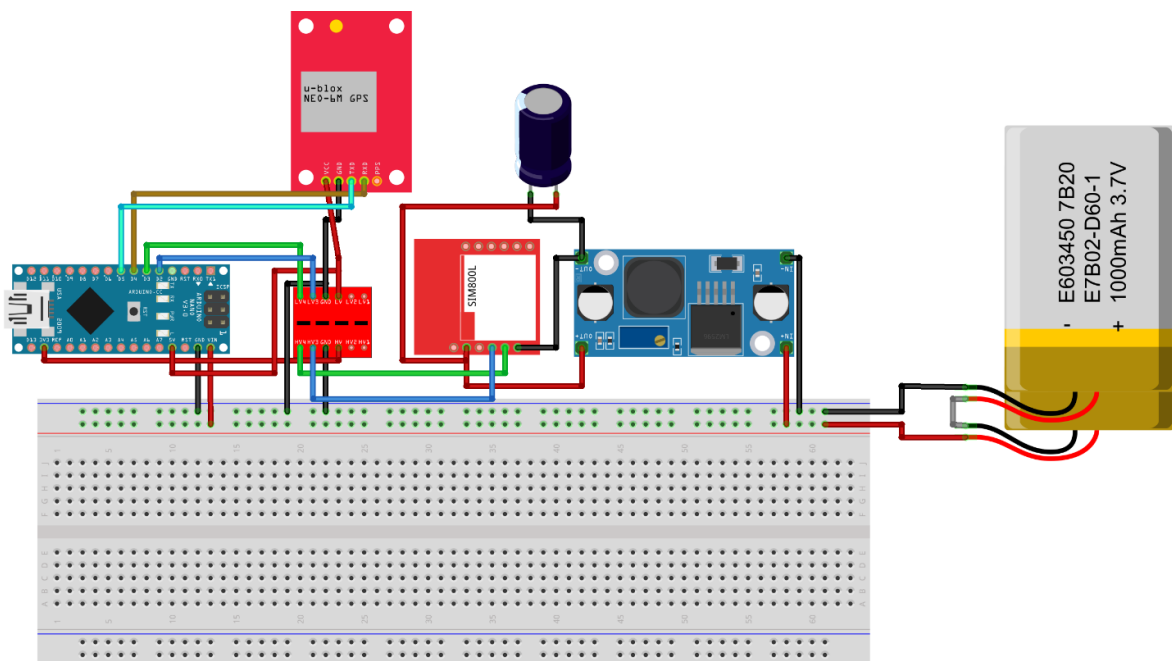


Figura 19. Diseño de circuitería del módulo de geolocalización con Arduino

Fuente: Elaboración propia

3.2.5.2. Componentes electrónicos

- **Arduino Nano:** Es el cerebro para el desarrollo de este sistema informático; en él se programará la lógica del funcionamiento del sistema informático.
- **U-blox NEO-6M GPS:** Es un GPS Satelital que captará las coordenadas de los satélites para luego enviarlas al Arduino por comandos de programación.
- **Sim800l:** Es un módulo GSM que nos permitirá conectarnos a internet haciendo uso de un chip de cualquier operadora utilizando tecnología 2G. El Arduino a través del módulo GSM enviará las coordenadas hacia la Base de datos de Firebase.
- **Convertidor de nivel lógico:** Es un módulo que nos permitirá convertir la señal de 5v a 3.3v. Esto se utilizará dado que los pines de comunicación del módulo GSM Sim800l trabaja a 3.3v.
- **LM2596:** Es un regulador de voltaje; nos permite bajar el voltaje de 7.4v a 4.1v para la alimentación del módulo GSM Sim800l
- **Batería Lipo 7.4V:** La batería nos alimentará todo nuestro sistema. Esta batería es recargable.
- **Capacitor 1000uf:** Es un componente electrónico que nos servirá para disminuir el ruido en la alimentación de nuestro sistema.

3.2.5.3. Programación de módulo de GPS con Arduino

```
#define TINY_GSM_MODEM_SIM800
#include <TinyGPS++.h>
#include <TinyGsmClient.h>
#include <ArduinoHttpClient.h>
#include <SoftwareSerial.h>
#define SerialMon Serial
#define DEVICE_ID "1"
#define modemBAUD 4800
#define gpsBAUD 9600
// RX, TX serial modem
SoftwareSerial SerialAT(3,2);
```

```

// RX,TX serial GPS
SoftwareSerial ss(5, 4);
// Internet setting
// GSM apn
const char apn[] = "entel.pe";
const char user[] = "entel";
const char pass[] = "entel";
//Firebase Realtime Database setting
const char server[] = "moviescool.firebaseio.com";
const int port = 443;
const String UPDATE_PATH = "UnidadEscolar/"+ String(DEVICE_ID);
//global variables
int count=0;
String fireData="";

#ifdef DUMP_AT_COMMANDS
#include <StreamDebugger.h>
StreamDebugger debugger(SerialAT, SerialMon);
TinyGsm modem(debugger);
#else
TinyGsm modem(SerialAT);
#endif
TinyGsmClientSecure client(modem,0);
HttpClient https(client, server, port);
TinyGPSPlus gps;

void setup() {
    // set console baud rate
    SerialMon.begin(115200);
    delay(10);
    // check modem communication
    initializeModem();
    // test internet connection
    tesKoneksi();
    // open gps serial
    ss.begin(gpsBAUD);
}

void loop(){
    if(ss.isListening()){
        Serial.println("gps listening");
        while(fireData.equals("")){
            scan();
        }
    }else{
        Serial.println("gps not listening");
        fireData="";
        ss.begin(gpsBAUD);
    }
}

```

```

    }
}

//send data to firebase
void sendData(const char* method, const String & path ,
              const String & data, HttpClient* http){
    String response;
    // Currently, this is needed for HTTPS
    http->connectionKeepAlive();
    String url;
    if (path[0] != '/'){
        url = "/";
    }
    url += path + ".json";
    url += "?print=silent";
    url += "&x-http-method-override=";
    url += String(method);
    String contentType = "application/json";
    http->post(url, contentType, data);
    int statusCode = http->responseStatusCode();
    Serial.print("Status code: ");
    Serial.println(statusCode);
    response = http->responseBody();
    Serial.print("Response: ");
    Serial.println(response);
    if (!http->connected()){
        Serial.println();
        // Shutdown
        http->stop();
        Serial.println("HTTP POST disconnected");
        SerialAT.begin(modemBAUD);
        tesKoneksi();
        fireData="";
        ss.begin(gpsBAUD);
    }
}

//scanning gps location
void scan(){
    while (ss.available() > 0)
        if (gps.encode(ss.read()))
            displayInfo();
}

void displayInfo(){
    Serial.print(F("Realtime Loc: "));
    if (gps.location.isValid()){
        Serial.print(gps.location.lat(), 6);
    }
}

```

```

        Serial.print(F(", "));
        Serial.print(gps.location.lng(), 6);
        data();
    }else{
        Serial.print(F("INVALID"));
        data();
    }
    Serial.println();
    SerialAT.begin(modemBAUD);
    updateData();
}

// connect to internet
void tesKoneksi(){
    tes:
    SerialMon.print(F("Koneksi ke Jaringan..."));
    if (!modem.waitForNetwork()) {
        SerialMon.println(" gagal");
        delay(300);
        goto tes;
    }
    SerialMon.println(" OK");
    SerialMon.print(F("Koneksi ke apn: "));
    SerialMon.print(apn);
    if (!modem.gprsConnect(apn, user, pass)) {
        SerialMon.println(" gagal");
        delay(300);
        goto tes;
    }
    SerialMon.println(" OK");
}

//update data firebase
void updateData(){
    infoModem();
    sendData("PUT", UPDATE_PATH, fireData, &https);
    // kirimData();
}

//initialize modem sim800
void initializeModem(){
    SerialAT.begin(modemBAUD);
    delay(3000);
    SerialMon.println(F("Initializing modem..."));
    modem.restart();
    String modemInfo = modem.getModemInfo();
    SerialMon.print(F("Modem: "));
    SerialMon.println(modemInfo);
}

```

```

    if (!modem.hasSSL()) {
        SerialMon.println(F("SSL is not supported by this
        modem"));
        while(true){
            delay(1000);
        }
    }
    Serial.println("wait 10 detik");
    delay(10000);
}

//get gps data
void data(){
    fireData="";
    fireData += "{";
    fireData += "\"gps\":\":" + String(gps.location.lat(),6)+
        "\",\"+ String(gps.location.lng(),6)+\"\", \"";
}

//get modem data
void infoModem(){
    int csq = modem.getSignalQuality();
    int battLevel = modem.getBattPercent();
    fireData += " \"id\" :\"" + String(DEVICE_ID) + "\", \"";
    fireData += " \"fechaHora\" :\"" +
        gsmData(modem.getGsmLocation(),3)+' '
        +gsmData(modem.getGsmLocation(),4)+\"\"";
    fireData += "}";
}

//parsing gsm location coordinates and gsm time
String gsmData(String data,int pos){
    char separator=',';
    int index=pos;
    int found = 0;
    int strIndex[] = { 0, -1 };
    int maxIndex = data.length() - 1;
    for (int i = 0; i <= maxIndex && found <= index; i++){
        if (data.charAt(i) == separator || i == maxIndex){
            found++;
            strIndex[0] = strIndex[1] + 1;
            strIndex[1] = (i == maxIndex) ? i+1 : i;
        }
    }
    return found > index ? data.substring(strIndex[0],
    strIndex[1]) : "";
}

```

3.3. Planificación y diseño

Desarrollaremos el sistema informático haciendo uso de la metodología Xtreme Programming, es decir, iniciaremos con el establecimiento de historias de usuario.

3.3.1. Establecimiento de historias de usuario

Tabla 4

Establecimiento de historias de usuario

Nº Historia	Historia de Usuario	Usuario
1	Mantenimiento de colegios	Administrador web
2	Mantenimiento de apoderados	Colegio
3	Mantenimiento de escolares	Colegio
4	Mantenimiento de transportistas	Colegio
5	Mantenimiento de unidades escolares	Colegio, Transportista
6	Mantenimiento de choferes	Colegio, Transportista
7	Mantenimiento de rutas	Transportista
8	Mantenimiento de asignaciones	Transportista
9	Evaluar conformidad de unidad escolar	Colegio
10	Evaluar conformidad de chofer	Colegio
11	Generar contratos	Transportista
12	Ordenar domicilios para el horario de ingreso escolar	Transportista
13	Ordenar domicilios para el horario de salida escolar	Transportista
14	Monitorear unidades escolares desde el sistema web	Administrador web, Colegio, Transportista
15	Ingresar al sistema web	Administrador web, Colegio, Transportista
16	Ingresar al aplicativo móvil	Apoderado, Chofer
17	Configurar domicilio	Apoderado
18	Registrar inasistencia	Apoderado

19	Historial de recorridos	Apoderado
20	Resetear contraseñas desde el aplicativo móvil	Apoderado, Chofer
21	Crear recorrido	Chofer
22	Notificar incidencias	Chofer
23	Marcar recogida	Chofer
24	Reporte de rutas	Colegio
25	Reporte de asignaciones	Colegio
26	Reporte de contratos	Colegio
27	Reporte de asistencia	Colegio
28	Resetear contraseñas desde el sistema web	Colegio, Transportista

Fuente: Elaboración propia

3.3.2. Iteración 1

En la iteración 1, se pretende comenzar el desarrollo del sistema web con las funcionalidades relacionadas al mantenimiento de las tablas maestras (operaciones CRUD).

Se trata de tener preparadas las funcionalidades básicas, por otro lado, se definió la arquitectura del sistema y las conexiones a los ambientes de prueba, por lo que esta iteración tiene la complejidad más alta.

3.3.2.1. Historias de usuario

Esta iteración consta de 8 historias de usuario:

- Mantenimientos de colegios
- Mantenimientos de apoderados
- Mantenimientos de escolares
- Mantenimientos de transportistas
- Mantenimientos de unidades escolares
- Mantenimientos de choferes
- Mantenimientos de rutas

- Mantenimientos de asignaciones

Tabla 5

Historia N° 01: Mantenimiento de colegios

Mantenimiento de colegios	
Número de historia: 01	Usuario: Administrador web
Nombre de historia: Mantenimiento de colegios	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos Estimados:	Iteración Asignada: 01
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: El administrador web ingresará al menú de mantenimiento de colegios, visualizará el listado de colegios, tendrá la opción de registrar, buscar, modificar y ver datos del colegio.	
Observaciones: Ninguna	

Fuente: Elaboración propia

Tabla 6

Tarea N° 01: Diseño de interfaz de mantenimiento de colegios

Diseño de interfaz de mantenimiento de colegios	
Número de tarea: 01	Número de historia: 01
Nombre: Diseño de interfaz de mantenimiento de colegios	
Tipo de tarea: Desarrollo	Puntos estimados:
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: Se diseñará una interfaz de usuario donde se podrá visualizar el listado de colegios, contará con botones básicos para poder realizar el registro, búsqueda, modificación y ver datos del colegio.	

Fuente: Elaboración propia

Tabla 7

Tarea N° 02: Gestión de mantenimiento de colegios

Gestión de mantenimiento de colegios	
Número de tarea: 02	Número de historia: 01
Nombre: Gestión de mantenimiento de colegios	
Tipo de tarea: Desarrollo	Puntos estimados:
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: Al pulsar el botón NUEVO COLEGIO se ejecutará una ventana modal en la que se deberá ingresar los datos básicos para el registro de colegios, al pulsar el botón VER se ejecutará una ventana modal en la que nos permitirá visualizar la información detallada del colegio y al pulsar el botón EDITAR se ejecutará una ventana modal en la que se nos permitirá modificar la información del colegio.	

Fuente: Elaboración propia

Tabla 8

Historia N° 02: Mantenimiento de apoderados

Mantenimiento de apoderados	
Número de historia: 02	Usuario: Colegio
Nombre de historia: Mantenimiento de apoderados	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos Estimados:	Iteración Asignada: 01
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: El usuario colegio ingresará al menú de mantenimiento de apoderados, visualizará el listado de apoderados, tendrá la opción de registrar, buscar, modificar, ver datos del apoderado.	
Observaciones: Ninguna	

Fuente: Elaboración propia

Tabla 9

Tarea N° 03: Diseño de interfaz de mantenimiento de apoderados

Diseño de interfaz de mantenimiento de apoderados	
Número de tarea: 03	Número de historia: 02
Nombre: Diseño de interfaz de mantenimiento de apoderados	
Tipo de tarea: Desarrollo	Puntos estimados:
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: Se diseñará una interfaz de usuario donde se podrá visualizar el listado de apoderados, contará con botones básicos para poder realizar el registro, búsqueda, modificación y ver datos del apoderado, además la interfaz permitirá añadir un escolar al apoderado.	
Fuente: Elaboración propia	

Tabla 10

Tarea N° 04: Gestión de mantenimiento de apoderados

Gestión de mantenimiento de apoderados	
Número de tarea: 04	Número de historia: 02
Nombre: Gestión de mantenimiento de apoderados	
Tipo de tarea: Desarrollo	Puntos estimados:
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: Al pulsar el botón NUEVO APODERADO se ejecutará una ventana modal en la que se deberá ingresar los datos básicos para el registro de apoderados, al pulsar el botón VER se ejecutará una ventana modal en la que nos permitirá visualizar la información detallada del apoderado, al pulsar el botón EDITAR se ejecutará una ventana modal en la que se nos permitirá modificar la información del apoderado y al pulsar el botón AÑADIR ESCOLAR se ejecutará una ventana modal en la que se nos permitirá agregar un escolar al apoderado.	
Fuente: Elaboración propia	

Tabla 11

Historia N° 03: Mantenimiento de escolares

Mantenimiento de escolares	
Número de historia: 03	Usuario: Colegio
Nombre de historia: Mantenimiento de escolares	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos Estimados:	Iteración Asignada: 01
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: El usuario colegio ingresará al menú de mantenimiento de escolares, visualizará el listado de escolares, tendrá la opción de registrar, buscar, modificar y ver datos del escolar.	
Observaciones: Ninguna	
Fuente: Elaboración propia	

Tabla 12

Tarea N° 05: Diseño de interfaz de mantenimiento de escolares

Diseño de interfaz de mantenimiento de escolares	
Número de tarea: 05	Número de historia: 03
Nombre: Diseño de interfaz de mantenimiento de escolares	
Tipo de tarea: Desarrollo	Puntos estimados:
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: Se diseñará una interfaz de usuario donde se podrá visualizar el listado de escolares, contará con botones básicos para poder realizar el registro, búsqueda, modificación y ver datos del escolar.	
Fuente: Elaboración propia	

Tabla 13

Tarea N° 06: Gestión para el mantenimiento de escolares

Gestión para el mantenimiento de escolares	
Número de tarea: 06	Número de historia: 03
Nombre: Gestión para el mantenimiento de escolares	
Tipo de tarea: Desarrollo	Puntos estimados:
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: Al pulsar el botón NUEVO ESCOLAR se ejecutará una ventana modal en la que se deberá ingresar los datos básicos para el registro de escolares, al pulsar el botón VER se ejecutará una ventana modal en la que nos permitirá visualizar la información detallada del escolar y al pulsar el botón EDITAR se ejecutará una ventana modal en la que se nos permitirá modificar la información del escolar.	
Fuente: Elaboración propia	

Tabla 14

Historia N° 04: Mantenimiento de transportistas

Mantenimiento de transportistas	
Número de historia: 04	Usuario: Colegio
Nombre de historia: Mantenimiento de transportistas	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos Estimados:	Iteración Asignada: 01
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: El usuario colegio ingresará al menú de mantenimiento de transportistas, visualizará el listado de transportista, tendrá la opción de registrar, buscar, modificar y ver datos del transportista.	
Observaciones: Ninguna	
Fuente: Elaboración propia	

Tabla 15

Tarea N° 07: Diseño de interfaz de mantenimiento de transportistas

Diseño de interfaz de mantenimiento de transportistas	
Número de tarea: 07	Número de historia: 04
Nombre: Diseño de interfaz de mantenimiento de transportistas	
Tipo de tarea: Desarrollo	Puntos estimados:
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: Se diseñará una interfaz de usuario donde se podrá visualizar el listado de transportistas, contará con botones básicos para poder realizar el registro, búsqueda, modificación y ver datos del transportista, además la interfaz permitirá agregarle una unidad escolar y un chofer al transportista	
Fuente: Elaboración propia	

Tabla 16

Tarea N° 08: Gestión para el mantenimiento de transportistas

Gestión para el mantenimiento de transportistas	
Número de tarea: 08	Número de historia: 04
Nombre: Gestión para el mantenimiento de transportistas	
Tipo de tarea: Desarrollo	Puntos estimados:
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: Al pulsar el botón NUEVO TRANSPORTISTA se ejecutará una ventana modal en la que se deberá ingresar los datos básicos para el registro de transportistas, al pulsar el botón VER se ejecutará una ventana modal en la que nos permitirá visualizar la información detallada del transportista, al pulsar el botón EDITAR se ejecutará una ventana modal en la que se nos permitirá modificar la información del transportista, al pulsar el botón AÑADIR CHOFER se ejecutará una ventana modal en la que se nos permitirá agregar un chofer al transportista y al pulsar el botón AÑADIR UNIDAD ESCOLAR se ejecutará una ventana modal en la que se nos permitirá agregar una unidad escolar al transportista.	
Fuente: Elaboración propia	

Tabla 17

Historia N° 05: Mantenimiento de unidades escolares

Mantenimiento de unidades escolares	
Número de historia: 05	Usuario: Colegio, Transportista
Nombre de historia: Mantenimiento de unidades escolares	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos Estimados:	Iteración Asignada: 01
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: El usuario colegio y usuario transportista ingresarán al menú de mantenimiento de unidades escolares, visualizarán el listado de unidades escolares, tendrán la opción de registrar, buscar, modificar y ver datos de la unidad escolar.	
Observaciones: Si es el colegio el que registra la unidad escolar su evaluación será aprobada. Si es el transportista el que registra la unidad escolar su evaluación será pendiente.	
Fuente: Elaboración propia	

Tabla 18

Tarea N° 09: Diseño de interfaz de mantenimiento de unidades escolares

Diseño de interfaz de mantenimiento de unidades escolares	
Número de tarea: 09	Número de historia: 05
Nombre: Diseño de interfaz de mantenimiento de unidades escolares	
Tipo de tarea: Desarrollo	Puntos estimados:
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: Se diseñará una interfaz de usuario donde se podrá visualizar el listado de unidades escolares, contará con botones básicos para poder realizar el registro, búsqueda, modificación y ver datos de las unidades escolares.	
Fuente: Elaboración propia	

Tabla 19

Tarea N° 10: Gestión para el mantenimiento de unidades escolares

Gestión para el mantenimiento de unidades escolares	
Número de tarea: 10	Número de historia: 05
Nombre: Gestión para el mantenimiento de unidades escolares	
Tipo de tarea: Desarrollo	Puntos estimados:
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: Al pulsar el botón NUEVA UNIDAD ESCOLAR se ejecutará una ventana modal en la que se deberá ingresar los datos básicos para el registro de la unidad escolar, al pulsar el botón VER se ejecutará una ventana modal en la que nos permitirá visualizar la información detallada de la unidad escolar y al pulsar el botón EDITAR se ejecutará una ventana modal en la que se nos permitirá modificar la información de la unidad escolar.	
Fuente: Elaboración propia	

Tabla 20

Historia N° 06: Mantenimiento de choferes

Mantenimiento de choferes	
Número de historia: 06	Usuario: Colegio, Transportista
Nombre de historia: Mantenimiento de choferes	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos Estimados:	Iteración Asignada: 01
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: El usuario colegio y usuario transportista ingresarán al menú de mantenimiento de unidades escolares, visualizará el listado de unidades escolares, tendrá la opción de registrar, buscar, modificar y ver datos de la unidad escolar.	
Observaciones: Si es el colegio el que registra al chofer su evaluación será aprobada. Si es el transportista el que registra al chofer su evaluación será pendiente.	
Fuente: Elaboración propia	

Tabla 21

Tarea N° 11: Diseño de interfaz de mantenimiento de choferes

Diseño de interfaz de mantenimiento de choferes	
Número de tarea: 11	Número de historia: 06
Nombre: Diseño de interfaz de mantenimiento de choferes	
Tipo de tarea: Desarrollo	Puntos estimados:
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: Se diseñará una interfaz de usuario donde se podrá visualizar el listado de choferes, contará con botones básicos para poder realizar el registro, búsqueda, modificación y ver datos del chofer.	
Fuente: Elaboración propia	

Tabla 22

Tarea N° 12: Gestión para el mantenimiento de choferes

Gestión para el mantenimiento de choferes	
Número de tarea: 12	Número de historia: 06
Nombre: Gestión para el mantenimiento de choferes	
Tipo de tarea: Desarrollo	Puntos estimados:
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: Al pulsar el botón NUEVO CHOFER se ejecutará una ventana modal en la que se deberá ingresar los datos básicos para el registro del chofer, al pulsar el botón VER se ejecutará una ventana modal en la que nos permitirá visualizar la información detallada del chofer y al pulsar el botón EDITAR se ejecutará una ventana modal en la que se nos permitirá modificar la información del chofer.	
Fuente: Elaboración propia	

Tabla 23

Historia N° 07: Mantenimiento de rutas

Mantenimiento de Rutas	
Número de historia: 07	Usuario: Transportista
Nombre de historia: Mantenimiento de rutas	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos Estimados:	Iteración Asignada: 01
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: El usuario transportista ingresará al menú de mantenimiento de rutas, visualizará el listado de rutas, tendrá la opción de registrar, buscar, modificar y ver datos de la ruta.	
Observaciones: Ninguna	
Fuente: Elaboración propia	

Tabla 24

Tarea N° 13: Diseño de interfaz de mantenimiento de rutas

Diseño de interfaz de mantenimiento de rutas	
Número de tarea: 13	Número de historia: 07
Nombre: Diseño de interfaz de mantenimiento de rutas	
Tipo de tarea: Desarrollo	Puntos estimados:
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: Se diseñará una interfaz de usuario donde se podrá visualizar el listado de rutas, contará con botones básicos para poder realizar el registro, búsqueda, modificación y ver datos de la ruta.	
Fuente: Elaboración propia	

Tabla 25

Tarea N° 14: Gestión para el mantenimiento de rutas

Gestión para el mantenimiento de rutas	
Número de tarea: 14	Número de historia: 07
Nombre: Gestión para el mantenimiento de rutas	
Tipo de tarea: Desarrollo	Puntos estimados:
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: Al pulsar el botón NUEVA RUTA se ejecutará una ventana modal en la que se deberá ingresar los datos básicos para el registro de la ruta, al pulsar el botón VER se ejecutará una ventana modal en la que nos permitirá visualizar la información detallada de la ruta y al pulsar el botón EDITAR se ejecutará una ventana modal en la que se nos permitirá modificar la información de la ruta.	
Fuente: Elaboración propia	

Tabla 26

Historia N° 08: Mantenimiento de asignaciones

Mantenimiento de Asignaciones	
Número de historia: 08	Usuario: Transportista
Nombre de historia: Mantenimiento de asignaciones	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos Estimados:	Iteración Asignada: 01
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: El usuario transportista ingresará al menú de mantenimiento de asignaciones, visualizará el listado de asignaciones, tendrá la opción de registrar, buscar, modificar y ver datos de la asignación.	
Observaciones: Ninguna	
Fuente: Elaboración propia	

Tabla 27

Tarea N° 15: Diseño de interfaz de mantenimiento de asignaciones

Diseño de interfaz de mantenimiento de asignaciones	
Número de tarea: 15	Número de historia: 08
Nombre: Diseño de interfaz de mantenimiento de asignaciones	
Tipo de tarea: Desarrollo	Puntos estimados:
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: Se diseñará una interfaz de usuario donde se podrá visualizar el listado de asignaciones, contará con botones básicos para poder realizar el registro, búsqueda, modificación y ver datos de las asignaciones.	
Fuente: Elaboración propia	

Tabla 28

Tarea N° 16: Gestión para el mantenimiento de asignaciones

Gestión para el mantenimiento de asignaciones	
Número de tarea: 16	Número de historia: 08
Nombre: Gestión para el mantenimiento de asignaciones	
Tipo de tarea: Desarrollo	Puntos estimados:
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: Al pulsar el botón NUEVA ASIGNACIÓN se ejecutará una ventana modal en la que se deberá ingresar los datos básicos para el registro de la asignación, al pulsar el botón VER se ejecutará una ventana modal en la que nos permitirá visualizar la información detallada de la asignación y al pulsar el botón EDITAR se ejecutará una ventana modal en la que se nos permitirá modificar la información de la asignación.	
Fuente: Elaboración propia	

3.3.2.2. Diseños de interfaces

admin

admin

INTERFACES

Mantenimiento

COLEGIOS

Colegio

Monitoreo

Reportes

Configuración

NUEVO COLEGIO

Mostrar 5 registros

Buscar:

N°	Nombre	Director	Email	Pagina Web	Opción
1	I.E. Nuestra Señora del Rosario	Sosa Lucia Rabanal Cruz	iensrosario@yahoo.com	https://iensrosario.edu.pe/web/	 

Mostrando registros del 1 al 1 de un total de 1 registros

Anterior 1 Siguiente

Todos los derechos reservados © 2019 MOVIESCOOL

https://startecperu.com/rest/moviescool/colegio

Figura 20. Módulo administrador: Mantenimiento de colegios

Fuente: Elaboración propia

I.E. Nuestra Señora del Rosario

110454543

INTERFACES

Mantenimiento

APODERADOS

ApoDERado

Escolar

TRANSPORTISTAS

Transportista

Unidad Escolar

Chofer

Monitoreo

Evaluación

Reportes

Configuración

NUEVO APODERADO

Mostrar 5 registros

Buscar:

N°	APODERADO	TIPO DOC.	N° DOCUMENTO	SEXO	OPCION
1	Jhordy Akito Supo Oliden	L.E/D.N.I	75250450	Hombre	 
2	Fiorela Katherine Acha Nuñez	L.E/D.N.I	12345678	Mujer	 
3	Aida Carmela Nuñez Chuzón	L.E/D.N.I	16472471	Mujer	 
4	José Ricardo Díaz Hernandez	L.E/D.N.I	16411517	Hombre	 
5	Bernardo Amado Bonilla Coronado	L.E/D.N.I	71874291	Hombre	 

Mostrando registros del 1 al 5 de un total de 5 registros

Anterior 1 Siguiente

Todos los derechos reservados © 2019 MOVIESCOOL

https://startecperu.com/rest/moviescool/apoderado

Figura 21. Módulo colegio: Mantenimiento de apoderados

Fuente: Elaboración propia

I.E Nuestra Señora del Rosario
110454543 

INTERFACES

- Mantenimiento ▾
- APODERADOS
 - ApoDERado**
 - Escolar
 - TRANSPORTISTAS
 - Transportista
 - Unidad Escolar
 - Chofer
- Monitoreo ▸
- Evaluación ▸
- Reportes ▸
- Configuración ▸

LISTADO DE ESCOLARES

NUEVO ESCOLAR

Mostrar registros

Buscar:

N°	TIPO DOC	N° DOCUMENTO	ESCOLAR	FECHA NAC.	APODERADO	OPCION
1	L.E/D.N.I	84235671	Silvana del Pilar Supo Ramirez	19/05/2009	Jhordy Akito Supo Oliden	 
2	L.E/D.N.I	85236741	Valentina Rivera Acha	27/10/2009	Fiorela Katherine Acha Nuñez	 
3	L.E/D.N.I	14725836	Melannie Alanna Reyes Nuñez	16/12/2008	Aida Carmela Nuñez Chuzón	 
4	L.E/D.N.I	87654321	Ayzza Estrella Díaz Puescas	28/01/2008	José Ricardo Díaz Hernandez	 
5	L.E/D.N.I	12345678	Larissa Bonilla Lopez	10/01/2008	Bernardo Amado Bonilla Coronado	 

Mostrando registros del 1 al 5 de un total de 5 registros

Anterior

1

Siguiente

Todos los derechos reservados © 2019 MOVIESCOOL

Figura 22. Módulo colegio: Mantenimiento de escolares

Fuente: Elaboración propia

moviescool
I.E Nuestra Señora del Rosario
110454543

INTERFACES

- Mantenimiento
- APODERADOS
 - Apoderado
 - Escolar
- TRANSPORTISTAS
 - Transportista**
 - Unidad Escolar
 - Chofer
- Monitoreo
- Evaluación
- Reportes
- Configuración

LISTADO DE TRANSPORTISTAS

Mostrar registros
Buscar:

Nº	RUC/DNI	RAZON SOC/NOMBRE	REPRESENTANTE	CELULAR	EMAIL	OPCION
1	10123456785	Bustamante Núñez Yasser	Bustamante Núñez Yasser	912345678	yasser.bunu@gmail.com	

Mostrando registros del 1 al 1 de un total de 1 registros

[Anterior](#)
[1](#)
[Siguiente](#)

Todos los derechos reservados © 2019 MOVIESCOOL

Figura 23. Módulo colegio: Mantenimiento de transportistas

Fuente: Elaboración propia

movi
esc
I.E Nuestra Señora del Rosario
110454543

INTERFACES

- Mantenimiento
- APODERADOS
- ApoDERado
- Escolar
- TRANSPORTISTAS
- Transportista
- Unidad Escolar**
- Chofer
- Monitoreo
- Evaluación
- Reportes
- Configuración

LISTADO DE UNIDADES ESCOLARES

NUEVA UNIDAD ESCOLAR

Buscar:

Mostrar registros

Nº	PLACA	Nº SOAT	REV. SOAT	CERT. TEC.	Nº ASIENTOS	ESTADO	TRANSP.	OPCION
1	A3K315	0020	12/12/2020	0015	10	Aprobado	Bustamante Núñez Yasser	
2	A4X5FX	C00212	13/12/2021	C0000ASA33	10	Aprobado	Bustamante Núñez Yasser	

Mostrando registros del 1 al 2 de un total de 2 registros

[Anterior](#)
1
[Siguiente](#)

Todos los derechos reservados © 2019 MOVIESCOOL

Figura 24. Módulo colegio: Mantenimiento de unidades escolares

Fuente: Elaboración propia

movi
esc
I.E Nuestra Señora del Rosario
ID0454543 

INTERFACES

LISTADO DE CHOFERES

+ NUEVO CHOFER

Mostrar

5

 registros

Buscar:

N°	DNI	CHOFER	N° LICENCIA	CATEGORIA	ESTADO	TRANSP.	OPCION
1	12345679	Sergio Bonilla Nuñez	C12345679	AIIA	Aprobado	Bustamante Núñez Yasser	 
2	87654321	Erick Rubio Díaz	C87654321	AIIA	Aprobado	Bustamante Núñez Yasser	 

Mostrando registros del 1 al 2 de un total de 2 registros

Anterior
1
Siguiente

APODERADOS
 Apoderado
 Escolar
 TRANSPORTISTA:
 Transportista
 Unidad Escolar
Chofer

Monitoreo
 Evaluación
 Reportes
 Configuración

Todos los derechos reservados © 2019 MOVIESCOOL

Figura 25. Módulo colegio: Mantenimiento de choferes

Fuente: Elaboración propia

movi
escool
I.E Nuestra Señora del Rosario
T110123456785

INTERFACES

- Mantenimiento
- TRANSPORTISTA
 - Unidades Escolares
 - Chofer
 - Ruta
 - Asignación
- Monitoreo
- Contratos
- Ordenar Horario
- Configuración

LISTADO DE UNIDADES ESCOLARES

Mostrar 5 registros

Buscar:

N°	PLACA	N° SOAT	REV. SOAT	CERT. TEC.	N° ASIENTOS	ESTADO	OPCION
1	A3K315	0020	12/12/2020	0015	10	Aprobado	
2	A4X5FX	C00212	13/12/2021	C0000ASA33	10	Aprobado	

Mostrando registros del 1 al 2 de un total de 2 registros

[Anterior](#)
1
[Siguiente](#)

https://starttecperu.com/rest/moviescool/unidadesescolar

Figura 26. Módulo transportista: Mantenimiento de unidades escolares

Fuente: Elaboración propia

I.E Nuestra Señora del Rosario
T110123456785 

moviescool

INTERFACES

- Mantenimiento
- TRANSPORTISTA
- Unidades Escolares
- Chofer
- Ruta
- Asignación
- Monitoreo
- Contratos
- Ordenar Horario
- Configuración

LISTADO DE CHOFERES

Mostrar registros

Buscar:

N°	DNI	CHOFER	N° LICENCIA	CATEGORIA	ESTADO	OPCION
1	12345679	Sergio Bonilla Nuñez	C12345679	AIIA	Aprobado	 
2	87654321	Erick Rubio Díaz	C87654321	AIIA	Aprobado	 

Mostrando registros del 1 al 2 de un total de 2 registros

[Anterior](#)
[1](#)
[Siguiente](#)

Figura 27. Módulo transportista: Mantenimiento de choferes

Fuente: Elaboración propia

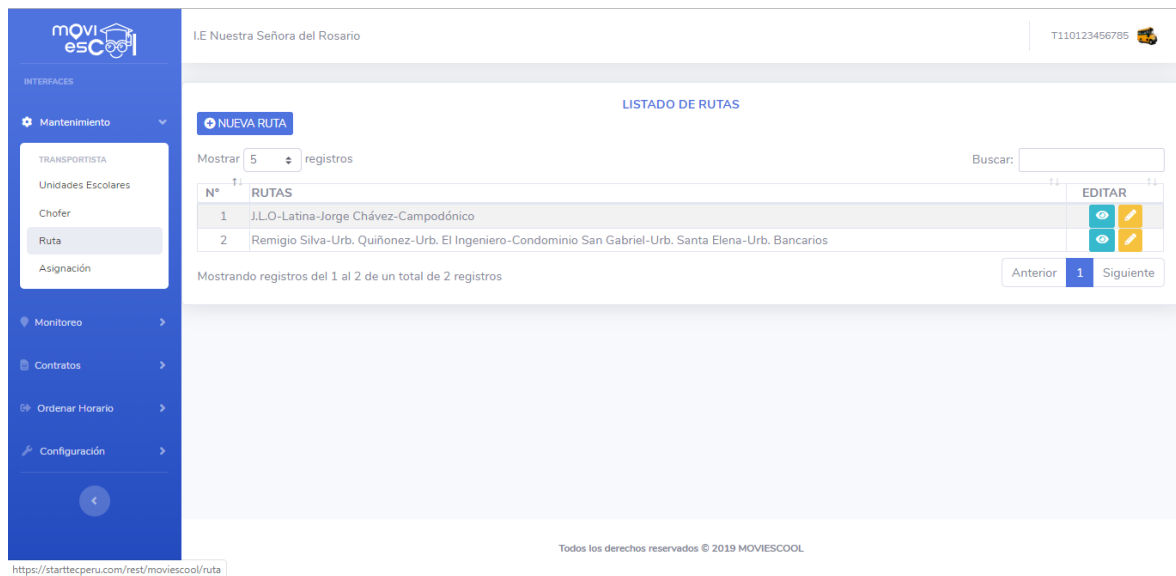


Figura 28. Módulo transportista: Mantenimiento de rutas

Fuente: Elaboración propia

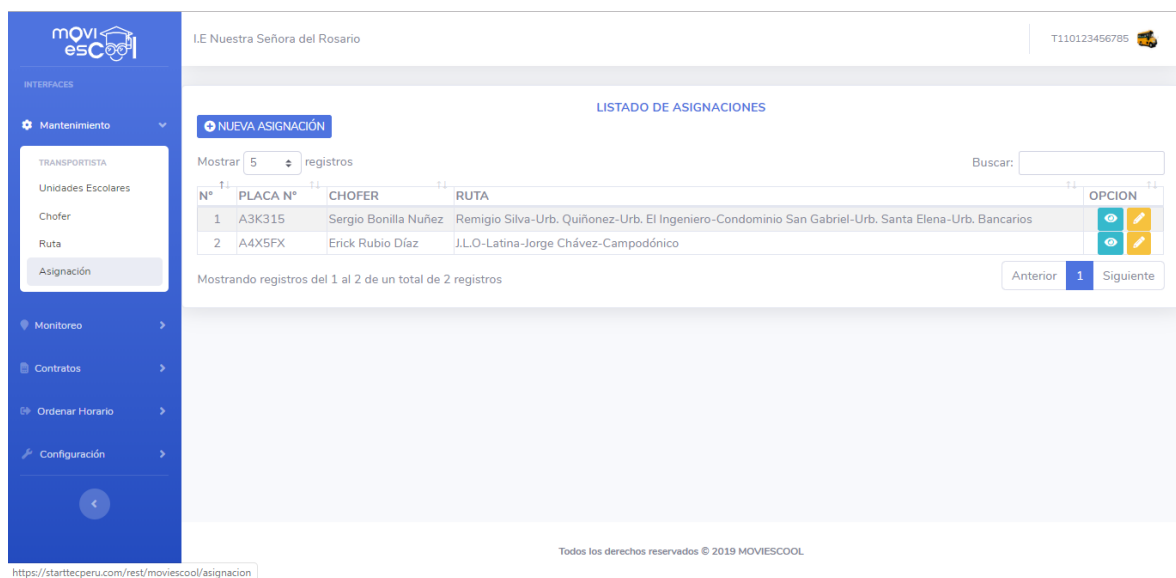


Figura 29. Módulo transportista: Mantenimiento de asignaciones

Fuente: Elaboración propia

3.3.2.3. Diagrama de base de datos

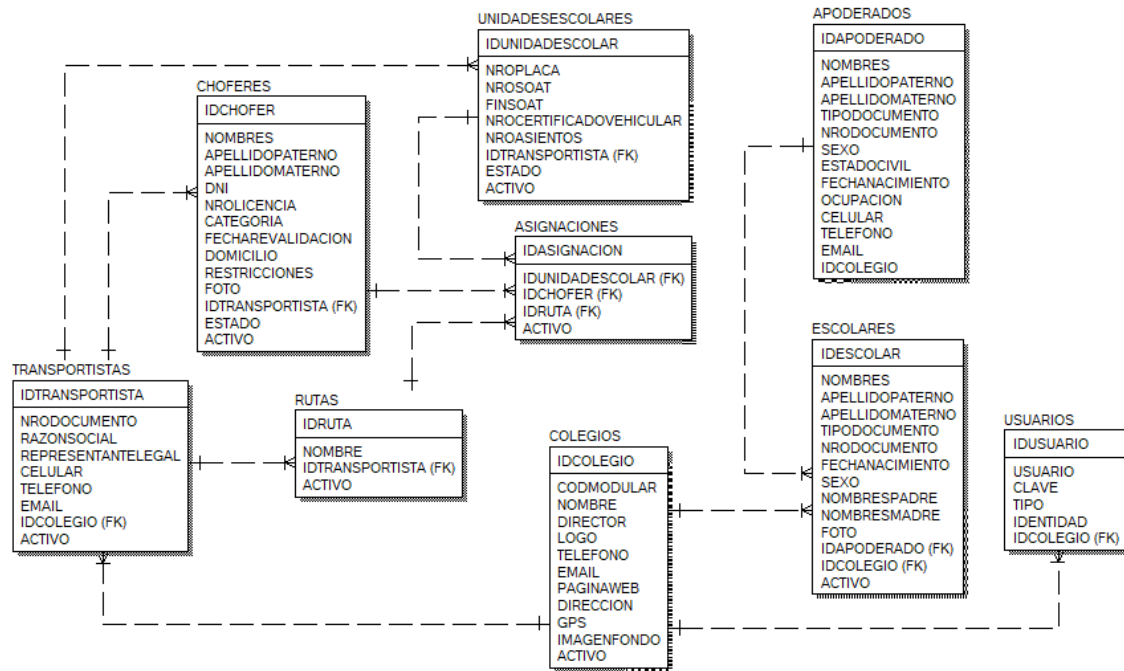


Figura 30. Diagrama de base de datos: Iteración 1

Fuente: Elaboración propia

3.3.3. Iteración 2

En la iteración 2 se pretende entregar parte del sistema web con las funcionalidades relacionadas a la evaluación de conformidad de las unidades escolares y de choferes por parte del colegio, generación de contratos entre apoderados y transportistas, ordenar las direcciones de domicilios de los escolares para el horario de ingreso y salida escolar, monitorear a las unidades escolares desde el sistema web y del ingreso de los usuarios colegio y transportistas al sistema web.

3.3.3.1. Historias de usuario

Esta iteración consta de 7 historias de usuario:

- Evaluar conformidad de unidades escolares
- Evaluar conformidad de choferes
- Generación de contratos
- Ordenar domicilios para el horario de ingreso escolar

- Ordenar domicilios para el horario de salida escolar
- Monitorear unidades escolares desde el sistema web
- Ingresar al sistema web

Tabla 29

Historia N° 09: Evaluar conformidad de unidades escolares

Evaluar conformidad de unidades escolares	
Número de historia: 09	Usuario: Colegio
Nombre de historia: Evaluar conformidad de unidades escolares	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos Estimados:	Iteración Asignada: 02
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: El usuario colegio ingresará al menú de evaluación de unidad escolar, visualizará el listado de unidades escolares que se encuentran pendiente de evaluación y podrá evaluar a la unidad escolar.	
Observaciones: El usuario colegio verificará los requisitos mínimos para que la evaluación de la unidad escolar pueda ser aprobado o desaprobado.	

Fuente: Elaboración propia

Tabla 30

Tarea N° 17: Diseño de interfaz de evaluación de conformidad de unidades escolares

Diseño de interfaz de evaluación de conformidad de unidades escolares	
Número de tarea: 17	Número de historia: 09
Nombre: Diseño de interfaz de evaluación de conformidad de unidades escolares	
Tipo de tarea: Desarrollo	Puntos estimados:
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: Se diseñará una interfaz de usuario donde se podrá visualizar el listado de unidades escolares pendiente de evaluación, contará con un botón para poder realizar la evaluación a la unidad escolar.	
Fuente: Elaboración propia	

Tabla 31

Tarea N° 18: Gestión de evaluación de conformidad de unidades escolares

Gestión de evaluación de conformidad de unidades escolares	
Número de tarea: 18	Número de historia: 09
Nombre: Gestión de evaluación de conformidad de unidades escolares	
Tipo de tarea: Desarrollo	Puntos estimados:
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: Al pulsar el botón EVALUAR se ejecutará una ventana modal en la que se deberá evaluar a la unidad escolar como aprobado o desaprobado.	
Fuente: Elaboración propia	

Tabla 32

Historia N° 10: Evaluar conformidad de choferes

Evaluar conformidad de choferes	
Número de historia: 10	Usuario: Colegio
Nombre de historia: Evaluar conformidad de choferes	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos Estimados:	Iteración Asignada: 02
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: El usuario colegio ingresará al menú de evaluación de chofer, visualizará el listado de choferes que se encuentran pendiente de evaluación y podrá evaluar al chofer.	
Observaciones: El usuario colegio verificará los requisitos mínimos para que la evaluación del chofer pueda ser aprobada o desaprobada.	
Fuente: Elaboración propia	

Tabla 33

Tarea N° 19: Diseño de interfaz de evaluación de conformidad de choferes

Diseño de interfaz de evaluación de conformidad de choferes	
Número de tarea: 19	Número de historia: 10
Nombre: Diseño de interfaz de evaluación de conformidad de choferes	
Tipo de tarea: Desarrollo	Puntos estimados:
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: Se diseñará una interfaz de usuario donde se podrá visualizar el listado de choferes pendiente de evaluación, contará con un botón para poder realizar la evaluación al chofer.	
Fuente: Elaboración propia	

Tabla 34

Tarea N° 20: Gestión de evaluación de conformidad de unidades escolares

Gestión de evaluación de conformidad de unidades escolares	
Número de tarea: 20	Número de historia: 10
Nombre: Gestión de evaluación de conformidad de unidades escolares	
Tipo de tarea: Desarrollo	Puntos estimados:
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: Al pulsar el botón EVALUAR se ejecutará una ventana modal en la que se deberá evaluar a la unidad escolar como aprobado o desaprobado.	
Fuente: Elaboración propia	

Tabla 35

Historia N° 11: Generación de contratos

Generación de contratos	
Número de historia: 11	Usuario: Transportista
Nombre de historia: Generación de contratos	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos Estimados:	Iteración Asignada: 02
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: El usuario transportista ingresará al menú de generación de contratos, visualizará el listado de contratos, tendrá la opción de registrar, buscar, modificar y ver datos del contrato.	
Observaciones: Ninguna	
Fuente: Elaboración propia	

Tabla 36

Tarea N° 21: Diseño de interfaz de generación de contratos

Diseño de interfaz de generación de contratos	
Número de tarea: 21	Número de historia: 11
Nombre: Diseño de interfaz de generación de contratos	
Tipo de tarea: Desarrollo	Puntos estimados:
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: Se diseñará una interfaz de usuario donde se podrá visualizar el listado de contratos, contará con botones básicos para poder realizar el registro, búsqueda, modificación y ver datos del contrato.	
Fuente: Elaboración propia	

Tabla 37

Tarea N° 22: Gestión de generación de contratos

Gestión de generación de contratos	
Número de tarea: 22	Número de historia: 11
Nombre: Gestión de generación de contratos	
Tipo de tarea: Desarrollo	Puntos estimados:
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: Al pulsar el botón NUEVO CONTRATO se ejecutará una ventana modal en la que se deberá ingresar los datos básicos para el registro de contratos, al pulsar el botón VER se ejecutará una ventana modal en la que nos permitirá visualizar la información detallada del contrato y al pulsar el botón EDITAR se ejecutará una ventana modal en la que se nos permitirá modificar la información del contrato.	
Fuente: Elaboración propia	

Tabla 38

Historia N° 12: Ordenar domicilios para el horario de ingreso escolar

Ordenar domicilios para el horario de ingreso escolar	
Número de historia: 12	Usuario: Transportista
Nombre de historia: Ordenar domicilios para el horario de ingreso escolar	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos Estimados:	Iteración Asignada: 02
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: El usuario transportista ingresará al menú de ordenar horario de ingreso escolar, visualizará el listado de choferes y ordenará el listado de escolares que recogerá a sus domicilios para el traslado al colegio.	
Observaciones: Ninguna	
Fuente: Elaboración propia	

Tabla 39

Tarea N° 23: Diseño de interfaz de ordenar domicilios para el horario de ingreso escolar

Diseño de interfaz de ordenar domicilios para el horario de ingreso escolar	
Número de tarea: 23	Número de historia: 12
Nombre: Diseño de interfaz de ordenar domicilios para el horario de ingreso escolar	
Tipo de tarea: Desarrollo	Puntos estimados:
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: Se diseñará una interfaz de usuario donde se podrá visualizar el listado de choferes, contará con un botón para poder realizar el ordenamiento de escolares que se recogerán en sus domicilios para el traslado al colegio.	
Fuente: Elaboración propia	

Tabla 40

Tarea N° 24: Gestión de ordenar domicilios para el horario de ingreso escolar

Gestión de ordenar domicilios para el horario de ingreso escolar	
Número de tarea: 24	Número de historia: 12
Nombre: Gestión de ordenar domicilios para el horario de ingreso escolar	
Tipo de tarea: Desarrollo	Puntos estimados:
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: Al pulsar el botón ORDENAR INGRESO se ejecutará una ventana modal en la que se visualizará el listado de escolares que han contratado el servicio con el transportista, el listado actual que se visualizará; se podrá arrastrar y pasar los ítems a otro listado para ordenar los escolares que se recogerán en sus domicilios para el traslado al colegio.	
Fuente: Elaboración propia	

Tabla 41

Historia N° 13: Ordenar domicilios para el horario de salida escolar

Ordenar domicilios para el horario de salida escolar	
Número de historia: 13	Usuario: Transportista
Nombre de historia: Ordenar domicilios para el horario de salida escolar	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos Estimados:	Iteración Asignada: 02
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: El usuario transportista ingresará al menú de ordenar horario de salida escolar, visualizará el listado de choferes y ordenará el listado de escolares que recogerá en el colegio para el traslado a sus domicilios.	
Observaciones: Ninguna	
Fuente: Elaboración propia	

Tabla 42

Tarea N° 25: Diseño de interfaz de ordenar domicilios para el horario de salida escolar

Diseño de interfaz de ordenar domicilios para el horario de salida escolar	
Número de tarea: 25	Número de historia: 13
Nombre: Diseño de interfaz de ordenar domicilios para el horario de salida escolar	
Tipo de tarea: Desarrollo	Puntos estimados:
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: Se diseñará una interfaz de usuario donde se podrá visualizar el listado de choferes, contará con un botón para poder realizar el ordenamiento de escolares que se recogerán en el colegio para el traslado a sus domicilios.	
Fuente: Elaboración propia	

Tabla 43

Tarea N° 26: Gestión de ordenar domicilios para el horario de salida escolar

Gestión de ordenar domicilios para el horario de salida escolar	
Número de tarea: 26	Número de historia: 13
Nombre: Gestión de ordenar domicilios para el horario de salida escolar	
Tipo de tarea: Desarrollo	Puntos estimados:
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: Al pulsar el botón ORDENAR SALIDA se ejecutará una ventana modal en la que se visualizará el listado de escolares que han contratado el servicio con el transportista, el listado actual que se visualizará; se podrá arrastrar y pasar los ítems a otro listado para ordenar los escolares que se recogerán en el colegio para el traslado a sus domicilios.	
Fuente: Elaboración propia	

Tabla 44

Historia N° 14: Monitorear unidades escolares desde el sistema web

Monitorear unidades escolares desde el sistema web			
Número de historia: 14	Usuario:	Administrador	Web, Colegio, Transportista
Nombre de historia: Monitorear unidades escolares desde el sistema web			
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta		
Puntos Estimados:	Iteración Asignada: 02		
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez			
Descripción: El administrador web, el usuario colegio y usuario transportista ingresará al menú de monitoreo, visualizarán en tiempo real las unidades escolares que transportarán a los escolares de casa al colegio y/o del colegio a casa.			
Observaciones: Ninguna			
Fuente: Elaboración propia			

Tabla 45

Tarea N° 27: Diseño de interfaz de monitorear unidades escolares desde el sistema web

Diseño de interfaz de monitorear unidades escolares desde el sistema web	
Número de tarea: 27	Número de historia: 14
Nombre: Diseño de interfaz de monitorear unidades escolares desde el sistema web	
Tipo de tarea: Desarrollo	Puntos estimados:
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: Se diseñará una interfaz de usuario donde se podrá visualizar en tiempo real las unidades escolares que transportarán a los escolares de casa al colegio y/o del colegio a casa.	
Fuente: Elaboración propia	

Tabla 46

Tarea N° 28: Visualización de monitoreo de unidades escolares desde el sistema web

Visualización de monitoreo de unidades escolares desde el sistema web	
Número de tarea: 28	Número de historia: 14
Nombre: Visualización de monitoreo de unidades escolares desde el sistema web	
Tipo de tarea: Desarrollo	Puntos estimados:
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: Al ingresar al menú monitoreo, se cargará el mapa que nos proporciona la API de Google Maps, a través de Firebase Realtime Database se visualizarán las unidades escolares que transportarán a los escolares de casa al colegio y/o del colegio a casa.	
Fuente: Elaboración propia	

Tabla 47

Historia N° 15: Ingresar al sistema web

Ingresar al sistema web	
Número de historia: 15	Usuario: Administrador Web, Colegio, Transportista
Nombre de historia: Ingresar al sistema web	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos Estimados:	Iteración Asignada: 02
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: El administrador web, usuario colegio y usuario transportista podrán ingresar al sistema web, tendrán acceso a distintas opciones que le corresponden a cada usuario.	
Observaciones: Ninguna	
Fuente: Elaboración propia	

Tabla 48

Tarea N° 29: Diseño de interfaz de ingresar al sistema web

Diseño de interfaz de ingresar al sistema web	
Número de tarea: 29	Número de historia: 15
Nombre: Diseño de interfaz de ingresar al sistema web	
Tipo de tarea: Desarrollo	Puntos estimados:
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: Se desarrollará una interfaz con los controles proporcionados por el usuario y se diseñará el acceso al sistema conforme a los requerimientos.	
Fuente: Elaboración propia	

Tabla 49

Tarea N° 30: Verificar autenticación de usuario para ingresar al sistema web

Verificar autenticación de usuario para ingresar al sistema web	
Número de tarea: 30	Número de historia: 15
Nombre: Verificar autenticación de usuario para ingresar al sistema web	
Tipo de tarea: Desarrollo	Puntos estimados:
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: Al pulsar el botón INGRESAR, se verificará las credenciales del usuario que ingresará al aplicativo web.	
Fuente: Elaboración propia	

3.3.3.2. Diseño de interfaces

MOVIESCOOL

I.E. Nuestra Señora del Rosario

110454543

INTERFACES

Mantenimiento

Monitoreo

Evaluación

TRANSPORTISTA

Unidad Escolar

Chofer

Reportes

Configuración

Mostrar 5 registros

Buscar:

N°	PLACA	N° SOAT	REV. SOAT	CERT. TEC.	N° ASIENTOS	TRANSP.	ESTADO
1	A3K315	0020	12/12/2020	0015	10	Bustamante Núñez Yasser	
2	A4X5FX	C00212	13/12/2021	C0000ASA33	10	Bustamante Núñez Yasser	

Mostrando registros del 1 al 2 de un total de 2 registros

Anterior 1 Siguiente

Todos los derechos reservados © 2019 MOVIESCOOL

<https://startecperu.com/rest/moviescool/evaluarunidad>

Figura 31. Módulo colegio: Interfaz de evaluar conformidad de unidades escolares

Fuente: Elaboración propia

MOVIESCOOL

I.E. Nuestra Señora del Rosario

110454543

INTERFACES

Mantenimiento

Monitoreo

Evaluación

TRANSPORTISTA

Unidad Escolar

Chofer

Reportes

Configuración

Mostrar 5 registros

Buscar:

N°	DNI	CHOFER	N° LICENCIA	CATEGORIA	TRANSP.	ESTADO
1	12345679	Sergio Bonilla Nuñez	C12345679	AIIA	Bustamante Núñez Yasser	
2	87654321	Erick Rubio Díaz	C87654321	AIIA	Bustamante Núñez Yasser	

Mostrando registros del 1 al 2 de un total de 2 registros

Anterior 1 Siguiente

Todos los derechos reservados © 2019 MOVIESCOOL

<https://startecperu.com/rest/moviescool/evaluar>

Figura 32. Módulo colegio: Interfaz de evaluar conformidad de choferes

Fuente: Elaboración propia

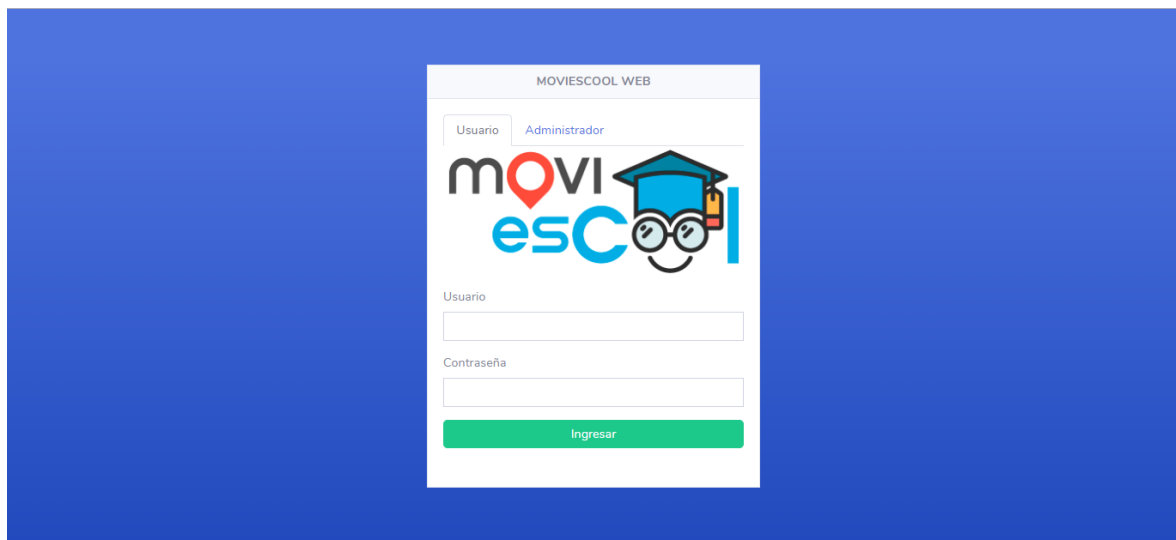


Figura 37. Módulo inicio de sesión: Interfaz de ingresar al sistema web

Fuente: Elaboración propia

3.3.3.3. Diagrama de base de datos

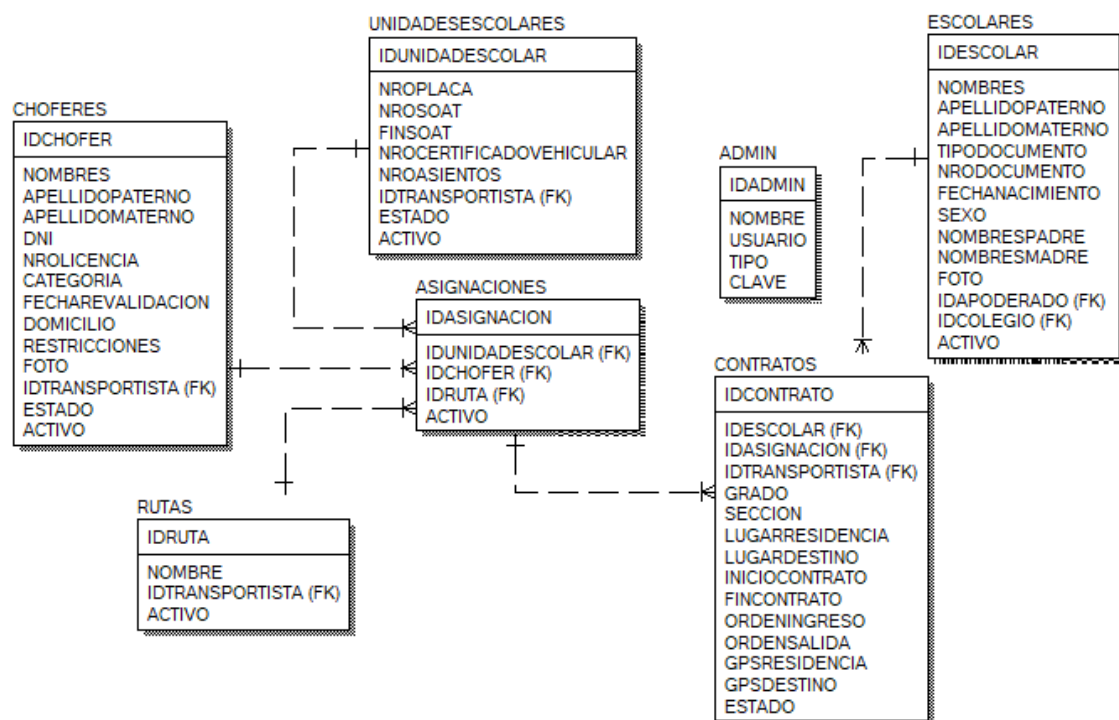


Figura 38. Diagrama de base de datos: Iteración 2

Fuente: Elaboración propia

3.3.4. Iteración 3

En la iteración 3 se pretende iniciar el desarrollo del aplicativo móvil con las funcionalidades relacionadas al ingreso de los usuarios apoderado y chofer al aplicativo

móvil, se implementó una funcionalidad para que los apoderados; cuando ingresen al aplicativo móvil por primera vez; puedan configurar la dirección del domicilio de recogida y llevada; de acuerdo al contrato que estableció con el transportista; para que así el chofer pueda recoger y llevar a su hijo al destino indicado, además el apoderado podrá registrar las inasistencias de su hijo cuando este no pueda ir al colegio, podrá visualizar el historial de recorridos realizando una búsqueda por fecha o rango de fechas, además los usuarios apoderado y transportista podrán resetear las contraseñas que por defecto se le estableció en el sistema web, el usuario chofer podrá crear recorridos tanto para el horario de ingreso (turno mañana o tarde) y salida escolar (turno mañana o tarde), notificará las incidencias que puedan surgir en el trayecto, además el usuario chofer podrá marcar la recogida del escolar cuando a este se le recogió en su domicilio.

3.3.4.1. Historias de usuario

Esta iteración consta de 8 historias de usuario:

- Ingresar al aplicativo móvil
- Configurar domicilio
- Registrar inasistencia
- Historial de recorridos
- Resetear contraseñas desde el aplicativo móvil
- Crear recorrido
- Notificar incidencias
- Marcar recogida

Tabla 50

Historia N° 16: Ingresar al aplicativo móvil

Ingresar al aplicativo móvil	
Número de historia: 16	Usuario: Apoderado, Chofer
Nombre de historia: Ingresar al aplicativo móvil	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos Estimados:	Iteración Asignada: 03
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: El usuario apoderado y usuario chofer podrán ingresar al aplicativo móvil, cada usuario tendrá acceso a distintas opciones.	
Observaciones: Ninguna	
Fuente: Elaboración propia	

Tabla 51

Tarea N° 31: Diseño de interfaz de ingresar al aplicativo móvil

Diseño de interfaz de ingresar al aplicativo móvil	
Número de tarea: 31	Número de historia: 16
Nombre: Diseño de interfaz de ingresar al aplicativo móvil	
Tipo de tarea: Desarrollo	Puntos estimados:
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: Se desarrollará dos interfaces, la primera interfaz con los controles adecuados para que el usuario pueda acceder al aplicativo móvil, la segunda interfaz; si el usuario autenticó los datos correctos mostrará un mensaje de bienvenida con los datos del apoderado o chofer.	
Fuente: Elaboración propia	

Tabla 52

Tarea N° 32: Verificar autenticación de usuario para ingresar al aplicativo móvil

Verificar autenticación de usuario para ingresar al aplicativo móvil	
Número de tarea: 32	Número de historia: 16
Nombre: Verificar autenticación de usuario para ingresar al aplicativo móvil	
Tipo de tarea: Desarrollo	Puntos estimados:
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: Al pulsar el botón INGRESAR, se verificará las credenciales del usuario que ingresará al aplicativo móvil.	
Fuente: Elaboración propia	

Tabla 53

Historia N° 17: Configurar domicilio

Configurar domicilio	
Número de historia: 17	Usuario: Apoderado
Nombre de historia: Configurar domicilio	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos Estimados:	Iteración Asignada: 03
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: El usuario apoderado cuando ingrese al aplicativo móvil, podrá configurar la dirección de su domicilio.	
Observaciones: Esta configuración se realizará la primera vez que el usuario apoderado ingrese al aplicativo móvil; activará el GPS del dispositivo móvil para proporcionar los datos necesarios.	
Fuente: Elaboración propia	

Tabla 54

Tarea N° 33: Diseño de interfaz de configurar domicilio

Diseño de interfaz de configurar domicilio	
Número de tarea: 33	Número de historia: 17
Nombre: Diseño de interfaz de configurar domicilio	
Tipo de tarea: Desarrollo	Puntos estimados:
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: Se desarrollará dos interfaces, la primera interfaz; con los controles adecuados para que el usuario pueda configurar la ubicación GPS de dirección de recogida y de destino, la segunda interfaz; para que el usuario pueda configurar la dirección de recogida y de destino se ejecutará una ventana modal con acceso a Google Maps.	
Fuente: Elaboración propia	

Tabla 55

Tarea N° 34: Gestión de configurar domicilio

Gestión de configurar domicilio	
Número de tarea: 34	Número de historia: 17
Nombre: Gestión de configurar domicilio	
Tipo de tarea: Desarrollo	Puntos estimados:
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: Al ingresar por primera vez en el aplicativo móvil, se ejecutará una ventana modal donde el usuario proporcionará permisos para poder activar el GPS del dispositivo móvil y configurar la dirección de recogida y de destino con acceso a Google Maps.	
Fuente: Elaboración propia	

Tabla 56

Historia N° 18: Reportar inasistencia

Registrar inasistencia	
Número de historia: 18	Usuario: Apoderado
Nombre de historia: Reportar inasistencia	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos Estimados:	Iteración Asignada: 03
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: El usuario apoderado visualizará el recorrido que el chofer realizará a los escolares y podrá registrar la inasistencia de su hijo.	
Observaciones: Ninguna.	
Fuente: Elaboración propia	

Tabla 57

Tarea N° 35: Diseño de interfaz de registrar inasistencia

Diseño de interfaz de registrar inasistencia	
Número de tarea: 35	Número de historia: 18
Nombre: Diseño de interfaz de registrar inasistencia	
Tipo de tarea: Desarrollo	Puntos estimados:
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: Se desarrollará dos interfaces, la primera interfaz; mostrará el recorrido que el chofer realiza al escolar, la segunda interfaz; mostrará una ventana modal donde el usuario apoderado podrá registrar el motivo de la inasistencia de su hijo.	
Fuente: Elaboración propia	

Tabla 58

Tarea N° 30: Gestión de registrar inasistencia

Gestión de registrar inasistencia	
Número de tarea: 36	Número de historia: 18
Nombre: Gestión de registrar inasistencia	
Tipo de tarea: Desarrollo	Puntos estimados:
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: Cuando el usuario apoderado ingresa al aplicativo móvil, se visualiza el recorrido que el chofer realizar al escolar, el botón REPORTAR FALTA; ejecuta una ventana modal con opciones personalizadas para que el usuario apoderado elija cual fue el motivo de la inasistencia, el botón CONFIRMAR; acepta la confirmación de la inasistencia.	
Fuente: Elaboración propia	

Tabla 59

Historia N° 19: Historial de recorridos

Historial de recorridos	
Número de historia: 19	Usuario: Apoderado
Nombre de historia: Historial de recorridos	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos Estimados:	Iteración Asignada: 03
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: El usuario apoderado podrá realizar una búsqueda de historial de recorridos por fecha o por rango de fechas que el chofer ha realizado.	
Observaciones: Ninguna.	
Fuente: Elaboración propia	

Tabla 60

Tarea N° 37: Diseño de interfaz de historial de recorridos

Diseño de interfaz de historial de recorridos	
Número de tarea: 37	Número de historia: 19
Nombre: Diseño de interfaz de historial de recorridos	
Tipo de tarea: Desarrollo	Puntos estimados:
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: Se desarrollará una interfaz que permitirá realizar una búsqueda de recorridos por fecha o por rangos de fechas.	
Fuente: Elaboración propia	

Tabla 61

Tarea N° 38: Gestión de historial de recorridos

Gestión de historial de recorridos	
Número de tarea: 38	Número de historia: 19
Nombre: Gestión de historial de recorridos	
Tipo de tarea: Desarrollo	Puntos estimados:
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: El botón BUSCAR realizará una búsqueda por fecha o por rango de fechas de los recorridos del escolar.	
Fuente: Elaboración propia	

Tabla 62

Historia N° 20: Resetear contraseñas desde el aplicativo móvil

Resetear contraseña desde el aplicativo móvil	
Número de historia: 20	Usuario: Apoderado, Chofer
Nombre de historia: Resetear contraseñas desde el aplicativo móvil	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos Estimados:	Iteración Asignada: 03
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: El usuario apoderado y usuario chofer podrán resetear sus contraseñas que se le crearon por defecto en el sistema web.	
Observaciones: Ninguna.	
Fuente: Elaboración propia	

Tabla 63

Tarea N° 39: Diseño de interfaz de resetear contraseñas desde el aplicativo móvil

Diseño de interfaz de resetear contraseñas desde el aplicativo móvil	
Número de tarea: 39	Número de historia: 20
Nombre: Diseño de interfaz de resetear contraseñas desde el aplicativo móvil	
Tipo de tarea: Desarrollo	Puntos estimados:
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: Se desarrollará una interfaz que permitirá al usuario apoderado y usuario chofer poder cambiar sus contraseñas que se crearon por defecto en el sistema web.	
Fuente: Elaboración propia	

Tabla 64

Tarea N° 40: Gestión de resetear contraseñas desde el aplicativo móvil

Gestión de resetear contraseñas desde el aplicativo móvil	
Número de tarea: 40	Número de historia: 20
Nombre: Gestión de resetear contraseñas desde el aplicativo móvil	
Tipo de tarea: Desarrollo	Puntos estimados:
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: El botón GUARDAR permite validar la contraseña antigua y además confirmará el cambio de la nueva contraseña.	
Fuente: Elaboración propia	

Tabla 65

Historia N° 21: Crear recorrido

Crear recorrido	
Número de historia: 21	Usuario: Chofer
Nombre de historia: Crear recorrido	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos Estimados:	Iteración Asignada: 03
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: El usuario chofer podrá crear un nuevo recorrido ya sea en el horario de ingreso escolar (turno mañana o tarde) y horario de salida escolar (turno mañana o tarde).	
Observaciones: El usuario chofer como máximo realizará 2 recorridos por día.	
Fuente: Elaboración propia	

Tabla 66

Tarea N° 41: Diseño de interfaz de crear recorrido

Diseño de interfaz de crear recorrido	
Número de tarea: 41	Número de historia: 21
Nombre: Diseño de interfaz de crear recorrido	
Tipo de tarea: Desarrollo	Puntos estimados:
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: Se desarrollará una interfaz que permitirá al usuario chofer crear un nuevo recorrido.	
Fuente: Elaboración propia	

Tabla 67

Tarea N° 42: Gestión de crear recorrido

Gestión de crear recorrido	
Número de tarea: 42	Número de historia: 21
Nombre: Gestión de crear recorrido	
Tipo de tarea: Desarrollo	Puntos estimados:
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: El botón CREAR permite crear un nuevo recorrido asignando un turno (mañana o tarde) y un horario (ingreso o salida).	
Fuente: Elaboración propia	

Tabla 68

Historia N° 22: Notificar incidencias

Notificar incidencias	
Número de historia: 22	Usuario: Chofer
Nombre de historia: Notificar incidencias	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos Estimados:	Iteración Asignada: 03
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: El usuario chofer podrá notificar incidencias personalizadas que ocurren durante el recorrido ya sea el horario de ingreso escolar y horario de salida escolar.	
Observaciones: Ninguna.	
Fuente: Elaboración propia	

Tabla 69

Tarea N° 43: Diseño de interfaz de notificar incidencias

Diseño de interfaz de notificar incidencias	
Número de tarea: 43	Número de historia: 22
Nombre: Diseño de interfaz de notificar incidencias	
Tipo de tarea: Desarrollo	Puntos estimados:
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: Se desarrollará tres interfaces, la primera interfaz; el usuario chofer visualiza el recorrido que está realizando, la segunda interfaz; permite que permitirá al usuario chofer notificar incidencias personalizadas durante su recorrido.	
Fuente: Elaboración propia	

Tabla 70

Tarea N° 44: Gestión de notificar incidencias

Gestión de notificar incidencias	
Número de tarea: 44	Número de historia: 22
Nombre: Gestión de notificar incidencias	
Tipo de tarea: Desarrollo	Puntos estimados:
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: El botón NUEVA INCIDENCIA permite crear una nueva incidencia que ha ocurrido durante el recorrido, el botón VER INCIDENCIA; permite visualizar las incidencias que han ocurrido durante el recorrido.	
Fuente: Elaboración propia	

Tabla 71

Historia N° 23: Marcar recogida

Marcar recogida	
Número de historia: 23	Usuario: Chofer
Nombre de historia: Marcar recogida	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos Estimados:	Iteración Asignada: 03
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: El usuario chofer visualizará el listado de escolares y marcará la recogida del escolar cuando este sea recogido de su hogar.	
Observaciones: El usuario chofer marcará recogida del escolar sólo si es en el horario de ingreso escolar.	
Fuente: Elaboración propia	

Tabla 72

Tarea N° 45: Diseño de interfaz de marcar recogida

Diseño de interfaz de marcar recogida	
Número de tarea: 45	Número de historia: 23
Nombre: Diseño de interfaz de marcar recogida	
Tipo de tarea: Desarrollo	Puntos estimados:
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: Se desarrollará dos interfaces, la primera interfaz; el usuario chofer visualiza el listado de escolares que podrá recoger a sus domicilios, la segunda interfaz; permitirá al usuario chofer confirmar la recogida del escolar a su domicilio.	
Fuente: Elaboración propia	

Tabla 73

Tarea N° 46: Gestión de marcar recogida

Gestión de marcar recogida	
Número de tarea: 46	Número de historia: 23
Nombre: Gestión de marcar recogida	
Tipo de tarea: Desarrollo	Puntos estimados:
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: El botón MARCAR RECOGIDA ejecuta una ventana modal que permitirá confirmar la recogida del escolar en su domicilio.	
Fuente: Elaboración propia	

3.3.4.2. Diseño de interfaces

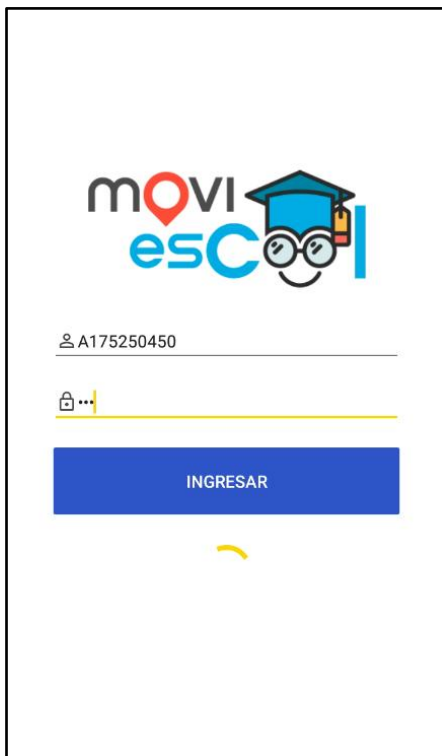


Figura 39. Módulo inicio de sesión: Interfaz de ingresar al aplicativo móvil

Fuente: Elaboración propia



Figura 40. Módulo apoderado: Interfaz de configurar domicilio

Fuente: Elaboración propia



Figura 41. Módulo apoderado: Interfaz de registrar inasistencia

Fuente: Elaboración propia



Figura 42. Módulo apoderado: Interfaz de historial de recorridos

Fuente: Elaboración propia

☰ Cambiar clave ⋮

Bernardo Amado Bonilla Coronado

Usuario: A171874291

Ingrese contraseña antigua

Ingrese nueva contraseña

Confirme nueva contraseña

GUARDAR

Figura 43. Módulo apoderado y chofer: Interfaz de cambiar clave desde el aplicativo móvil

Fuente: Elaboración propia

Nuevo Recorrido

Turno:
Mañana ▼

Horario:
Ingreso ▼

Observaciones

CREAR

Figura 44. Módulo chofer: Interfaz de crear recorrido

Fuente: Elaboración propia



Figura 45. Módulo chofer: Interfaz de notificar incidencias

Fuente: Elaboración propia

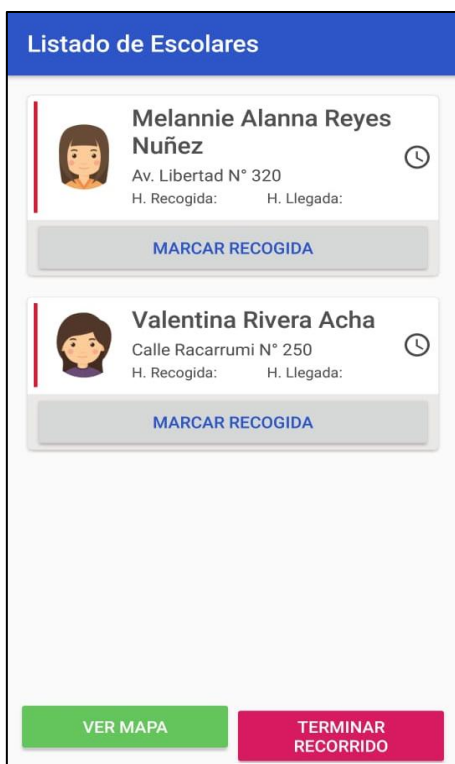


Figura 46. Módulo chofer: Interfaz de marcar recogida

Fuente: Elaboración propia

3.3.4.3. Diagrama de base de datos

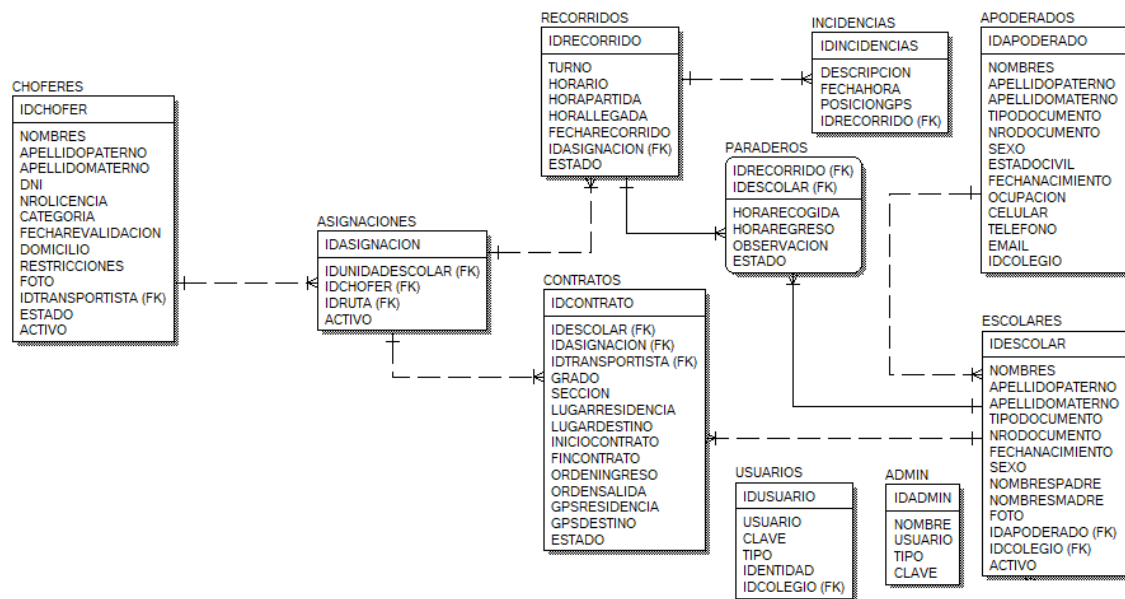


Figura 47. Diagrama de base de datos: Iteración 3

Fuente: Elaboración propia

3.3.5. Iteración 4

En la iteración 4 se pretende finalizar el desarrollo del sistema web con funcionalidades relacionados a los reportes: listado de rutas, listado de asignaciones, listado de contratos, listado de asistencias, reseteo de contraseñas.

3.3.5.1. Historias de usuario

Esta iteración consta de 5 historias de usuario:

- Reporte de rutas
- Reportes de asignaciones
- Reporte de contratos
- Reporte de asistencia
- Resetear contraseñas desde el sistema web

Tabla 74

Historia N° 24: Reporte de rutas

Reporte de rutas	
Número: 24	Usuario: Colegio
Nombre de historia: Reporte de rutas	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos Estimados:	Iteración Asignada: 04
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: El usuario colegio ingresará al menú de reporte de rutas, visualizará el listado de rutas que los transportistas han registrados.	
Observaciones: Ninguna	
Fuente: Elaboración propia	

Tabla 75

Tarea N° 47: Diseño de interfaz de reporte de rutas

Diseño de interfaz de reporte de rutas	
Número de tarea: 47	Número de historia: 24
Nombre: Diseño de interfaz de reportes de rutas	
Tipo de tarea: Desarrollo	Puntos estimados:
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: Se diseñará una interfaz de usuario donde se podrá visualizar el listado de rutas que los transportistas han registrado.	
Fuente: Elaboración propia	

Tabla 76

Tarea N° 48: Gestión de reportes de rutas

Gestión de reportes de rutas	
Número de tarea: 48	Número de historia: 24
Nombre: Gestión de reportes de rutas	
Tipo de tarea: Desarrollo	Puntos estimados:
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: Al pulsar el botón GENERAR PDF se abrirá una nueva pestaña en el navegador web donde se visualizará el listado de rutas que los transportistas han registrado.	
Fuente: Elaboración propia	

Tabla 77

Historia N° 25: Reporte de asignaciones

Reporte de asignaciones	
Número: 25	Usuario: Colegio
Nombre de historia: Reporte de asignaciones	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos Estimados:	Iteración Asignada: 04
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: El usuario colegio ingresará al menú de reporte de asignaciones, visualizará el listado de asignaciones que los transportistas han asignado a los choferes para determinadas rutas.	
Observaciones: Ninguna	
Fuente: Elaboración propia	

Tabla 78

Tarea N° 49: Diseño de interfaz de reporte de asignaciones

Diseño de interfaz de reporte de asignaciones	
Número de tarea:	Número de historia:
Nombre: Diseño de interfaz de reportes de asignaciones	
Tipo de tarea: Desarrollo	Puntos estimados:
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: Se diseñará una interfaz de usuario donde se podrá visualizar el listado de asignaciones que los transportistas han asignado a los choferes para determinadas rutas.	
Fuente: Elaboración propia	

Tabla 79

Tarea N° 50: Gestión de reportes de asignaciones

Gestión de reportes de asignaciones	
Número de tarea: 50	Número de historia: 25
Nombre: Gestión de reportes de asignaciones	
Tipo de tarea: Desarrollo	Puntos estimados:
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: Al pulsar el botón GENERAR PDF se abrirá una nueva pestaña en el navegador web donde se visualizará el listado de asignaciones que los transportistas han asignado a los choferes para determinadas rutas.	
Fuente: Elaboración propia	

Tabla 80

Historia N° 26: Reporte de contratos

Reporte de contratos	
Número: 26	Usuario: Colegio
Nombre de historia: Reporte de contratos	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos Estimados:	Iteración Asignada: 04
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: El usuario colegio ingresará al menú de reporte de contratos, visualizará el listado de contratos que los transportistas han registrado, tendrá la opción de generar archivos pdf de manera general e individual y ver datos del contrato.	
Observaciones: Ninguna	
Fuente: Elaboración propia	

Tabla 81

Tarea N° 51: Diseño de interfaz de reportes de contratos

Diseño de interfaz de reporte de contratos	
Número de tarea: 51	Número de historia: 26
Nombre: Diseño de interfaz de reportes de contratos	
Tipo de tarea: Desarrollo	Puntos estimados:
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: Se diseñará una interfaz de usuario donde se podrá visualizar el listado de contratos que los transportistas han registrado, contará con botones básicos para poder generar archivos pdf de manera general e individual y ver más información del contrato.	
Fuente: Elaboración propia	

Tabla 82

Tarea N° 52: Gestión de reportes de contratos

Gestión de reportes de contratos	
Número de tarea: 52	Número de historia: 26
Nombre: Gestión de reportes de contratos	
Tipo de tarea: Desarrollo	Puntos estimados:
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: Al pulsar el botón GENERAR PDF se abrirá una nueva pestaña en el navegador web donde se visualizará el listado general de contratos que los transportistas han asignado, al pulsar el botón PDF se abrirá una nueva pestaña en el navegador web donde se visualizará el contrato que el transportista ha realizado con el apoderado y el botón VER ejecutará una ventana modal en la que nos permitirá visualizar la información detallada del contrato.	
Fuente: Elaboración propia	

Tabla 83

Historia N° 27: Reporte de asistencia

Reporte de asistencia	
Número: 27	Usuario: Colegio
Nombre de historia: Reporte de asistencia	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos Estimados:	Iteración Asignada: 04
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: El usuario colegio ingresará al menú de reporte de asistencias, visualizará el listado de asistencias que emitirá el chofer cuando termine su recorrido en el traslado del escolar al colegio.	
Observaciones: Ninguna	
Fuente: Elaboración propia	

Tabla 84

Tarea N° 53: Diseño de interfaz de reportes de asistencia

Diseño de interfaz de reporte de asistencia	
Número de tarea: 53	Número de historia: 27
Nombre: Diseño de interfaz de reportes de asistencia	
Tipo de tarea: Desarrollo	Puntos estimados:
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: Se diseñará una interfaz de usuario donde se podrá visualizar el listado de asistencias que emitirá el chofer cuando termine su recorrido en el traslado de los escolares al colegio.	
Fuente: Elaboración propia	

Tabla 85

Tarea N° 54: Gestión de reportes de asignaciones

Gestión de reportes de asistencias	
Número de tarea: 54	Número de historia: 27
Nombre: Gestión de reportes de asignaciones	
Tipo de tarea: Desarrollo	Puntos estimados:
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: Al pulsar el botón GENERAR PDF se abrirá una nueva pestaña en el navegador web donde se visualizará el listado de asistencias que emitirá el chofer cuando termine su recorrido en el traslado del escolar al colegio.	
Fuente: Elaboración propia	

Tabla 86

Historia N° 28: Resetear contraseñas desde el sistema web

Resetear contraseñas desde el sistema web	
Número: 28	Usuario: Colegio, Transportista
Nombre de historia: Resetear contraseñas desde el sistema web	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos Estimados:	Iteración Asignada: 04
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: El usuario colegio y usuario transportista ingresarán al menú de configuración de contraseñas y podrán resetear las contraseñas de los usuarios.	
Observaciones: Si es el usuario colegio podrá cambiar la contraseña del usuario apoderado, transportista, chofer y de su mismo usuario. Si es el usuario transportista podrá cambiar la contraseña del usuario transportista y de su mismo usuario.	
Fuente: Elaboración propia	

Tabla 87

Tarea N° 55: Diseño de interfaz de resetear contraseñas desde el sistema web

Diseño de interfaz de resetear contraseñas desde el sistema web	
Número de tarea: 55	Número de historia: 28
Nombre: Diseño de interfaz de resetear contraseñas desde el sistema web	
Tipo de tarea: Desarrollo	Puntos estimados:
Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez	
Descripción: Se diseñará una interfaz donde los usuarios podrán modificar sus contraseñas que se crearon por defecto.	
Fuente: Elaboración propia	

Tabla 88

Tarea N° 56: Gestión de resetear contraseñas desde el sistema web

Gestión de resetear contraseñas desde el sistema web

Número de tarea: 56

Número de historia: 28

Nombre: Gestión de resetear contraseñas desde el sistema web

Tipo de tarea: Desarrollo

Puntos estimados:

Programador Responsable: Bernardo Amado Bonilla Coronado, Hans Pritton Diaz Nuñez

Descripción: Los usuarios al presionar el botón CAMBIAR CONTRASEÑA, modificaran sus contraseñas que el sistema les creó por defecto.

Fuente: Elaboración propia

3.3.5.2. Diseño de interfaces

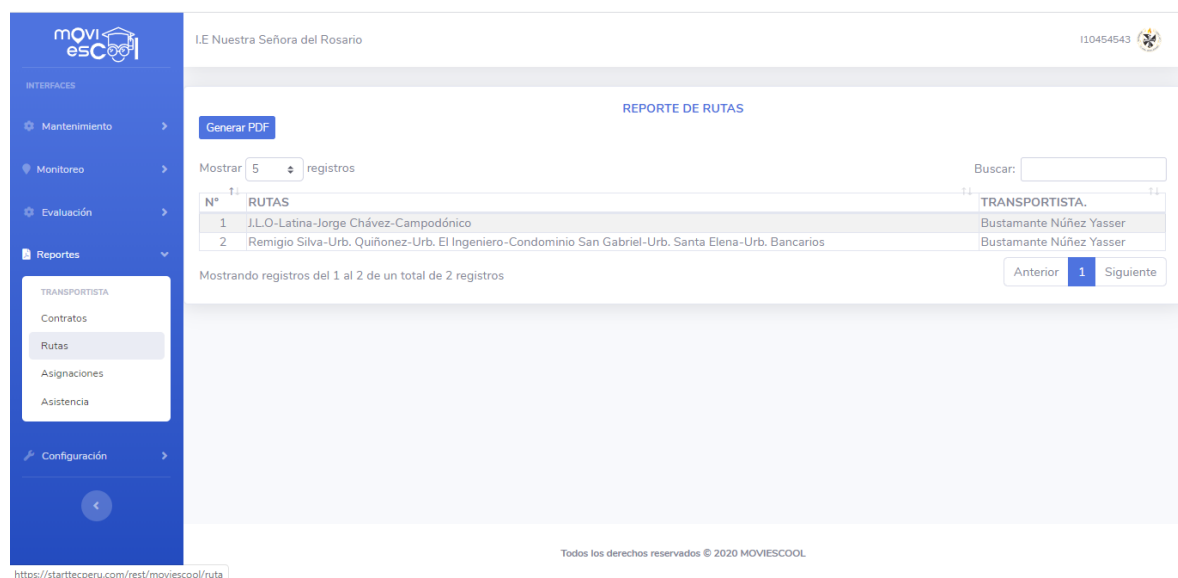


Figura 48. Módulo colegio: Interfaz de reporte de rutas

Fuente: Elaboración propia

I.E Nuestra Señora del Rosario
110454543

**movi
escool**

INTERFACES

- Mantenimiento >
- Monitoreo >
- Evaluación >
- Reportes >

TRANSPORTISTA
Contratos
Rutas
Asignaciones
Asistencia
- Configuración >

LISTADO DE ASIGNACIONES

Mostrar 5 registros

N°	PLACA	CHOFER	RUTA	TRANSP.
1	A3K315	Sergio Bonilla Nuñez	Remigio Silva-Urb. Quiñonez-Urb. El Ingeniero-Condominio San Gabriel-Urb. Santa Elena-Urb. Bancarios	Bustamante Núñez Yasser
2	A4X5FX	Erick Rubio Díaz	J.L.O-Latina-Jorge Chávez-Campodónico	Bustamante Núñez Yasser

Buscar:

Mostrando registros del 1 al 2 de un total de 2 registros

Anterior
1
Siguiente

Todos los derechos reservados © 2020 MOVIESCOOL

Figura 49. Módulo colegio: Interfaz de reporte de asignaciones

Fuente: Elaboración propia

movi
esc
I.E Nuestra Señora del Rosario
ID0454543 

INTERFACES

- Mantenimiento >
- Monitoreo >
- Evaluación >
- Reportes >

TRANSPORTISTA
Contratos
 Rutas
 Asignaciones
 Asistencia
- Configuración >

REPORTE DE CONTRATOS

[Generar PDF](#)

Mostrar registros
 Buscar:

N°	ESCOLAR	APODERADO	RUTA	F. CONTRATO	TRANSP.	PDF	OPCION
1	Larissa Bonilla Lopez	Bernardo Amado Bonilla Coronado	Jorge Chavez N° 956-Jorge Chavez N° 956	04/04/2019 - 28/12/2019	Bustamante Núñez Yasser		
2	Ayza Estrella Díaz Puentes	José Ricardo Díaz Hernandez	Jorge Chavez N° 680-Jorge Chavez N° 680	10/04/2019 - 28/12/2019	Bustamante Núñez Yasser		
3	Silvana del Pilar Supo Ramirez	Jhordy Akito Supo Oliden	Jorge Chavez N° 340-Jorge Chavez N° 340	06/12/2019 - 28/12/2019	Bustamante Núñez Yasser		

Mostrando registros del 1 al 3 de un total de 3 registros

[Anterior](#)
1
[Siguiente](#)

Todos los derechos reservados © 2020 MOVIESCOOL

Figura 50. Módulo colegio: Interfaz de reporte de contratos

Fuente: Elaboración propia

3.3.5.3. Diagrama de base de datos

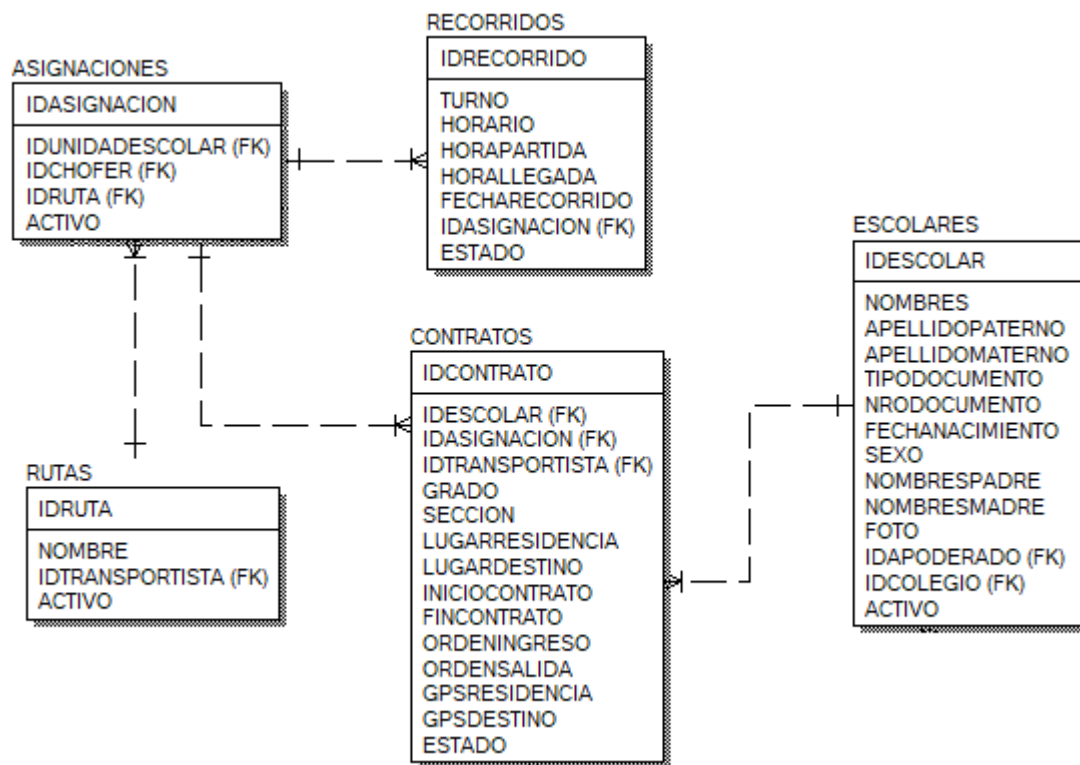


Figura 51. Diagrama de base de datos: Iteración 4

Fuente: Elaboración propia

3.4. Diagrama de componentes

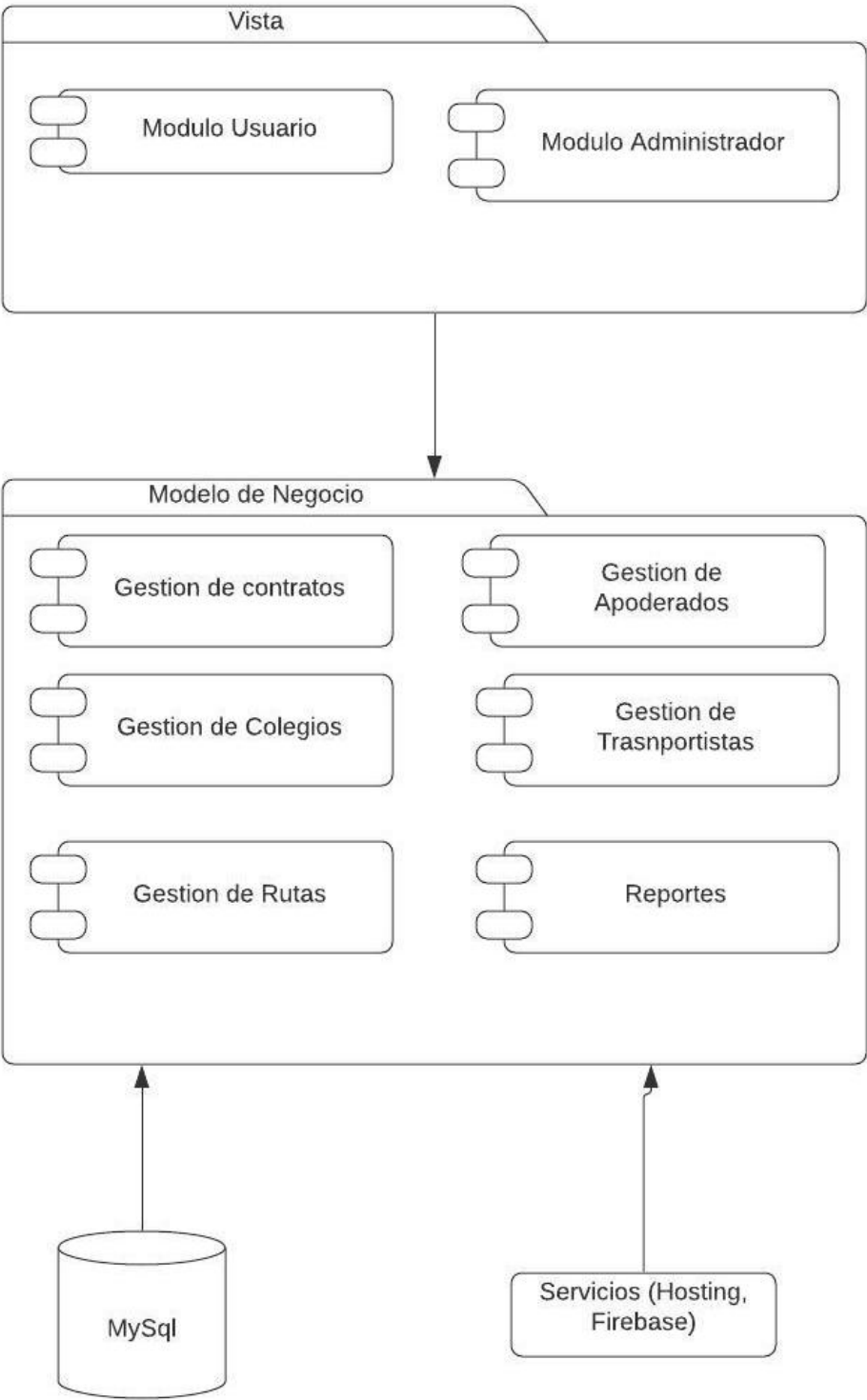


Figura 52. Diagrama de componentes

Fuente: Elaboración propia

3.5. Diagrama de despliegue

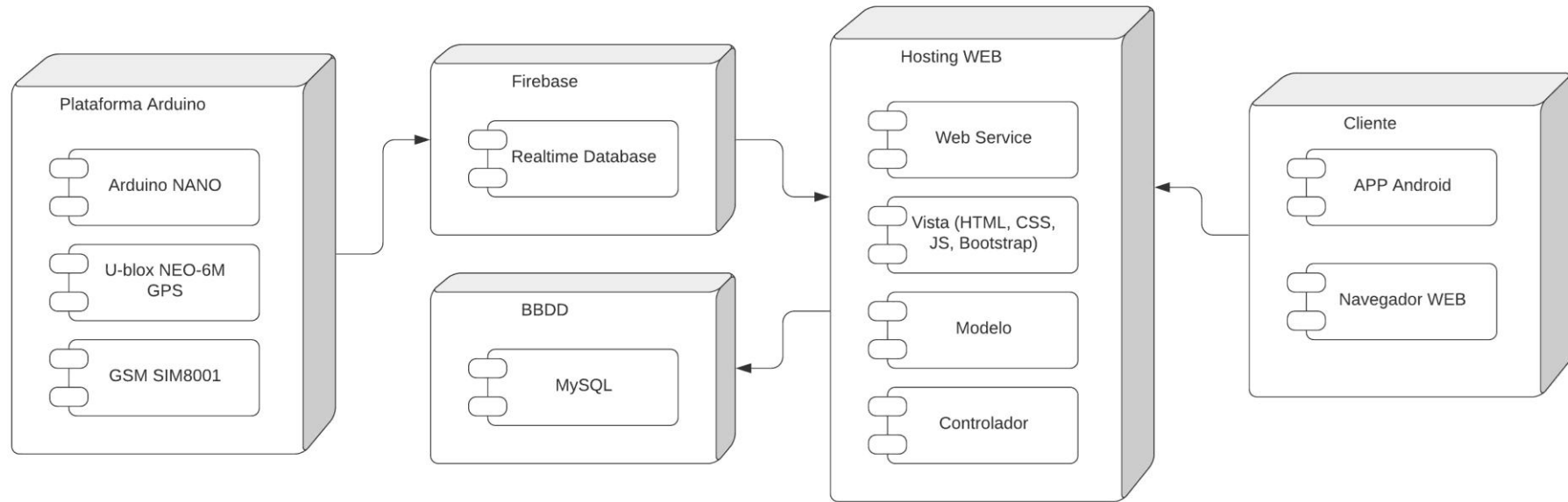


Figura 53. Diagrama de despliegue

Fuente: Elaboración propia

3.6. Diagrama de modelo relacional

3.6.1. Diseño lógico de base de datos

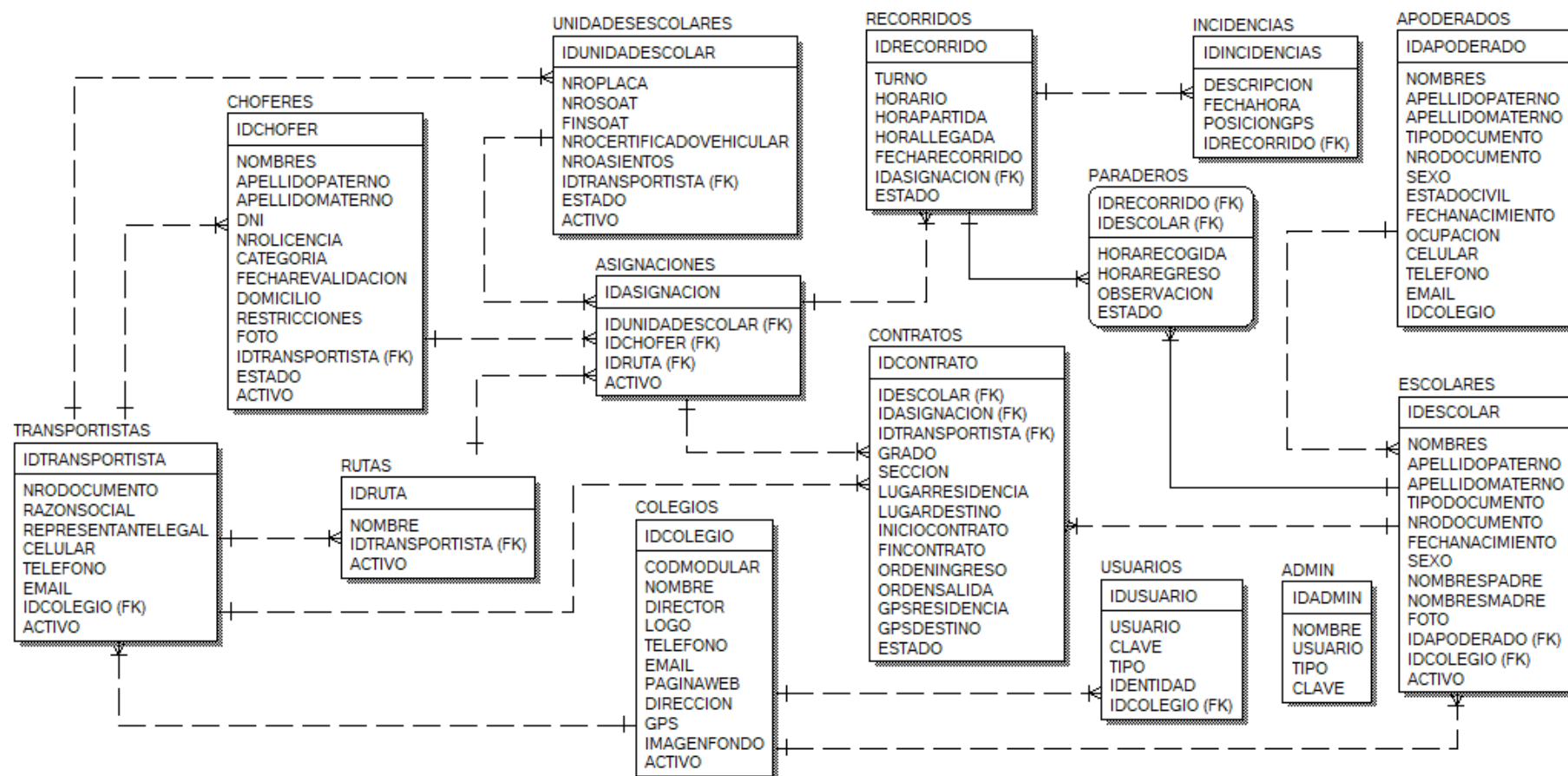


Figura 54. Diseño de modelo relacional de base de datos

Fuente: Elaboración propia

3.6.2. Script de la base de datos

3.6.2.1. Script DDL para crear base de datos MOVIESCOOL

```
-- CREACIÓN DE BASE DE DATOS BDMOVILESCOOL --  
DROP DATABASE IF EXISTS BDMOVILESCOOL;  
CREATE DATABASE BDMOVILESCOOL;  
USE BDMOVILESCOOL;
```

3.6.2.2. Script DDL para crear tablas

```
-- CREACIÓN DE TABLA ADMIN --  
CREATE TABLE ADMIN(  
    IDADMIN INT AUTO_INCREMENT PRIMARY KEY,  
    NOMBRE VARCHAR(200) ,  
    USUARIO VARCHAR(40) ,  
    TIPO VARCHAR(15) DEFAULT 'Admin',  
    CLAVE VARCHAR(40)  
) ENGINE=INNODB DEFAULT CHARSET=utf8;  
  
-- CREACIÓN DE TABLA COLEGIOS --  
CREATE TABLE COLEGIOS(  
    IDCOLEGIO INT AUTO_INCREMENT PRIMARY KEY,  
    NOMBRE VARCHAR(80) ,  
    LOGO VARCHAR(100) ,  
    DIRECTOR VARCHAR(80) ,  
    TELEFONO CHAR(6) ,  
    EMAIL VARCHAR(60) ,  
    PAGINAWEB VARCHAR(100) ,  
    DIRECCION VARCHAR(80) ,  
    CODMODULAR CHAR(7) ,  
    IMAGENFONDO VARCHAR(100) ,  
    ACTIVO BOOLEAN DEFAULT TRUE ,  
    GPS VARCHAR(200)  
) ENGINE=INNODB DEFAULT CHARSET=utf8;  
  
-- CREACIÓN DE TABLA APODERADOS --  
CREATE TABLE APODERADOS(  
    IDAPODERADO INT AUTO_INCREMENT PRIMARY KEY,  
    NOMBRES VARCHAR(40) ,  
    APELLIDOPATERNO VARCHAR(30) ,  
    APELLIDOMATERNO VARCHAR(30) ,  
    TIPODOCUMENTO VARCHAR(30) ,  
    NDOCUMENTO VARCHAR(12) ,  
    ESTADOCIVIL VARCHAR(30) ,  
    SEXO CHAR(10) ,  
    FECHANACIMIENTO DATE ,  
    OCUPACION VARCHAR(40) ,
```

```

        CELULAR CHAR(9) ,
        TELEFONO CHAR(6) ,
        EMAIL VARCHAR(60) ,
        IDCOLEGIO INT
    ) ENGINE=INNODB DEFAULT CHARSET=utf8;

-- CREACIÓN DE TABLA ESCOLARES --
CREATE TABLE ESCOLARES (
    IDESCOLAR INT AUTO_INCREMENT PRIMARY KEY,
    TIPODOCUMENTO VARCHAR(30) ,
    DNI CHAR(8) ,
    NOMBRES VARCHAR(40) ,
    APELLIDOPATERNO VARCHAR(50) ,
    APELLIDOMATERNO VARCHAR(50) ,
    FECHANACIMIENTO DATE,
    SEXO CHAR(10) ,
    FOTO VARCHAR(100) ,
    NOMBREPADRE VARCHAR(80) ,
    NOMBREMADRE VARCHAR(80) ,
    IDAPODERADO INT,
    IDCOLEGIO INT,
    ACTIVO BOOLEAN DEFAULT TRUE,
    CONSTRAINT FK_ESCOLAR_APODERADO FOREIGN KEY (IDAPODERADO)
        REFERENCES APODERADOS (IDAPODERADO) ,
    CONSTRAINT FK_ESCOLAR_COLEGIO FOREIGN KEY (IDCOLEGIO)
        REFERENCES COLEGIOS (IDCOLEGIO)
    ) ENGINE=INNODB DEFAULT CHARSET=utf8;

-- CREACIÓN DE TABLA USUARIOS --
CREATE TABLE USUARIOS (
    IDUSUARIO INT AUTO_INCREMENT PRIMARY KEY,
    USUARIO VARCHAR(40) ,
    CLAVE VARCHAR(40) ,
    TIPO VARCHAR(30) ,
    IDENTIDAD INT,
    IDCOLEGIO INT,
    CONSTRAINT FK_USUARIO_COLEGIO FOREIGN KEY (IDCOLEGIO)
        REFERENCES COLEGIOS (IDCOLEGIO)
    ) ENGINE=INNODB DEFAULT CHARSET=utf8;

-- CREACIÓN DE TABLA TRANSPORTISTAS --
CREATE TABLE TRANSPORTISTAS (
    IDTRANSPORTISTA INT AUTO_INCREMENT PRIMARY KEY,
    RUC CHAR(11) ,
    RAZONSOCIAL VARCHAR(40) ,
    REPRESENTANTELEGAL VARCHAR(70) ,
    TELEFONO CHAR(6) ,
    CELULAR CHAR(9) ,

```

```

EMAIL VARCHAR(60) ,
IDCOLEGIO INT,
ACTIVO BOOLEAN DEFAULT TRUE,
CONSTRAINT FK_TRANSPORTISTA_COLEGIO FOREIGN KEY (IDCOLEGIO)
REFERENCES COLEGIOS (IDCOLEGIO)
) ENGINE=INNODB DEFAULT CHARSET=utf8;

```

-- CREACIÓN DE TABLA CHOFERES --

```

CREATE TABLE CHOFERES (
    IDCHOFER INT AUTO_INCREMENT PRIMARY KEY,
    DNI CHAR(8) ,
    NOMBRES VARCHAR(40) ,
    APELLIDOPATERNO VARCHAR(30) ,
    APELLIDOMATERNO VARCHAR(30) ,
    NROLICENCIA CHAR(12) ,
    CATEGORIA CHAR(4) ,
    DOMICILIO VARCHAR(100) ,
    FECHAREVALIDACION DATE,
    FOTO VARCHAR(100) ,
    RESTRICCIONES VARCHAR(20) ,
    IDTRANSPORTISTA INT,
    ACTIVO BOOLEAN DEFAULT TRUE,
    ASIGNADO BOOLEAN DEFAULT FALSE,
    ESTADO VARCHAR(30) DEFAULT 'Pendiente',
    CONSTRAINT FK_CHOFER_TRANSPORTISTA
    FOREIGN KEY (IDTRANSPORTISTA)
    REFERENCES TRANSPORTISTAS (IDTRANSPORTISTA)
) ENGINE=INNODB DEFAULT CHARSET=utf8;

```

-- CREACIÓN DE TABLA UNIDADES ESCOLARES --

```

CREATE TABLE UNIDADES ESCOLARES (
    IDUNIDADES COLAR INT AUTO_INCREMENT PRIMARY KEY,
    NROPLACA CHAR(6) ,
    NROSOAT CHAR(12) ,
    FINSOAT DATE,
    NROCERTIFICADOVEHICULAR VARCHAR(10) ,
    NROASIENTOS INT,
    IDTRANSPORTISTA INT,
    ESTADO VARCHAR(30) DEFAULT 'Pendiente',
    CONSTRAINT FK_UNIDADES COLAR_TRANSPORTISTA
    FOREIGN KEY (IDTRANSPORTISTA)
    REFERENCES TRANSPORTISTAS (IDTRANSPORTISTA)
) ENGINE=INNODB DEFAULT CHARSET=utf8;

```

-- CREACIÓN DE TABLA RUTAS --

```

CREATE TABLE RUTAS (
    IDRUTA INT AUTO_INCREMENT PRIMARY KEY,
    NOMBRE VARCHAR(100) ,

```

```

IDTRANSPORTISTA INT,
ACTIVO BOOLEAN DEFAULT TRUE,
CONSTRAINT FK_RUTAS_TRANSPORTISTA
    FOREIGN KEY(IDTRANSPORTISTA)
    REFERENCES TRANSPORTISTAS (IDTRANSPORTISTA)
) ENGINE=INNODB DEFAULT CHARSET=utf8;

-- CREACIÓN DE TABLA ASIGNACIONES --
CREATE TABLE ASIGNACIONES (
    IDASIGNACION INT AUTO_INCREMENT PRIMARY KEY,
    IDUNIDADESCOLAR INT,
    IDCHOFER INT,
    IDRUTA INT,
    ACTIVO BOOLEAN DEFAULT TRUE,
    CONSTRAINT FK_ASIGNACION_UNIDADESCOLAR
        FOREIGN KEY(IDUNIDADESCOLAR)
        REFERENCES UNIDADESESCOLARES (IDUNIDADESCOLAR) ,
    CONSTRAINT FK_ASIGNACION_CHOFER FOREIGN KEY(IDCHOFER)
        REFERENCES CHOFERES (IDCHOFER) ,
    CONSTRAINT FK_ASIGNACION_RUTA FOREIGN KEY(IDRUTA)
        REFERENCES RUTAS (IDRUTA)
) ENGINE=INNODB DEFAULT CHARSET=utf8;

-- CREACIÓN DE TABLA CONTRATOS --
CREATE TABLE CONTRATOS (
    IDCONTRATO INT AUTO_INCREMENT PRIMARY KEY,
    IDTRANSPORTISTA INT,
    IDESCOLAR INT,
    GRADO CHAR(2) ,
    SECCION CHAR(3) ,
    LUGARRESIDENCIA VARCHAR(60) ,
    LUGARDESTINO VARCHAR(69) ,
    INICIOCONTRATO DATE,
    FINCONTRATO DATE,
    ORDENINGRESO BOOLEAN DEFAULT FALSE,
    ORDENSALIDA BOOLEAN DEFAULT FALSE,
    ESTADO VARCHAR(20) DEFAULT 'Vigente',
    IDASIGNACION INT,
    GPSRESIDENCIA VARCHAR(50) ,
    GPSDESTINO VARCHAR(50) ,
    CONSTRAINT FK_CONTRATO_TRANSPORTISTA
        FOREIGN KEY(IDTRANSPORTISTA)
        REFERENCES TRANSPORTISTAS (IDTRANSPORTISTA) ,
    CONSTRAINT FK_CONTRATO_ESCOLAR
        FOREIGN KEY(IDESCOLAR) REFERENCES ESCOLARES (IDESCOLAR) ,
    CONSTRAINT FK_CONTRATO_ASIGNACION
        FOREIGN KEY(IDASIGNACION)
        REFERENCES ASIGNACIONES (IDASIGNACION)

```

```

)ENGINE=INNODB DEFAULT CHARSET=utf8;

-- CREACIÓN DE TABLA RECORRIDOS --
CREATE TABLE RECORRIDOS (
    IDRECORRIDO INT AUTO_INCREMENT PRIMARY KEY,
    TURNO CHAR(10),
    HORAPARTIDA TIME,
    HORALLEGADA TIME,
    LUGARSALIDA VARCHAR(60),
    LUGARLLEGADA VARCHAR(60),
    FECHARECORRIDO DATE,
    IDASIGNACION INT,
    OBSERVACION VARCHAR(100),
    TIPO VARCHAR(15),
    ESTADO VARCHAR(15) DEFAULT 'En curso',
    CONSTRAINT FK_RECORRIDO_ASIGNACION FOREIGN KEY(IDASIGNACION)
        REFERENCES ASIGNACIONES(IDASIGNACION)
)ENGINE=INNODB DEFAULT CHARSET=utf8;

-- CREACIÓN DE TABLA PARADEROS --
CREATE TABLE PARADEROS (
    IDRECORRIDO INT,
    IDESCOLAR INT,
    HORARECOGIDA TIME,
    HORAREGRESO TIME,
    OBSERVACION VARCHAR(200),
    ESTADO VARCHAR(30) DEFAULT 'Pendiente',
    CONSTRAINT FK_PARADERO_RECORRIDO FOREIGN KEY(IDRECORRIDO)
        REFERENCES RECORRIDOS(IDRECORRIDO),
    CONSTRAINT FK_PARADERO_ESCOLAR FOREIGN KEY(IDESCOLAR)
        REFERENCES ESCOLARES(IDESCOLAR)
)ENGINE=INNODB DEFAULT CHARSET=utf8;

ALTER TABLE PARADEROS
ADD CONSTRAINT PK_PARADEROS PRIMARY KEY(IDRECORRIDO, IDESCOLAR);

-- CREACIÓN DE TABLA INCIDENCIAS --
CREATE TABLE INCIDENCIAS (
    IDINCIDENCIA INT AUTO_INCREMENT PRIMARY KEY,
    DESCRIPCION VARCHAR(100),
    FECHAHORA DATETIME,
    IDRECORRIDO INT,
    POSICIONGPS VARCHAR(200),
    CONSTRAINT FK_INCIDENCIA_RECORRIDO FOREIGN KEY(IDRECORRIDO)
        REFERENCES RECORRIDOS(IDRECORRIDO)
)ENGINE=INNODB DEFAULT CHARSET=utf8;

```

3.7. Implementación de funcionalidades

Se detallará la implementación de algunas funcionalidades de las historias de usuario de mantenimiento de apoderados, generación de contratos, reporte de asistencia para demostrar cómo se ha realizado el trabajo de integración con las diferentes herramientas de desarrollo de software para que el sistema informático sea lo más potente y robusta posible.

3.7.1. Mantenimiento de apoderados

3.7.1.1. Tabla de base de datos

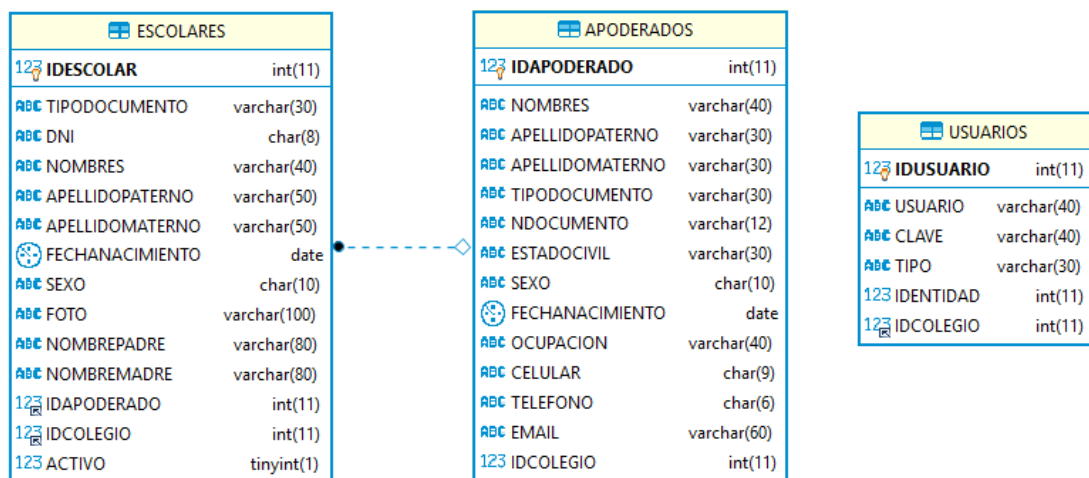


Figura 55. Tablas: ESCOLARES, APODERADOS Y USUARIOS

Fuente: Elaboración propia

3.7.1.2. Procedimientos almacenados

```
-- INSERTAR APODERADOS --
DELIMITER $$
DROP PROCEDURE IF EXISTS SP_INSERTAR_APODERADO$$
CREATE PROCEDURE SP_INSERTAR_APODERADO
(P_NOMBRES VARCHAR(40), P_APELLIDOPATERNO VARCHAR(30),
P_APELLIDOMATERNO VARCHAR(30), P_TIPODOCUMENTO VARCHAR(30),
P_NDOCUMENTO VARCHAR(11), P_ESTADOCIVIL VARCHAR(30),
P_SEXO CHAR(10), P_FECHANACIMIENTO DATE,
P_OCUPACION VARCHAR(40), P_CELULAR CHAR(9),
P_TELEFONO CHAR(6), P_EMAIL VARCHAR(60), P_ID INT,
P_TIPO VARCHAR(30), P_IDCOLEGIO INT, OUT MENSAJE VARCHAR(100))
BEGIN
    DECLARE ID INT;
    IF (EXISTS (SELECT * FROM APODERADOS
                WHERE NDOCUMENTO=P_NDOCUMENTO
                AND IDCOLEGIO=P_IDCOLEGIO)) THEN
```

```

SET MENSAJE='EL NUMERO DE DOCUMENTO INGRESADO YA
EXISTE';

ELSE
    INSERT INTO APODERADOS (NOMBRES, APELLIDOPATERNO,
    APELLIDOMATERNO, TIPODOCUMENTO, NDOCUMENTO, ESTADOCIVIL,
    SEXO, FECHANACIMIENTO, OCUPACION, CELULAR, TELEFONO, EMAIL,
    IDCOLEGIO)
    VALUES (P_NOMBRES, P_APELLIDOPATERNO, P_APELLIDOMATERNO,
    P_TIPODOCUMENTO, P_NDOCUMENTO, P_ESTADOCIVIL, P_SEXO,
    P_FECHANACIMIENTO, P_OCUPACION, P_CELULAR, P_TELEFONO,
    P_EMAIL, P_ID);
    SET ID=(SELECT LAST_INSERT_ID());
    INSERT INTO USUARIOS (USUARIO, TIPO, CLAVE, IDENTIDAD,
    IDCOLEGIO)
    VALUES (CONCAT('A', P_IDCOLEGIO, P_NDOCUMENTO), P_TIPO,
    '123', ID, P_IDCOLEGIO);
    SET MENSAJE='DATOS DEL APODERADOS REGISTRADOS
    CORRECTAMENTE';

END IF;
END $$

-- ACTUALIZAR APODERADOS --
DELIMITER $$
DROP PROCEDURE IF EXISTS SP_ACTUALIZAR_APODERADO$$
CREATE PROCEDURE SP_ACTUALIZAR_APODERADO
(P_IDAPODERADO INT, P_NOMBRES VARCHAR(40),
P_APELLIDOPATERNO VARCHAR(30), P_APELLIDOMATERNO VARCHAR(30),
P_TIPODOCUMENTO VARCHAR(30), P_NDOCUMENTO VARCHAR(11),
P_ESTADOCIVIL VARCHAR(30), P_SEXO CHAR(10),
P_FECHANACIMIENTO DATE, P_OCUPACION VARCHAR(40), P_CELULAR CHAR(9),
P_TELEFONO CHAR(6), P_EMAIL VARCHAR(60), P_ID INT,
OUT MENSAJE VARCHAR(100))
BEGIN
    IF (EXISTS (SELECT * FROM APODERADOS
    WHERE NDOCUMENTO=P_NDOCUMENTO
    AND IDAPODERADO<>P_IDAPODERADO)) THEN
        SET MENSAJE='EL NUMERO DE DOCUMENTO INGRESADO YA
        EXISTE';
    ELSE
        UPDATE APODERADOS
        SET NOMBRES=P_NOMBRES,
        APELLIDOPATERNO=P_APELLIDOPATERNO,
        APELLIDOMATERNO=P_APELLIDOMATERNO,
        TIPODOCUMENTO=P_TIPODOCUMENTO,
        NDOCUMENTO=P_NDOCUMENTO, ESTADOCIVIL=P_ESTADOCIVIL,
        SEXO=P_SEXO, FECHANACIMIENTO=P_FECHANACIMIENTO,
        OCUPACION=P_OCUPACION, CELULAR=P_CELULAR,
        TELEFONO=P_TELEFONO, EMAIL=P_EMAIL, IDCOLEGIO=P_ID

```

```

        WHERE IDAPODERADO=P_IDAPODERADO;
        SET MENSAJE='DATOS DEL APODERADOS ACTUALIZADOS
CORRECTAMENTE';
    END IF;
END $$

-- LISTAR APODERADOS --
DELIMITER $$
DROP PROCEDURE IF EXISTS SP_LISTAR_APODERADOS$$
CREATE PROCEDURE SP_LISTAR_APODERADOS
(P_ID INT)
BEGIN
    IF (P_ID<>0) THEN
        SELECT IDAPODERADO,CONCAT(NOMBRES,' ',APELLIDOPATERNO,
        ' ',APELLIDOMATERNO)AS APODERADO,TIPODOCUMENTO,
        NDOCUMENTO,ESTADOCIVIL,SEXO,
        DATE_FORMAT(FECHANACIMIENTO,'%d/%m/%Y')AS FECHA,
        OCUPACION,CELULAR,TELEFONO,EMAIL
        FROM APODERADOS
        WHERE IDCOLEGIO=P_ID
        ORDER BY IDAPODERADO DESC;
    ELSE
        SELECT A.IDAPODERADO,CONCAT(A.NOMBRES,' ',
        A.APELLIDOPATERNO,' ',A.APELLIDOMATERNO)AS APODERADO,
        A.TIPODOCUMENTO,A.NDOCUMENTO,A.ESTADOCIVIL,A.SEXO,
        DATE_FORMAT(A.FECHANACIMIENTO,'%d/%m/%Y')AS FECHA,
        A.OCUPACION,A.CELULAR,A.TELEFONO,A.EMAIL,C.NOMBRE
        FROM APODERADOS A
        INNER JOIN COLEGIOS C ON A.IDCOLEGIO=C.IDCOLEGIO
        ORDER BY IDAPODERADO DESC;
    END IF;
END $$

-- BUSCAR APODERADO --
DELIMITER $$
DROP PROCEDURE IF EXISTS SP_BUSCAR_APODERADO$$
CREATE PROCEDURE SP_BUSCAR_APODERADO
(P_ID INT)
BEGIN
    SELECT *
    FROM APODERADOS
    WHERE IDAPODERADO=P_ID;
END $$

-- BUSCAR USUARIO --
DELIMITER $$
DROP PROCEDURE IF EXISTS SP_BUSCAR_USUARIO$$

```

```

CREATE PROCEDURE SP_BUSCAR_USUARIO
(P_ID INT,P_TIPO VARCHAR(30))
BEGIN
    SELECT *
    FROM USUARIOS
    WHERE IDENTIDAD=P_ID AND TIPO=P_TIPO;
END $$

-- INSERTAR ESCOLAR --
DELIMITER $$
DROP PROCEDURE IF EXISTS SP_INSERTAR_ESCOLAR$$
CREATE PROCEDURE SP_INSERTAR_ESCOLAR
(P_TIPODOCUMENTO VARCHAR(30),P_DNI CHAR(11),P_NOMBRESVARCHAR(40),
P_APELLIDOPATERNO VARCHAR(50),P_APELLIDOMATERNO VARCHAR(50),
P_FECHANACIMIENTO DATE,P_SEXO CHAR(10),P_FOTO VARCHAR(100),
P_NOMBREPADRE VARCHAR(80),P_NOMBREMADRE VARCHAR(80),
P_IDAPODERADO INT,P_IDCOLEGIO INT,OUT MENSAJE VARCHAR(100))
BEGIN
    IF(EXISTS(SELECT * FROM ESCOLARES WHERE DNI=P_DNI
AND IDCOLEGIO=P_IDCOLEGIO)) THEN
        SET MENSAJE='SET EL DNI YA EXISTE';
    ELSE
        INSERT INTO ESCOLARES(TIPODOCUMENTO,DNI,NOMBRES,
        APELLIDOPATERNO,APELLIDOMATERNO,FECHANACIMIENTO,SEXO,
        FOTO,NOMBREPADRE,NOMBREMADRE,IDAPODERADO,IDCOLEGIO)
        VALUES (P_TIPODOCUMENTO,P_DNI,P_NOMBRES,
        P_APELLIDOPATERNO,P_APELLIDOMATERNO,P_FECHANACIMIENTO,
        P_SEXO,P_FOTO,P_NOMBREPADRE,P_NOMBREMADRE,
        P_IDAPODERADO,P_IDCOLEGIO);
        SET MENSAJE='DATOS DEL ESCOLAR REGISTRADOS
CORRECTAMENTE';
    END IF;
END $$

```

3.7.1.3. Interfaz de usuario

MOVIESCOOL

I.E. Nuestra Señora del Rosario

110454543

INTERFACES

Mantenimiento

APODERADOS

Apoderado

Escolar

TRANSPORTISTAS

Transportista

Unidad Escolar

Chofer

Monitoreo

Evaluación

Reportes

Configuración

NUEVO APODERADO

Mostrar 5 registros

Buscar:

N°	APODERADO	TIPO DOC.	N° DOCUMENTO	SEXO	OPCION
1	Jhordy Akito Supo Oliden	L.E/D.N.I	75250450	Hombre	  
2	Fiorela Katherine Acha Nuñez	L.E/D.N.I	12345678	Mujer	  
3	Aida Carmela Nuñez Chuzón	L.E/D.N.I	16472471	Mujer	  
4	José Ricardo Díaz Hernandez	L.E/D.N.I	16411517	Hombre	  
5	Bernardo Amado Bonilla Coronado	L.E/D.N.I	71874291	Hombre	  

Mostrando registros del 1 al 5 de un total de 5 registros

Anterior 1 Siguiente

Todos los derechos reservados © 2019 MOVIESCOOL

<https://startecperu.com/rest/moviescool/apoderado>

Figura 56. Módulo colegio: Interfaz de listado de apoderados

Fuente: Elaboración propia

MOVIESCOOL

I10454543

INTERFACES

Mantenimiento

APODERADOS

Apoderado

Escolar

TRANSPORTISTAS

Transportista

Unidad Escolar

Chofer

Monitoreo

Evaluación

Reportes

Configuración

REGISTRAR DATOS DEL APODERADO

Nombres(*):

A. Paterno(*):

A. Materno(*):

Tipo Documento(*):

Seleccione una Opción

N° Documento(*):

Sexo(*):

Seleccione una Opción

Estado Civil(*):

Seleccione una Opción

F. Nacimiento(*):

dd/mm/aaaa

Ocupación(*):

Celular(*):

Telefono:

Email:

(*) Campo Obligatorio

GUARDAR CERRAR

Todos los derechos reservados © 2019 MOVIESCOOL

Figura 57. Módulo colegio: Interfaz de registrar apoderado

Fuente: Elaboración propia

EDITAR DATOS DEL APODERADO

Nombres(*):

A. Paterno(*): A. Materno(*):

Tipo Documento(*): N° Documento(*):

Sexo(*): Estado Civil(*):

F. Nacimiento(*): Ocupación(*):

Celular(*): Telefono:

Email:

(*) Campo Obligatorio

ACTUALIZAR **CERRAR**

Todos los derechos reservados © 2019 MOVIESCOOL

Figura 58. Módulo colegio: Interfaz de editar apoderado

Fuente: Elaboración propia

INFORMACION GENERAL DEL APODERADO

Nombres : José Ricardo Díaz Hernandez

Tipo Documento : LE/D.N.I.

N° Documento : 16411517

Estado Civil : Casado

Sexo : Hombre

F. Nacimiento : 10/01/1957

Ocupación : Maestro Soldador

Celular : 922501144

Telefono :

Email : jricardo57@gmail.com

Cerrar

Todos los derechos reservados © 2019 MOVIESCOOL

Figura 59. Módulo colegio: Interfaz de ver información del apoderado

Fuente: Elaboración propia

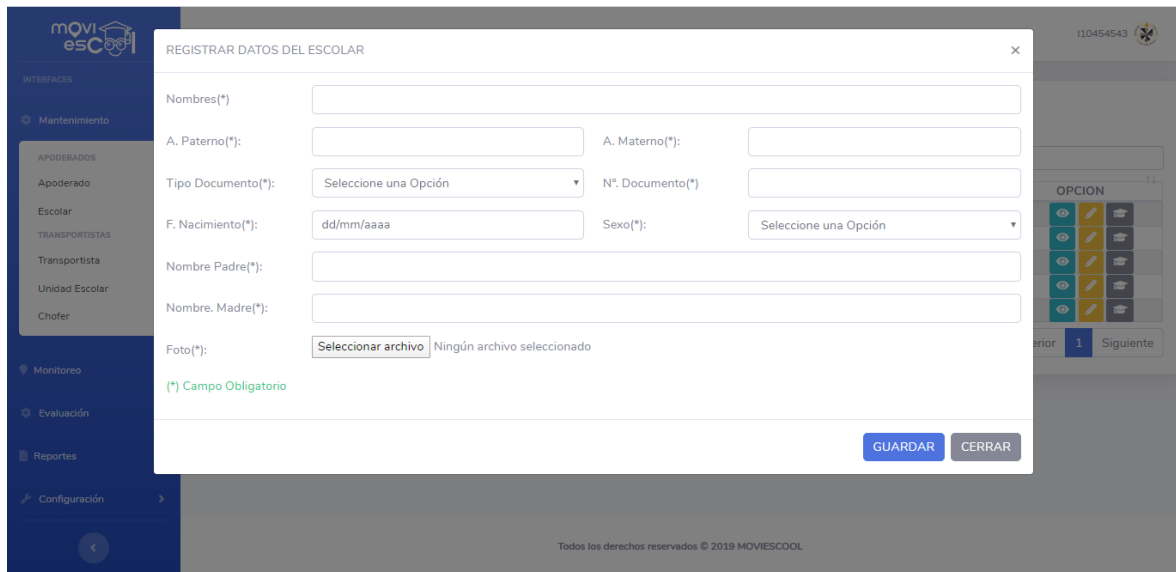


Figura 60. Módulo colegio: Interfaz de registrar escolar

Fuente: Elaboración propia

3.7.1.4. Codificación

3.7.1.4.1. Clase: Apoderado_Model

```
<?php
class Apoderado_Model extends Model
{
    public function __construct()
    {
        parent::__construct();
    }
    public function Listar()
    {
        if($_SESSION["TIPO"]=="Colegio") {
            $id=$_SESSION["IDCOLEGIO"];
            return $this->db->query("CALL
            SP_LISTAR_APODERADOS ($id) ");
        } else {
            $id=0;
            return $this->db->query("CALL
            SP_LISTAR_APODERADOS ($id) ");
        }
    }
    public function Buscar_Usuario()
    {
        $id = $_POST['id'];
        $sth = $this->db->prepare("CALL
        SP_BUSCAR_USUARIO ($id, 'Apoderado' )");
        $sth->execute();
    }
}
```

```

        $data = $sth->fetch();
        echo json_encode($data);
    }
    public function Buscar_Apoderado()
    {
        $id = $_POST['id'];
        $sth = $this->db->prepare("CALL
        SP_BUSCAR_APODERADO($id)");
        $sth->execute();
        $data = $sth->fetch();
        echo json_encode($data);
    }
    public function Reporte_General()
    {
        $id=0;
        return $this->db->query("CALL
        SP_LISTAR_APODERADOS($id)");
    }
    public function Insertar()
    {
        $nombre          = $_POST['nombres'];
        $paterno          = $_POST['paterno'];
        $materno          = $_POST['materno'];
        $tipodocumento   = $_POST['tipo'];
        $numero           = $_POST['numero'];
        $estadocivil      = $_POST['estadocivil'];
        $sexo             = $_POST['sexo'];
        $fecha            = $_POST['fecha'];
        $ocupacion        = $_POST['ocupacion'];
        $celular          = $_POST['celular'];
        $telefono         = $_POST['telefono'];
        $email            = $_POST['email'];
        $idcole           = $_SESSION["IDCOLEGIO"];
        $tipouser         = "Apoderado";
        $idcolegio        = $_SESSION["IDCOLEGIO"];
        $sql = "CALL SP_INSERTAR_APODERADO(:nom,:pat,:mat,:tip,
        :num,:est,:sex,:fec,:ocu,:cel,:tel,:ema,:idc,:tipusu,
        :idcol,@MENSAJE)";
        $stm = $this->db->prepare($sql);
        $stm->bindParam(':nom', $nombre, PDO::PARAM_STR, 100);
        $stm->bindParam(':pat', $paterno, PDO::PARAM_STR, 100);
        $stm->bindParam(':mat', $materno, PDO::PARAM_STR, 100);
        $stm->bindParam(':tip', $tipodocumento, PDO::PARAM_STR,
        100);
        $stm->bindParam(':num', $numero, PDO::PARAM_STR, 100);
        $stm->bindParam(':est', $estadocivil, PDO::PARAM_STR,
        100);
        $stm->bindParam(':sex', $sexo, PDO::PARAM_STR, 100);
    }

```

```

$stmt->bindParam(':fec', $fecha, PDO::PARAM_STR, 100);
$stmt->bindParam(':ocu', $ocupacion, PDO::PARAM_STR,
100);
$stmt->bindParam(':cel', $celular, PDO::PARAM_STR, 100);
$stmt->bindParam(':tel', $telefono, PDO::PARAM_STR,
100);
$stmt->bindParam(':ema', $email, PDO::PARAM_STR, 100);
$stmt->bindParam(':idc', $idcole, PDO::PARAM_STR, 100);
$stmt->bindParam(':tipusu', $tipouser, PDO::PARAM_STR,
100);
$stmt->bindParam(':idcol', $idcolegio, PDO::PARAM_STR,
100);
$stmt->execute(); $stmt->closeCursor();
$msg = $this->db->query("SELECT @MENSAJE AS mensaje")
->fetch(PDO::FETCH_ASSOC);
$msgaje = sprintf($msg["mensaje"]);
echo json_encode($mensaje);
}

public function Actualizar()
{
    $idapoderado = $_POST['idapoderado'];
    $nombre = $_POST['nombres'];
    $paterno = $_POST['paterno'];
    $materno = $_POST['materno'];
    $tipodocumento = $_POST['tipo'];
    $numero = $_POST['numero'];
    $estadocivil = $_POST['estadocivil'];
    $sexo = $_POST['sexo'];
    $fecha = $_POST['fecha'];
    $ocupacion = $_POST['ocupacion'];
    $celular = $_POST['celular'];
    $telefono = $_POST['telefono'];
    $email = $_POST['email'];
    $idcole = $_SESSION["IDCOLEGIO"];
    $sql = "CALL SP_ACTUALIZAR_APODERADO(:idapo,:nom,:pat,
:mat,:tip,:num,:est,:sex,:fec,:ocu,:cel,:tel,:ema,:idc,
@MENSAJE) ";
    $stmt = $this->db->prepare($sql);
    $stmt->bindParam(':idapo', $idapoderado, PDO::PARAM_STR,
100);
    $stmt->bindParam(':nom', $nombre, PDO::PARAM_STR, 100);
    $stmt->bindParam(':pat', $paterno, PDO::PARAM_STR, 100);
    $stmt->bindParam(':mat', $materno, PDO::PARAM_STR, 100);
    $stmt->bindParam(':tip', $tipodocumento, PDO::PARAM_STR,
100);
    $stmt->bindParam(':num', $numero, PDO::PARAM_STR, 100);
    $stmt->bindParam(':est', $estadocivil, PDO::PARAM_STR,
100);

```

```

        $stm->bindParam(':sex', $sexo, PDO::PARAM_STR, 100);
        $stm->bindParam(':fec', $fecha, PDO::PARAM_STR, 100);
        $stm->bindParam(':ocu', $ocupacion, PDO::PARAM_STR,
100);
        $stm->bindParam(':cel', $celular, PDO::PARAM_STR, 100);
        $stm->bindParam(':tel', $telefono, PDO::PARAM_STR,
100);
        $stm->bindParam(':ema', $email, PDO::PARAM_STR, 100);
        $stm->bindParam(':idc', $idcole, PDO::PARAM_STR, 100);
        $stm->execute(); $stm->closeCursor();
        $ms = $this->db->query("SELECT @MENSAJE AS mensaje")
->fetch(PDO::FETCH_ASSOC);
        $mensaje = sprintf($ms["mensaje"]);
        echo json_encode($mensaje);
    }
}

```

La clase `Apoderado_Model` extiende de la clase base `Model` y contiene los siguientes métodos: `Listar()`, `Buscar_Usuario()`, `Buscar_Apoderado()`, `Reporte_General()`, `Insertar()`, `Actualizar()`. Estos métodos sirven para realizar consultas a la base datos MySQL.

El método `Listar()`, realiza la llamada al procedimiento almacenado `SP_LISTAR_APODERADOS` y retorna el listado de apoderados.

El método `Buscar_Usuario()`, captura el id del usuario que proviene del método `POST`, realiza la llamada al procedimiento almacenado `SP_BUSCAR_USUARIO`, retorna la consulta y la convierte en formato JSON.

El método `Buscar_Apoderado()`, captura el id del apoderado que proviene del método `POST`, realiza la llamada al procedimiento almacenado `SP_BUSCAR_APODERADO`, retorna la consulta y la convierte en formato JSON.

El método `Reporte_General()`, realiza la llamada al procedimiento almacenado `SP_LISTAR_APODERADOS` y retorna el listado de apoderados.

El método `Insertar()`, captura los datos que provienen del método `POST`, realiza la llamada al procedimiento almacenado `SP_INSERTAR_APODERADO` que contiene un parametro de salida, retorna la consulta y la convierte en formato JSON.

El método Actualizar(), captura los datos que provienen del método POST, realiza la llamada al procedimiento almacenado SP_ACTUALIZAR_APODERADO que contiene un parametro de salida, retorna la consulta y la convierte en formato JSON.

3.7.1.4.2. Clase: Escolar_Model

```
<?php
error_reporting(0);
class Escolar_Model extends Model
{
    public function __construct()
    {
        parent::__construct();
    }
    public function Insertar()
    {
        if ($_FILES['imagen']['size'] != 0 &&
            $_FILES['imagen']['error'] == 0) {
            $Imagen = $_FILES['imagen'];
            $file_name = $Imagen['name'];
            $file_size = $Imagen['size'];
            $extension = explode(".", $file_name);
            $num = count($extension) - 1;
            $random = (date('d:m:y:h:i') * rand() * 9999999);
            $ruta_img = './public/img/escolar/' . $random .
                '.' . $extension[$num];
            if(move_uploaded_file($_FILES['imagen']
                ['tmp_name'], $ruta_img)) {
                $nombre_img = $random . '.' .
                    $extension[$num];
            } else {
                $nombre_img = 'escolar.png';
            }
        } else {
            $nombre_img = 'escolar.png';
        }
        $tipodoc = $_POST["tipodoc"];
        $dni = $_POST["dni"];
        $nombres = $_POST["nombres"];
        $paterno = $_POST["paterno"];
        $materno = $_POST["materno"];
        $fecha = $_POST["fecha"];
        $sexo = $_POST["sexo"];
        $foto = $nombre_img;
        $padre = $_POST["padre"];
        $madre = $_POST["madre"];
        $idapod = $_POST["idapoderado"];
```

```

        $idcole = $_SESSION["IDCOLEGIO"];
        $sql = "CALL SP_INSERTAR_ESCOLAR(:tipodoc,:dni,:nom,
        :pat,:mat,:fec,:sex,:fot,:pad,:mad,:ida,:idc,@MENSAJE)";
        $stm = $this->db->prepare($sql);
        $stm->bindParam(':tipodoc', $tipodoc,
        PDO::PARAM_STR,100);
        $stm->bindParam(':dni', $dni, PDO::PARAM_STR, 100);
        $stm->bindParam(':nom', $nombres, PDO::PARAM_STR, 100);
        $stm->bindParam(':pat', $paterno, PDO::PARAM_STR, 100);
        $stm->bindParam(':mat', $materno, PDO::PARAM_STR, 100);
        $stm->bindParam(':fec', $fecha, PDO::PARAM_STR, 100);
        $stm->bindParam(':sex', $sexo, PDO::PARAM_STR, 100);
        $stm->bindParam(':fot', $foto, PDO::PARAM_STR, 100);
        $stm->bindParam(':pad', $padre, PDO::PARAM_STR, 100);
        $stm->bindParam(':mad', $madre, PDO::PARAM_STR, 100);
        $stm->bindParam(':ida', $idapod, PDO::PARAM_STR, 100);
        $stm->bindParam(':idc', $idcole, PDO::PARAM_STR, 100);
        $stm->execute();
        $stm->closeCursor();
        $ms = $this->db->query("SELECT @MENSAJE AS mensaje")
        ->fetch(PDO::FETCH_ASSOC);
        $mensaje = sprintf($ms["mensaje"]);
        echo json_encode($mensaje);
    }
}

```

La clase `Escolar_Model` extiende de la clase base `Model`, contiene un método `Insertar()`, captura los datos que provienen del método `POST`, realiza la llamada al procedimiento almacenado `SP_INSERTAR_ESCOLAR` que contiene un parametro de salida, retorna la consulta y la convierte en formato `JSON`.

3.7.1.4.3. Clase: Apoderado

```

<?php
class Apoderado extends Controller
{
    public function __construct()
    {
        parent::__construct();
        Session::init();
        $slogged = Session::get('loggedIn');
        if ($slogged == false) {
            Session::destroy();
            header('location: ../login');
            exit;
        }
    }
}

```

```

        $this->view->js = array('apoderado/js/apoderado.js');
    }
    public function index()
    {
        $this->view->render('apoderado/index');
    }
    public function Listar_Apoderado()
    {
        $this->view->Listar = $this->model->Listar();
        $this->view->render('apoderado/listado', true);
    }
    public function Buscar_Apoderado()
    {
        $this->model->Buscar_Apoderado();
    }
    public function Buscar_Usuario()
    {
        $this->model->Buscar_Usuario();
    }
    public function Insertar_Apoderado()
    {
        $this->model->Insertar();
    }
    public function Actualizar_Apoderado()
    {
        $this->model->Actualizar();
    }
    public function Reporte_Apoderado()
    {
        $this->Listar = $this->model->Reporte_General();
        $this->view->render('require
        'views/apoderado/reportes/Reporte_general.php');
    }
}

```

La clase Apoderado extiende de la clase base Controller, contiene el método: index(), Listar_Apoderado(), Buscar_Apoderado(), Buscar_Usuario(), Insertar_Apoderado(), Actualizar_Apoderado(), Reporte_Apoderado(). Estos métodos sirven para recibir las peticiones que realiza la vista.

3.7.1.4.4. Clase: Escolar

```

<?php
class Escolar extends Controller
{
    public function Insertar_Escolar()

```

```

    {
        $this->model->Insertar();
    }
}

```

La clase Escolar extiende de la clase base Controller, contiene el método: Insertar_Escolar(). Este métodos sirve para recibir la petición que realiza la vista.

3.7.1.4.5. Vista apoderado/index: index.php

```

<div class="card shadow mb-4">
    <div class="card-body">
        <?php if ($_SESSION["TIPO"] == "Admin"): ?>
            <h6 class="m-0 font-weight-bold text-primary text-center">REPORTE DE APODERADOS</h6>
        <?php else: ?>
            <h6 class="m-0 font-weight-bold text-primary text-center">LISTADO DE APODERADOS</h6>
        <?php endif; ?>
        <?php if ($_SESSION["TIPO"] == "Admin"): ?>
            <a href="apoderado/Reporte_Apoderado/" target="_blank" class="btn btn-primary btn-sm">
                <i></i> Generar PDF</a>
        <?php endif; ?>
        <?php if ($_SESSION["TIPO"]=="Colegio"): ?>
            <button class="btn btn-primary btn-sm float-left" data-toggle="modal" data-target="#nuevo"><span
                class="fa fa-plus-circle"></span>&nbsp; NUEVO
                APODERADO</button>
        <?php endif; ?>
    <div id="apoderado"></div>
</div>
</div>

<!-- ventana modal nuevo apoderado -->
<div class="modal fade" id="nuevo" tabindex="-1" role="dialog"
    aria-labelledby="myExtraLargeModalLabel" aria-hidden="true">
    <div class="modal-dialog modal-xl">
        <div class="modal-content">
            <div class="modal-header">
                <h6 class="modal-title" id="exampleModalLabel">
                    REGISTRAR DATOS DEL APODERADO</h6>
                <button type="button" class="close" data-dismiss="modal" arial-label="Close">
                    <span aria-hidden="true">&times;</span></button>
            </div>
            <div class="modal-body">
                <form action="<?php echo URL;?>apoderado/Insertar_Apoderado/" name="frmnuevoapoderado"
                    id="frmnuevoapoderado" method="post">

```

```

<div class="modal-body">
  <div class="form-group row">
    <label for="inputPassword"
      class="col-sm-2 col-form-label">
      Nombres (*) :</label>
    <div class="col-sm-10">
      <input type="text"
        class="form-control"
        name="nombres" id="lnombres"
        required autocomplete="off"
        onkeypress="return
          soloLetras(event)">
    </div>
  </div>
  <div class="form-group row">
    <label for="inputPassword"
      class="col-sm-2 col-form-label">
      A. Paterno (*) :</label>
    <div class="col-sm-4">
      <input type="text"
        class="form-control"
        name="paterno" id="lpaterno"
        autocomplete="off" required
        onkeypress="return
          soloLetras(event) ;" required
        placeholder="">
    </div>
    <label for="inputPassword"
      class="col-sm-2 col-form-label">
      A. Materno (*) :</label>
    <div class="col-sm-4">
      <input type="text"
        class="form-control"
        name="materno" id="lmaterno"
        autocomplete="off" required
        onkeypress="return
          soloLetras(event) ;" required
        placeholder="">
    </div>
  </div>
  <div class="form-group row">
    <label for="inputPassword"
      class="col-sm-2 col-form-label">
      Tipo Documento (*) :</label>
    <div class="col-sm-4">
      <select name="tipo" id="ltipo"
        class="form-control cat"
        required
        onchange="getval(this) ;">
        <option value=""
          selected="Seleccione Una
            Opción">Seleccione una
            Opción</option>
    </div>
  </div>

```

```

        <option
        value="L.E/D.N.I">L.E/D.N.
        I
        </option>
        <option value="Carnet de
        Extranjeria">Carnet de
        Extranjeria</option>
    </select>
</div>
<label for="inputPassword"
class="col-sm-2 col-form-label">
N° Documento (*):</label>
<div class="col-sm-4">
    <input type="text"
    class="form-control"
    name="numero" id="lnumero"
    onkeypress="return
    esInteger(event) ;"
    placeholder=""
    autocomplete="off">
</div>
</div>
<div class="form-group row">
    <label for="inputPassword"
    class="col-sm-2 col-form-label">
    Sexo (*):</label>
    <div class="col-sm-4">
        <select name="sexo" id=""
        class="form-control cat"
        required>
            <option value=""
            selected="Seleccione Una
            Opción">Seleccione una
            Opción</option>
            <option value="H">
            Hombre</option>
            <option value="M">
            Mujer</option>
        </select>
    </div>
    <label for="inputPassword"
    class="col-sm-2 col-form-label">
    Estado Civil (*):</label>
    <div class="col-sm-4">
        <select name="estadocivil"
        id="" class="form-control cat"
        required>
            <option value=""
            selected="Seleccione Una
            Opción">Seleccione una
            Opción</option>
            <option value="S">
            Soltero</option>

```

```

        <option value="C">
            Casado</option>
        <option value="V">
            Viudo</option>
        <option value="D">
            Divorciado</option>
    </select>
</div>
</div>
<div class="form-group row">
    <label for=""
        class="col-sm-2 col-form-label">
        F. Nacimiento(*):</label>
    <div class="col-sm-4">
        <input type="date"
            class="form-control"
            name="fecha" id="lfecha">
    </div>
    <label for=""
        class="col-sm-2 col-form-label">
        Ocupación(*):</label>
    <div class="col-sm-4">
        <input type="text"
            class="form-control"
            name="ocupacion"
            id="locupacion" required
            placeholder=""
            autocomplete="off">
    </div>
</div>
<div class="form-group row">
    <label for=""
        class="col-sm-2 col-form-label">
        Celular(*):</label>
    <div class="col-sm-4">
        <input type="text"
            class="form-control"
            onkeypress="return
            esInteger(event) ;"
            name="celular" id="lcelular"
            maxlength="9" required
            placeholder=""
            autocomplete="off">
    </div>
    <label for=""
        class="col-sm-2 col-form-label">
        Telefono:</label>
    <div class="col-sm-4">
        <input type="text"
            class="form-control"
            name="telefono"
            onkeypress="return
            esInteger(event) ;"

```

```

        maxlength="6" id="ltelefono"
        placeholder=""
        autocomplete="off">
    </div>
</div>
<div class="form-group row">
    <label for=""
        class="col-sm-2 col-form-label">
        Email:</label>
    <div class="col-sm-10">
        <input type="email"
            class="form-control"
            name="email" id="lemail"
            autocomplete="off">
    </div>
</div>
<p class="text-success">
    (*) Campo Obligatorio</p>
</div>
<div class="mensaje"></div>
<div class="modal-footer">
    <button class="btn btn-primary"
        type="submit">
        GUARDAR</button>
    <button class="btn btn-secondary"
        data-dismiss="modal">CERRAR</button>
</div>
</form>
</div>
</div>
</div>

<!-- ventana modal actualizar apoderado -->
<div class="modal fade" id="update" tabindex="-1" role="dialog"
    aria-labelledby="myExtraLargeModalLabel" aria-hidden="true">
    <div class="modal-dialog modal-xl">
        <div class="modal-content">
            <div class="modal-header">
                <h6 class="modal-title" id="exampleModalLabel">
                EDITAR DATOS DEL APODERADO</h6>
                <button type="button" class="close"
                    data-dismiss="modal" arial-label="Close">
                    <span aria-hidden="true">&times;</span>
                </button>
            </div>
            <form action="<?php echo URL; ?>apoderado/
                Actualizar_Apoderado/" name="frmupdateapoderado"
                id="frmupdateapoderado" method="post">
                <div class="modal-body">
                    <div class="form-group row">
                        <label for="inputPassword"
                            class="col-sm-2 col-form-label">
                            Nombres (*):</label>

```

```

<div class="col-sm-10">
    <input type="hidden"
        name="idapoderado"
        id="idapoderado">
    <input type="text"
        class="form-control"
        name="nombres" id="nombres"
        autocomplete="off" required
        onkeypress="return
            soloLetras(event) ;">
    </div>
</div>
<div class="form-group row">
    <label for=""
        class="col-sm-2 col-form-label">
A. Paterno(*) :</label>
    <div class="col-sm-4">
        <input type="text"
            class="form-control"
            name="paterno" id="paterno"
            autocomplete="off" required
            onkeypress="return
                soloLetras(event) ;" >
    </div>
    <label for=""
        class="col-sm-2 col-form-label">
A. Materno(*) :</label>
    <div class="col-sm-4">
        <input type="text"
            class="form-control"
            name="materno" id="materno"
            autocomplete="off" required>
    </div>
</div>
<div class="form-group row">
    <label for=""
        class="col-sm-2 col-form-label">
Tipo Documento (*) :</label>
    <div class="col-sm-4">
        <select name="tipo" id="tipo"
            class="form-control cat"
            required onchange=
                "getvalupdateapo(this)">
            <option value=""
                selected="Seleccione Una
                    Opción">Seleccione una
                    Opción</option>
            <option
                value="L.E/D.N.I">
                L.E/D.N.I</option>
            <option value="Carnet de
                Extranjeria">Carnet de
                Extranjeria</option>
    </div>
</div>

```

```

        </select>
    </div>
    <label for=""
class="col-sm-2 col-form-label">
N° Documento (*) :</label>
    <div class="col-sm-4">
        <input type="text"
class="form-control"
name="numero" id="numero"
required autocomplete="off"
onkeypress="return
esInteger(event) ;">
    </div>
</div>
<div class="form-group row">
    <label for=""
class="col-sm-2 col-form-label">
Sexo (*) :</label>
    <div class="col-sm-4">
        <select name="sexo" id="sexo"
class="form-control cat"
required>
            <option value=""
selected="Seleccione Una
Opción">Seleccione una
Opción</option>
            <option value="H">
Hombre</option>
            <option value="M">
Mujer</option>
        </select>
    </div>
    <label for=""
class="col-sm-2 col-form-label">
Estado Civil (*) :</label>
    <div class="col-sm-4">
        <select name="estadocivil"
id="estadocivil" class="form-
control cat" required>
            <option value=""
selected="Seleccione Una
Opción">Seleccione una
Opción</option>
            <option value="S">
Soltero</option>
            <option value="C">
Casado</option>
            <option value="V">
Viudo</option>
            <option value="D">
Divorciado</option>
        </select>
    </div>

```

```

</div>
<div class="form-group row">
  <label for=""
    class="col-sm-2 col-form-label">
F. Nacimiento(*):</label>
  <div class="col-sm-4">
    <input type="date"
      class="form-control"
      name="fecha" id="fecha">
  </div>
  <label for=""
    class="col-sm-2 col-form-label">
Ocupación(*):</label>
  <div class="col-sm-4">
    <input type="text"
      class="form-control"
      name="ocupacion" id="ocupacion"
      required>
  </div>
</div>
<div class="form-group row">
  <label for=""
    class="col-sm-2 col-form-label">
Celular(*):</label>
  <div class="col-sm-4">
    <input type="text"
      class="form-control"
      name="celular" id="celular"
      required maxlength="9"
      onkeypress="return
        esInteger(eevent) ;">
  </div>
  <label for=""
    class="col-sm-2 col-form-label">
Telefono:</label>
  <div class="col-sm-4"
    onkeypress="return esInteger(event) ">
    <input type="text"
      class="form-control"
      name="telefono" id="telefono"
      maxlength="6"
      onkeypress="return
        esInteger(event) ;">
  </div>
</div>
<div class="form-group row">
  <label for=""
    class="col-sm-2 col-form-label">
Email:</label>
  <div class="col-sm-10">
    <input type="email"
      class="form-control"
      name="email" id="email">

```

```

        </div>
    </div>
    <p class="text-success">
        (*) Campo Obligatorio</p>
    <div class="mensaje"></div>
</div>
<div class="modal-footer">
    <button class="btn btn-primary"
        type="submit">
        ACTUALIZAR</button>
    <button class="btn btn-secondary"
        data-dismiss="modal">CERRAR</button>
</div>
</form>
</div>
</div>
</div>

<!-- ventana modal nuevo escolar -->
<div class="modal fade" id="nuevoescolar" tabindex="-1" role="dialog"
    aria-labelledby="myExtraLargeModalLabel" aria-hidden="true">
    <div class="modal-dialog modal-xl">
        <div class="modal-content">
            <div class="modal-header">
                <h6 class="modal-title" id="exampleModalLabel">
                    REGISTRAR DATOS DEL ESCOLAR</h6>
                <button type="button" class="close"
                    data-dismiss="modal" arial-label="Close">
                    <span aria-hidden="true">&times;</span>
                </button>
            </div>
            <form action="" name="frmnuevoescolar"
                id="frmnuevoescolar" method="post">
                <div class="modal-body">
                    <div class="form-group row">
                        <label for=""
                            class="col-sm-2 col-form-label">
                            Nombres (*)</label>
                        <div class="col-sm-10">
                            <input type="text"
                                class="form-control"
                                name="nombres" id="nom"
                                autocomplete="off" required
                                placeholder=""
                                onkeypress="return
                                    soloLetras(event)">
                        </div>
                    </div>
                    <div class="form-group row">
                        <label for=""
                            class="col-sm-2 col-form-label">
                            A. Paterno (*)</label>
                        <div class="col-sm-4">

```

```

        <input type="text"
            class="form-control"
            name="paterno" id="ap"
            autocomplete="off" required
            placeholder=""
            onkeypress="return
                soloLetras(event)">
    </div>
<label for=""
    class="col-sm-2 col-form-label">
A. Materno (*) :</label>
<div class="col-sm-4">
    <input type="text"
        class="form-control"
        name="materno" id="am"
        autocomplete="off" required
        placeholder=""
        onkeypress="return
            soloLetras(event)">
</div>
</div>
<div class="form-group row">
    <input type="hidden"
        name="idapoderado" id="idop">
    <label for=""
        class="col-sm-2 col-form-label">
Tipo Documento (*) :</label>
    <div class="col-sm-4">
        <select name="tipodoc"
            id="ltipo" class="form-control
            cat" required
            onchange="getvalEsc(this)">
            <option value=""
                selected="Seleccione Una
                Opción">Seleccione una
                Opción</option>
            <option value="L.E/D.N.I">
                L.E/D.N.I</option>
            <option value="Carnet de
                Extranjeria">Carnet de
                Extranjeria</option>
        </select>
    </div>
    <label for=""
        class="col-sm-2 col-form-label">
Nº. Documento (*)</label>
    <div class="col-sm-4">
        <input type="text"
            class="form-control" name="dni"
            id="dn" autocomplete="off"
            required maxlength="8"
            placeholder=""

```

```

        onkeypress="return
        esInteger(event) ">
    </div>
</div>
<div class="form-group row">
    <label for=""
    class="col-sm-2 col-form-label">
F. Nacimiento(*):</label>
    <div class="col-sm-4">
        <input type="date"
        class="form-control"
        name="fecha" id="fe"
        autocomplete="off" required>
    </div>
    <label for=""
    class="col-sm-2 col-form-label">
Sexo(*):</label>
    <div class="col-sm-4">
        <select name="sexo" id=""
        class="form-control cat">
            <option value=""
            selected="Seleccione Una
            Opción">Seleccione una
            Opción</option>
            <option value="H">
            Hombre</option>
            <option value="M">
            Mujer</option>
        </select>
    </div>
</div>
<div class="form-group row">
    <label for=""
    class="col-sm-2 col-form-label">
Nombre Padre(*):</label>
    <div class="col-sm-10">
        <input type="text"
        class="form-control"
        name="padre" id="np"
        autocomplete="off" required
        placeholder=""
        onkeypress="return
        soloLetras(event) ">
    </div>
</div>
<div class="form-group row">
    <label for=""
    class="col-sm-2 col-form-label">
Nombre. Madre(*):</label>
    <div class="col-sm-10">
        <input type="text"
        class="form-control"
        name="madre" id="nm"

```

```

        autocomplete="off" required
        placeholder=""
        onkeypress="return
        soloLetras(event) ">
    </div>
</div>
<div class="form-group row">
    <label for=""
    class="col-sm-2 col-form-label">
        Foto(*) :</label>
    <div class="col-sm-10">
        <input type="file"
        name="imagen" id="imgfoto">
    </div>
</div>
<p class="text-success">
    (*) Campo Obligatorio</p>
<div class="mensaje"></div>
</div>
<div class="modal-footer">
    <button class="btn btn-primary"
    type="submit">
        GUARDAR</button>
    <button class="btn btn-secondary"
    data-dismiss="modal">CERRAR</button>
</div>
</form>
</div>
</div>
</div>

```

El archivo index.php de la vista apoderado/index contiene lo siguiente:

- Ventanas modales de los formularios registrar datos del apoderado (ver Figura 57. Módulo colegio: Interfaz de registrar apoderado), formulario editar datos del apoderado (ver Figura 58. Módulo colegio: Interfaz de editar apoderado) y formulario registrar datos del escolar (ver figura 60. Módulo colegio: Interfaz de registrar escolar).

3.7.1.4.6. Vista apoderado/listado: listado.php

```
<!-- tabla con listado de apoderados -->
<table class="table-responsive-sm table-striped table-bordered
table-hover" width="100%" cellpadding="0" id="tbapoderado">
  <thead class="">
    <tr>
      <th class="text-center">N°</th>
      <th class="">APODERADO</th>
      <th class="">TIPO DOC.</th>
      <th class="">N° DOCUMENTO</th>
      <th class="">SEXO</th>
      <?php if($_SESSION["TIPO"]=="Admin"): ?>
        <th class="">COLEGIO</th>
      <?php endif; ?>
      <?php if($_SESSION["TIPO"]=="Colegio"): ?>
        <th class="text-center">OPCION</th>
      <?php endif; ?>
    </tr>
  </thead>
  <tbody>
    <?php $contador = 0;?>
    <?php foreach ($this->Listar as $key => $value): ?>
    <?php $contador++;?>
    <tr>
      <td class="text-center">
        <?php echo $contador; ?></td>
      <td class="">
        <?php echo $value['APODERADO']; ?></td>
      <td class="">
        <?php echo $value['TIPODOCUMENTO']; ?></td>
      <td class="">
        <?php echo $value['NDOCUMENTO']; ?></td>
      <td class="">
        <?=$value['SEXO']=="M" ? 'Mujer': 'Hombre' ?></td>
      <?php if($_SESSION["TIPO"]=="Admin"): ?>
        <td class="">
          <?php echo $value['NOMBRE']; ?></td>
        <?php endif; ?>
      <?php if($_SESSION["TIPO"]=="Colegio"): ?>
        <td class="text-center">
          <span class="btn btn-info btn-sm view"
            onclick="<?php echo $value['IDAPODERADO'];
            ?>" data-toggle="modal" data-target="#info"
            title="Ver más Información"><i class="fa
            fa-eye"></i></span>
          <span class="btn btn-warning btn-sm del"
            onclick="<?php echo $value['IDAPODERADO'];
            ?>" data-toggle="modal" data-
            target="#update" title="Editar Datos"><i
            class="fa fa-pencil-alt"></i></span>
          <span class="btn btn-secondary btn-sm esc"
            onclick="<?php echo $value['IDAPODERADO'];
```

```

        ?>" data-toggle="modal" data-
        target="#nuevoescolar" title="Agregar
        Escolar"><i class="fa fa-graduation-
        cap"></i></span>

        </td>
        <?php endif; ?>
    </tr>
    <?php endforeach; ?>
</tbody>
</table>

<!-- ventana modal ver información apoderado -->
<div class="modal fade" id="info" tabindex="-1" role="dialog"
aria-labelledby="myExtraLargeModalLabel" aria-hidden="true">
    <div class="modal-dialog modal-lg">
        <div class="modal-content">
            <div class="modal-header">
                <h6 class="modal-title"
                id="exampleModalLabel">INFORMACION GENERAL DEL
                APODERADO</h6>
                <button type="button" class="close"
                data-dismiss="modal" aria-label="Close">
                    <span aria-hidden="true">&times;</span>
                </button>
            </div>
            <div class="modal-body">
                <img src="" alt="" id="foto"
                class="float-right" width="120px" height="120px">
                <dl class="row">
                    <dt class="col-sm-4">Nombres</dt>
                    <dd class="col-sm-8" id="dnombres"></dd>
                    <dt class="col-sm-4">Tipo Documento</dt>
                    <dd class="col-sm-8" id="dtipo"></dd>
                    <dt class="col-sm-4">N° Documento</dt>
                    <dd class="col-sm-8" id="dnumero"></dd>
                    <dt class="col-sm-4">Estado Civil</dt>
                    <dd class="col-sm-8" id="destado"></dd>
                    <dt class="col-sm-4">Sexo</dt>
                    <dd class="col-sm-8" id="dsexo"></dd>
                    <dt class="col-sm-4">F. Nacimiento</dt>
                    <dd class="col-sm-8" id="dfecha"></dd>
                    <dt class="col-sm-4">Ocupación</dt>
                    <dd class="col-sm-8" id="docupacion"></dd>
                    <dt class="col-sm-4">Celuar</dt>
                    <dd class="col-sm-8" id="dcelular"></dd>
                    <dt class="col-sm-4">Telefono</dt>
                    <dd class="col-sm-8" id="dtelefono"></dd>
                    <dt class="col-sm-4">Email</dt>
                    <dd class="col-sm-8" id="demail"></dd>
                </dl>
            </div>
            <div class="modal-footer">
                <button type="button" class="btn btn-secondary"
                data-dismiss="modal">Cerrar</button>
            </div>
        </div>
    </div>
</div>

```

</div>

<script>

```
$("#del").on("click",function(){
    let id=$(this).attr("onclick");
    $.post('apoderado/Buscar_Apoderado',{'id':id},function(r){
        $("#idapoderado").val(r.IDAPODERADO);
        $("#nombres").val(r.NOMBRES);
        $("#paterno").val(r.APELLIDOPATERNO);
        $("#materno").val(r.APELLIDOMATERNO);
        if (r.TIPODOCUMENTO=="L.E/D.N.I") {
            document.getElementById("numero")
                .setAttribute("maxlength", 8);
        } else {
            document.getElementById("numero")
                .setAttribute("maxlength", 11);
        }
        $("#tipo").val(r.TIPODOCUMENTO);
        $("#numero").val(r.NDOCUMENTO);
        $("#estadocivil").val(r.ESTADOCIVIL);
        $("#sexo").val(r.SEXO);
        $("#fecha").val(r.FECHANACIMIENTO);
        $("#ocupacion").val(r.OCUPACION);
        $("#celular").val(r.CELULAR);
        $("#telefono").val(r.TELEFONO);
        $("#email").val(r.EMAIL);
    },'json');
});
$("#esc").on("click",function(){
    let id=$(this).attr("onclick");
    $("#idop").val(id);
});
```

</script>

<script>

```
$("#view").on("click",function() {
    $("#titulo").empty();
    $("#dato").empty();
    let id=$(this).attr("onclick");
    $.post("apoderado/Buscar_Apoderado",{id:id},function(v) {
        let sex,est;
        if(v.SEXO=="M"){
            sex="Mujer";
        } else {
            sex="Hombre";
        }
        if(v.ESTADOCIVIL=="C") {
            est="Casado";
        } else if(v.ESTADOCIVIL=="S") {
            est="Soltero";
        } else if(v.ESTADOCIVIL=="D") {
            est="Divorciado";
        } else {
```

```

        est="Viudo";
    }
    $('#dnombres').html('<dd>+': '+v.NOMBRES+
    ' '+v.APELLIDOPATERNO +' '+
    v.APELLIDOMATERNO+'</dd>');
    $('#dtipo').html('<dd>+': '+v.TIPODOCUMENTO +</dd>');
    $('#dnumero').html('<dd>+': '+v.NDOCUMENTO +</dd>');
    $('#destado').html('<dd>+': '+est+'</dd>');
    $('#dsexo').html('<dd>+': '+sex+'</dd>');
    $('#dfecha').html('<dd>+': '+
    formato(v.FECHANACIMIENTO)+'</dd>');
    $('#docupacion').html('<dd>+': '+
    v.OCUPACION +'</dd>');
    $('#dcelular').html('<dd>+': '+v.CELULAR +'</dd>');
    $('#dtelefono').html('<dd>+': '+v.TELEFONO +'</dd>');
    $('#demail').html('<dd>+': '+v.EMAIL +'</dd>');
    }, "json");
});
function formato( fecha ) {
    return
    fecha.replace(/^(\\d{4})-(\\d{2})-(\\d{2})$/g, '$3/$2/$1');
}
</script>

```

El archivo listado.php de la vista apoderado/listado contiene lo siguiente:

- Tabla con estilo de Bootstrap para mostrar el listado de apoderados (ver Figura 56. Módulo colegio: Interfaz de listado de apoderados).
- Ventana modal para visualizar información general del apoderado (ver Figura 59. Módulo colegio: Interfaz de ver información del apoderado).
- Archivos javaScript con jQuery, Ajax y JSON.

3.7.1.4.7. javaScript: apoderado/js/apoderado.js

```

$(function() {
    listado();
    function listado() {
        $.post('apoderado/Listar_Apoderado', function(w) {
            $('#apoderado').html(w);
        });
    }
    function limpiar() {
        $('#lnombres').val("");
        $('#lpaterno').val("");
        $('#lmaterno').val("");
        $('#lnumero').val("");
        $('#lfecha').val("");
        $('#lcelular').val("");
        $('#ltelefono').val("");
    }

```

```

        $("#lemail").val("");
        $("#locupacion").val("");
        $("#lusuario").val("");
        $("#lclave").val("");
    }
    function limpiar_cajas_escolar() {
        $("#nom").val("");
        $("#ap").val("");
        $("#am").val("");
        $("#dn").val("");
        $("#fe").val("");
        $("#np").val("");
        $("#nm").val("");
        $("#imgfoto").val("");
    }
    $('#frmNuevoapoderado').submit(function() {
        var url=$(this).attr('action');
        var data=$(this).serialize();
        $.post(url,data,function(e) {
            if (e=='DATOS DEL APODERADOS REGISTRADOS CORRECTAMENTE') {
                $(".mensaje").html('<div class="alert alert-info" role="alert">'+e+'</div>');
                listado();
                $(".mensaje").show();
                $(".mensaje").fadeOut(4000);
                setTimeout(function() {
                    limpiar();
                    $('#nuevo').modal('hide');
                }, 3000);
            }else{
                $(".mensaje").html('<div class="alert alert-danger" role="alert">'+e+'</div>');
                $(".mensaje").show();
                $(".mensaje").fadeOut(5000);
            }
        },'json');
        return false;
    });
    $('#frmupdateapoderado').submit(function() {
        var url=$(this).attr('action');
        var data=$(this).serialize();
        $.post(url,data,function(e) {
            if (e=='DATOS DEL APODERADOS ACTUALIZADOS CORRECTAMENTE') {
                $(".mensaje").html('<div class="alert alert-info" role="alert">'+e+'</div>');
                listado();
                $(".mensaje").show();
                $(".mensaje").fadeOut(4000);
                setTimeout(function() {
                    $('#update').modal('hide');
                }, 3000);
            }
        });
    });

```

```

        }else{
            $(".mensaje").html('<div class="alert
            alert-danger" role="alert">'+e+'</div>');
            $(".mensaje").show();
            $(".mensaje").fadeOut(5000);
        }
    },'json');
    return false;
});
$("#frmnuevoescolar").on('submit', function(e) {
    e.preventDefault();
    $.ajax({
        type:'POST',
        url:'escolar/Insertar_Escolar/',
        data: new FormData(this),
        contentType: false, cache: false, processData:false,
        success: function(msg) {
            console.log(msg);
            var data=JSON.parse(msg);
            if (data=='DATOS DEL ESCOLAR REGISTRADOS
            CORRECTAMENTE') {
                $(".mensaje").html('<div class="alert
                alert-info
                alert-dismissible">'+data+'</div>');
                $(".mensaje").show();
                $(".mensaje").fadeOut(4000);
                setTimeout(function() {
                    limpiar_cajas_escolar()
                    $('#nuevoescolar').modal('hide');
                }, 3000);
            }else{
                $(".mensaje").html('<div class="alert
                alert-danger alert-
                dismissible">'+data+'</div>');
                $(".mensaje").show();
                $(".mensaje").fadeOut(5000);
            }
        }
    });
});
});

```

El archivo apoderado.js de la vista apoderado/js/ contiene:

- Función listado(): realiza el llamado al método Listar_Apoderado() del controlador Apoderado, retorna el listado de apoderados y se carga en un elemento HTML <div id="apoderado"></div> del archivo index.php.

- Función limpiar(), limpiar_cajas_escolar(): realiza la limpieza de los controles de los formulario nuevo apoderado y nuevo escolar.
- Código jQuery, Ajax y JSON de los formularios: Nuevo apoderado, actualizar apoderado y nuevo escolar. Los formularios envían peticiones a través del método POST de los controladores Apoderado y Escolar de los métodos Insertar_Apoderado(), Actualizar_Apoderado() e insertar_Escolar().

3.7.2. Generación de contratos

3.7.2.1. Tabla de base de datos

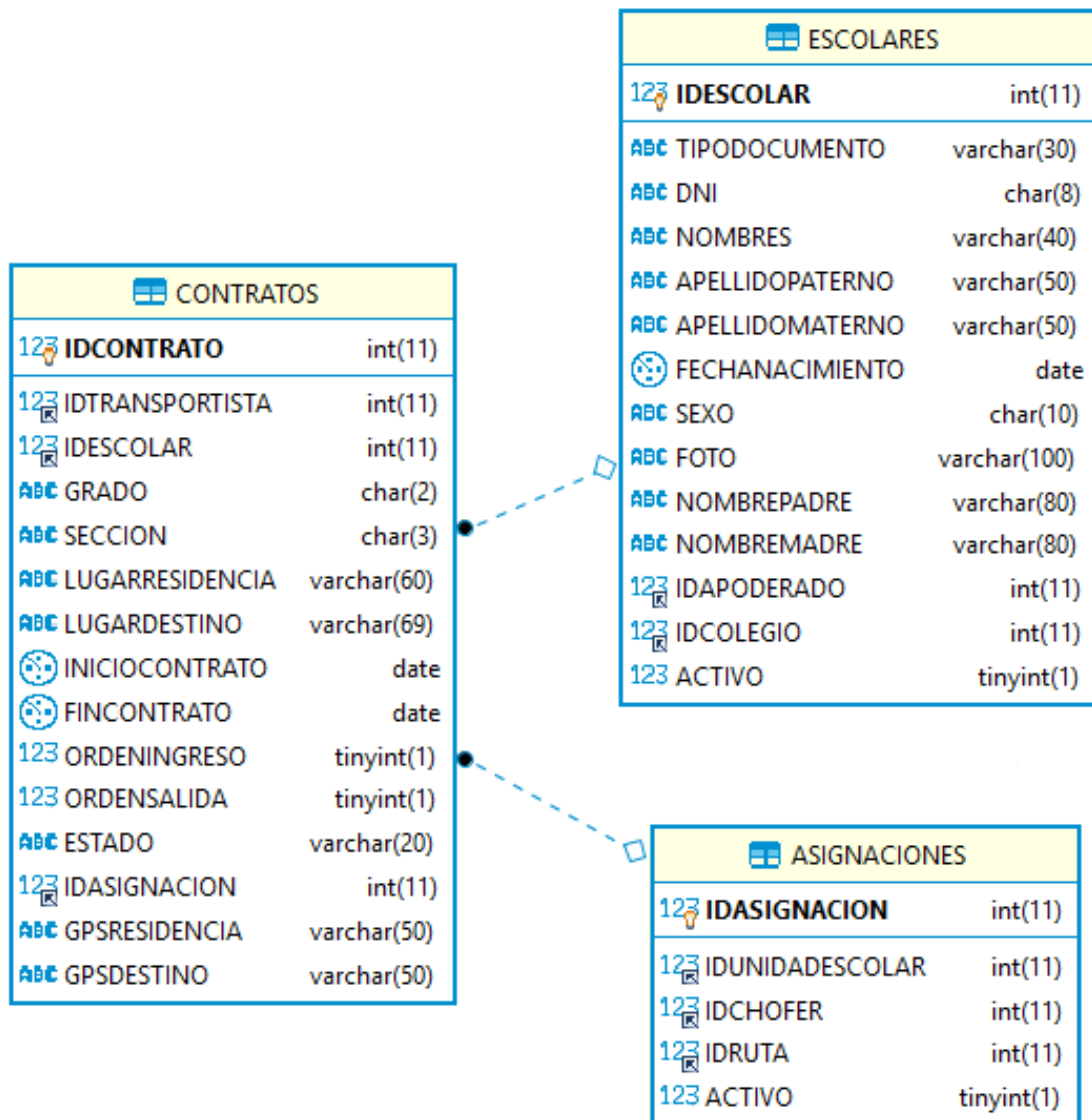


Figura 61. Tablas: CONTRATOS, ESCOLARES, ASIGNACIONES

Fuente: Elaboración propia

3.7.2.2. Procedimientos almacenados

```
-- INSERTAR CONTRATO --
DELIMITER $$
DROP PROCEDURE IF EXISTS SP_INSERTAR_CONTRATO$$
CREATE PROCEDURE SP_INSERTAR_CONTRATO
(P_IDTRANSPORTISTA INT,P_IDESCOLAR INT,P_GRADO CHAR(3) ,
P_SECCION CHAR(2) ,P_LUGARRESIDENCIA VARCHAR(60) ,
P_LUGARDESTINO VARCHAR(69) ,P_INICIOCONTRATO DATE,
P_FINCONTRATO DATE,P_IDASIGNACION INT,P_FECHA DATE,
```

```

OUT MENSAJE VARCHAR(100))
BEGIN
    IF (EXISTS (SELECT * FROM CONTRATOS
    WHERE IDESCOLAR=P_IDESCOLAR AND FINCONTRATO>P_FECHA) ) THEN
        SET MENSAJE='NO PUEDE REALIZAR LA OPERACION,
        EL ESCOLAR TIENE UN CONTRATO VIGENTE';
    ELSE
        INSERT INTO CONTRATOS (IDTRANSPORTISTA, IDESCOLAR, GRADO,
        SECCION, LUGARRESIDENCIA, LUGARDESTINO, INICIOCONTRATO,
        FINCONTRATO, IDASIGNACION)
        VALUES (P_IDTRANSPORTISTA, P_IDESCOLAR, P_GRADO, P_SECCION,
        P_LUGARRESIDENCIA, P_LUGARDESTINO, P_INICIOCONTRATO,
        P_FINCONTRATO, P_IDASIGNACION);
        SET MENSAJE='OPERACION REALIZADA CORRECTAMENTE';
    END IF;
END $$

```

3.7.2.3. Interfaz de usuario

The screenshot shows the MOVIESCOOL web application interface for transportists. On the left is a blue sidebar with the 'moviescool' logo and a menu with options: 'Mantenimiento', 'Monitoreo', 'Contratos', 'Ordenar Horario', and 'Configuración'. The main content area is titled 'LISTADO DE CONTRATOS' and features a 'NUEVO CONTRATO' button. Below this is a search bar and a table with 3 columns: 'N°', 'ESCOLAR', 'APODERADO', 'RUTA', 'F. CONTRATO', and 'OPCION'. The table lists 3 contracts. At the bottom of the table area, it says 'Mostrando registros del 1 al 3 de un total de 3 registros' and has navigation buttons for 'Anterior', '1', and 'Siguiente'. The footer of the page states 'Todos los derechos reservados © 2019 MOVIESCOOL'.

| N° | ESCOLAR | APODERADO | RUTA | F. CONTRATO | OPCION |
|----|--------------------------------|---------------------------------|---|-------------------------|---|
| 1 | Larissa Bonilla Lopez | Bernardo Amado Bonilla Coronado | Jorge Chavez N° 956-Jorge Chavez N° 956 | 04/04/2019 - 28/12/2019 |   |
| 2 | Ayzza Estrella Díaz Puestas | José Ricardo Díaz Hernandez | Jorge Chavez N° 680-Jorge Chavez N° 680 | 10/04/2019 - 28/12/2019 |   |
| 3 | Silvana del Pilar Supo Ramirez | Jhordy Akito Supo Oliden | Jorge Chavez N° 340-Jorge Chavez N° 340 | 06/12/2019 - 28/12/2019 |   |

Figura 62. Módulo transportista: Interfaz de listado de contratos

Fuente: Elaboración propia

REGISTRAR DATOS DEL CONTRATO

Escolar(*): Buscar

Grado(*): Seccion(*):

Residencia(*):

Destino(*):

Fecha de Inicio(*): Fecha Fin(*):

Chofer(*): Buscar

(*) Campo Obligatorio

GUARDAR CERRAR

Figura 63. Módulo transportista: Interfaz de registrar contrato

Fuente: Elaboración propia

LISTADO DE ESCOLARES

Mostrar 5 registros Buscar:

| N° | Ti | Dni | Ti | Escolar | Seleccionar |
|----|----|----------|----|--------------------------------|-------------------------------------|
| 1 | | 12345678 | | Larissa Bonilla Lopez | ☒ |
| 2 | | 87654321 | | Ayzza Estrella Díaz Puestas | ☒ |
| 3 | | 14725836 | | Melannie Alanna Reyes Nuñez | ☒ |
| 4 | | 85236741 | | Valentina Rivera Acha | ☒ |
| 5 | | 84235671 | | Silvana del Pilar Supo Ramirez | ☒ |

Mostrando registros del 1 al 5 de un total de 5 registros Anterior 1 Siguiente

CERRAR

Figura 64. Módulo transportista: Interfaz de cargar listado de escolares

Fuente: Elaboración propia

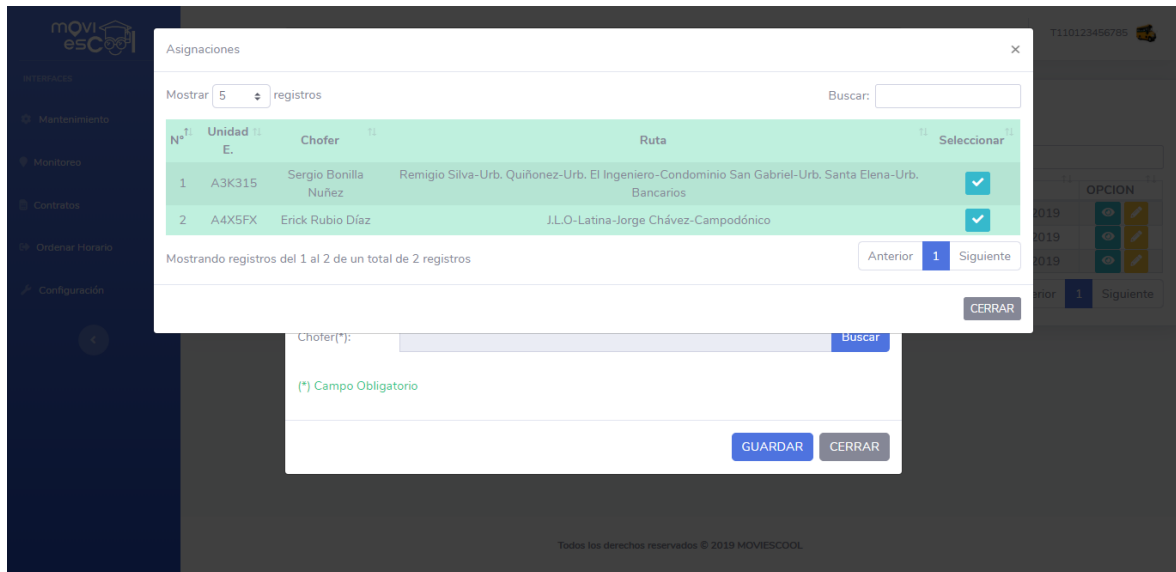


Figura 65. Módulo transportista: Interfaz de cargar listado de asignaciones

Fuente: Elaboración propia

3.7.2.4. Codificación

3.7.2.4.1. Clase: Contrato_Model

```
<?php
class Contrato_Model extends Model
{
    public function __construct()
    {
        parent::__construct();
    }
    public function Listar()
    {
        if ($_SESSION["TIPO"] == "Transportista") {
            $id = $_SESSION["IDENTIDAD"];
            return $this->db
            ->query("CALL
            SP_LISTAR_CONTRATOS($id,'Transportista')");
        } else {
            $id = $_SESSION["IDCOLEGIO"];
            return $this->db
            ->query("CALL
            SP_LISTAR_CONTRATOS($id,'Colegio')");
        }
    }
    public function Reporte()
    {
        if ($_SESSION["TIPO"] == "Transportista") {
            $id = $_SESSION["IDENTIDAD"];

```

```

        return $this->db
        ->query("CALL
        SP_LISTAR_CONTRATOS($sid,'Transportista')");
    } else {
        $sid = $_SESSION["IDCOLEGIO"];
        return $this->db
        ->query("CALL
        SP_LISTAR_CONTRATOS($sid,'Colegio')");
    }
}

public function Cargar_Escolares()
{
    $sid = $_SESSION["IDCOLEGIO"];
    return $this->db
    ->query("CALL SP_CARGAR_ESCOLARES($sid)");
}

public function Cargar_Asignacion()
{
    $sid = $_SESSION["IDENTIDAD"];
    return $this->db->query("CALL
    SP_CARGAR_ASIGNACION($sid)");
}

public function Insertar()
{
    $sidtransp = $_SESSION["IDENTIDAD"];
    $idescolar = $_POST["idescolar"];
    $grado = $_POST["grado"];
    $seccion = $_POST["seccion"];
    $reside = $_POST["reside"];
    $destino = $_POST["destino"];
    $inicio = $_POST["inicio"];
    $fin = $_POST["fin"];
    $asig = $_POST["idasignacion"];
    date_default_timezone_set('America/Lima');
    $fecha_actual = date('Y-m-d H:i:s');
    $sql = "CALL SP_INSERTAR_CONTRATO(:idt,:ides,:gra,:sec,
    :res,:des,:in,:fi,:as,:fa,@MENSAJE)";
    $stm = $this->db->prepare($sql);
    $stm->bindParam(':idt', $sidtransp, PDO::PARAM_STR,
    100);
    $stm->bindParam(':ides', $idescolar, PDO::PARAM_STR,
    100);
    $stm->bindParam(':gra', $grado, PDO::PARAM_STR, 100);
    $stm->bindParam(':sec', $seccion, PDO::PARAM_STR, 100);
    $stm->bindParam(':res', $reside, PDO::PARAM_STR, 100);
    $stm->bindParam(':des', $destino, PDO::PARAM_STR, 100);
    $stm->bindParam(':in', $inicio, PDO::PARAM_STR, 100);
    $stm->bindParam(':fi', $fin, PDO::PARAM_STR, 100);

```

```

        $stm->bindParam(':as', $asig, PDO::PARAM_STR, 100);
        $stm->bindParam(':fa', $fecha_actual, PDO::PARAM_STR,
        100);
        $stm->execute();
        $stm->closeCursor();
        $ms      = $this->db
        ->query("SELECT @MENSAJE AS mensaje")
        ->fetch(PDO::FETCH_ASSOC);
        $mensaje = sprintf($ms["mensaje"]);
        echo json_encode($mensaje);
    }
}

```

La clase Contrato_Model extiende de la clase base Model y contiene los siguientes métodos: Listar(), Cargar_Escolares(), Cargar_Asignacion(), Reporte(), Insertar(). Estos métodos sirven para realizar consultas a la base datos MySQL.

El método Listar(), realiza la llamada al procedimiento almacenado SP_LISTAR_CONTRATOS y retorna el listado de contratos. El método Cargar_Escolares(); captura el id del usuario que proviene del método POST, realiza la llamada al procedimiento almacenado SP_CARGAR_ESCOLARES, retorna la consulta. El método Cargar_Asignacion(); captura el id del usuario que proviene del método POST, realiza la llamada al procedimiento almacenado SP_CARGAR_ASIGNACION, retorna la consulta.

El método Insertar(), captura los datos que provienen del método POST, realiza la llamada al procedimiento almacenado SP_INSERTAR_CONTRATO que contiene un parametro de salida, retorna la consulta y la convierte en formato JSON.

3.7.2.4.2. Clase: Contrato

```

<?php
class Contrato extends Controller
{
    public function __construct()
    {
        parent::__construct();
        Session::init();
        $slogged = Session::get('loggedIn');
        if ($slogged == false) {

```

```

        Session::destroy();
        header('location: ../login');
        exit;
    }
    $this->view->js = array('contrato/js/contrato.js');
}
public function index()
{
    $this->view->render('contrato/index');
}
public function Listar_Contratos()
{
    $this->view->Listar = $this->model->Listar();
    $this->view->render('contrato/listado', true);
}
public function Reporte_General()
{
    $this->Listar = $this->model->Reporte();
    $this->view->render(require
        'views/contrato/reportes/Reporte_general.php');
}
public function Cargar_Escolares()
{
    $this->view->Listar = $this->model->Cargar_Escolares();
    $this->view
        ->render('contrato/listado_escolares', true);
}
public function Cargar_Asignacion()
{
    $this->view->Listar = $this->model
        ->Cargar_Asignacion();
    $this->view
        ->render('contrato/listado_asignacion', true);
}
public function Insertar_Contrato()
{
    $this->model->Insertar();
}
}

```

La clase Contrato extiende de la clase base Controller, contiene el método: index(), Listar_Contratos(), Reporte_General(), Cargar_Escolares(), Cargar_Asignacion(), Insertar_Contrato(). Estos métodos sirven para recibir las peticiones que realiza la vista.

3.7.2.4.3. Vista contrato/index: index.php

```
<div class="card shadow mb-4">
  <div class="card-body">
    <?php if ($_SESSION["TIPO"] == "Colegio"): ?>
      <h6 class="m-0 font-weight-bold text-primary text-center">REPORTE DE CONTRATOS</h6>
    <?php else: ?>
      <h6 class="m-0 font-weight-bold text-primary text-center">LISTADO DE CONTRATOS</h6>
    <?php endif; ?>
    <?php if ($_SESSION["TIPO"] == "Colegio"): ?>
      <a href="contrato/Reporte_General/" target="_blank" class="btn btn-primary btn-sm"><i></i> Generar PDF</a>
    <?php endif; ?>
    <?php if ($_SESSION["TIPO"] == "Transportista"): ?>
      <span class="btn btn-primary btn-sm float-left" data-toggle="modal" data-target="#nuevo">
        <i class="fa fa-plus-circle"></i>&nbsp;  NUEVO CONTRATO</span>
    <?php endif; ?>
  <div id="contrato"></div>
</div>

<!-- ventana modal nuevo contrato -->
<div class="modal fade" id="nuevo" tabindex="-1" role="dialog" aria-labelledby="myExtraLargeModalLabel" aria-hidden="true">
  <div class="modal-dialog modal-lg">
    <div class="modal-content">
      <div class="modal-header">
        <h6 class="modal-title" id="exampleModalLabel">REGISTRAR DATOS DEL CONTRATO</h6>
        <button type="button" class="close" data-dismiss="modal" arial-label="Close">
          <span aria-hidden="true">&times;</span>
        </button>
      </div>
      <form action="<?php echo URL; ?>contrato/Insertar_Contrato/" name="frmNuevoContrato" id="frmNuevoContrato" method="post">
        <div class="modal-body">
          <div class="form-group row">
            <label for="" class="col-sm-2 col-form-label">Escolar (*):</label>
            <div class="col-sm-10">
              <div class="input-group mb-3">
                <input type="hidden" name="idescolar" id="es">
                <input type="text" class="form-control" placeholder="" aria-label="Recipient's username" aria-describedby="button-addon2" name="escolar">
              </div>
            </div>
          </div>
        </div>
      </form>
    </div>
  </div>
</div>
```

```

        id="lescolar" required
        readonly>
        <div class="input-group-
        append">
            <button class="btn
            btn-primary btnapo"
            type="button"
            data-toggle="modal"
            data-target="#apo"
            id="button-
            addon2">Buscar
        </button>
        </div>
    </div>
</div>
<div class="form-group row">
    <label for="" class="col-sm-2
    col-form-label">Grado (*):</label>
    <div class="col-sm-4">
        <input type="text"
        class="form-control"
        name="grado" id="lgrado"
        required autocomplete="off">
    </div>
    <label for="" class="col-sm-2
    col-form-label">Seccion (*):</label>
    <div class="col-sm-4">
        <input type="text"
        class="form-control"
        name="seccion" id="lseccion"
        required autocomplete="off"
        onkeypress="return
        soloLetras(event)">
    </div>
</div>
<div class="form-group row">
    <label for="" class="col-sm-2
    col-form-label">Residencia (*):</label>
    <div class="col-sm-10">
        <input type="text"
        class="form-control"
        name="reside" id="lreside"
        required autocomplete="off">
    </div>
</div>
<div class="form-group row">
    <label for="" class="col-sm-2
    col-form-label">Destino (*):</label>
    <div class="col-sm-10">
        <input type="text"
        class="form-control"
        name="destino" id="ldestino"
        required autocomplete="off">
    </div>
</div>
<div class="form-group row">
    <label for="" class="col-sm-2
    col-form-label">
    Fecha de Inicio (*):</label>

```

```

        <div class="col-sm-4">
            <input type="date"
                class="form-control"
                name="inicio" id="linicio">
        </div>
        <label for="" class="col-sm-2
col-form-label">Fecha Fin(*):</label>
        <div class="col-sm-4">
            <input type="date"
                class="form-control" name="fin"
                id="lfin">
        </div>
    </div>
    <div class="form-group row">
        <label for="" class="col-sm-2
col-form-label">Chofer(*):</label>
        <div class="col-sm-10">
            <div class="input-group mb-3">
                <input type="hidden"
                    name="idasignacion"
                    id="as">
                <input type="text"
                    class="form-control"
                    placeholder="" aria-
                    label="Recipient's
                    username" aria-
                    describedby="button-
                    addon2" name="asignacion"
                    id="asignacion" required
                    readonly>
                <div class="input-group-
                    append">
                    <button class="btn
                    btn-primary btnas"
                    type="button"
                    data-toggle="modal"
                    data-target="#asig"
                    id="button-
                    addon2">Buscar
                    </button>
                </div>
            </div>
        </div>
    </div>
    <div class="text-success">
        (*) Campo Obligatorio</p>
    <div class="mensaje"></div>
</div>
<div class="modal-footer">
    <button class="btn btn-primary"
        type="submit">GUARDAR</button>
    <button class="btn btn-secondary"
        data-dismiss="modal">CERRAR</button>
</div>
</form>
</div>
</div>
</div>

```

```

<!-- ventana modal listado de escolares -->
<div class="modal fade" id="apo" tabindex="-1" role="dialog"
aria-labelledby="myExtraLargeModalLabel" aria-hidden="true">
  <div class="modal-dialog modal-xl">
    <div class="modal-content">
      <div class="modal-header">
        <h6 class="modal-title"
id="exampleModalLabel">LISTADO DE ESCOLARES</h6>
        <button type="button" class="close"
data-dismiss="modal" aria-label="Close">
          <span aria-hidden="true">&times;</span>
        </button>
      </div>
      <div class="modal-body">
        <div id="esco"></div>
      </div>
      <div class="modal-footer">
        <button type="button" class="btn btn-secondary"
data-dismiss="modal">Cerrar</button>
      </div>
    </div>
  </div>
</div>

<!-- ventana modal listado de asignaciones -->
<div class="modal fade" id="asig" tabindex="-1" role="dialog"
aria-labelledby="myExtraLargeModalLabel" aria-hidden="true">
  <div class="modal-dialog modal-xl">
    <div class="modal-content">
      <div class="modal-header">
        <h6 class="modal-title" id="exampleModalLabel">
LISTADO DE ASIGNACIONES</h6>
        <button type="button" class="close"
data-dismiss="modal" aria-label="Close">
          <span aria-hidden="true">&times;</span>
        </button>
      </div>
      <div class="modal-body">
        <div id="asg"></div>
      </div>
      <div class="modal-footer">
        <button type="button" class="btn btn-secondary
btn-sm" data-dismiss="modal">CERRAR</button>
      </div>
    </div>
  </div>
</div>

```

El archivo index.php de la vista contrato/index contiene lo siguiente:

- Ventanas modales del formulario registrar datos del contrato (ver Figura 65.

Módulo transportista: Interfaz de registrar contrato), ventana modal de listado de

escolares (ver Figura 66. Módulo transportista: Interfaz de cargar listado de escolares) y ventana modal de listado de asignaciones (ver figura 67. Módulo transportista: Interfaz de cargar listado de asignaciones).

3.7.2.4.4. Vista contrato/listado: listado.php

```
<!-- tabla con listado de contratos -->
<table class="table-striped table-responsive-sm table-hover
table-bordered" width="100%" cellpadding="0" id="tbcontrato">
    <thead>
        <tr>
            <th class="text-center">N°</th>
            <th class="">ESCOLAR</th>
            <th class="">APODERADO</th>
            <th class="">RUTA</th>
            <th class="">F. CONTRATO</th>
            <?php if ($_SESSION["TIPO"] == "Colegio"): ?>
                <th class="">TRANSP.</th>
                <th class="text-center">PDF</th>
            <?php endif; ?>
            <th class="text-center">OPCION</th>
        </tr>
    </thead>
    <tbody>
        <?php $contador = 0; ?>
        <?php foreach ($this->Listar as $key => $value): ?>
            <?php $contador++; ?>
            <tr>
                <td class="text-center"><?php echo $contador; ?></td>
                <td class=""><?php echo $value['ESCOLAR']; ?></td>
                <td class=""><?php echo $value['APODERADO']; ?></td>
                <td class=""><?php echo $value['RUTA']; ?></td>
                <td class="">
                    <?php echo $value['FECHACONTRATO']; ?></td>
                    <?php if ($_SESSION["TIPO"] == "Colegio"): ?>
                        <td class="">
                            <?php echo $value['RAZONSOCIAL']; ?></td>
                            <td class="text-center">
                                <form action="contrato/report/"
                                target="_blank">
                                    <input type="hidden" name="idc"
                                    value="<?php echo
                                    $value['IDCONTRATO'] ?>">
                                    <button type="submit"
                                    class="fa fa-file-pdf"></button>
                                </form>
                            </td>
                        <?php endif; ?>
                    <td class="text-center">
                        <span class="btn btn-info btn-sm view"
                        onclick="<?php echo $value['IDCONTRATO']; ?>"
```

```

        data-toggle="modal" data-target="#info"
        title="Ver más Información"><i class="fa fa-eye">
    </i></span>
    <?php if ($_SESSION["TIPO"] == "Transportista"): ?>
    <span class="btn btn-warning btn-sm del"
    onclick="<?php echo $value['IDCONTRATO']; ?>"
    data-toggle="modal" data-target="#update"
    title="Editar Datos">
    <i class="fa fa-pencil-alt"></i></span>
    <?php endif; ?>
    </td>
</tr>
<?php endforeach; ?>
</tbody>
</table>

```

El archivo listado.php de la vista contrato/listado contiene lo siguiente:

- Tabla con estilo de Bootstrap para mostrar el listado de contratos (ver Figura 64.

Módulo transportista: Interfaz de listado de contratos).

3.7.2.4.5. Vista contrato/listado_escolares: listado_escolares.php

```

<!-- tabla con listado de escolares -->
<table class="table-striped table-sm table-hover table-success"
id="tbcontratoesco" width="100%" cellpadding="0">
    <thead>
        <tr>
            <th class="text-center">N°</th>
            <th class="text-center">Dni</th>
            <th class="text-center">Escolar</th>
            <th class="text-center">Seleccionar</th>
        </tr>
    </thead>
    <tbody>
        <?php $contador = 0; ?>
        <?php foreach ($this->Listar as $key => $value): ?>
        <?php $contador++; ?>
        <tr>
            <td class="text-center"><?php echo $contador; ?></td>
            <td class="text-center">
                <?php echo $value['DNI']; ?></td>
            <td class="text-center">
                <?php echo $value['ESCOLAR']; ?></td>
            <td class="text-center">
                <span title="Seleccionar Escolar"
                class="btn btn-sm btn-info"
                onclick="AgregarCond_ed('<?=$value['IDESCOLAR']; ?>', '<?=$value['ESCOLAR']; ?>');" data-toggle='modal'><i
                class="fa fa-check"></i></span></td>
        </tr>
        <?php endforeach; ?>
    </tbody>
</table>

```

```

        </tbody>
    </table>

    <script>
        function AgregarCond_ed(cod,nombres){
            $('#apo').modal('hide');
            document.frmnuevocontrato.idescolar.value=cod;
            document.frmnuevocontrato.escolar.value=nombres;
        }
    </script>

```

El archivo listado_escolares.php de la vista contrato/listado_escolares contiene lo siguiente:

- Tabla con estilo de Bootstrap para mostrar el listado de escolares (ver Figura 66. Módulo transportista: Interfaz de cargar listado de escolares) que se cargará en un contenedor <div id="esco"></div>.

3.7.2.4.6. Vista contrato/listado_asignaciones: listado_asignaciones.php

```

<!-- tabla con listado de asignaciones -->
<table class="table-striped table-sm table-hover table-success"
id="tbcontratoasig" width="100%" cellpadding="0">
    <thead>
        <tr>
            <th class="text-center">N°</th>
            <th class="text-center">Unidad E.</th>
            <th class="text-center">Chofer</th>
            <th class="text-center">Ruta</th>
            <th class="text-center">Seleccionar</th>
        </tr>
    </thead>
    <tbody>
        <?php $contador = 0;?>
        <?php foreach ($this->Listar as $key => $value): ?>
        <?php $contador++;?>
        <tr>
            <td class="text-center"><?php echo $contador; ?></td>
            <td class="text-center">
                <?php echo $value['NROPLACA']; ?></td>
            <td class="text-center">
                <?php echo $value['CHOFER']; ?></td>
            <td class="text-center">
                <?php echo $value['NOMBRE']; ?></td>
            <td class="text-center">
                <span title="Seleccionar Chofer Asignado"
                class="btn btn-sm btn-info"
                onclick="AgregarCond_ed('<?=$value['IDASIGNACION'];?>','
                '<?=$value['CHOFER'];?>');" data-toggle='modal'>

```

```

        <i class="fa fa-check"></i></span>
      </td>
    </tr>
  <?php endforeach;?>
</tbody>
</table>

<script>
  function AgregarCond_ed(cod,nombres){
    $('#asig').modal('hide');
    document.frmnuevocontrato.idasignacion.value=cod;
    document.frmnuevocontrato.asignacion.value=nombres;
  }
</script>

```

El archivo listado_asignaciones.php de la vista contrato/listado_asignaciones contiene lo siguiente:

- Tabla con estilo de Bootstrap para mostrar el listado de asignaciones(ver Figura 67. Módulo transportista: Interfaz de cargar listado de asignaciones) que se cargará en un contenedor <div id="asg"></div>.

3.7.2.4.7. javaScript: contrato/js/contrato.js

```

$(function(){
  listado();
  function listado(){
    $.post('contrato/Listar_Contratos',function(r){
      $('#contrato').html(r);
    });
  }
  function limpiar(){
    $('#es').val("");
    $('#lescolar').val("");
    $('#lgrado').val("");
    $('#lseccion').val("");
    $('#lreside').val("");
    $('#ldestino').val("");
    $('#linicio').val("");
    $('#lfin').val("");
    $('#asignacion').val("");
  }
  $('.btnapo').on('click',function(){
    $.post('contrato/Cargar_Escolares',function(data){
      $('#esco').html(data);
    });
  });
});

```

```

$($('.btnas')).on('click',function(){
    $.post('contrato/Cargar_Asignacion', function(data) {
        $("#asg").html(data);
    });
});
$('#frmnuevocontrato').submit(function(){
    let url=$(this).attr("action");
    let data=$(this).serialize();
    $.post(url,data,function(r){
        if (r=="OPERACION REALIZADA CORRECTAMENTE") {
            listado();
            $(".mensaje").html('<div class="alert alert-info" role="alert">'+r+'</div>');
            $(".mensaje").show();
            $(".mensaje").fadeOut(4000);
            setTimeout(function(){
                limpiar();
                $('#nuevo').modal('hide');
            }, 2000);
        }else{
            $(".mensaje").html('<div class="alert alert-danger" role="alert">'+r+'</div>');
            $(".mensaje").show();
            $(".mensaje").fadeOut(4000);
        }
    },'json');
    return false
});
});

```

El archivo contrato.js de la vista apoderado/js/ contiene:

- Función listado(): realiza el llamado del método Listar_Contratos() al controlador Contrato, retorna el listado de contratos y se carga en un elemento HTML <div id="contrato"></div>.
- Función limpiar(): realiza la limpieza de los controles del formulario nuevo contrato.
- Código jQuery y Ajax de listado de asignaciones: realiza el llamado del método Cargar_Asignacion al controlador Contrato, retorna el listado de asignaciones y se carga en un elemento HTML <div id="asg"></div>

- Código jQuery y Ajax de listado de escolares: realiza el llamado del método Cargar_Escolares al controlador Contrato, retorna el listado de escolares y se carga en un elemento HTML <div id="esco"></div>
- Código jQuery, Ajax y JSON del formulario nuevo contrato. Este envía la petición a través del método POST al controlador Contrato del método Insertar_Contrato().

3.7.3. Reporte de asignaciones

3.7.3.1. Tabla de base de datos

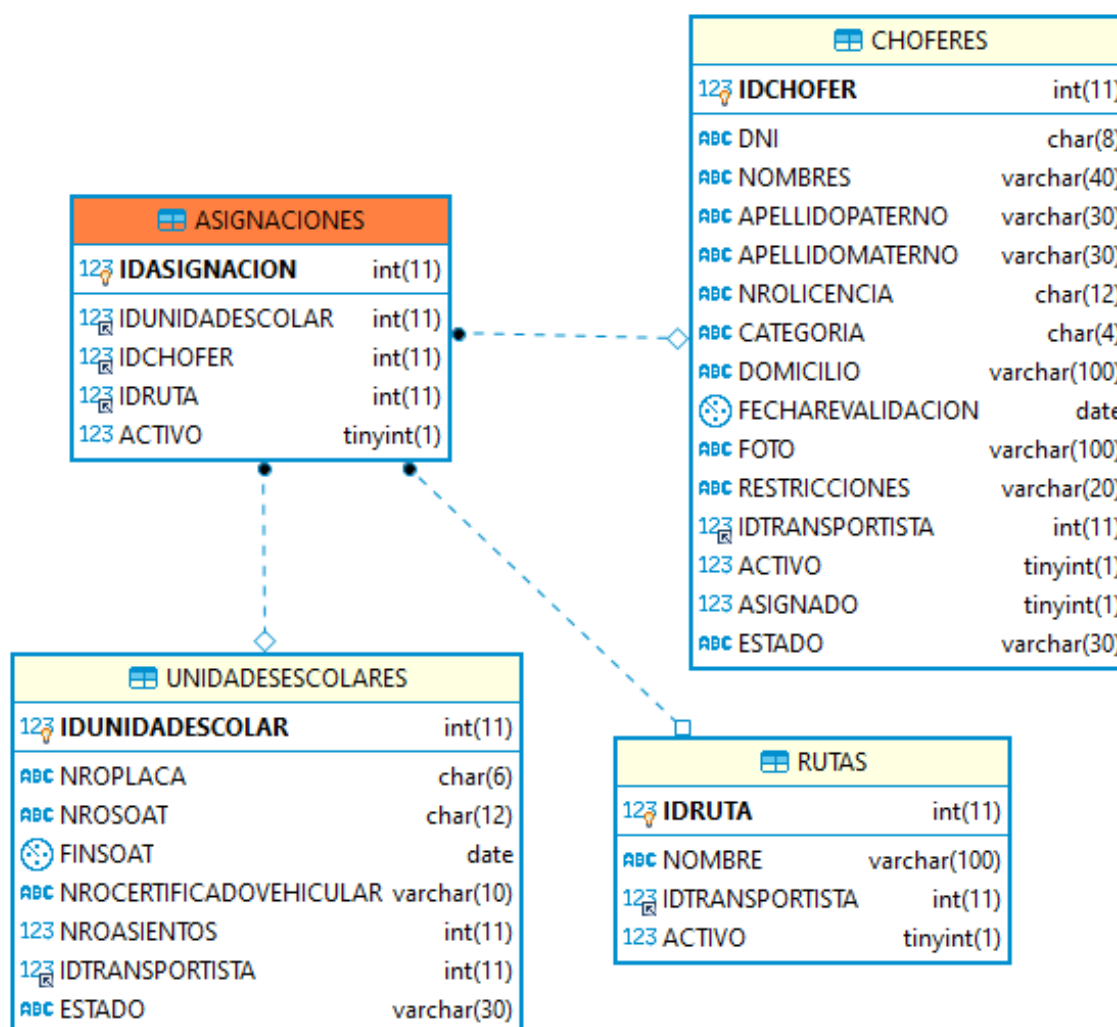


Figura 66. Tablas: ASIGNACIONES, UNIDADESESCOLARES, CHOFERES, RUTAS

Fuente: Elaboración propia

3.7.3.2. Procedimientos almacenados

```
-- LISTAR ASIGNACIONES --
DELIMITER $$
DROP PROCEDURE IF EXISTS SP_LISTAR_ASIGNACIONES$$
CREATE PROCEDURE SP_LISTAR_ASIGNACIONES
(P_ID INT,P_TIPO VARCHAR(30))
BEGIN
    IF(P_TIPO='Transportista') THEN
        SELECT A.IDASIGNACION,U.NROPLACA,
               CONCAT(C.NOMBRES,' ',C.APELLIDOPATERNO,
                      ' ',C.APELLIDOMATERNO) AS CHOFER,R.NOMBRE,
               T.RAZONSOCIAL
        FROM ASIGNACIONES A
        INNER JOIN UNIDADESESCOLARES U ON
            A.IDUNIDADESCOLAR=U.IDUNIDADESCOLAR
        INNER JOIN CHOFERES C ON C.IDCHOFER=A.IDCHOFER
        INNER JOIN RUTAS R ON A.IDRUTA=R.IDRUTA
        INNER JOIN TRANSPORTISTAS T ON
            C.IDTRANSPORTISTA=T.IDTRANSPORTISTA
        WHERE C.IDTRANSPORTISTA=P_ID;
    ELSE
        SELECT A.IDASIGNACION,U.NROPLACA,CONCAT(C.NOMBRES,
            ' ',C.APELLIDOPATERNO,' ',C.APELLIDOMATERNO)
            AS CHOFER,R.NOMBRE,T.RAZONSOCIAL
        FROM ASIGNACIONES A
        INNER JOIN UNIDADESESCOLARES U ON
            A.IDUNIDADESCOLAR=U.IDUNIDADESCOLAR
        INNER JOIN CHOFERES C ON C.IDCHOFER=A.IDCHOFER
        INNER JOIN RUTAS R ON A.IDRUTA=R.IDRUTA
        INNER JOIN TRANSPORTISTAS T ON
            C.IDTRANSPORTISTA=T.IDTRANSPORTISTA
        WHERE T.IDCOLEGIO=P_ID;
    END IF;
END $$
```

3.7.3.3. Interfaz de usuario

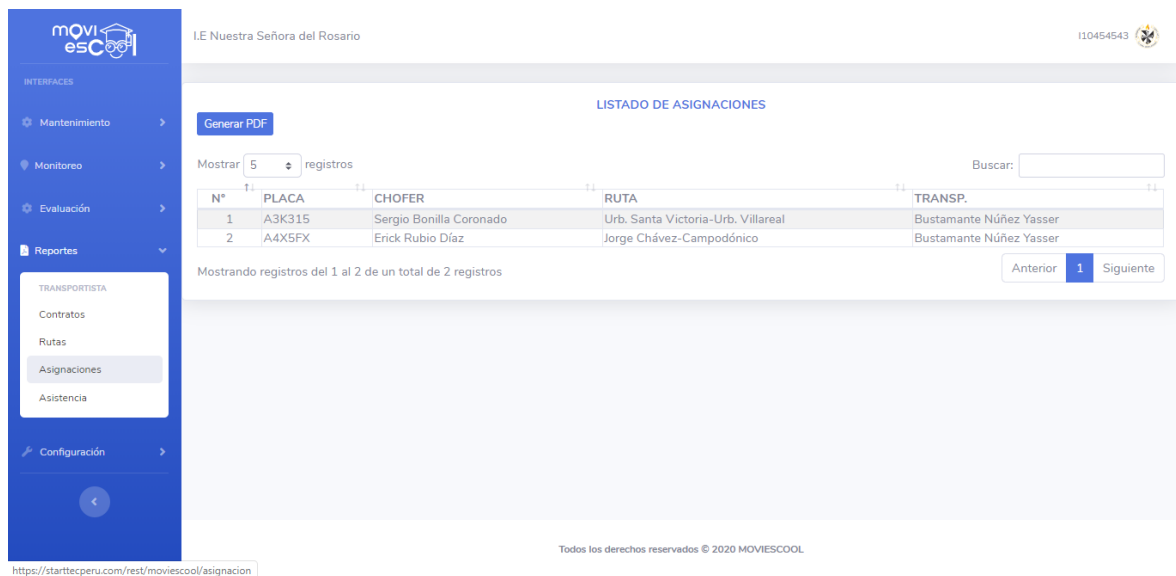


Figura 67. Módulo colegio: Interfaz de listado de asignaciones

Fuente: Elaboración propia

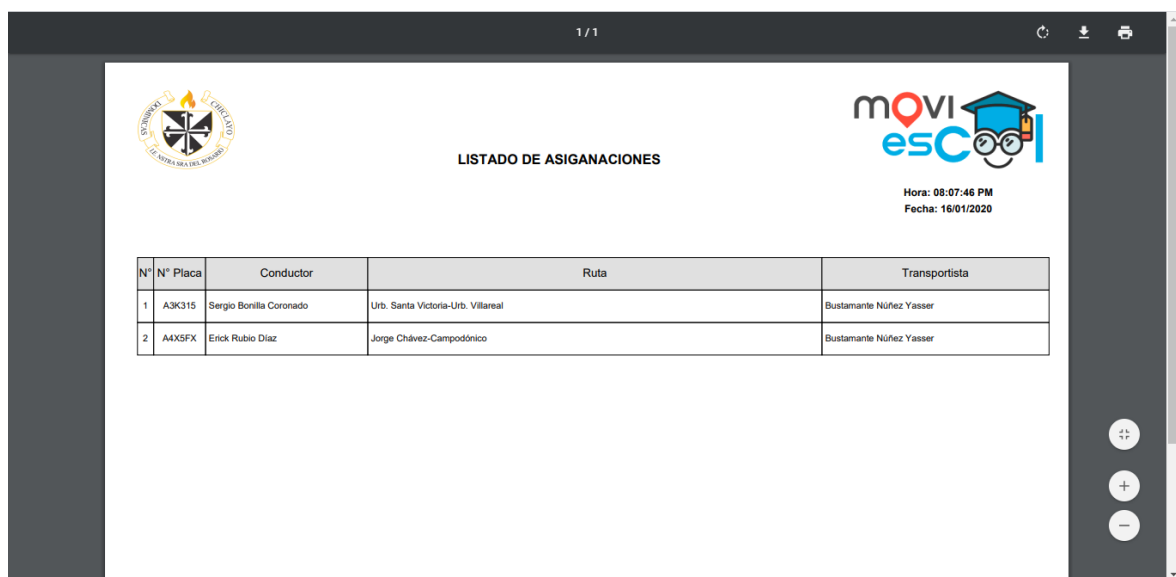


Figura 68. Módulo colegio: Interfaz de reporte de asignaciones PDF

Fuente: Elaboración propia

3.7.3.4. Codificación

3.7.3.4.1. Clase: Asignacion_Model

```
<?php
class Asignacion_Model extends Model
{
    public function __construct()
```

```

{
    parent::__construct();
}
public function Listar()
{
    if ($_SESSION["TIPO"] == "Transportista") {
        $id = $_SESSION["IDENTIDAD"];
        return $this->db->query("CALL
        SP_LISTAR_ASIGNACIONES($id, 'Transportista')");
    } else {
        $id = $_SESSION["IDCOLEGIO"];
        return $this->db->query("CALL
        SP_LISTAR_ASIGNACIONES($id, 'Colegio')");
    }
}
public function Reporte()
{
    if ($_SESSION["TIPO"] == "Transportista") {
        $id = $_SESSION["IDENTIDAD"];
        return $this->db->query("CALL
        SP_LISTAR_ASIGNACIONES($id, 'Transportista')");
    } else {
        $id = $_SESSION["IDCOLEGIO"];
        return $this->db->query("CALL
        SP_LISTAR_ASIGNACIONES($id, 'Colegio')");
    }
}
}

```

La clase Asignacion_Model extiende de la clase base Model y contiene los siguientes métodos: Listar(), Reporte(). Estos métodos sirven para realizar consultas a la base datos MySQL.

El método Listar() o Reporte(), realiza la llamada al procedimiento almacenado SP_LISTAR_ASIGNACIONES y retorna el listado de asignaciones.

3.7.3.4.2. Clase: Asignacion

```

<?php
class Asignacion extends Controller
{
    public function __construct()
    {
        parent::__construct();
        Session::init();
        $slogged = Session::get('loggedIn');
    }
}

```

```

        if ($slogged == false) {
            Session::destroy();
            header('location: ../login');
            exit;
        }
        $this->view->js = array('asignacion/js/asignacion.js');
    }
    public function index() {
        $this->view->render('asignacion/index');
    }
    public function Listar_Asignacion()
    {
        $this->view->Listar = $this->model->Listar();
        $this->view->render('asignacion/listado', true);
    }
    public function Reporte_Asignaciones()
    {
        $this->Listar = $this->model->Listar();
        $this->view->render(require
            'views/asignacion/reportes/asignacion_general.php');
    }
}

```

La clase Asignacion extiende de la clase base Controller, contiene el método: index(), Listar_Asignacion(), Reporte_General(). Estos métodos sirven para recibir las peticiones que realiza la vista.

3.7.3.4.3. Vista asignacion/index: index.php

```

<!-- reporte de asignaciones -->
<div class="card shadow mb-4">
    <div class="card-body">
        <?php if ($_SESSION["TIPO"] == "Admin"): ?>
            <h6 class="m-0 font-weight-bold text-primary text-center">REPORTE DE ASIGNACIONES</h6>
        <?php else: ?>
            <h6 class="m-0 font-weight-bold text-primary text-center">LISTADO DE ASIGNACIONES</h6>
        <?php endif; ?>
        <?php if ($_SESSION["TIPO"] == "Colegio"): ?>
            <a href="asignacion/Reporte_Asignaciones/"
            target="_blank" class="btn btn-primary btn-sm"><i
            >>/i> Generar PDF</a>
        <?php endif; ?>
        <?php if ($_SESSION["TIPO"] == "Transportista"): ?>
            <span class="btn btn-primary btn-sm float-left"
            data-toggle="modal" data-target="#nuevo">
            <i class="fa fa-plus-circle"></i>
            &nbsp;  NUEVO ASIGNACION</span>
        </if>
    </div>
</div>

```

```

        <?php endif;?>
        <div id="asignacion"></div>
    </div>
</div>

```

3.7.3.4.4. Reporte asignacion/resportes/asignacion_general.php

```

<?php
date_default_timezone_set("America/Lima");
class PDF extends FPDF
{
    function Header()
    {
        $this->Image('./public/img/moviescool_png_small.png' ,
        230 ,8, 60 , 25,'png');
        $this->SetFont('Arial','B',12);
        if ($_SESSION["TIPO"]=="Colegio" ) {
            $this->Image('./public/img/colegio/'.$_SESSION["LOGO"],
            10,8, 30 , 25,'PNG');
        }
        $this->SetFont('Arial','B',12);
        $this->Cell(30);
        $this->Cell(200,40,iconv('utf-8', 'cp1252',
        'LISTADO DE ASIGNACIONES'),0,1,'C');
        $this->SetFont('Arial','B',9);
        $this->Cell(500,-10,"Fecha: ".date("d/m/Y"),0,1,'C');
        $this->Cell(500,0,"Hora: ".date("h:i:s A"),0,1,'C');
    }
    function Footer() {
        $this->SetY(-37);
        $this->SetFont('Arial','I',8);
        $this->Cell(0,10,iconv('utf-8', 'cp1252', 'Página ').$this
        ->PageNo(). ' / {nb}',0,0,'C');
        $this->Line(15,260,200,260);
    }
    function __construct() {
        parent::__construct('L','mm','A4');
    }
}

$pdf=new PDF();
$pdf->AddPage();
$pdf->SetFont('Arial','',10);
$pdf->SetFillColor(224,224,224);
$pdf->ln(20);
$pdf->Cell(5,10,iconv('utf-8', 'cp1252', 'N°'),1,0,'C','true');
$pdf->Cell(16,10,iconv('utf-8', 'cp1252', 'N° Placa'),1,0,'C','true');
$pdf->Cell(50,10,'Conductor',1,0,'C','true');
$pdf->Cell(140,10,'Ruta',1,0,'C','true');
$pdf->Cell(70,10,'Transportista',1,1,'C','true');
$pdf->SetFont('Arial','',8);
$contador =0;
foreach ($this->Listar as $key => $value) {

```

```

$contador++;
$placa=$value['NROPLACA'];
$chofer=$value['CHOFER'];
$nombre=$value['NOMBRE'];
$razon=$value['RAZONSOCIAL'];
$pdf->Cell(5,10,iconv('utf-8','cp1252',$contador),1,0,'C');
$pdf->Cell(16,10,iconv('utf-8','cp1252',$placa),1,0,'C');
$pdf->Cell(50,10,iconv('utf-8','cp1252',$chofer),1,0,'L');
$y = $pdf->GetY();
$pdf->Cell(140,10,iconv('utf-8','cp1252',$nombre),1,0,'L');
$pdf->Cell(70,10,iconv('utf-8','cp1252',$razon),1,1,'L');
}
$pdf->Output();
?>

```

CAPÍTULO IV: COSTOS Y BENEFICIOS

4.1. Análisis de costos

4.1.1. Costo de software

Tabla 89

Costo de software

Software	Importe (S/)	
ERWin Data Modeler Community Edition	S/	-
DBeaver 6.3.1	S/	-
Sublime Text 3	S/	-
Android Studio 3.5.1	S/	-
WampServer 3.2	S/	-
Total (S/)	S/	-

Fuente: Elaboración propia

4.1.2. Costo de hardware

Tabla 90

Costo de hardware

Hardware	Importe (S/)	
Lenovo IdeaPad 510-15IKB Notebook	S/	2,800.00
Mouse Inalámbrico Logitech M280	S/	80.00
Memoria USB SanDisk Cruzer Force 16GB	S/	27.00
Total (S/)	S/	2,907.00

Fuente: Elaboración propia

4.1.3. Costo de implementación módulo GPS

Tabla 91

Costo de implementación módulo de GPS con Arduino

Componentes	Importe (S/)	
Arduino Nano	S/	25.00
U-blox NEO-6M GPS	S/	70.00
SIM800L GPRS	S/	52.00
Convertidor de nivel lógico	S/	8.00
LM2596	S/	12.00
Batería Lipo 7.4V 1000mA	S/	65.00
Chip entel prepago	S/	5.00
Cables jumper m-m	S/	8.00
Cables jumper h-m	S/	8.00
Total (S/)	S/	253.00

Fuente: Elaboración propia

4.1.4. Costo de personal

Tabla 92

Costo de personal

Personal	Meses	Sueldo (S/)		Remuneración (S/)	
Programador web	2	S/	1,500.00	S/	3,000.00
Programador android	1	S/	1,500.00	S/	1,500.00
Analista de base de datos	1	S/	1,500.00	S/	1,000.00
Ingeniero electrónico	1	S/	1,000.00	S/	1,000.00
Total (S/)				S/	6,500.00

Fuente: Elaboración propia

4.1.5. Costo de servicios

Tabla 93

Costo de servicio

Servicio	Importe (S/)
Firebase Realtime Database (Plan Spark)	S/ -
Dominio y alojamiento web	S/ 200.00
Correo electrónico gmail	S/ -
Recarga chip prepago	S/ 30.00
Total (S/)	S/ 230.00

Fuente: Elaboración propia

4.1.6. Resumen de costos

Tabla 94

Resumen de costos

Tipo de costo	Importe (S/)
Costo de software	S/ -
Costo de hardware	S/ 2,907.00
Costo de implementación módulo de GPS con Arduino	S/ 253.00
Costo de personal	S/ 6,500.00
Costo de servicios	S/ 230.00
Total (S/)	S/ 9,890.00

Fuente: Elaboración propia

4.2. Análisis de beneficios

4.2.1. Beneficios tangibles

Tabla 95

Beneficios tangibles

Descripción	Importe (S/)
Soporte multiplataforma y entorno visual moderno	S/ 6,000.00
Fácil control de las unidades de transporte en tiempo real	S/ 1,200.00
Mejora el proceso del control de rutas	S/ 800.00
Generacion de reportes estadisticos	S/ 800.00
Total (S/)	S/ 8,800.00

Fuente: Elaboración propia

4.2.2. Beneficios intangibles

- ✓ Mejora en la seguridad de traslado del escolar (colegio y/o hogar).
- ✓ Ahorro de tiempo en el proceso de asistencia del escolar.
- ✓ Mejora la comunicación entre centros educativos, padres de familia y transportistas.
- ✓ Agiliza los procesos en la generación de contratos.

4.3. Cálculo de VAN y TIR

4.3.1. Valor actual neto (VAN)

Es un valor actual de los flujos neto, que permiten conocer cuánto se va a ganar o perder con la inversión. La evaluación del proyecto se está tomando un periodo de 02 años, con una tasa de interes del 5%. Para calcularlo se utiliza la siguiente formula:

$$VAN = -I_0 + \sum_{t=1}^n \frac{f_n}{(1+k)^n}$$

Donde:

f_n = representa los flujos de caja en cada periodo n

$I_0 = \text{valor del desembolso inicial de la inversión}$

$n = \text{número de periodos considerados}$

$k = \text{tipo de interés o descuento}$

Entonces:

$$VAN = -9,890 + \frac{8,800}{(1 + 0.05)^1} + \frac{8,800}{(1 + 0.05)^2}$$

$$VAN = -9,890 + 8,380.95 + 7,981.85$$

$$VAN = 6,472.80$$

El VAN es S/ 6,472.80 y es mayor a 0, por lo tanto, se acepta el proyecto, ya que se genera un beneficio.

4.3.2. Tasa interno de retorno (TIR)

Es la tasa de descuento o rentabilidad que ofrece una inversión, hace que el valor actual de los flujos beneficios (positivos) sea igual al de los flujos de costo (negativos), generando un VAN = 0. Para calcularlo, igualamos el VAN a cero y despejamos TIR.

$$VAN = \sum_{t=1}^n \frac{f_n}{(1 + I_0)^n} = 0$$

Donde:

$f_n = \text{representa los flujos de caja en cada periodo } n$

$I_0 = \text{valor del desembolso inicial de la inversión}$

$n = \text{representa el número de período}$

Entonces:

$$0 = \frac{8800}{(1 + TIR)^2} - 9890$$

Según la evaluación con el método del TIR, vamos a recuperar la inversión con una tasa del 48.79 %, en un periodo de dos años. Por lo que concluimos que el proyecto es viable.

4.3.3. Período de recuperación de inversión (PRI)

Es el periodo de tiempo requerido para recuperar el capital inicial de una inversión. Para calcularlo, se utiliza la siguiente fórmula:

$$PRI = \frac{I_0}{f_n}$$

Donde:

f_n = *representa los flujos de caja en cada periodo n*

I_0 = *valor del desembolso inicial de la inversión*

Entonces:

$$PRI = \frac{9,890}{8800}$$

$$PRI = 1,12$$

El PRI es 1.12, es decir que la recuperación de la inversión inicial del proyecto tomará 1 años, 2 meses y 26 días.

CAPÍTULO V: CONCLUSIONES

- La Municipalidad Provincial de Chiclayo, no cuenta con un padrón oficial de los choferes o transportistas que ofrecen el servicio de movilidad escolar, por lo tanto; refleja la informalidad en este servicio. Además, las instituciones educativas tampoco cuentan con datos oficiales para averiguar si las unidades escolares están debidamente registradas. Finalmente, los padres de familia se limitan a contratar el servicio de movilidad escolar a unidades escolares sin registro, lo que repercute en una mala comunicación entre estos: padres de familia, colegios y municipalidad, trayendo como consecuencia posibles riesgos para la integridad de los escolares.
- La arquitectura implementada, incluye un sistema web que administra la información de los procesos del servicio de transporte escolar, de la misma manera se tiene un aplicativo móvil que se encarga de realizar el seguimiento en tiempo real de las unidades escolares a través del módulo de geolocalización con Arduino.
- La base de datos implementada en MySQL, administra la información de las incidencias que ocurren en el recorrido, y estas a su vez se visualizan en tiempo real en el aplicativo móvil permitiendo controlar las incidencias de una manera óptima.
- La implementación del módulo de control de rutas, asegura que el chofer cumpla el orden de las recogidas de los escolares en sus domicilios para llevarlos al colegio (a quien recoge primero, segundo, tercero, etc, y a quien recoge último), si por algún motivo durante la ruta; el chofer no cumple el orden, el chofer debe crear una incidencia para saber que sucedió con el escolar en dicha ruta.

- La implementación del modulo de notificaciones; es de gran ayuda para el padre de familia, colegio y transportista; ya que alerta sobre las recogidas del escolar, inasistencia del escolar, incidencias durante el traslado, entre otras notificaciones.
- El aplicativo móvil, permite realizar un seguimiento más detallado de las unidades escolares al momento de completar las rutas en los destinos establecidos, además que supervisa el recorrido en tiempo real y de que manera los escolares salen y regresan a sus hogares.

CAPÍTULO VI: RECOMENDACIONES

- La Municipalidad Provincial de Chiclayo debe de formalizar el servicio de transporte de estudiantes con alianza de los colegios, de manera que los padres de familia puedan acceder a información veráz y oportuna que les permita contratar el servicio como MOVILIDAD ESCOLAR SEGURA.
- El armado del Arduino y sus componentes, se puede realizar en circuito impreso en una placa baquelita, dado que el modelo de muestra; se ha armado en una placa de pruebas (protoboard).
- Las configuraciones de los servidores y de la base de datos en tiempo real; deben ser revisado cada cierto tiempo para verificar el correcto uso del sistema informático.

BIBLIOGRAFÍA REFERENCIADA

- Anton, C. (2015). *Platzi*. Obtenido de <https://platzi.com/blog/que-es-postgresql/>
- Complutense, F. d. (s.f.). *Fundamentos de Programacion*. Obtenido de <https://www.fdi.ucm.es/profesor/luis/fp/devtools/devtools.html>
- Data, P. (05 de 2016). *Power Data*. Obtenido de <https://blog.powerdata.es/el-valor-de-la-gestion-de-datos/que-es-un-gestor-de-datos-y-para-que-sirve>
- Developers. (25 de 4 de 2018). *Developers*. Obtenido de <https://developer.android.com/studio/intro/?hl=es-419>
- García, M. (5 de Octubre de 2017). *MVC (Modelo-Vista-Controlador): ¿qué es y para qué sirve?* Obtenido de coding or not: <https://codingornot.com/mvc-modelo-vista-controlador-que-es-y-para-que-sirve>
- Gustavo B. (14 de Noviembre de 2019). *¿Qué es un dominio web?* Obtenido de Hostinger: <https://www.hostinger.mx/tutoriales/que-es-un-dominio-web>
- Gustavo B. (10 de Diciembre de 2019). *¿Qué es un hosting?* Obtenido de Hostinger: <https://www.hostinger.es/tutoriales/que-es-un-hosting>
- How, K. (20 de 07 de 2016). Obtenido de Digital Guide: <https://www.ionos.es/digitalguide/servidores/know-how/que-es-el-servidor-dns-y-como-funciona/>
- ICTEA. (2019). *¿Qué es el lenguaje de programación JAVA?* Obtenido de ICTEA: <http://www.ictea.com/cs/knowledgebase.php?action=displayarticle&id=8790>
- Infootec. (2018). *Infootec.net* . Obtenido de <https://www.infootec.net/arduino/#1--Descripcion-y-caracteristicas-tecnicas>
- Infortelecom. (29 de 09 de 2016). *INFORTELECOM*. Obtenido de <https://infortelecom.es/blog/que-es-un-servidor-y-para-que-sirve/>
- ISACA. (31 de 07 de 2017). *Evaluando Software*. Obtenido de <https://www.evaluandosoftware.com:https://www.evaluandosoftware.com/la-geolocalizacion-funciona/>
- Java. (s.f.). *Java*. Obtenido de https://www.java.com/es/download/faq/whatis_java.xml
- Magazine, S. M. (s.f.). Obtenido de <https://sistemas.com/gprs.php>

Manuel, J. (20 de 09 de 2010). *Duplika*. Obtenido de <https://duplika.com/blog/que-son-los-servidores-web-y-por-que-son-necesarios/>

Mechatronics, N. (2016). *Naylamp Mechatronics*. Obtenido de <https://naylampmechatronics.com/arduino-tarjetas/8-arduino-uno-r3.html>

Pelhon, L. (27 de Mayo de 2019). *Qué es JavaScript: primeros pasos*. Obtenido de HostGator: <https://www.hostgator.mx/blog/que-es-javascript/>

Peña, J. (1 de Febrero de 2018). *¿Qué es firebase y qué nos aporta?* Obtenido de Arpen Technologies: <https://arpen technologies.com/es/blog/aplicaciones-movil/que-es-firebase-y-que-nos-aporta/>

Pérez Cardona, M. (14 de Octubre de 2016). *Firebase, qué es y para qué sirve la plataforma de Google*. Obtenido de iebschool: <https://www.iebschool.com/blog/firebase-que-es-para-que-sirve-la-plataforma-desarrolladores-google-seo-sem/>

Porto, J. P. (2012). *Definicion De* . Obtenido de <https://definicion.de/lenguaje-de-programacion/>

RBjuarez. (3 de Octubre de 2019). *Cómo descarga e instalar Bootstrap 4*. Obtenido de RBjuarez: <https://www.rbjuarez.com/bootstrap-descarga-e-instalacion/>

Reifman, J. (17 de Marzo de 2016). *¿Qué es HTML5?* Obtenido de Tutsplus: <https://code.tutsplus.com/es/tutorials/what-is-html5--cms-25803>

Rosero, W. (s.f.). *SUPERTECNODROID*. Obtenido de <https://supertecnodroid.com/que-es-el-jdk/>

Senso, J. (1 de Octubre de 2019). *Servidores WAMP para producción local*. Obtenido de blogs.ugr.es: <https://blogs.ugr.es/tecweb/servidores-wamp/>

Silva, I. (22 de Febrero de 2019). *AJAX: ¿Qué es y cómo utilizar en el Scriptcase*. Obtenido de scriptcase blog: <https://www.scriptcaseblog.net/es/scriptcase-es/ajax-que-es-y-como-utilizar-en-el-scriptcase/>

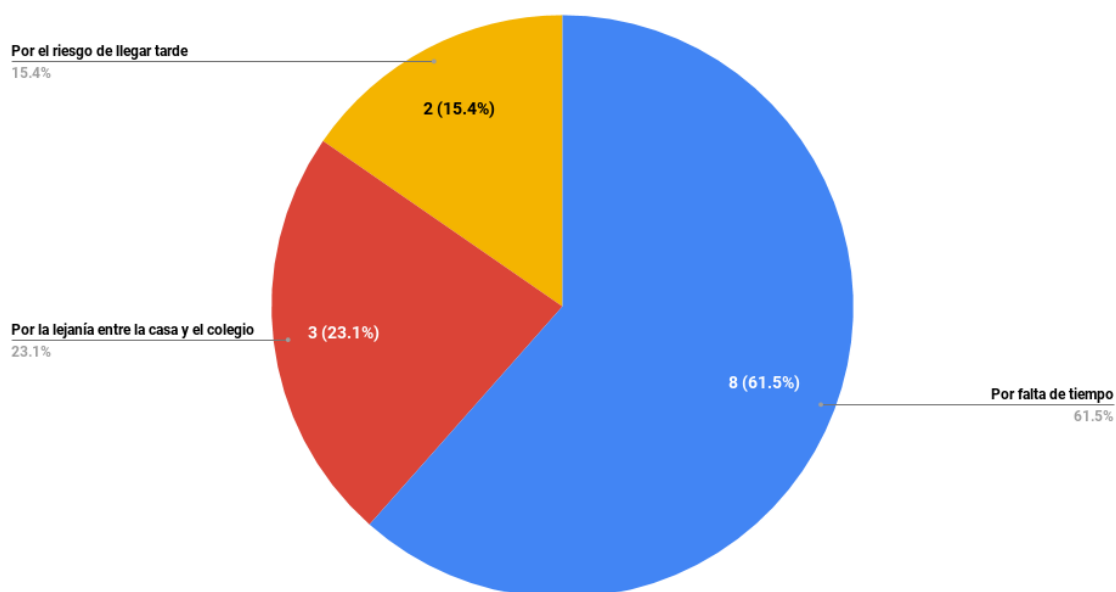
Yubal FM. (3 de Agosto de 2018). *¿Qué es Arduino, cómo funciona y qué puedes hacer con uno?* Obtenido de Xataka: <https://www.xataka.com/basics/que-arduino-como-funciona-que-puedes-hacer-uno>

ANEXOS

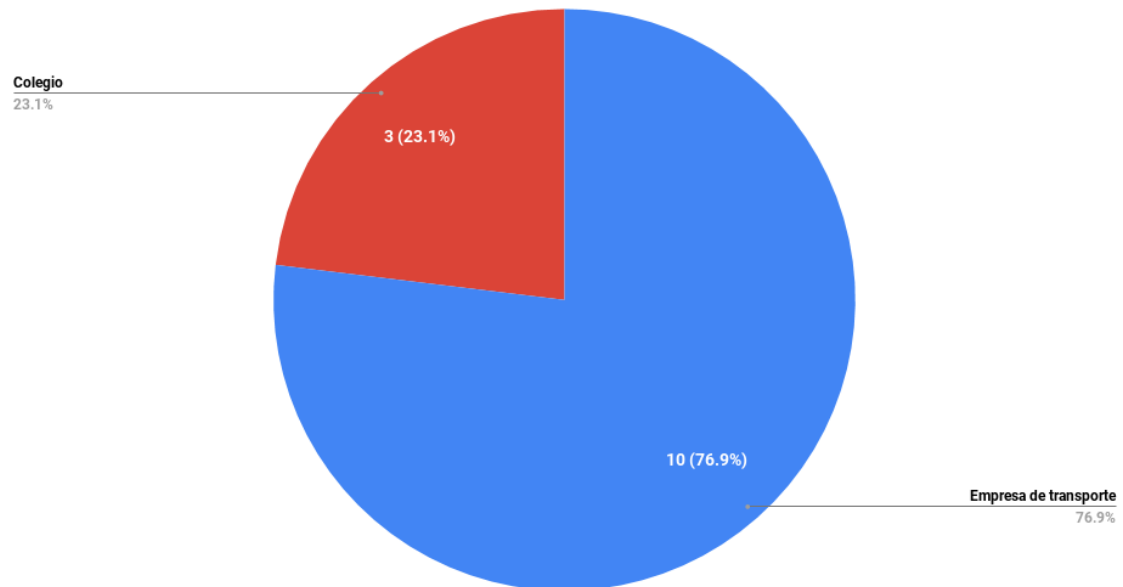
La encuesta se realizó a los padres de familia o apoderados, el cual tiene por objetivo identificar los principales riesgos que existen a diario en el servicio de transporte de estudiantes en la ciudad de Chiclayo. Esto constituye un instrumento para recopilar información significativa sobre la movilidad escolar que contrató, llegada de la movilidad escolar al lugar de destino indicado, incidentes durante el recorrido, entre otros.

Enlace para participar de la encuesta: <https://forms.gle/388dwYbR4JnHyWCV6>

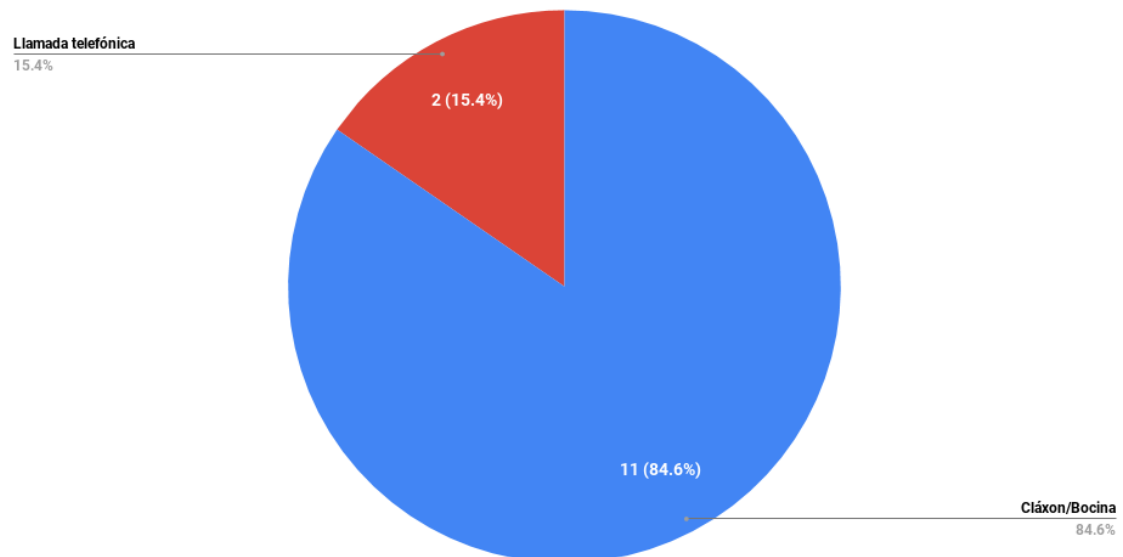
1.- ¿Por qué motivo utiliza el servicio de movilidad escolar para llevar a su hijo al colegio?



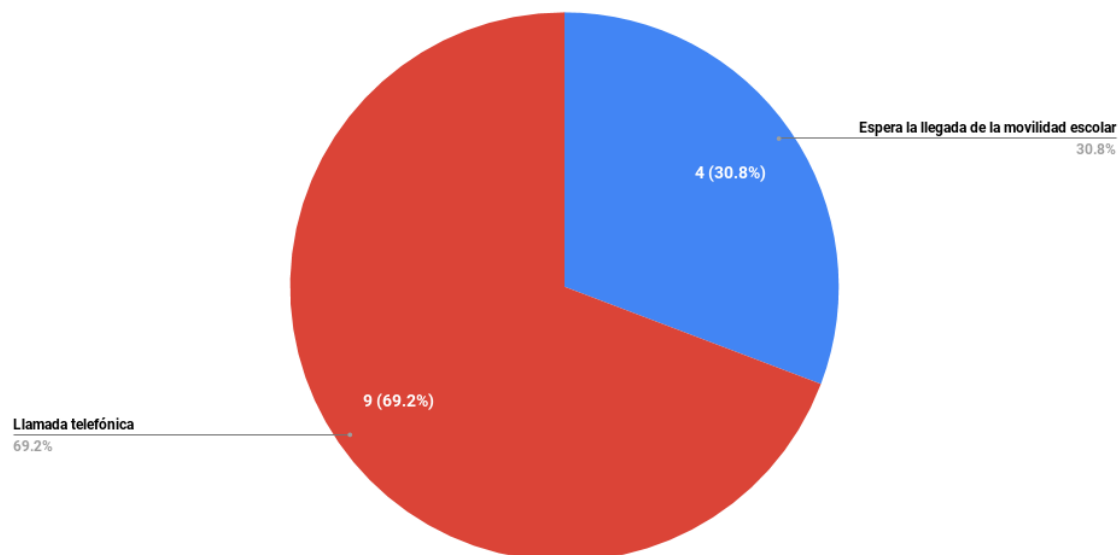
2.- ¿A quién le pertenece la movilidad escolar que contrató?



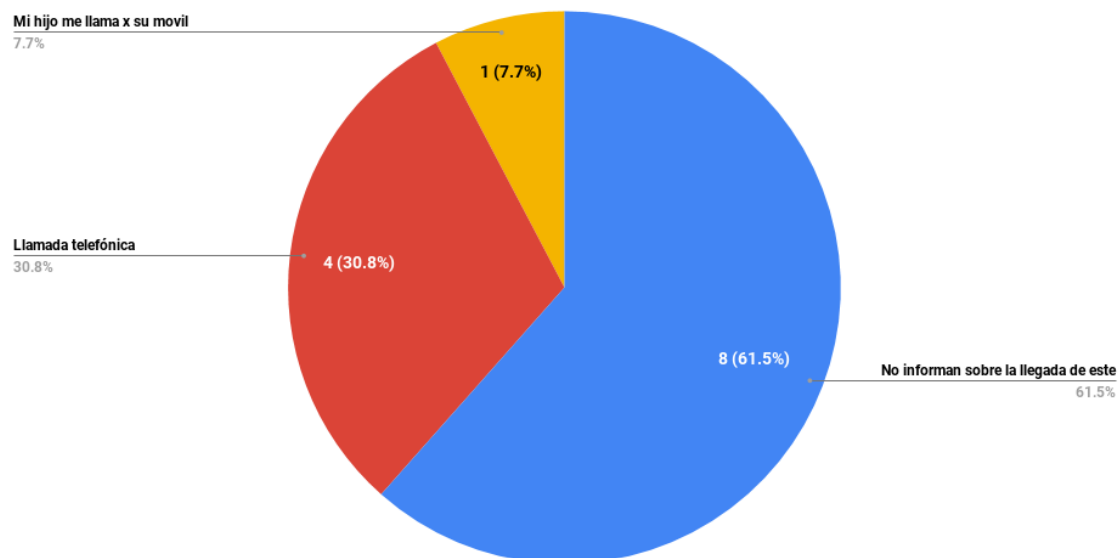
3.- ¿De qué manera le informan que la movilidad escolar llegó a su casa para llevar a su hijo al colegio?



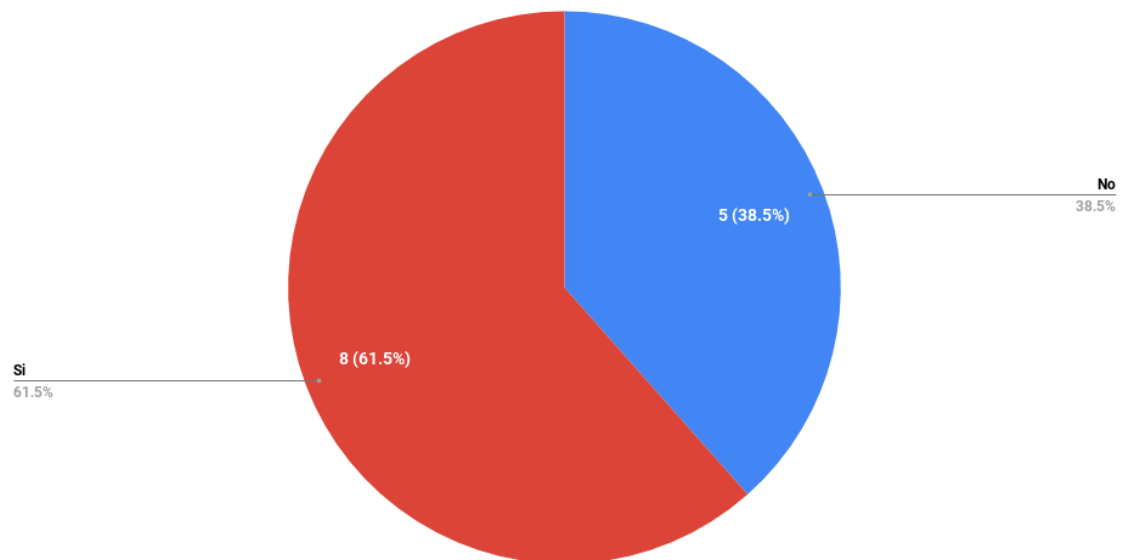
4.- En caso de que su hijo no pudiera ir al colegio por situación de enfermedad o de otra índole ¿Cómo informa a la movilidad escolar sobre esta situación?



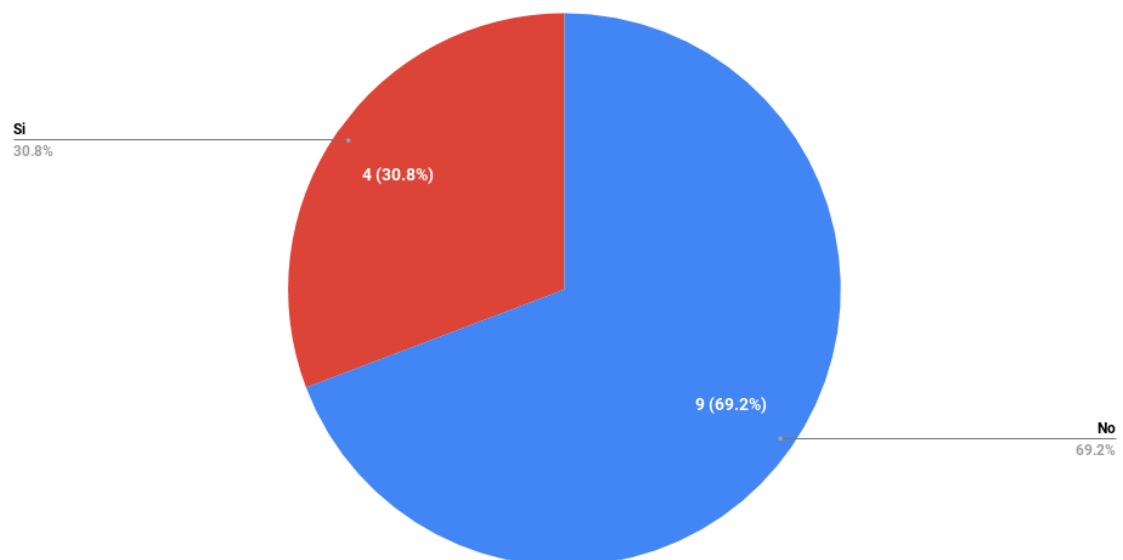
5.- ¿De que manera le informan que la movilidad escolar llegó al colegio a recoger a su hijo para llevarlo a su casa?



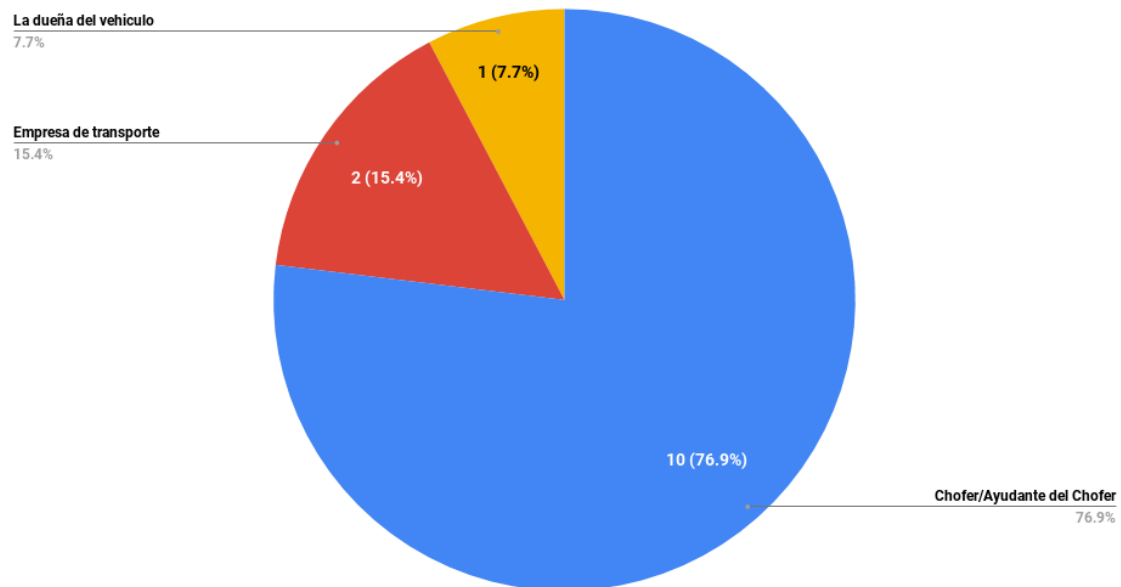
6.- Durante el trayecto de su casa al colegio y/o del colegio a su casa ¿Sabe si su hijo llegó al lugar de destino indicado?



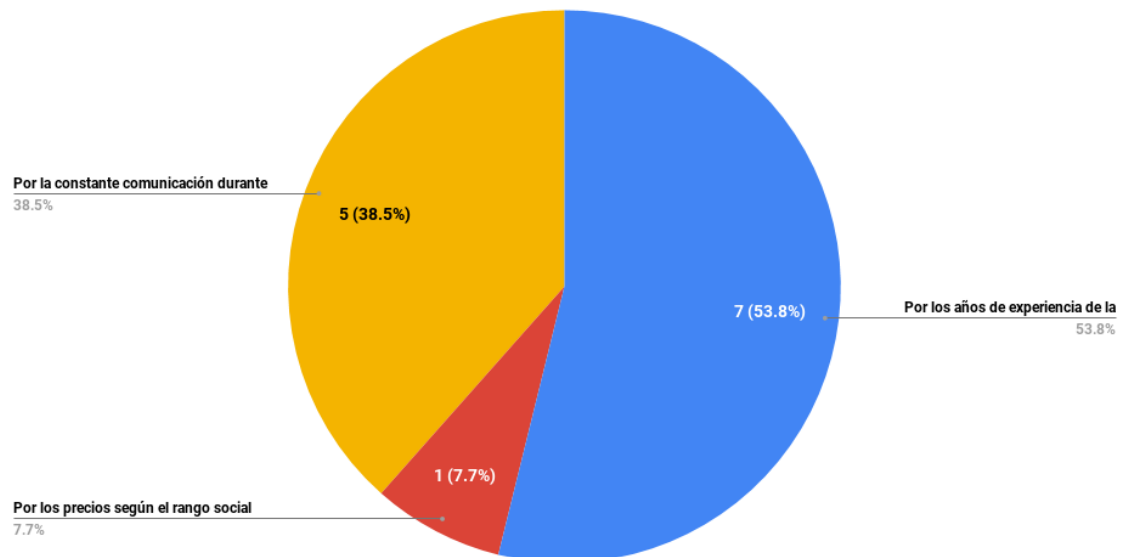
7.- Durante el trayecto de su casa al colegio y/o del colegio a su casa ¿Sabe sobre el estado actual que se encuentra su hijo?



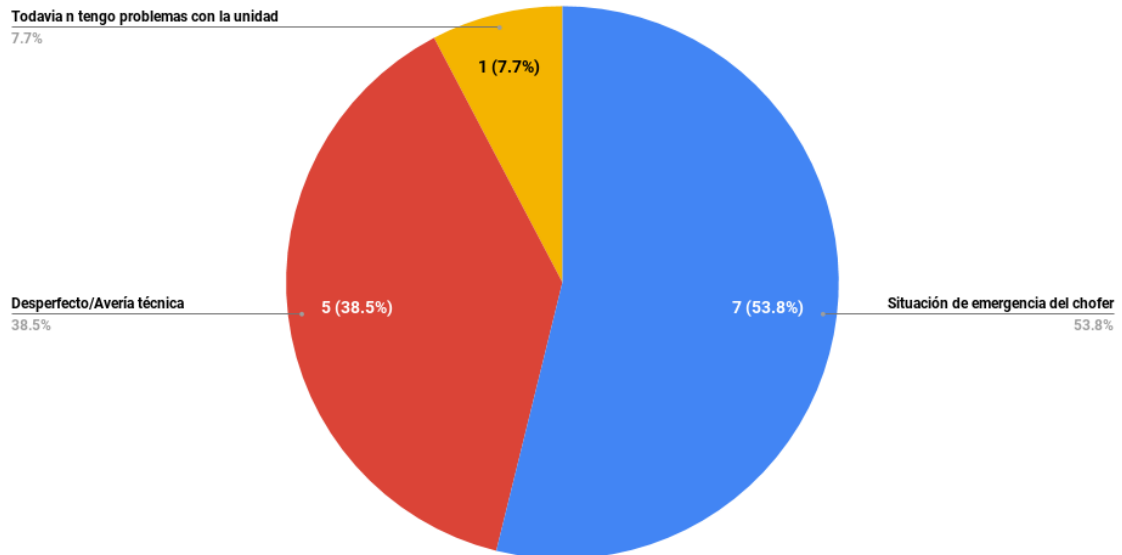
8.- En caso de que ocurriera un incidente durante el trayecto ¿Quién informa sobre estos incidentes?



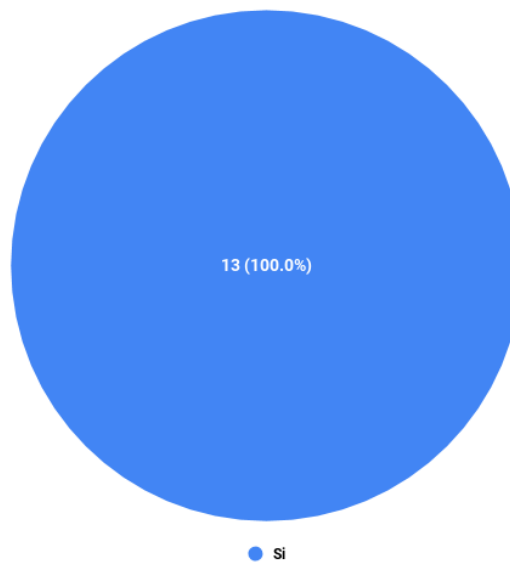
9.- En caso de que tuviera que recomendar usted el servicio de movilidad escolar, ¿Por qué motivo recomendaría este servicio?



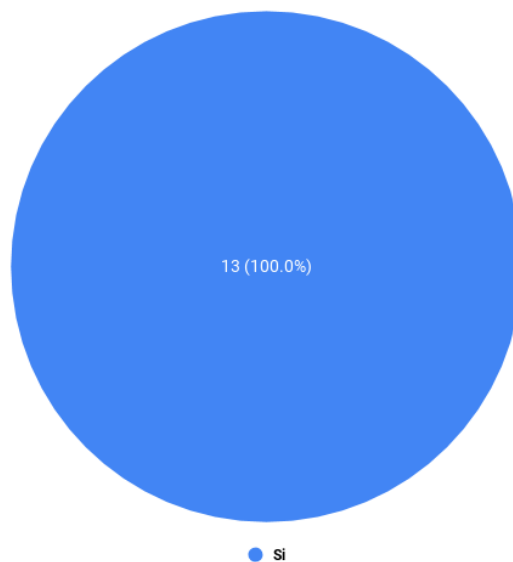
10.- En caso de que la movilidad escolar no llegue a recoger a su hijo a casa o no llegue a recoger a su hijo al colegio, ¿Cuál fue el motivo del porqué no llegó la movilidad escolar?



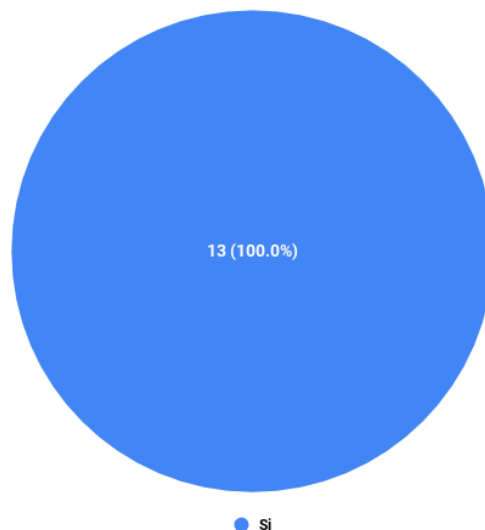
11.- ¿Le gustaría ver en tiempo real a través de una aplicación web de geolocalización el trayecto de la movilidad escolar que traslada a su hijo de casa al colegio y/o del colegio a su casa?



12.- ¿Le gustaría que el aplicativo web le informe la llegada de la movilidad escolar a su casa o al colegio para el traslado de su hijo?



13.- ¿Le gustaría que la aplicación web le alerte con mensajes predeterminados los constantes incidentes que realiza la movilidad escolar durante la trayectoria (semáforo en rojo, congestión vehicular, avería técnica, etc.)?



14.- ¿Considera que la aplicación web sería de gran ayuda para una mejor comunicación entre los padres de familia, empresas de transporte y colegios?

