

ESCUELA DE INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN

**Tesis previa a la obtención de título de
Ingeniera en Tecnologías de la Información
y Comunicación**

AUTOR: Karla Paulina Conde Paccha

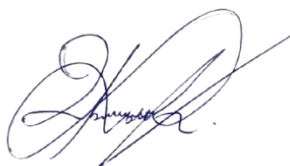
TUTOR: Mgs. Ing. Lorena Conde

**“Desarrollo de una aplicación web aplicando
modelos predictivos para la gestión y notificación
de siniestros de tránsito”**

CERTIFICACIÓN DE AUTORÍA

Yo, **Karla Paulina Conde Paccha**, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido presentado anteriormente para la obtención de ningún grado académico ni calificación profesional, y que he consultado la bibliografía que se detalla.

Cedo mis derechos de propiedad intelectual a la **Universidad Internacional del Ecuador**, para que pueda publicarlo y difundirlo en internet, de conformidad con lo establecido en la Ley de Propiedad Intelectual, su reglamento y demás normativas aplicables.



Karla Paulina Conde Paccha
Autor

APROBACIÓN DEL TUTOR

Yo Lorena Elizabeth Conde Zhingre, certifico que conozco a la autora del presente trabajo de titulación **“Desarrollo de una aplicación web aplicando modelos predictivos para la gestión y notificación de siniestros de tránsito”** Karla Paulina Conde Paccha, siendo la responsable exclusiva tanto de su originalidad y autenticidad, como de su contenido.

.....

Mgs. Ing. Lorena Elizabeth Conde Zhingre,
DIRECTOR DEL TRABAJO DE TITULACIÓN

DEDICATORIA

Dedico este trabajo a mi familia, a mis papás que siempre me han apoyado y me han brindado los mejores estudios, han estado en todas las cosas que he querido hacer, ya sea en los estudios, baile, deportes, teatro y más, siempre han sido mi pilar de donde puedo sostenerme si estoy dudando, para poder volver y hacerlo con más fuerzas para cumplir mis metas.

Dedico esto a mi hermana Mónica que me ha apoyado en cada paso que he dado, así mismo a mis hermanos menores Catalina y Alejandro que son un apoyo espero poder ser un ejemplo. También lo dedico a mis abuelitos que por sus consejos y apoyo he podido ir construyendo un camino, mejorando constantemente.

Este trabajo también dedico a mis docentes y de manera especial a mi tutora, por compartir sus conocimientos, guía y paciencia, aportando a mi formación académica, profesional y personal.

AGRADECIMIENTO

Agradezco, a la Universidad Internacional del Ecuador, por permitirme formarme profesionalmente dentro de sus instalaciones, así como también a cada uno de los docentes quienes han transmitido sus conocimientos, los cuales me han ayudado durante todos los años de estudio y me ayudaran en un futuro en mi vida profesional.

A mi directora de titulación Mgs. Ing. Lorena Conde quien ha sido mi guía en el desarrollo de mi proyecto. Extiendo mi gratitud a todas las personas que me han apoyado durante todo este proceso y haberme brindado el apoyo necesario.

Agradezco a mis compañeros que han sido parte de mi formación académica, quienes me han brindado su amistad y han sido una ayuda en todo el camino transcurrido. Finalmente, agradezco a mi familia que son mi motor, mi impulso para continuar.

INDICE DE CONTENIDO

CARÁTULA.....	1
CERTIFICACIÓN DE AUTORÍA	II
APROBACIÓN DEL TUTOR.....	III
DEDICATORIA.....	IV
AGRADECIMIENTO	V
INDICE DE CONTENIDO	VI
ÍNDICE DE TABLAS	XI
ÍNDICE DE FIGURAS.....	XIII
RESUMEN	15
ABSTRACT.....	16
INTRODUCCIÓN	17
PLANTEAMIENTO DEL PROBLEMA.	18
OBJETIVOS.	19
OBJETIVO GENERAL.	19
OBJETIVOS ESPECÍFICOS.....	19
CAPÍTULO I: ESTADO DEL ARTE	20
ANTECEDENTES	21
1.1 UNIDAD DE CONTROL OPERATIVO DE TRÁNSITO MUNICIPAL	21
1.1.1 ACCIDENTES Y SINIESTROS DE TRÁNSITO	21
1.1.2 SINIESTROS REGISTRADOS.....	22
1.2.3 UNIDAD DE CONTROL OPERATIVO	22
1.2 DATOS	23
1.2.1 RECOLECCIÓN.....	23
1.2.2 CONSERVACIÓN DE DATOS	23
1.2.3 GESTIÓN DE DATOS	23
1.3 ARQUITECTURA.....	24
1.3.1 ARQUITECTURA DE SOFTWARE.....	24
1.3.3 CLIENTE – SERVIDOR.....	24
1.4 MODELADO	24

1.4.1 LENGUAJE UNIFICADO DE MODELADO.....	25
1.5 BACK-END	25
1.5.1 VULNERABILIDAD	25
1.5.2 SEGURIDAD.....	26
1.6 APLICACIONES WEB	26
1.6.1 FUNCIONALIDAD.....	26
1.6.2 VENTAJAS.....	26
1.6.4 EMBEBER APLICACIONES WEB	27
1.7 APLICACIONES MÓVILES	27
1.7.1 NOTIFICACIONES.....	27
1.7.2 PROTOTIPADO Y FRONT-END.....	28
1.7.3 ANDROID	28
1.7.4 VERSIONES DE ANDROID	28
1.7.5 CANTIDAD DE USO DE ANDROID.....	28
1.8 SOFTWARE EFICIENTE.....	29
1.8.1 CALIDAD.....	29
1.8.2 RENDIMIENTO	29
1.9 METODOLOGÍA	29
1.9.1 SCRUM.....	29
1.9.2 SCRUM MASTER.....	30
1.9.3 METODOLOGÍA KANBAN	30
1.10 MODELO PREDICTIVO	30
1.11 TIPOS DE MODELOS PREDICTIVOS.....	31
1.11.1 CLUSTERING	31
1.11.2 REGRESIÓN LOGÍSTICA.....	31
1.11.3 ÁRBOLES DE DECISIÓN	31
1.11.4 RANDOM FOREST	31
1.11.5 XGBOOST	32
1.12 MÉTRICAS DE EVALUACIÓN	32
1.13 RESULTADOS DE MODELADOS	32
1.14 TABLA COMPARATIVA DE MODELOS.....	33

1.15 EVALUACIÓN DEL MODELO.....	34
CAPITULO II: ANÁLISIS Y DISEÑO.....	35
2.1 INVESTIGACIÓN PRELIMINAR	36
2.1.1 RECOPIACIÓN DE DATOS.....	36
2.2 REQUERIMIENTOS.....	37
2.2.1 FUNCIONALES.....	37
2.2.2 NO FUNCIONALES	39
2.3 DESARROLLO DE LA METODOLOGÍA.....	40
2.3.1 COMPARATIVA DE METODOLOGÍAS DE DESARROLLO DE SOFTWARE.....	44
2.3.2 FASES DE METODOLOGÍA.....	45
2.3.3 ENTREGABLES	48
2.4 FASE 1: INICIACIÓN	51
2.4.1 VISIÓN DEL PROYECTO.....	51
2.4.1.1 PROPÓSITO	51
2.4.1.2 ALCANCE.....	51
2.4.1.3 DEFINICIONES, ACRÓNIMOS Y ABREVIACIONES.....	52
2.4.1.4 BENEFICIOS.....	53
2.4.2 OPORTUNIDADES DE NEGOCIO	53
2.4.3 REGLAS DE NEGOCIO	53
2.4.4 MODELO DE BASE DE DATOS.....	54
2.4.5 CASO DE USO	55
2.4.5.1 ESPECIFICACIONES DE CASO DE USO	56
2.4.6 ARQUITECTURA LÓGICA.....	59
2.4.7 PRODUCT BACKLOG	60
2.4.8 LISTA PRIORIZADA.....	62
2.5 FASE 2: TABLA KANBAN.....	62
2.5.1 DISEÑO Y ESTRUCTURA	62
2.5.2 REQUERIMIENTOS VS CASOS DE USO	63
2.6 FASE 3: LANZAMIENTO.....	64
2.6.1 LIBERACIÓN DE ACTIVIDADES EN LA TABA KANBAN	64
2.6.2 RETROSPECTIVA DEL PROYECTO.....	66

2.7 FASE 4: PRUEBAS.....	66
2.7.1 INFORME DE PRUEBAS.....	66
2.8 DESARROLLO DEL MODELO PREDICTIVO	69
2.8.1 MODELO DE CLASIFICACIÓN RANDOM FOREST	71
2.9 ARQUITECTURA DEL MODELO	71
2.10 TÉCNICAS DE ANÁLISIS	71
2.10.1 INTEGRACIÓN DE TÉCNICAS DE AGRUPAMIENTO	72
CAPÍTULO III: DESARROLLO E IMPLEMENTACIÓN	73
3.1 DESCRIPCIÓN DEL PROBLEMA	74
3.2 PROTOTIPADO	74
3.2.1 INICIO DE SESIÓN.....	74
3.2.2 DASHBOARD.....	75
3.2.3 CREAR ROL	76
3.2.4 LISTA DE ROLES.....	77
3.2.5 EDITAR ROL	78
3.2.6 CREAR USUARIO	79
3.2.7 LISTA DE USUARIOS	80
3.2.8 EDITAR USUARIO	81
3.2.9 CREAR GRUPO DE CONTROL	82
3.2.10 LISTA DE GRUPO DE CONTROL.....	83
3.2.11 EDITAR GRUPO DE CONTROL.....	84
3.2.12 CREAR AGENTES	85
3.2.13 LISTA DE AGENTES	86
3.2.14 EDITAR AGENTE	87
3.2.15 AGREGAR SINIESTRO INFORMATIVO	88
3.2.16 LISTA DE SINIESTROS INFORMATIVOS.....	89
3.2.17 EDITAR SINIESTROS INFORMATIVOS.....	90
3.2.18 AGREGAR SINIESTRO OFICIAL CON PARTE.....	91
3.2.19 LISTA DE SINIESTRO OFICIAL CON PARTE.....	92
3.2.20 EDITAR SINIESTRO OFICIAL CON PARTE.....	93
3.3 ANALISIS DE CÓDIGO.....	93

___ 3.3.1 ARQUITECTURA DEL SISTEMA	93
___ 3.3.2 ROLES	94
___ 3.3.3 CONFIGURACIÓN DEL ENTORNO	94
___ 3.3.3.1 REQUISITOS PREVIOS	94
___ 3.3.4 BACKEND – LARAVEL	95
___ 3.3.4.1 CONFIGURACIÓN INICIAL	96
___ 3.3.4.2 CONTROLADORES BASE	96
___ 3.3.4.3 MODELOS	97
___ 3.3.5 FRONTEND – ANGULAR	97
___ 3.3.5.1 ESTRUCTURA DE DIRECTORIOS	97
___ 3.3.5.2 CONFIGURACIÓN ANGULAR	98
___ 3.3.5.3 SERVICIOS	99
___ 3.3.6 COMANDOS PARA EJECUCIÓN	100
3.4 EFICIENCIA DEL PROCESO	100
___ 3.4.1 PLAN DE PRUEBAS E IMPLEMENTACIÓN	100
___ 3.4.2 EVALUACIÓN DE USABILIDAD DEL SISTEMA.	100
___ 3.4.3 EVALUACIÓN DE CALIDAD DE RESPUESTA	102
___ 3.4.4 EVALUACIÓN DE DESEMPEÑO TÉCNICO.	104
___ 3.4.5 EVALUACIÓN DE SATISFACCIÓN GENERAL	105
CAPÍTULO IV: CONCLUSIONES Y RECOMENDACIONES	108
4.1 CONCLUSIONES.	109
4.1.1 CONCLUSIÓN GENERAL	109
4.1.2 CONCLUSIONES SEGÚN OBJETIVOS.	109
BIBLIOGRAFÍA.	111
V: ANEXOS	115

ÍNDICE DE TABLAS

Tabla 1: Comparativa de algoritmos de predicción.....	33
Tabla 2 Resultados de modelos predictivos	42
Tabla 3: Requerimiento funcional 1	37
Tabla 4: Requerimiento funcional 2.....	37
Tabla 5: Requerimiento funcional 3.....	38
Tabla 6: Requerimiento funcional 4.....	38
Tabla 7: Requerimiento funcional 5.....	38
Tabla 8: Requerimiento funcional 6.....	38
Tabla 9: Requerimiento funcional 7.....	39
Tabla 10: Requerimiento funcional 8.....	39
Tabla 11: Requerimiento no funcional 1	39
Tabla 12: Requerimiento no funcional 2.....	39
Tabla 13: Requerimiento no funcional 3.....	40
Tabla 14: Requerimiento no funcional 4.....	40
Tabla 15: Comparativa de metodologías de desarrollo de software	44
Tabla 16: Procesos de fase de iniciación.....	49
Tabla 17: Procesos tabla kanban	49
Tabla 18: Procesos de fase de lanzamiento.....	50
Tabla 19: Procesos de fase de pruebas.....	50
Tabla 20 : Lista Product backlog	60
Tabla 21: Lista Priorizada	62
Tabla 22: CU-001: Iniciar Sesión	56
Tabla 23: CU-002: Registro Siniestro Informativo.....	56
Tabla 24: CU-003: Registro Siniestro con Parte.....	56
Tabla 25: CU-004: Consultar Modelo Predictivo	57
Tabla 26: CU-005: Asignación de Código	57
Tabla 27: CU-006: Lista de Siniestros Informativos	57
Tabla 28: CU-007: Lista de Siniestros con Parte	58
Tabla 29: CU-008: Lista Pública de Siniestros Informativos	58

Tabla 30: CU-09: Notificaciones de Siniestros.....	58
Tabla 31: Casos de uso y requerimientos.....	63
Tabla 32: Liberación sprint 1 - 3	64
Tabla 33: Liberación sprints 4 - 5	64
Tabla 34: Liberación sprint 6	65
Tabla 35: Liberación de sprints 7 - 9	65
Tabla 36: Liberación sprint 10	65
Tabla 37: Liberación general	66
Tabla 38: Módulos y Funciones a evaluar	67
Tabla 39: Evaluación de rendimiento	67
Tabla 40: Datos de restricciones	68
Tabla 41: Pruebas de Seguridad.....	68
Tabla 42: Configuración Inicial	96
Tabla 43: Comandos para ejecución	100

ÍNDICE DE FIGURAS

Figura 1 Resultados gráficos de evaluación de modelos	34
Figura 2: Extracto de código.....	41
Figura 3: Algoritmo K-means.....	43
Figura 4: Fase de iniciación	45
Figura 5: Fase 2 tabla kanban	46
Figura 6: Fase 3 de lanzamiento	47
Figura 7: Fase de pruebas	48
Figura 8: Diagrama de base de datos.....	54
Figura 9: Caso de uso	55
Figura 10: Arquitectura lógica.....	59
Figura 11: Proceso de scrum con product backlog.....	60
Figura 12: Arquitectura del modelo predictivo.....	69
Figura 13: Estructura gráfica de Random Forest.....	71
Figura 14: Comparación de Modelos en función a target Alto riesgo.....	72
Figura 15: Prototipo de login.....	74
Figura 16: Dashboard de administrador	75
Figura 17: Prototipo para crear rol	76
Figura 18: Prototipo de lista de roles	77
Figura 19: Prototipo editar roles	78
Figura 20: Prototipo de agregar usuario	79
Figura 21: Prototipo de lista de usuarios	80
Figura 22: Prototipo de editar usuarios.....	81
Figura 23: Prototipo de agregar grupo de control.....	82
Figura 24: Prototipo de lista de grupo de control	83
Figura 25: Prototipo editar grupo de control	84
Figura 26: Prototipo de agregar agentes	85
Figura 27: Prototipo lista de agentes.....	86
Figura 28: Prototipo editar agente	87
Figura 29: Prototipo agregar siniestro informativo	88

Figura 30: Prototipo lista de siniestros informativos	89
Figura 31: Prototipo editar siniestro informativo	90
Figura 32: Prototipo agregar siniestro oficial con parte	91
Figura 33: Prototipo lista de siniestros oficiales con parte	92
Figura 34: Prototipo editar siniestros oficiales con parte	93
Figura 35: Extensiones Recomendadas para VS Code.....	94
Figura 36: Configuración del Workspace	95
Figura 37: Estructura de directorios	95
Figura 38: Referencia de controller	96
Figura 39: Referencia de un model.....	97
Figura 40: Estructura de frontend.....	97
Figura 41: Estructura de app.module.ts	98
Figura 42: Referencia al archivo auth.service.ts.....	99
Figura 43: Métricas de evaluación de usabilidad	101
Figura 44: Métricas de rendimiento.....	101
Figura 45: Grafica de resultados de encuesta realizada.....	101
Figura 46: Prueba de navegación.....	102
Figura 47: prueba de calidad de respuesta.....	103
Figura 48: Prueba de carga de datos	103
Figura 49: Prueba técnica	104
Figura 50: Evaluación de exportación	105
Figura 51: Funcionalidad del sistema.....	105
Figura 52: Usabilidad para los conductores	106
Figura 53: Prueba de aceptación.....	107

RESUMEN

El presente proyecto aborda la necesidad de identificar de manera temprana los lugares con más siniestros de tránsito, los mismos que afectan la eficiencia operativa a la Unidad de Control Operativo de Tránsito (UCOT) de la provincia de Loja, según los datos oficiales publicados por el municipio de Loja, en el año 2024 reflejando un total de incidentes de 630 siniestros. Estos datos ponen en evidencia la magnitud del problema estos siniestros han dejado un total de 465 heridos. Los siniestros se dan las veinticuatro horas del día los siete días de la semana, esto representa una creciente preocupación para las autoridades gubernamentales y la ciudadanía. Mediante el uso de modelos predictivos basados en aprendizaje automático, se da una solución analítica apoyada en modelos de inteligencia artificial de detección y predicción sustentándose en datos históricos y la comparación de distintos algoritmos de aprendizaje automático, se diseñó una arquitectura híbrida que combina técnicas de clustering (K-Means) para identificar zonas denominadas como “puntos negros” es decir lugares con incidencias en siniestros y algoritmos de clasificación (Random Forest) para predecir la probabilidad de ocurrencia de dichos sucesos bajo ciertas condiciones específicas, permitiendo anticipar eventos y visualización de patrones de riesgo de manera estadística, además con un aplicativo web que recolecte todos estos datos obteniendo una base de datos centralizada se hará más fácil el trabajo realizado por los agentes de tránsito esto será una herramienta eficaz y de apoyo estratégico para la planificación de operativos, toma de decisiones y el despliegue de recurso humano, siendo un avance significativo para el control vial permitiendo a las instituciones anticiparse a problemas críticos que afectan la seguridad.

Palabras Clave: Modelo predictivo, Inteligencia artificial, Detección, Siniestros, Tránsito.

ABSTRACT

This project addresses the need to identify early on the locations with the highest number of traffic accidents, which affect the operational efficiency of the Traffic Control Unit (UCOT) in the province of Loja, according to official data published by the municipality of Loja, which in 2024 reflected a total of 630 accidents. These data highlight the magnitude of the problem, as these accidents have left a total of 465 people injured. Accidents occur 24 hours a day, seven days a week, which is a growing concern for government authorities and citizens. Using predictive models based on machine learning, an analytical solution was provided, supported by artificial intelligence detection and prediction models based on historical data and the comparison of different machine learning algorithms. A hybrid architecture was designed that combines clustering techniques (K-Means) to identify areas known as “black spots” i.e., places with high accident rates, and classification algorithms (Random Forest) to predict the probability of such events occurring under certain specific conditions, allowing events to be anticipated and risk patterns to be visualized statistically. In addition, a web application that collects all this data into a centralized database will make the work easier.

Keywords: Predictive model, Artificial Intelligence, Detection, Accidents, Traffic.

INTRODUCCIÓN

Una aplicación web desarrollada para el registro, notificación y predicción de siniestros de tránsito en Loja - Ecuador esto representa un avance tecnológico significativo para mejorar la comunicación entre la unidad de control operativa de tránsito y la población y aportando a la prevención. Dando como resultado una aplicación web que recolecta datos sobre accidentes con rapidez y hace un análisis predictivo de esta forma ayudar a las autoridades a reducir tiempos de respuesta y mejorar vías.

Los agentes de tránsito diariamente se enfrentan a grandes retos debido a la tasa de accidentes viales que se elevan por imprudencia de conductores o peatones, fallos en las vías, problemas con las señales de tránsito, entre otros. Y ya que no cuentan con una aplicación unificada en donde se puedan reflejar esta data no se tiene un dato real de los puntos negros de la ciudad (lugares con más incidencias de tránsito), por ello el planteamiento de este proyecto no solo ayudará a agilizar los datos, sino que también contribuirá en gran medida a plantear estrategias de prevención para siniestros, recopilar de las zonas más propensas a accidentes y detectar razones más comunes entre los conductores.

Como primera solución esta aplicación web será propuesta para el uso de los agentes de tránsito de la ciudad de Loja, de esta forma se probara su efectividad y se obtiene datos reales en cuanto a las mejoras que se pueden realizar en la ciudad, trabajando con la unidad de control del municipio de la ciudad el proyecto tendrá beneficios claros por un lado se registrara accidentes de manera eficiente, permitiría a los ciudadanos consumir la información actualizada sobre siniestros y posible vías cerradas además de realizar un análisis predictivo de los siniestros.

En conclusión, este planteamiento tecnológico representa una innovación en la gestión de siniestros y un aporte para mejorar la seguridad vial junto con la comunicación efectiva con la sociedad, brindando como aporte a las autoridades de la ciudad una herramienta capaz de registrar y realizar análisis predictivo en base a los datos históricos ingresados, en las versiones posteriores del aplicativo se logrará implementar a nivel nacional.

PLANTEAMIENTO DEL PROBLEMA.

En la ciudad de Loja los siniestros de tránsito se han dado la mayoría de las veces por descuido o por alguna falta cometida por los conductores, por lo que representan una de las mayores causas de mortalidad a pesar de los esfuerzos realizados por las autoridades con propuestas que minimicen estos peligros en la vía, una mejora significativa para el problema se puede presentar como una herramienta tecnológica como el presente proyecto, que permitirá el registro y notificación de siniestros en tiempo real, alertando a otros conductores de lo sucedido obteniendo como resultado una comunicación óptima entre los organismos gubernamentales y la ciudadanía. (Ecu911, 2023)

En la actualidad el registro llevado por parte de los agentes de tránsito se realiza de forma manual, por lo que debido a este ineficiente proceso puede existir el riesgo de pérdida, adulteración o daño en estos documentos, y tampoco es un proceso eficaz y eficiente, por este motivo se presenta dificultad para la llegada de los servicios de emergencia como es asistencia médica y asistencia vehicular a los lugares del accidente.

A nivel de países se puede sacar en relevancia a Colombia que tiene algunas aplicaciones que están ligadas ligeramente al presente proyecto, a nivel mundial se puede destacar diferentes sistemas digitales para el registro de siniestros y gracias a esto de acuerdo con los datos presentados se puede analizar una reducción significativa en los tiempos de respuesta de los servicios.

En conclusión, debido a esta problemática surge la necesidad de desarrollar una aplicación web que simplifique el proceso que realizan los agentes de tránsito al momento de levantar un parte después de un accidente, adicional a esto se presentara la notificación de siniestros a los conductores y la ciudadanía en tiempo real, esta tecnología permitirá un mejor manejo de los datos.

De preguntas de investigación en las que este proyecto se basará será: ¿Cuál es el impacto que tiene la ausencia de una herramienta tecnológica para el manejo de siniestros de tránsito en Loja?, ¿De qué manera una aplicación exclusiva para agentes de tránsito puede mejorar la comunicación entre la Unidad de control operativo de tránsito (UCOT) y la ciudadanía?, ¿Cómo influye el tiempo de respuesta de los agentes de tránsito y las líneas de emergencia con una aplicación?

OBJETIVOS.

Objetivo General.

Desarrollar una aplicación web aplicando modelos predictivos para la gestión, y notificación de siniestros de tránsito.

Objetivos Específicos.

- Analizar la situación actual del proceso que realizan los agentes de la UCOT de la ciudad de Loja enfocado a siniestros de tránsito.
- Diseñar la arquitectura de software y el modelo de análisis predictivo para el desarrollo de la aplicación web.
- Desarrollar e implementar la aplicación web basándose en la arquitectura de software propuesta.
- Realizar un plan de pruebas y validar la aplicación web con los agentes de tránsito.

CAPÍTULO I: ESTADO DEL ARTE

ANTECEDENTES

En la ciudad de Loja – Ecuador no se ha encontrado ninguna aplicación relevante que tenga funcionalidades similares a las presentadas en este proyecto, sin embargo, tenemos algunas como la de Ecu-911 aunque en esta aplicación pueden realizar denuncias sobre diferentes emergencias cualquier usuario, por lo que esta aplicación no es igual a la presentada.

Los agentes de tránsito ya no deberán llevar un control manual y tener que subir la información en un centro de acopio, podrían realizar el reporte conforme se va desarrollando y levantando datos en el accidente, dando un servicio extra a los usuarios conductores que obtienen información de la aplicación para su lectura sobre el cierre de vía en diferentes sectores de la ciudad, la digitalización se agilizaría y se trabajaría con una base de datos específica para los siniestros evaluando la eficacia de las unidades y de las señales de tránsito ubicadas en los diferentes sitios de la ciudad.

InfoTránsito Ecuador la misma que tiene un enfoque de recolección de datos para dar informe a la ciudadanía a pesar de no ser una aplicación oficial es usada para dar informes policiacos y personales, está enfocada al ámbito de noticias para todas las personas que la usen y tengan necesidad de revisar la información.

A diferencia que el país vecino Colombia que tiene waze aunque esta es una aplicación global es altamente usada en Colombia y Perú para la navegación proporcionando información sobre tránsito y tráfico, pero al igual que otras aplicaciones los reportes son realizados por usuarios de la aplicación por lo que los convierte en una fuente no oficial.

1.1 Unidad de control operativo de tránsito municipal

1.1.1 Accidentes y siniestros de tránsito

Los siniestros de tránsito se dan en todas las ciudades y los países siendo sucesos que son evitables o predecibles, que se dan por acciones y sucesos premeditados o consecuentes, mientras que si hablamos de un accidente se entiende como circunstancias que son totalmente impredecibles

que no tienen un precedente como situaciones de afectaciones por el clima o totalmente extremas al control humano, según Sarra por esta razón la mayoría de los choques o situaciones desafortunadas de tránsito son consideradas siniestros, ya que generalmente son evitables o predecibles por acciones de los diferentes conductores. (Sarra, 2024)

1.1.2 Siniestros registrados

El INEC en Ecuador es un instituto nacional de estadísticas y censos que es la encargada de coordinar y evaluar información proveniente de operaciones estadísticas oportunas produciendo data para el uso público siendo un referente para estudios especializados o toma de decisiones para instituciones públicas y privadas o planificación nacional. (INEC, El nuevo Ecuador, 2023)

De acuerdo con el Inec existen estadísticas de los siniestros de tránsito que provén información sobre todos los sucesos ocurridos en territorio nacional, de los últimos datos recolectados podemos tener una tabla comparativa entre el año 2020 y 2021 teniendo como resultado en el año 2021 se tiene un mayor resultado con datos 21352 mientras que en el 2020 se mantenía con 16972, siendo ya un número elevado. (INEC, INEC, 2022)

1.2.3 Unidad de Control Operativo

El análisis del estado del arte está basado en el estudio de la unidad de control operativa de tránsito UCOT en la ciudad de Loja que es la encargada del manejo y control del transporte terrestre, tránsito y seguridad vial de todo el cantón, la misma que es una institución perteneciente al municipio de Loja siendo uno de sus objetivos fomentar la conciencia vial disminuir los siniestros de tránsito e incrementar la eficiencia y la cobertura dentro de Loja para mantener una capacitación constante de todo su personal y la ciudadanía de esta forma ayudara la aplicación desarrollada dentro de este proyecto. (Loja, 2024)

1.2 Datos

1.2.1 Recolección

Debido a este precedente, el proyecto web va a realizar una recolección de datos que servirán en un futuro para tener un número más exacto de estos acontecimientos además de lugares con más frecuencia y vulnerables dejando abierta la posibilidad de realizar minería de datos gracias a la información obtenida; Clientify habla sobre la recolección de datos como un proceso totalmente necesario dentro de cualquier empresa o software que busque tomar una decisión o generar un dataset, siendo un proceso fundamental en investigación científica, estudio de mercado y análisis de comportamiento. (Clientify, 2024)

1.2.2 Conservación de datos

Una vez recolectados los datos del aplicativo web se debe guardar lo que consiste en conservar la información empleando tecnología específica para este proceso, de esta forma se encuentran disponibles siempre que se necesite realizar alguna consulta evitando la pérdida de información o redundancia de datos. (Packard, 2025)

1.2.3 Gestión de datos

El proceso de gestión de datos mientras más grande se vuelva la base de datos más difícil será garantizar la consistencia, precisión y seguridad es por eso que en la arquitectura de software se debe definir un buen flujo y modelos aptos para preservar el origen y destino de los datos, en concreto la gestión de datos es una disciplina diseñada para ayudar a los usuarios a operar y consultar un aplicativo. (Pedersen, 2021)

1.3 Arquitectura

1.3.1 Arquitectura de software

Arquitectura de software se entiende como la estructura esquelética de un sistema y las relaciones que existen entre las partes que se van a construir y el flujo que tendrá entre ellas. Todo esto de acuerdo con Software Engineering Institute (SEI) que también menciona que influye en el comportamiento global del aplicativo y en el proceso de construcción ayuda a que todas las personas involucradas puedan entender y analizar el sistema. (Institute, 2025)

1.3.2 Tipos de arquitectura

Los tipos de arquitectura entenderlos y definir un tipo de forma adecuada puede direccionar el camino que va a tener el proyecto de software cada tipo tiene su propia filosofía y aplicaciones específicas como orientada a servicios, orientada a objetos, arquitectura en capas, basada en microservicios, cliente servidor, arquitectura sin servicio, para esto se debe definir bien los requisitos y los factores que afectaran al desarrollo. (UNIR, 2024)

1.3.3 Cliente – servidor

Arquitectura tipo cliente servidor, vamos a ampliar el tema, ya que el presente proyecto será desarrollado es un modelo de comunicación que permite distribuir las tareas dentro de una red de ordenadores entre el servidor que podría ser un hardware o almacenamiento en la nube que almacenara todo el contenido de la aplicación un servidor siempre aceptara las peticiones de un cliente que en este caso los usuarios de frontend que usan el aplicativo. (IONOS, 2023)

1.4 Modelado

Mientras que la arquitectura de software es la estructura general del proyecto el modelado es un diseño de alto nivel que es usado para describir con mayor precisión datos abstractos sobre sistemas de esta forma se facilita la comprensión y el análisis del software para de esta forma

entender con más precisión la arquitectura, si el modelo de software es idóneo el flujo y gestión de datos está bien estructurada. (Mert Ozkaya, 2020)

1.4.1 Lenguaje unificado de modelado

Para modelado el más usado es el UML, sin embargo existen diferentes tipos de modelado, pero el estándar es el mencionado, lenguaje Unificado de Modelado, que fue creado para ser algo visual y sistemático basándonos en eso para la implementación y desarrollo de sistemas de software esto es comparable a unos planos de una casa que se genera antes de la construcción, lo que permite describir límites y comportamiento del sistema. (IBM, 2021)

1.5 Back-end

Back-end es la parte del desarrollo que nace desde los modelos uml que se generan es aquel código que se ejecuta en la parte del servidor administrador funcionalidades, interacciones del usuario mediante el front-end y solicitudes de API, para el desarrollo de esto se suelen tener bugs durante la codificación que pueden darse por alguna mala interacción entre backend y frontend o errores de lógica, los bugs son aquellos errores que pueden tener algún impacto en el sistema o pueden afectar a la seguridad del sistema siendo una vulnerabilidad. (Service, 2024)

1.5.1 Vulnerabilidad

Las vulnerabilidades en los proyectos de software se dan con frecuencia ya que de acuerdo con Incibe son fallos técnicos o deficientes de algún aplicativo en donde un usuario no legítimo pueda acceder a la información. Actualmente, se descubren vulnerabilidades dentro de las aplicaciones gracias a los hackers o personas que se encargan de vulnerar sistemas y robar información valiosa. (INCIBE, 2020)

1.5.2 Seguridad

La seguridad de un proyecto de software es esencial en este contexto el proyecto tendrá un acceso exclusivo de modificación para los agentes de UCOT, pero para esto debemos entender sobre la seguridad de datos siendo la práctica de proteger toda la información digital del acceso no autorizado esto implica implementar herramientas y tecnológicas que mejoren la seguridad del aplicativo esto según IBM. (IBM, IBM, 2024)

1.6 Aplicaciones web

Una aplicación web son programas con grandes funciones y diferentes módulos que por lo general se almacenan en un servidor remoto de internet a la que puedes acceder mediante un navegador a través de una URL, estas son desarrolladas con tecnologías web como html, css entre otros para poder cargar este tipo de servicios lo único que se necesita es algún dispositivo con conexión a internet wifi y tus datos de acceso. (Esic, 2024)

1.6.1 Funcionalidad

Una de las principales funcionalidades son su capacidad de sincronización en multidispositivo, ya que trabajan con servidor web tienen la facilidad de intercambio de información para que todos reflejen los mismos cambios aun que se conecten desde un celular, tableta o computadora además de contar con altos estándares de seguridad, como mecanismos integrados en la aplicación. (Lymanska, 2023)

1.6.2 Ventajas

La ventaja significativa para las aplicaciones web es que son una solución rápida y barata, además no es necesario usar versiónamiento porque se pueden ir actualizando en la nube sin afectar al uso de los usuarios, para dar soporte a aplicaciones web es más fácil trabajar con desarrolladores de diferentes categorías, ya que estas usan lenguajes conocidos y comunes, la navegación es sencilla porque pueden verse de diferentes sistemas operativos. (Rivera, 2024)

1.6.3 Terminología de web

Cuando cualquier servicio es accesible desde un navegador de internet de acuerdo a Adán Volle se escriben en lenguajes compatibles con los principales navegadores utilizando dos partes de código enfocadas a diferentes funcionalidades como es el backend para el lado del servidor y la parte de frontend o interfaces para la parte del cliente. (Volle, 2022)

1.6.4 Embeber aplicaciones web

Para incrustar o insertar contenido elaborado en cierto lenguaje de programación compatible como html que es uno de los más usado, esto se lo conoce como embeber también es un tipo específico de programación que permite la creación de dispositivos para consumidores es ideal para facilitar el uso o la forma de acceder a ella en resumen de acuerdo con Ingrith código o parte de código fuente desarrollado en otro lenguaje de programación. (Gomez I. , 2023)

1.7 Aplicaciones móviles

Las aplicaciones móviles son un software diseñado para funcionar netamente en dispositivos móviles como celulares inteligentes o tabletas que requieren ser instaladas a diferencia de las aplicaciones web que ingresan directamente a un navegador para su funcionamiento, por lo cual se denomina app que proviene de la palabra en ingles application en abreviación “APP”. (Calvo, 2024)

1.7.1 Notificaciones

Actualmente se usa las notificaciones push en tiempo real de diferentes anuncios, dependiendo del tipo de desarrollo va su implementación ya que existen para web y móvil, según IBM en aplicaciones web que usan el navegador para su funcionamiento pueden aparecer en el computador conectándose con los sistemas operativos o navegadores compatibles, mientras que en aplicaciones móviles en donde se presencian con mayor frecuencia suelen ser ventanas emergentes creando una

mayor participación con los usuarios que han descargado la aplicación o hacer que por la influencia de esta notificación algún usuario haga una acción específica. (Statista, 2021)

1.7.2 Prototipado y front-end

Para la codificación del aplicativo se debe trabajar con un prototipado de front-end que según Tom Lowry, Director de Advocacy de Figma dice que "Prototipado es una forma rápida de ir desde bocetos o planos hasta una simulación donde puedes obtener retroalimentación de los usuarios" identificándose como una parte esencial entre la experiencia de usuario para de esta manera poder probar ideas y diseños más rápido sin tener que realizar todo un proceso de desarrollo obteniendo una alta calidad de software. (Rikke Friis Dam, 2021)

1.7.3 Android

Android es un sistema operativo que fue diseñado para dispositivos móviles inteligentes principalmente con pantalla táctil, pero también se lo puede encontrar en relojes o en equipos para carros la primera versión salió en el 2005 pero al mercado se presentó en el año 2007 el código fuente es conocido como Android open y destaca por ser el sistema operativo más usado con el 78,5% de usuarios en el mundo usándolo. (Adeva, 2025)

1.7.4 Versiones de Android

Versiones de Android en el transcurso de todos estos años se han presentado diferentes versiones desde la kitkat, pie, android 10 hasta actualmente la versión 15 y Android 16 que está en beta, para el testeado que realicen diferentes usuarios, lo recomendable para el desarrollo de aplicaciones según el sitio oficial de desarrollo en Android es de la versión 10 para arriba, ya que cuenta con diferentes parches de seguridad. (Android, 2025)

1.7.5 Cantidad de uso de android

La mayoría de las personas usuarias de un dispositivo móvil presentan el uso del sistema

operativo Android por esta razón la primera versión del proyecto será solo para estos móviles, según un estudio realizado por el ministerio de telecomunicaciones en Ecuador el 93,75% de los usuarios de teléfonos móviles usan Android. (Telecomunicaciones, 2021)

1.8 Software eficiente

1.8.1 Calidad

En el desarrollo de un aplicativo se debe tomar en cuenta la calidad de software que se puede categorizar entre cualidades como su rendimiento, escalabilidad, eficiencia y fiabilidad u otras cualidades de cómo se debe estructurar u organización del producto de software, para medir la calidad se debe especificar la facilidad con la que se puede resolver o atender algún problema que se dé mientras esté en uso. (Gomez, 2024)

1.8.2 Rendimiento

El rendimiento de software es un aspecto del desarrollo que se especializa en medir y mejorar el código, es la eficiencia del aplicativo usando recursos para hacer lo que has modelado, sin usar más recursos de los que se debe como consumo de energía uso de cpu, etc. Con un buen rendimiento se tiene una buena respuesta de las aplicaciones para mejorar la experiencia lo más posible. (Bridge, 2023)

1.9 Metodología

1.9.1 Scrum

La metodología Scrum es un marco de trabajo o framework que se utiliza dentro de equipos que manejan proyectos complejos, la misma que tiene como finalidad la entrega de valor en periodos cortos de tiempo y para ello se basa en tres pilares: la transparencia, inspección y adaptación. (Abellán, 2020)

Al considerarse dentro de las metodologías ágiles, Scrum se basa en aspectos como, la flexibilidad en la adopción de cambios y nuevos requisitos durante un proyecto complejo, el factor humano, la colaboración e interacción con el cliente y sobre todo el desarrollo iterativo como forma de asegurar buenos resultados.

1.9.2 Scrum Master

El Scrum Master es quien lidera el equipo de Scrum y mantiene a los miembros enfocados en los principios del marco de trabajo. Además, ayuda a los encargados del proyecto para la distribución y sus organizaciones al compartir las prácticas de esta metodología con otras personas. (Asana, 2021)

1.9.3 Metodología Kanban

La metodología Kanban es tan eficiente como efectivo. Forma parte de las metodologías ágiles y su objetivo es gestionar la realización de las tareas hasta su finalización, al ser un método visual permite que con un simple vistazo se conozca el estado de los proyectos y se puedan asignar nuevas tareas de manera ordenada.

1.10 Modelo predictivo

Para hacer una predicción usamos modelos de inteligencia artificial con técnicas para analizar datos históricos y predecir sucesos futuros esto segun Honan 2017 donde explica que por lo general cuando se usan estas técnicas es para tratar con grandes cantidades de datos y de esta manera evitar sucesos o tomar mejores decisiones de acuerdo con los resultados obtenidos adelantándose a eventos. Los modelos usan diferentes algoritmos como: regresión logística, árboles de decisión, random forest, XGBoost, redes neuronales. (B.H.B. Honan, 2017)

1.11 Tipos de modelos predictivos

1.11.1 Clustering

Un tipo de modelo de predicción es clustering que según la universidad de Oviedo es un algoritmo de aprendizaje no supervisado que se usa para modelos de predicción, se desarrolla con la clasificación de objetos o datos dando como resultado una agrupación de objetos en k conjuntos basándose en sus características, esto se realiza minimizando la suma de distancias entre cada objeto y el centroide, el algoritmo cuenta de tres pasos principales: Inicialización, asignación de objetos a los centroides y la actualización de centroides. (Oviedo, n.d.)

1.11.2 Regresión logística

De acuerdo con el artículo de Martínez y Pérez el tipo de modelo de predicción conocido como regresión logística es una técnica estadística multivariante que relaciona las variables dependientes no métricas o categóricas y variables dependientes métricas cuantitativas o categorías, permitiendo comprobar una hipótesis o relaciones casuales. (J.A Martínez Pérez, 2024)

1.11.3 Árboles de decisión

Según Ferrero en el 2020 los árboles de decisión sirven para organizar decisiones y sus posibles resultados o consecuencias formando un diagrama similar a un árbol con muchas ramas, esto se realiza mediante preguntas binarias o categóricas que van dividiendo los datos en subconjuntos facilitando la clasificación y la visualización de los resultados. Sin embargo, cuando se trata de grandes volúmenes de datos el resultado es mejor con un algoritmo como random forest. (Ferrero, 2020)

1.11.4 Random forest

En el artículo de Niklas Donges explica que random forest es un algoritmo de aprendizaje automático, fácil y flexible para el uso que refleja excelentes resultados sin demasiados ajustes en

la mayoría de los casos, este es usado para clasificación o regresión, ya que crea un conjunto de múltiples árboles de decisión entrenados por el método bagging para obtener una predicción o resultado más preciso, otra forma de trabajar con árboles de decisión es con el algoritmo xgboosts. (Niklas Donges, 2024)

1.11.5 XGBoost

Tyagi menciona que XGBoost es un algoritmo de aprendizaje automático que se centra en el marco de potenciación de gradiente utilizando árboles de decisión como referentes base para el aprendizaje y empleando técnicas de regularización mejorando la generalización del modelo, ya que se reconoce por la eficiencia computacional dando como resultado un procesamiento eficiente y un análisis profundo de las características de los datos. Para estos algoritmos se usan diferentes métricas de evaluación para de esta forma comprobar que se encuentra en un estado óptimo el modelo de predicción. (Tyagi, 2025)

1.12 Métricas de evaluación

Cuando hablamos de modelos predictivos se pueden obtener diferentes tipos de salidas de acuerdo con Chauhan esto puede ser por clase o por probabilidad, existen algunas métricas de evaluación del modelo de aprendizaje automático como: matriz de confusión, exactitud, precisión, exhaustividad, puntuación F1, especificidad, curva de características operativas del receptor, pérdida logarítmica, índice Jaccard y gráfico de kolmogorov smirnov. Los modelos predictivos con estas métricas pueden ser utilizados en diferentes ámbitos y ramas, como el caso de estudio de la unidad de control en Loja. (Chauhan, 2023)

1.13 Resultados de modelados

Existen métricas de evaluación del modelo predictivo como se explicó en el párrafo anterior para la matriz de confusión los resultados son comparados mediante tablas con valores reales de ahí obtenemos un valor más acertado en cuanto a los resultados, exactitud esta es obtenida mediante una fórmula indicado el porcentaje de las predicciones correctas, puntuación F1, esto es

representa la fórmula entre precisión, el valor de los datos positivos, con la exhaustividad los valores positivos reales mientras que especificidad, mide que tan bien el modelo reacciona e identifica a los negativos reales. (Chauhan, 2023)

1.14 Tabla comparativa de modelos

Tabla 1: Comparativa de algoritmos de predicción

Algoritmo	Tipo	Ventajas	Desventajas	Adecuación al caso
Clustering (K-Means)	Aprendizaje no supervisado	Detecta patrones y agrupa zonas de alto riesgo sin etiquetas previas.	Requiere análisis previo de número de clusters o densidad.	Ideal para encontrar “puntos negros” o zonas de riesgo.
Regresión Logística	Supervisado (clasificación)	Fácil de interpretar; útil si se tienen variables categóricas claras.	Poco eficaz con relaciones no lineales o datos complejos.	Útil para estudios simples con variables bien definidas.
Árboles de Decisión	Supervisado (clasificación)	Interpretable; estructura clara para tomar decisiones.	Riesgo de sobreajuste; menor precisión frente a métodos más robustos.	Bueno si se busca explicabilidad en zonas con alto número de casos.
Random Forest	Supervisado (clasificación)	Maneja grandes volúmenes de datos; reduce sobreajuste.	Modelo más pesado; menos interpretable que un solo árbol.	Alta precisión para predicción de zonas o condiciones de siniestros.
XGBoost	Supervisado (clasificación)	Alto rendimiento y eficiencia; maneja outliers y valores faltantes.	Requiere ajuste fino de hiperparámetros.	Excelente si se busca un modelo de alta precisión.
Redes Neuronales	Supervisado (clasificación / regresión)	Muy potente para relaciones complejas y grandes volúmenes de datos.	“Caja negra”; requiere mucho procesamiento y datos.	Útil si se tienen muchos datos etiquetados, pero difícil de explicar.

Nota: Esta tabla muestra las características de cada uno de los modelos

En base a la *tabla 1 comparativa de algoritmos de predicción* se detectó que la mejor opción a seguir de acuerdo con ventajas y desventajas es el algoritmo ***Clustering K-means + random forest*** para clasificar zonas de alto riesgo y predecir los factores influyentes en siniestros de tránsito en función de datos históricas. Para estar seguros de que el modelo predictivo que se eligió es la mejor

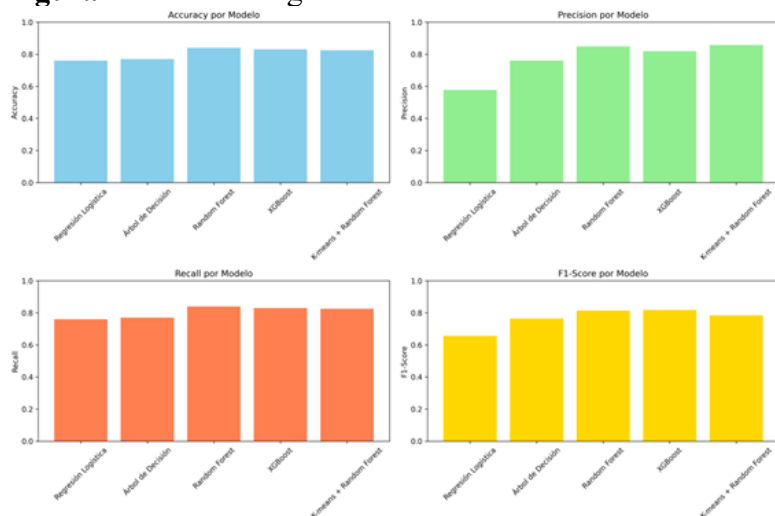
opción se realiza una tabla comparativa de resultados obtenidos de cada uno de los modelos predictivos.

1.15 Evaluación del modelo

Para la evaluación del modelo predictivo se usan diferentes métricas como precisión que evalúa la proporción de las predicciones correctas sobre el total además la exhaustividad que refleja el porcentaje de verdaderos positivos, también se obtiene una matriz de confusión para visualizar el comportamiento de las predicciones frente a los valores reales.

Como los resultados obtenidos como se muestra en la tabla 14 y las métricas que se usan en este caso tenemos de **accuracy**, **precisión**, **recall** y **f1** métricas que ayudan a obtener una idea general de cuál es el mejor modelo de predicción que se debe usar para este caso de estudio y finalmente presentar gráficos con las métricas obtenidas.

Figura 1 Resultados gráficos de evaluación de modelos



CAPITULO II: ANÁLISIS Y DISEÑO

2.1 Investigación preliminar

Los siniestros de tránsito actualmente se registran mediante el proceso del Ecu-911 o cuando un agente se encuentra insitu en el lugar en donde se acopian todas las emergencias y se van catalogando de acuerdo con lo que se necesita, una vez que el operador recibe toda la información pertinente la cataloga enviando las unidades de respuesta correspondientes como agentes de tránsito y unidades médicas para que lleguen a la ubicación de la emergencia, cuando las unidades llegan realizan el levantamiento correspondiente con documentación y registro fotográfico.

El registro llevado por parte de los agentes de tránsito se realiza de forma manual, por lo que debido a este ineficiente proceso puede existir el riesgo de pérdida, adulteración o daño en estos documentos, y tampoco es un proceso eficaz y eficiente, por este motivo se presenta dificultad para la llegada de los servicios de emergencia como es asistencia médica y asistencia.

En la ciudad de Loja los siniestros de tránsito se han dado la mayoría de las veces por descuido o por alguna falta cometida por los conductores vehiculares, por lo que representan una de las mayores causas de mortalidad a pesar de los esfuerzos realizados por las autoridades por presentar propuestas que minimicen estos peligros en la vía, una mejora significativa para el problema es una herramienta tecnológica como el presente proyecto, que permitirá el registro y análisis de datos dando un referente sobre cuáles son las circunstancias en las que se da un siniestro y cuáles son los puntos negros.(Ecu911, 2023)

Preguntas de investigación en las que este proyecto se basará será: ¿Cuál es el impacto que tiene la ausencia de una herramienta tecnológica para el manejo de siniestros de tránsito en Loja?, ¿De qué manera una aplicación exclusiva para agentes de tránsito puede mejorar la comunicación entre la UCOT y la ciudadanía?, ¿Cómo influye el tiempo de respuesta de los agentes de tránsito y las líneas de emergencia con una aplicación?

2.1.1 Recopilación de Datos

La recolección de datos se realizó mediante:

- Extracción de informes realizados por parte de los agentes de tránsito.
- Entrevistas con agentes de tránsito en funciones y supervisores.
- Datos de los agentes de tránsito que trabajan en la provincia de Loja.
- Análisis de la seguridad vial en la provincia de Loja.
- Recopilación de datos sobre frecuencia de accidentes en Loja.

Los datos se estructuraron en una base de datos relacional para facilitar su procesamiento y análisis.

2.2 Requerimientos

2.2.1 Funcionales.

Tabla 2: Requerimiento funcional 1

Número de requisito	RF01
Nombre de requisito	El sistema debe permitir login con correo institucional o con Gmail y clave con notificación de creación de cuenta
Tipo	Requisito
Fuente del requisito	Documento base
Prioridad del requisito	Alta/Esencial

Nota: De autoría

Tabla 3: Requerimiento funcional 2

Número de requisito	RF02
Nombre de requisito	El sistema debe tener un módulo de gestión de siniestros mediante formularios
Tipo	Requisito
Fuente del requisito	Documento base
Prioridad del requisito	Alta/Esencial

Nota: De autoría

Tabla 4: Requerimiento funcional 3

Número de requisito	RF03
Nombre de requisito	El sistema debe permitir guardar cada tipo de siniestro informativo y oficial con id y código de registro único
Tipo	Requisito
Fuente del requisito	Documento base
Prioridad del requisito	Alta/Esencial

Nota: De autoría

Tabla 5:Requerimiento funcional 4

Número de requisito	RF04
Nombre de requisito	El sistema debe permitir guardar automáticamente los siniestros en borrador para la publicación posterior
Tipo	Requisito
Fuente del requisito	Documento base
Prioridad del requisito	Alta/Esencial

Nota: De autoría

Tabla 6: Requerimiento funcional 5

Número de requisito	RF05
Nombre de requisito	La aplicación permitirá ver y editar los siniestros generados por los agentes únicamente el propietario y cambiar estado borrador/activo
Tipo	Requisito
Fuente del requisito	Documento base
Prioridad del requisito	Alta/Esencial

Nota: De autoría

Tabla 7: Requerimiento funcional 6

Número de requisito	RF06
Nombre de requisito	La aplicación web permitirá a los conductores ver el mapa con los siniestros de cierre de vía, hora y ubicación con notificaciones
Tipo	Requisito
Fuente del requisito	Documento base
Prioridad del requisito	Alta/Esencial

Nota: De autoría

Tabla 8: Requerimiento funcional 7

Número de requisito	RF07
Nombre de requisito	La aplicación web debe tener la funcionalidad de geolocalización en el formulario también adjuntar evidencia fotográfica.
Tipo	Requisito
Fuente del requisito	Documento base
Prioridad del requisito	Alta/Esencial

Nota: De autoría

Tabla 9: Requerimiento funcional 8

Número de requisito	RF08
Nombre de requisito	La aplicación web contará con un modelo predictivo para detectar siniestros de tránsito y puntos negros de la provincia
Tipo	Requisito
Fuente del requisito	Documento base
Prioridad del requisito	Alta/Esencial

Nota: De autoría

2.2.2 No Funcionales

Tabla 10: Requerimiento no funcional 1

Número de requisito	RNF01
Nombre de requisito	La aplicación garantiza seguridad en el acceso mediante roles de acceso y asignación de grupos de control
Tipo	Requisito No Funcional
Fuente del requisito	Buenas prácticas de usabilidad
Prioridad del requisito	Alta/Esencial

Nota: De autoría

Tabla 11:Requerimiento no funcional 2

Número de requisito	RNF02
Nombre de requisito	El diseño de la aplicación debe ser responsivo y que se adapte a diferentes pantallas para su uso
Tipo	Requisito No Funcional
Fuente del requisito	Buenas prácticas de confiabilidad
Prioridad del requisito	Alta/Esencial

Nota: De autoría

Tabla 12: Requerimiento no funcional 3

Número de requisito	RNF03
Nombre de requisito	El modelo predictivo debe procesar datos históricos
Tipo	Requisito No Funcional
Fuente del requisito	Buenas prácticas de usabilidad y accesibilidad
Prioridad del requisito	Media

Nota: De autoría

Tabla 13: Requerimiento no funcional 4

Número de requisito	RNF03
Nombre de requisito	La aplicación debe funcionar en un navegador web con base de datos normalizada y relacional
Tipo	Requisito No Funcional
Fuente del requisito	Buenas prácticas de usabilidad y accesibilidad
Prioridad del requisito	Media

Nota: De autoría

2.3 Desarrollo de la metodología

Mediante el desarrollo de un comparador automático de algoritmo de machine learning que ayude a predecir en este caso los puntos negros de la ciudad y obtener las circunstancias más comunes por las que ocurre un siniestro probando diferentes algoritmos con el dataset que se solicitó a la UCOT de acuerdo como se especifica en el anexo 2, con la ayuda de estos datos se procede a probar 5 algoritmos diferentes para determinar cuál predice mejor los accidentes graves simulando a un laboratorio automatizado que prueba múltiples enfoques.

Para realizar el preprocesamiento de datos primero se debería hacer una limpieza adecuada de los datos obtenidos para esto se debe eliminar los datos duplicados o valores nulos en variables importantes como tipos de siniestros, evidencias, ubicación, tipo de vía y condiciones climáticas, se deben transformar a formato numérico además se debe normalizar los datos para un mejor resultado y entrenamiento del modelo y de esta forma escalar las variables categóricas.

El trabajo con el dataset y para el entrenamiento del modelo predictivo se debe dividir el dataset en el 70% de su contenido para el uso del entrenamiento y el 30% sobrante para realizar pruebas como una semilla fija para garantizar la efectividad de sus resultados.

Figura 2: Extracto de código

```

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split, cross_val_score, StratifiedKFold
from sklearn.preprocessing import LabelEncoder, StandardScaler, MinMaxScaler
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score,
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.cluster import KMeans
from sklearn.decomposition import PCA
import xgboost as xgb
import warnings
warnings.filterwarnings('ignore')

class ComparadorModelosSiniestros:
    """
    Clase para comparar diferentes algoritmos de ML en predicción de siniestros de tránsito
    """

    def __init__(self, archivo_csv):
        """Inicializar con el dataset"""
        self.df = pd.read_csv(archivo_csv)
        self.df_procesado = None
        self.X = None
        self.y = None
        self.scaler = StandardScaler()
        self.label_encoders = {}
        self.resultados = {}

        print(f"Dataset cargado: {len(self.df)} registros")

    def preprocesar_datos(self):
        """Preprocesar datos para machine learning"""
        print("=== PREPROCESAMIENTO DE DATOS ===")

        df = self.df.copy()

```

Nota: estructura de código para comparar modelos predictivos

Tabla 14 Resultados de modelos predictivos

Técnica	Competencia	Salida	Accuracy	Precision	Recall	F1
Regresión Logística	Predicción de Siniestros	0	0.76000	0.76000	1.00000	0.86364
Regresión Logística	Predicción de Siniestros	1	0.76000	0.00000	0.00000	0.00000
Árbol de Decisión	Predicción de Siniestros	0	0.77000	0.83544	0.86842	0.85161
Árbol de Decisión	Predicción de Siniestros	1	0.77000	0.52381	0.45833	0.48889
Random Forest	Predicción de Siniestros	0	0.84000	0.83333	0.98684	0.90361
Random Forest	Predicción de Siniestros	1	0.84000	0.90000	0.37500	0.52941
XGBoost	Predicción de Siniestros	0	0.83000	0.85119	0.94079	0.89375
XGBoost	Predicción de Siniestros	1	0.83000	0.71875	0.47917	0.57500
K-means	Predicción de Siniestros	0	0.82500	0.81283	1.00000	0.89676
K-means	Predicción de Siniestros	1	0.82500	1.00000	0.27083	0.42623

Nota: Tabla de resultados de los modelos predictivos

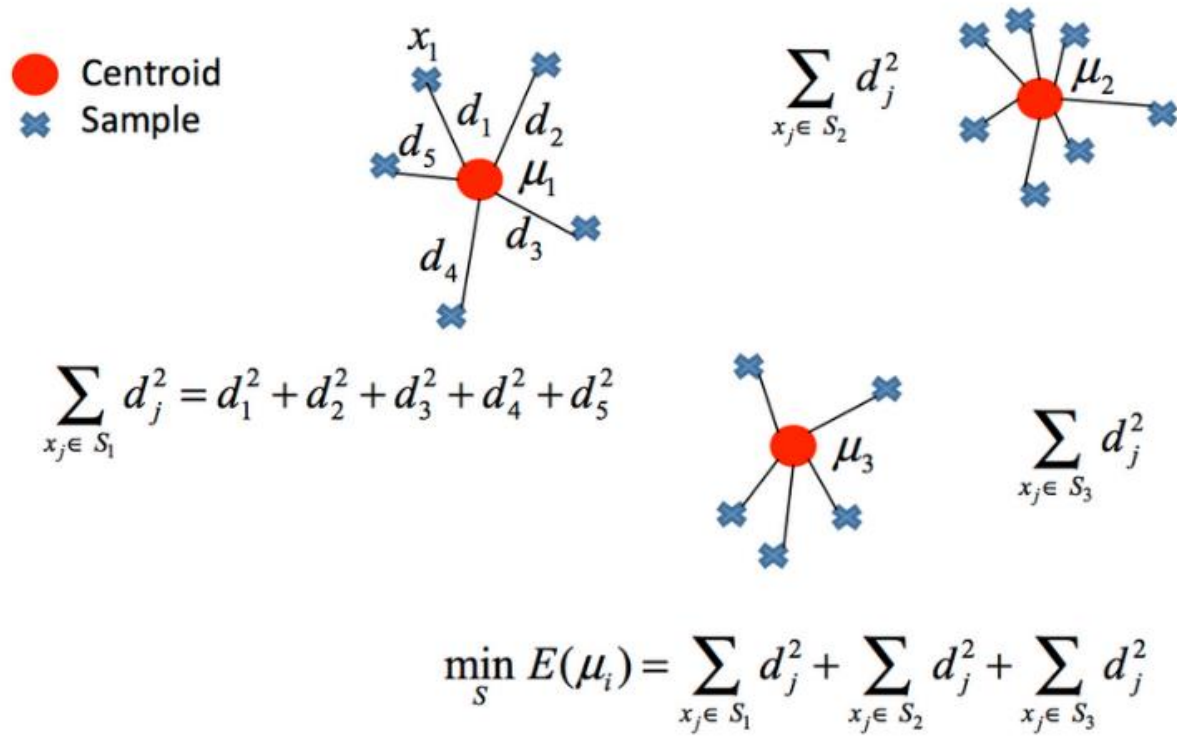
Con el algoritmo K-means se puede identificar los puntos negros, ya que permite la agrupación espacial de los incidentes sin requerir etiquetas previas de esta forma se facilita el análisis geoespacial para luego ubicar en un mapa de calor, para predecir la probabilidad de ocurrencia de un siniestro bajo condiciones específicas la mejor opción de algoritmo es random forest porque permite el manejo de grandes volúmenes de datos y su resistencia al overfitting.

De acuerdo con los resultados obtenidos en los modelos predictivos las mejores opciones son las que se eligieron con un resultado de Mejor Accuracy: Random Forest (0.8400) Mejor Precision: K-means (0.8578) Mejor Recall: Random Forest (0.8400) Mejor F1-Score: XGBoost (0.8173).

Para el desarrollo del proyecto se usará una metodología híbrida Scrumban, que será la mezcla entre la metodología Scrum y metodología Kanban obteniendo lo mejor de cada una, esta metodología tiene las siguientes fases: iniciación, implementación de requerimientos en la tabla

kanban, lanzamiento y por último la fase de pruebas que nos facilitara el proceso de desarrollo del modelo. Acorde a esto, se presenta la siguiente arquitectura utilizada.

Figura 3: Algoritmo K-means



Nota: Describe la lógica de k-means

2.3.1 Comparativa de metodologías de desarrollo de software

Tabla 15: Comparativa de metodologías de desarrollo de software

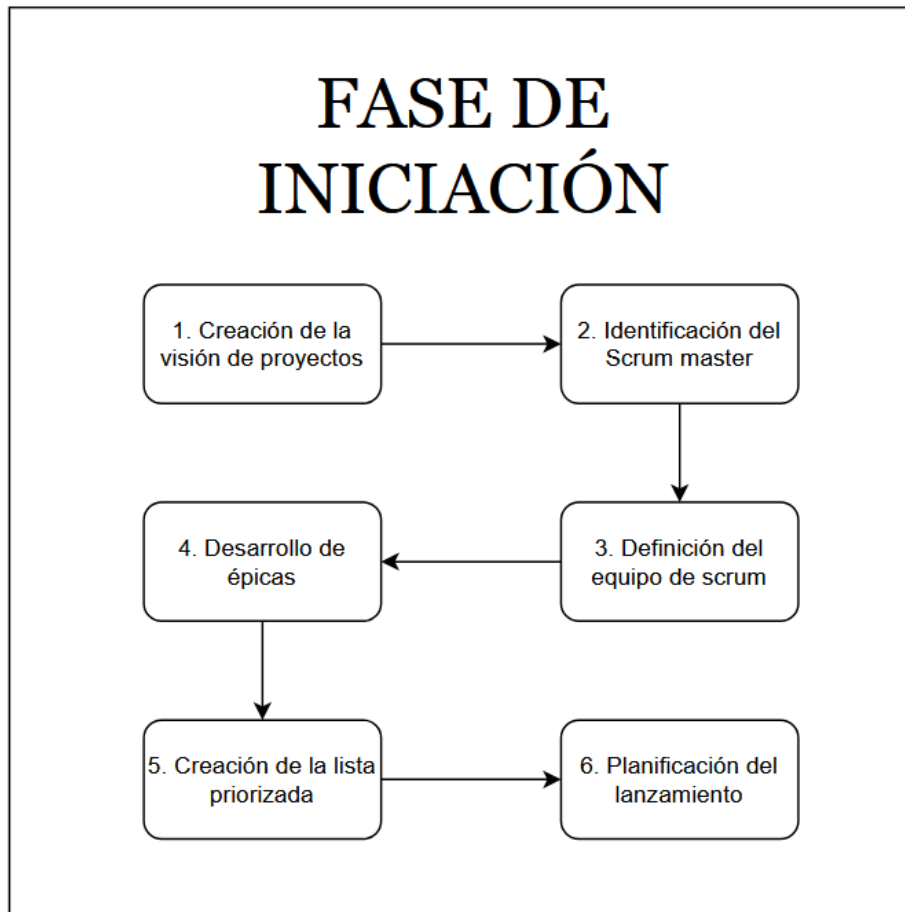
Aspecto	Scrum	Kanban	Scrumban	XP	Waterfall	Lean	RAD	DevOps
Duración ciclos	1-4 semanas	Continuo	Flexible	1-2 semanas	6+ meses	Continuo	2-6 meses	Continuo
Flexibilidad	Media	Alta	Alta	Media	Muy baja	Alta	Media	Alta
Documentación	Mínima	Mínima	Mínima	Mínima	Extensa	Mínima	Media	Mínima
Tiempo setup inicial	2-4 semanas	1 semana	1-2 semanas	3-4 semanas	2-6 meses	2-3 semanas	4-8 semanas	4-12 semanas
Velocidad entrega	Media-Alta	Alta	Muy Alta	Media	Muy baja	Alta	Alta	Muy alta
Cambios requisitos	Final sprint	Cualquier	Cualquier	Limitados	Muy difícil	Cualquier	Difícil	Continuo
Curva aprendizaje	Media	Baja	Baja	Alta	Baja	Media	Media	Alta
Ceremonias/Reuniones	Muchas	Pocas	Flexibles	Muchas	Muchas	Pocas	Medias	Variables
Testing	Al final sprint	Continuo	Continuo	Continuo	Al final	Continuo	Por fases	Automatizado
Escalabilidad	Media	Alta	Muy alta	Baja	Alta	Alta	Media	Muy alta
Costo implementación	Medio	Bajo	Bajo	Alto	Alto	Medio	Alto	Muy alto

Nota: Tabla de metodologías de software para elegir la más conveniente

De acuerdo con la tabla comparativa realizada en diferentes aspectos importantes para evaluar una metodología la mejor opción para el desarrollo de una aplicación web es scrumban, ya que la velocidad de implementación es óptima porque no necesita definir roles específicos, trabaja con un tablero de toma de decisiones “To do -> In progress -> Done”, dependiendo de la dimensión del proyecto se puede definir el tiempo de desarrollo.

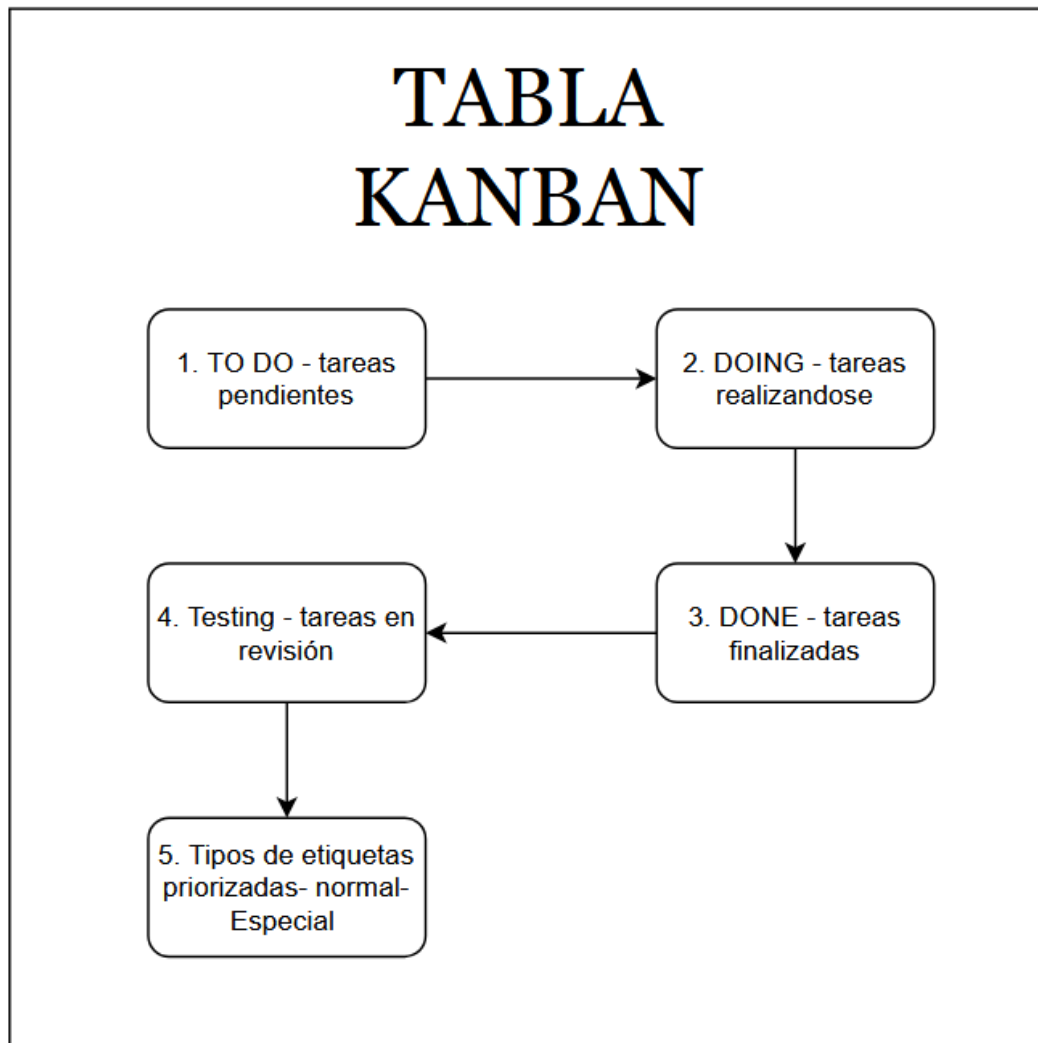
2.3.2 Fases de metodología

Figura 4: Fase de iniciación



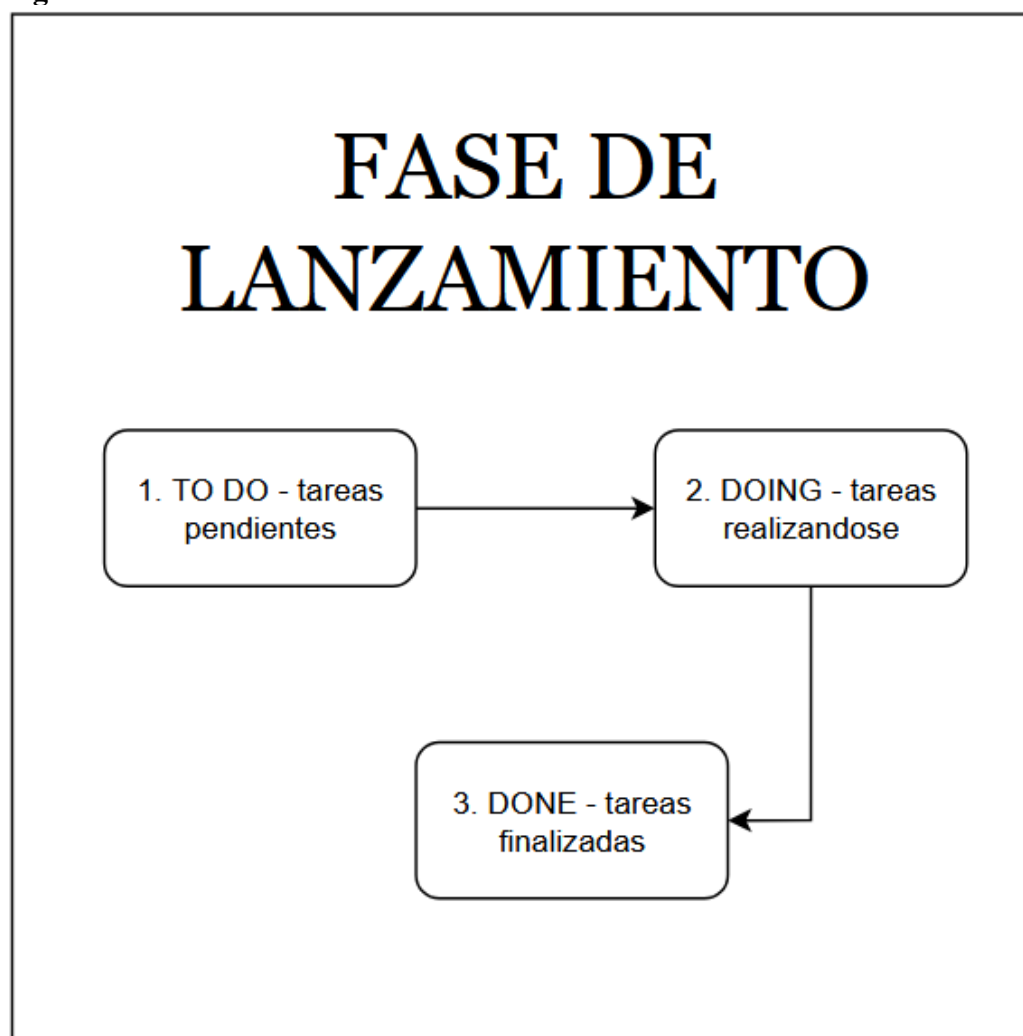
Nota: Describe por pasos toda la fase de iniciación en cuanto a la metodología Scrumban

1. Creación de la visión del proyecto. _ se define todo lo que va y lo que no va dentro del proyecto.
2. Identificación del scrum master. _ podríamos basarnos en la experiencia que tenga el desarrollador.
3. Definición del Equipo de Scrum. _ encontrar o definir todos los involucrados.
4. Desarrollo de Épicas. Aclaración del cliente de acuerdo con las necesidades que presenta.
5. Creación de la lista priorizada. _ El responsable es el product owner, pero se puede socializar con el equipo.
6. Planificación del lanzamiento. _ el cual puede ser incremental o ya del proyecto final.

Figura 5: Fase 2 tabla kanban

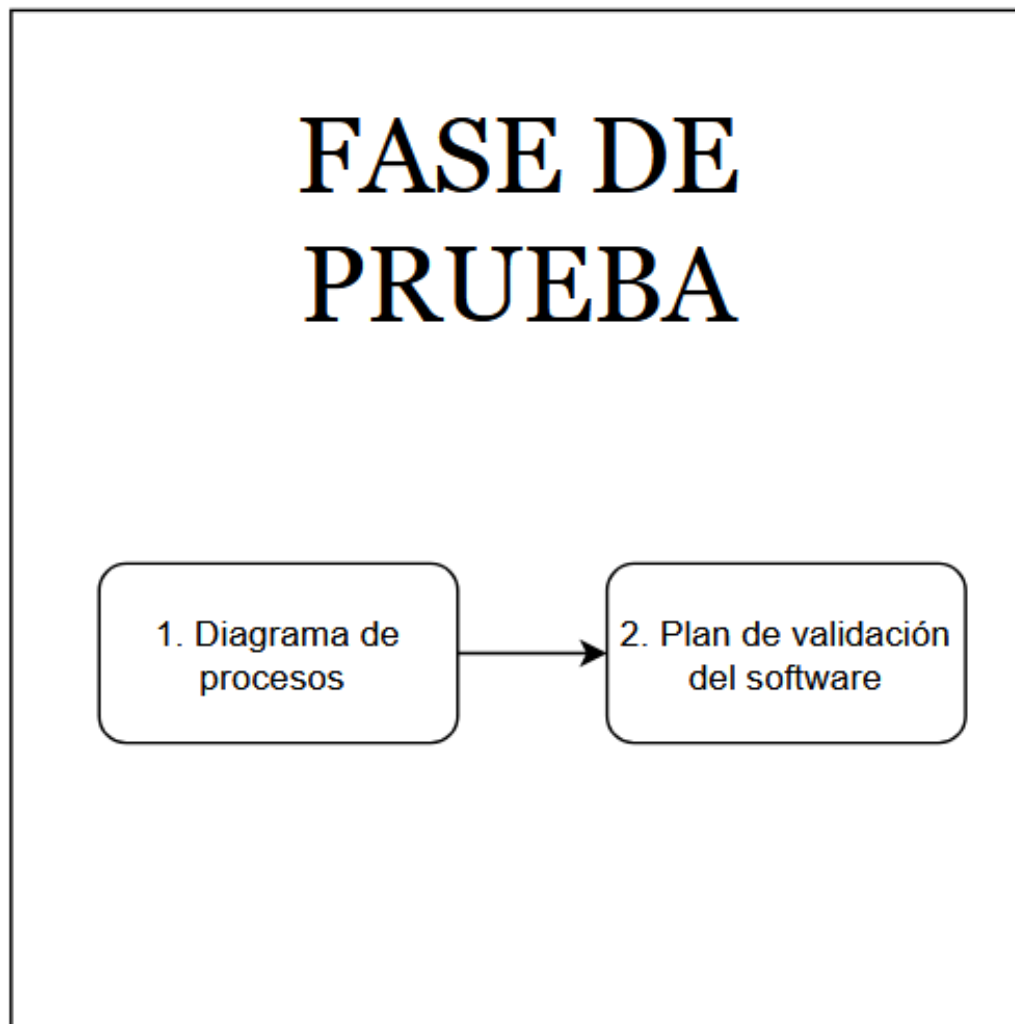
Nota: Imagen que describe como se debe hacer la tabla kanban y que se debe tomar en cuenta para la metodología Scrumban

1. To Do. _ tareas pendientes de iniciar su desarrollo, determinando requerimientos.
2. Doing. _ documentos que se están realizando actualmente.
3. Testing. _ documentos que están en proceso de revisión y aprobación, dándonos un registro como documento controlado.
4. Done. _ aquellos documentos que están acabados, registrados y controlados.
5. Para la tabla tenemos etiquetas de Prioridad (Azul), Normal (Verde), Especial (Naranja).

Figura 6: Fase 3 de lanzamiento

Nota: Imagen que describe las tareas a realizar para la fase de lanzamiento en la metodología Scrumban.

1. Liberación de tareas. _ Se dan por finalizadas todas las tareas enlistadas en la lista priorizada.
2. Retrospectiva del proyecto. _ Tablas presentando todo el proyecto y sus modelos conforme a los requerimientos.
3. Recomendaciones. _ Presentar recomendaciones relacionadas con el proyecto.

Figura 7: Fase de pruebas

Nota: Imagen que describe las últimas actividades para culminar con éxito la fase de pruebas de la metodología

1. Plan de pruebas. _ Realizar un plan de pruebas efectivo y eficaz.
2. Pruebas de software. _ Dentro del entorno real en el que se usará.

2.3.3 Entregables

En las siguientes tablas se detallan las fases de la metodología híbrida, los recursos involucrados tanto humanos como tecnológicos, las actividades que se realizan dentro cada fase y los entregables para el desarrollo del proyecto.

Tabla 16: Procesos de fase de iniciación

Fase de Iniciación	
Objetivo:	Realizar un análisis exhaustivo identificando la visión del proyecto y la viabilidad, aplicando técnicas de investigación y metodologías de software para la recopilación de información y requerimientos.
Recursos Humanos	• Jefe de proyectos • Analistas • Agentes de tránsito
Recursos tecnológicos	
Hardware	• Computadoras
Software	• Office Libre
Actividades	• Reunión con los involucrados en la fase para definir la visión del proyecto • Reuniones generales con los profesionales de la agencia de tránsito, para determinar las necesidades y requerimientos e identificar las partes que llevará el product backlog.
Entregables	• Visión del proyecto • Product backlog • Lista Priorizada

Nota: Tabla auto generada

Tabla 17: Procesos tabla kanban

Tabla Kanban	
Objetivo:	Dividir los Sprints o tareas que se van a realizar mediante una tabla, llamada tabla de Kanban que se divide en tres grandes grupos; pendiente, haciendo y completado, para tener una mejor organización y visibilidad de los sprints.
Recursos Humanos	• Desarrollador del proyecto • Analistas
Recursos tecnológicos	
Hardware	• Computadoras
Software	• Office Libre • Trello • Visual Code • PhpMyAdmin
Actividades	• Obtener lista priorizada y el product backlog para definir tabla en “Por Hacer”, “Haciendo” y “Completado” • Definir Sprints y apilar en la tabla según la lista priorizada
Entregables	• Diseño y estructura de la tabla kanban • Prototipo del aplicativo • Desarrollo del Software (codificación del aplicativo)

Nota: Tabla auto generada

Tabla 18: Procesos de fase de lanzamiento

Fase de Lanzamiento	
Objetivo:	Liberar los sprints pendientes realizando una retrospectiva del proyecto a través de informes y tablas para completar el proyecto y revisar errores cometidos durante el proceso.
Recursos Humanos	• Product owner • Project manager • Team Scrum
Recursos tecnológicos	
Hardware	• Computadoras
Software	• Office Libre • Servidor linux • Visual Code • File zilla
Actividades	• Reunión con los desarrolladores y el product owner para la liberación de sprints y la retrospectiva del proyecto realizado.
Entregables	• Liberación de actividades en la tabla kanban • Retrospectiva del proyecto • Recomendaciones

Nota: Tabla auto generada

Tabla 19: Procesos de fase de pruebas

Fase de Pruebas	
Objetivo:	Realizar el testeo de software mediante los métodos de pruebas más utilizados en el desarrollo determinando la respuesta que da a los problemas en un escenario real.
Recursos Humanos	• Team Scrum • Tester • Agentes de tránsito
Recursos tecnológicos	
Hardware	• Computadoras
Software	• Office Libre • Servidor linux
Actividades	• Testers realizan las pruebas al proyecto. • Realizar plan de pruebas • Ejecutar el plan de pruebas
Entregables	• Plan de pruebas • Informe del plan de pruebas ejecutado

Nota: Tabla auto generada

2.4 Fase 1: Iniciación

En esta primera fase se crea la visión del proyecto que sirve de enfoque y dirección de este. En base a la metodología utilizada se identifican el scrum master, stakeholder, usuarios además se define el producto backlog en base a las historias de usuario y la lista priorizada.

2.4.1 Visión del proyecto

2.4.1.1 Propósito

En la actualidad tenemos una cultura digitalizada y competitiva por ende es importante tener avances tecnológicos. El presente proyecto se basa en el desarrollo e implementación de una aplicación web que permita a los agentes de tránsito de la ciudad de Loja pertenecientes a la UCOT registrar siniestros viales de forma segura y estructurada.

Adicional la aplicación permitirá a los conductores tener una vista pública informativa para evitar cuellos de botella en caso de que alguna vía se cierre por circunstancias de siniestros. El sistema facilitará la recolección y análisis de datos para proveer un modelo predictivo que ayude a identificar puntos negros de la ciudad y circunstancias en las que es más probable que suceda un siniestro, fortaleciendo los procesos de planificación y prevención en la ciudad.

2.4.1.2 Alcance

- Para la aplicación web se desarrollará dos vistas principales, módulo de gestión exclusivo para agentes de tránsito y módulo de vista para usuarios externos que será solo informativo.
- Módulo para administración de todos los datos registrados y reportados.
- La aplicación será desarrollada en backend con PHP Laravel y frontend con Angular para la base de datos se usará MySQL, la primera versión para el embebido presentada en tesis será desarrollado únicamente para Android.
- Desarrollo de un modelo predictivo para detección de siniestros de tránsito y puntos negros de la provincia.

- Construir un UI y UX responsiva intuitiva con un enfoque en las necesidades y facilidades para los agentes de tránsito.
- Desarrollar un CRUD del registro de siniestros de tránsito mediante formularios estructurados generales.
- Integración de geolocalización para capturar la ubicación del siniestro de tránsito, además capturar evidencia fotográfica.
- Mostrará notificaciones para los usuarios conductores que usen la aplicación y de esta forma puedan obtener datos precisos del día.
- Contará con una base de datos relacional normalizada Mysql para el manejo efectivo de los datos guardando toda la información sobre los siniestros.
- Seguridad para la aplicación con el control de roles y usuarios para los agentes de tránsito y los usuarios conductores.
- El proyecto no contempla la integración con sistemas existentes de la agencia nacional de tránsito.
- La aplicación contempla únicamente el registro de los siniestros no se tiene logística para el manejo de las unidades de emergencia.

2.4.1.3 Definiciones, acrónimos y abreviaciones

- **PHP:** Es un lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico.
- **MYSQL:** My Structured Query Language, por sus siglas en inglés, la cual traduce Lenguaje de Consulta Estructurado. Es un sistema de gestión de bases de datos relacional, licenciado bajo la GPL de la GNU. Su diseño multihilo le permite soportar una gran carga de forma muy eficiente.
- **LOCALHOST:** En desarrollo de software, localhost es un nombre reservado que tienen todas las computadoras, router o dispositivo independientemente de que disponga o no de una tarjeta de red ethernet.
- **ANGULAR:** Es un framework de desarrollo web basado en código abierto que permite crear aplicación con arquitectura modelo, vista, controlador.

2.4.1.4 Beneficios

- **Disponibilidad.** _ al contar con una base de datos, los datos estarán registrados en tiempo real y asegurados para el uso de los agentes o de conductores.
- **Optimización.** _ con una herramienta tecnológica el registro de siniestros será realizado en menor tiempo, mejorando la eficacia y eficiencia de los agentes.
- **Identificación.** _ analizar zonas críticas y patrones de siniestros de tránsito mediante el uso de un modelo predictivo aportando en la prevención y la toma de decisiones.

2.4.2 Oportunidades de negocio

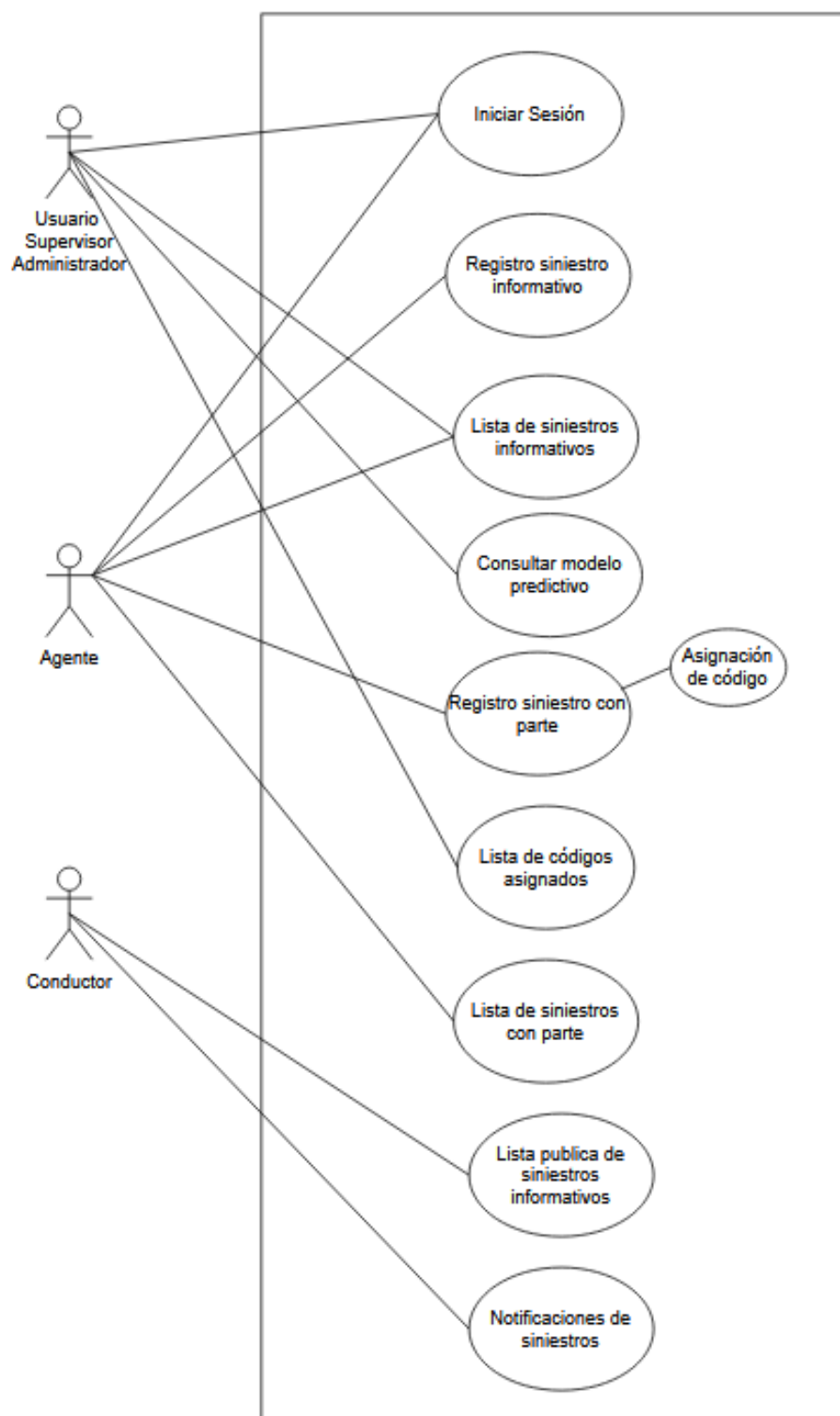
El desarrollo de un aplicativo web para los agentes de tránsito pertenecientes a la UCOT representa una oportunidad estrategia que ofrece un producto escalable que se podría implementar en otras ciudades que enfrentan problemas similares agregando un gran valor de responsabilidad vial a las unidades de control con un modelo predictivo analítico apoyando a la toma de decisiones.

2.4.3 Reglas de negocio

- Por formulario iniciado se debe agregar un código único que no se repite por ninguna circunstancia.
- Los agentes solo pueden visualizar los reportes con parte que ellos hayan generado, no comparten información con otros agentes.
- El usuario administrador puede visualizar únicamente los reportes informativos y la lista de códigos únicos generados.
- Los conductores tienen acceso a una visualizando de mapa únicamente datos generales sin comprometer la integridad de la investigación.

2.4.5 Caso de uso

Figura 9: Caso de uso



Nota: diagrama de caso de uso

2.4.5.1 Especificaciones de caso de uso

Tabla 20: CU-001: Iniciar Sesión

Elemento	Descripción
Actor Principal	Usuario Supervisor/Administrador
Objetivo	Autenticarse en el sistema con credenciales administrativas
Precondiciones	El usuario debe tener credenciales válidas registradas
Flujo Principal	• Usuario ingresa credenciales • Sistema válido credenciales y otorga permisos • Redirige a dashboard administrativo
Flujos Alternativos	A1. Credenciales inválidas → Mostrar error y permitir reintento
Postcondiciones	Usuario autenticado con permisos administrativos

Nota: De autoría

Tabla 21: CU-002: Registro Siniestro Informativo

Elemento	Descripción
Actor Principal	Usuario Supervisor/Administrador
Objetivo	Registrar un siniestro sin parte oficial
Precondiciones	Usuario autenticado con permisos
Flujo Principal	• Seleccionar “Registro siniestro informativo” • Completar formulario básico • Adjuntar fotos (opcional) • Confirmar registro
Flujos Alternativos	A1. Datos incompletos → Validar y solicitar información faltante
Postcondiciones	Siniestro informativo creado en base de datos

Nota: De autoría

Tabla 22: CU-003: Registro Siniestro con Parte

Elemento	Descripción
Actor Principal	Agente
Objetivo	Registrar siniestro oficial con documentación legal
Precondiciones	Agente autenticado, parte oficial disponible
Flujo Principal	• Iniciar registro oficial • Completar datos del siniestro y fotos • Registrar vehículos involucrados • Registrar personas (conductores/acompañantes) • Generar parte oficial
Flujos Alternativos	A1. Múltiples vehículos → Registrar cada vehículo
Postcondiciones	Siniestro oficial registrado con código asignado

Nota: De autoría

Tabla 23: CU-004: Consultar Modelo Predictivo

Elemento	Descripción
Actor Principal	Usuario Supervisor/Administrador
Objetivo	Acceder a análisis predictivos y estadísticas
Precondiciones	Usuario con permisos administrativos, datos históricos disponibles
Flujo Principal	• Acceder a módulo de análisis • Sistema procesa datos históricos • Visualizar resultados
Flujos Alternativos	A1. Datos insuficientes → Mostrar mensaje informativo
Postcondiciones	Reportes predictivos generados

Nota: De autoría

Tabla 24: CU-005: Asignación de Código

Elemento	Descripción
Actor Principal	Agente
Objetivo	Generar código único para identificación de siniestro
Precondiciones	Siniestro en proceso de registro
Flujo Principal	• Sistema genera código automáticamente • Válida unicidad del código • Asigna código al siniestro • Actualiza base de datos
Flujos Alternativos	A1. Código duplicado → Generar nuevo código automáticamente
Postcondiciones	Siniestro con código único asignado

Nota: De autoría

Tabla 25: CU-006: Lista de Siniestros Informativos

Elemento	Descripción
Actor Principal	Usuario Supervisor/Administrador
Objetivo	Visualizar y gestionar todos los siniestros informativos
Precondiciones	Usuario administrador autenticado, siniestros registrados
Flujo Principal	• Acceder a "Lista siniestros informativos" • Sistema muestra listado paginado • Aplicar filtros (fecha) • Seleccionar siniestro para ver detalles • Opciones: Editar, Eliminar, Ver
Flujos Alternativos	A1. Sin registros → Mostrar mensaje y opción de crear nuevo
Postcondiciones	Lista de siniestros informativos visualizada

Nota: De autoría

Tabla 26: CU-007: Lista de Siniestros con Parte

Elemento	Descripción
Actor Principal	Agente
Objetivo	Consultar siniestros oficiales registrados
Precondiciones	Agente autenticado, siniestros oficiales existentes
Flujo Principal	• Acceder a "Siniestros con parte" • Ver listado de siniestros oficiales • Seleccionar para ver detalles completos • Opciones: Editar, Eliminar, Ver -> Exportar
Flujos Alternativos	A1. Buscar por código específico
Postcondiciones	Información de siniestros oficiales consultada

Nota: De autoría

Tabla 27: CU-008: Lista Pública de Siniestros Informativos

Elemento	Descripción
Actor Principal	Conductor
Objetivo	Consultar información pública de siniestros informativos y con parte
Precondiciones	Conductor autenticado, siniestros informativos públicos
Flujo Principal	• Acceder a "Siniestros públicos" • Ver mapa con ubicaciones de siniestros • Filtrar por zona de interés • Solo ver con cierres de vía • Consultar recomendaciones de seguridad
Flujos Alternativos	A1. Filtrar por tipo de vía A2. Ver tendencias por horarios
Postcondiciones	Información preventiva consultada por conductor

Nota: De autoría

Tabla 28: CU-09: Notificaciones de Siniestros

Elemento	Descripción
Actor Principal	Conductor
Objetivo	Recibir alertas sobre siniestros relevantes
Precondiciones	Conductor registrado, configuración de notificaciones activa
Flujo Principal	• Sistema identifica siniestros ingresados con cierre de vía • Evalúa criterios del conductor (zona, vehículo) • Genera notificación personalizada • Envía por canal preferido (email, SMS, push) • Registra entrega de notificación
Flujos Alternativos	A1. Conductor involucrado → Notificación especial A2. Zona frecuente → Alerta preventiva
Postcondiciones	Conductor notificado sobre siniestros relevantes

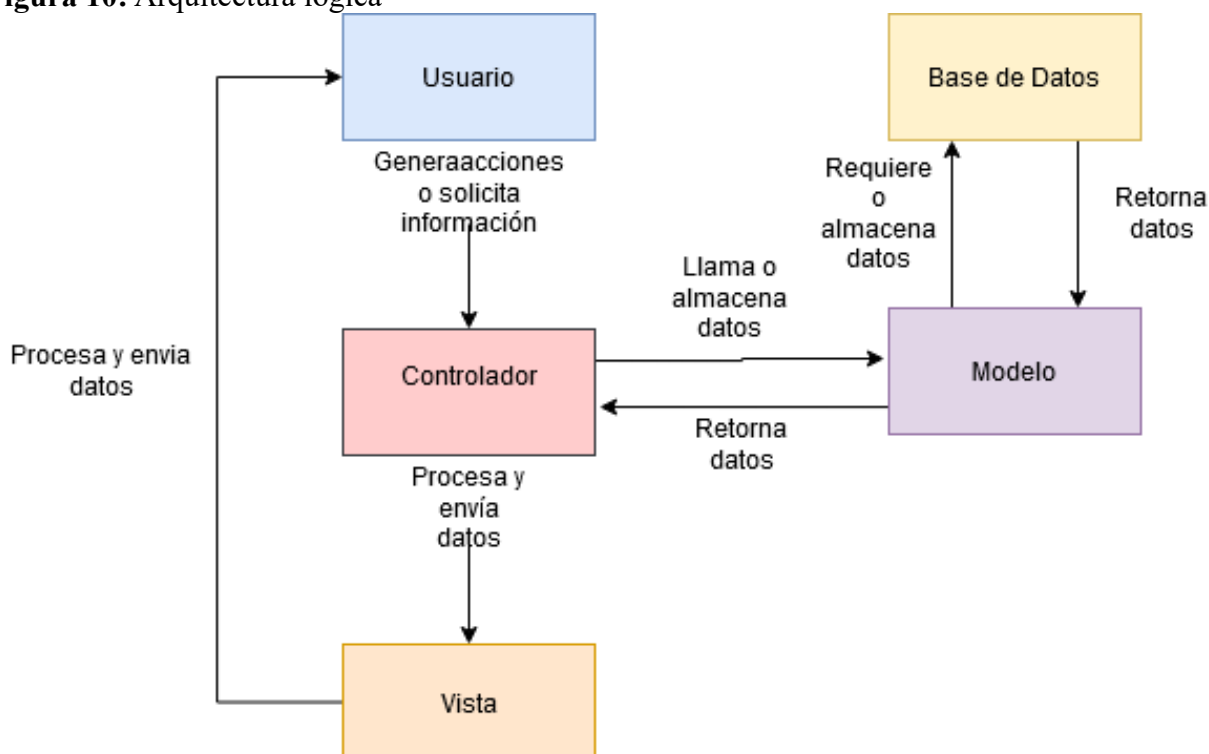
Nota: De autoría

2.4.6 Arquitectura Lógica

La arquitectura lógica de esta basada en la modelo vista controlador, ya que al usar php laravel se trabaja con modelos y con controladores lo que consume la vista en este caso angular, todos los componentes se comunican en ambas direcciones permitiendo un intercambio de información precisa y sin retrasos, los controladores actúan como coordinadores que toda la lógica delimitando lo específico para los siniestros viales, cada capa tiene responsabilidades específicas.

La vista es la presentación e interacción, el controlador es la lógica de negocio o coordinación mientras del modelo y la base de datos es las especificaciones de campos o atributos con los que se va a trabajar, la ventaja de usar esta arquitectura lógica es que es escalable porque cada componente puede crecer de forma individual, además se pueden realizar cambios aislados por capas.

Figura 10: Arquitectura lógica



Nota: Diagrama lógica

2.4.7 Product backlog

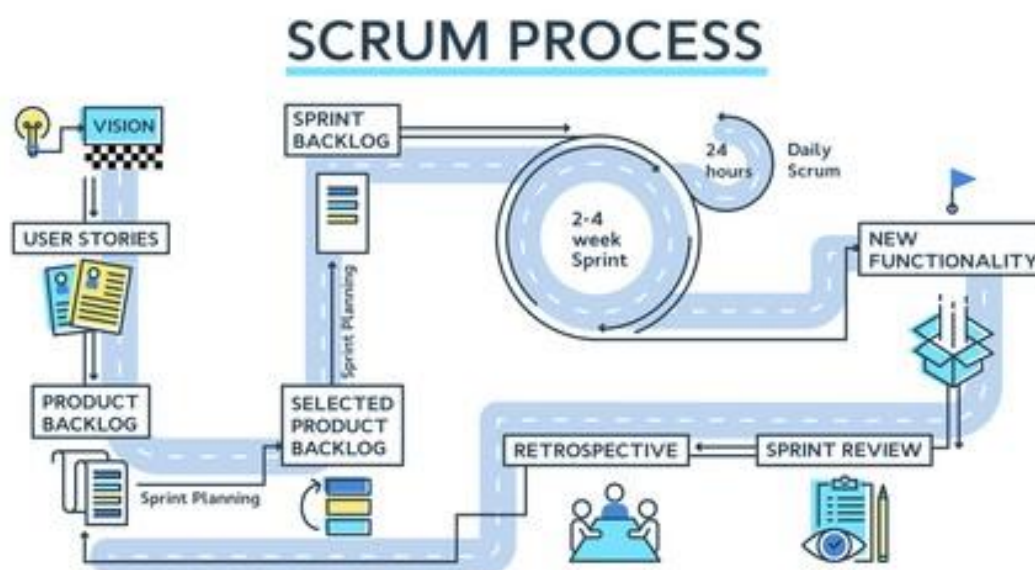
Como el proyecto es desarrollado con una metodología ágil mezcla de scrum y Kanban, se requiere generar un Product Backlog, permitiendo de esta forma organizar tareas o actividades por módulos en base a los requerimientos obtenidos con anterioridad. Para construir la tabla y definir el esfuerzo relativo de cada tarea vamos a usar una metodología de story points, estos puntos no representan tiempo exactamente sino hace referencia a la complejidad de la tarea.

Tabla 29: Tabla de story points

2 puntos	Simple	1 día
3 puntos	Moderado	2 días y medio
5 puntos	Complejo	De 3 a 5 días
8 puntos	Muy complejo	1 semana y media
13 puntos	Épico	3 semanas
21 puntos	Muy épico	Más de 3 semanas

Nota: De autoría

Figura 11: Proceso de scrum con product backlog



Nota: Esta imagen muestra el proceso que lleva scrum

Tabla 30: Lista Product backlog

Req. Original	Backlog ID	Descripción	Tipo	Prioridad	Estimación
RF01 + RNF01	PBI-01	Como usuario, quiero autenticarme con Gmail/correo institucional con seguridad basada en roles	Requerimiento	Alta	8 puntos
RF02	PBI-02	Como agente, quiero registrar siniestros mediante formularios para documentar incidentes	Requerimiento	Alta	8 puntos
RF03	PBI-03	Como sistema, debo generar códigos únicos para cada siniestro para identificación legal	Requerimiento	Alta	3 puntos
RF04	PBI-04	Como agente, quiero que mis formularios se guarden automáticamente en borrador	Requerimiento	Alta	2 puntos
RF05	PBI-05	Como agente, quiero ver/editar solo mis siniestros y cambiar su estado	Requerimiento	Alta	5 puntos
RF06	PBI-06	Como conductor, quiero ver un mapa con siniestros y recibir notificaciones	Requerimiento	Alta	13 puntos
RF07	PBI-07	Como agente, quiero capturar geolocalización y adjuntar fotos en el formulario	Requerimiento	Alta	8 puntos
RF08 + RNF03	PBI-08	Como administrador, quiero un modelo predictivo que procese datos históricos	Requerimiento	Alta	21 puntos
RNF02	PBI-09	Como desarrollador, implemento diseño responsivo para todos los dispositivos	Requerimiento	Alta	5 puntos
RNF04	PBI-10	Como desarrollador, configuro la aplicación web con base de datos relacional	Requerimiento	Media	2 puntos

Nota: Lista del product backlog desarrollada en base a los requerimientos

2.4.8 Lista priorizada

La lista priorizada sirve para ordenar todas las actividades (Sprints) de forma descendente desde las que tiene una prioridad alta en base a lo obtenido en la *tabla 20 el producto backlog*.

Tabla 31: Lista Priorizada

Prioridad	PBI	Tipo	Alias	Estimación	Sprint
Crítica	PBI-10	Técnica	Setup: App web + BD relacional	3 puntos	Sprint 0
Crítica	PBI-01	Usuario	Autenticación segura con roles	8 puntos	Sprint 1
Crítica	PBI-02	Usuario	Formularios de registro	8 puntos	Sprint 1
Crítica	PBI-09	Técnica	Diseño responsivo transversal	5 puntos	Sprint 1-2
Alta	PBI-03	Técnica	Códigos únicos automáticos	3 puntos	Sprint 2
Alta	PBI-04	Usuario	Autoguardado en borradores	2 puntos	Sprint 2
Alta	PBI-05	Usuario	Gestión personal de registros	5 puntos	Sprint 2-3
Alta	PBI-07	Usuario	Geolocalización + evidencia	8 puntos	Sprint 3
Alta	PBI-06	Usuario	Mapa público + notificaciones	13 puntos	Sprint 4
Alta	PBI-08	Usuario	Modelo predictivo con datos históricos	21 puntos	Sprint 5+

Nota: Tabla lista priorizada

2.5 Fase 2: Tabla Kanban

En esta fase se realiza la planificación de desarrollo (codificación) del proyecto en base a la lista priorizada definida con anterioridad, se diseña la estructura de la tabla utilizando herramientas como trello y Monday que ayudara a organizar las actividades y llevar un control dentro del scrum team finalmente se termina presentando el diseño de interfaces del sistema.

2.5.1 Diseño y estructura

Con ayuda de la tabla Kanban en donde su estructura básica tiene como títulos por hacer una columna en donde se enlistan los sprints en base a la lista priorizada obtenida anteriormente tabla 25 para de esta manera tener claro lo que se va a realizar durante todo el proceso de codificación, el siguiente título que tenemos es haciendo columna en donde se enlistan las actividades (sprint)

que se están desarrollando en el presente y para finalizar tenemos la columna titulada completado que es referente a las actividades que se han terminado con éxito en base a sprints.

2.5.2 Requerimientos vs casos de uso

Tabla 32: Casos de uso y requerimientos

Caso de Uso	RF01	RF02	RF03	RF04	RF05	RF06	RF07	RF08
CU01 - Iniciar Sesión	✓							
CU02 - Registro Siniestro Informativo		✓	✓	✓			✓	
CU03 - Lista Siniestros Informativos		✓			✓			
CU04 - Consultar Modelo Predictivo								✓
CU05 - Registro Siniestro con Parte		✓	✓	✓			✓	
CU06 - Asignación de Código			✓					
CU07 - Lista Códigos Asignados		✓	✓		✓			
CU08 - Lista Siniestros con Parte		✓			✓			
CU09 - Lista Pública Siniestros						✓		
CU10 - Notificaciones Siniestros						✓		

Nota: esta tabla demuestra en que parte del caso de uso se está cumpliendo con los requerimientos funcionales

Mediante el desarrollo de una matriz de trazabilidad entre casos de uso y requerimientos funcionales, lo que permite mostrar que requisito funciona se cumple en cada caso de usos en el aplicativo web, entendiendo a la primera columna como las acciones principales de los casos de uso y las demás columnas son todos los requerimientos funcionales y cada celda indica con un check a que caso de uso y requerimiento estan relacionados

2.6 Fase 3: Lanzamiento

La fase de lanzamiento hace hincapié a la finalización del proyecto, entrega del sistema.

2.6.1 Liberación de actividades en la taba Kanban

Tabla 33: Liberación sprint 1 - 3

Módulo/Funcionalidad	Sprint 1	Sprint 2	Sprint 3	Estado
Módulo de Autenticación	100%			Completado
- Login con Gmail/Institucional	100%			Completado
- Gestión de roles	100%			Completado
- Grupos de control	100%			Completado
Módulo de Formularios	60%	100%		Completado
- Formularios siniestros informativos	30%	100%		Completado
- Formularios siniestros con parte		80%	100%	Completado
Códigos Únicos para parte		100%		Completado
- Generación automática códigos ligados a los siniestros con parte		100%		Completado
- Validación unicidad		100%		Completado
Módulo de Autoguardado		100%		Completado
- Guardado automático borradores		100%		Completado
- Recuperación de sesión		100%		Completado

Nota: Esta tabla demuestra cómo se fueron trabajando en los sprints en diferentes procesos hasta llegar al estado de completado

Tabla 34: Liberación sprints 4 - 5

Módulo/Funcionalidad	Sprint 4	Sprint 5	Estado
Módulo de Gestión Personal	80%	100%	Completado
- Lista siniestros propios	100%		Completado
- Edición siniestros propios	60%	100%	Completado
- Cambio estado borrador→activo		100%	Completado
Módulo de Geolocalización	70%	100%	Completado
- Captura ubicación tiempo real	100%		Completado
- Integración mapas	40%	100%	Completado
Módulo de Evidencias	50%	100%	Completado
- Upload fotos	80%	100%	Completado
- Validación archivos	20%	100%	Completado
- Compresión imágenes		100%	Completado

Nota: Explica cómo se fue avanzando con el proyecto las funcionalidades propias de los agentes

Tabla 35: Liberación sprint 6

Módulo/Funcionalidad	Sprint 6	Estado
Módulo de Mapa Público	90%	Completado
- Visualización mapa interactivo	100%	Completado
- Filtros por ubicación	80%	Completado
- Información estado vías	90%	Completado
Sistema de Notificaciones	60%	Completado
- Notificaciones push	80%	Completado
- Notificaciones email	40%	Completado
- Configuración preferencias	60%	Completado

Nota: En esta tabla se ve cómo se avanzó progresivamente con la funcionalidad pública del aplicativo web, la parte a la que el conductor tendría acceso.

Tabla 36: Liberación de sprints 7 - 9

Módulo/Funcionalidad	Sprint 7	Sprint 8	Sprint 9	Estado
Módulo Predictivo	30%	70%	100%	Completo
- Procesamiento datos históricos	40%	80%	100%	Completo
- Algoritmos de predicción	20%	60%	100%	Completo
- Identificación puntos negros		70%	100%	Completo
Dashboard Analítico		50%	100%	Completo
- Reportes estadísticos		60%	100%	Completo
- Gráficos predictivos		40%	100%	Completo
- Exportación reportes			100%	Completo

Nota: En esta tabla se muestra cómo se fue trabajando en el cumplimiento de diferentes sprints relacionados con el modelo predictivo

Tabla 37: Liberación sprint 10

Módulo/Funcionalidad	Sprint 10	Estado
Diseño Responsivo	100%	Completo
- Optimización móvil	100%	Completo
- Optimización tablet	100%	Completo
- Tests cross-browser	100%	Completo
Performance y Seguridad	100%	Completo
- Optimización base datos	100%	Completo
- Auditoria seguridad	100%	Completo
- Tests de carga	100%	Completo

Nota: Los sprints que no son de complejidad alta como el diseño responsive se pudo completar en su totalidad

Tabla 38: Liberación general

Release	Progreso	Fecha Estimada	Estado
Release 1 - MVP Core	100%	Semana 9	Completado
Release 2 - Gestión Personal	75%	Semana 10	Completado
Release 3 - Portal Público	15%	Semana 12	Completado
Release 4 - IA Predictiva	5%	Semana 18	Completado
Release 5 - Optimización	0%	Semana 20	Completado

Nota: Resumen de los realease es decir las liberaciones de los sprints de acuerdo con el avance del proyecto

2.6.2 Retrospectiva del proyecto

Al concluir el proyecto de desarrollo e implementación de un aplicativo web para la UCOT Unidad de control operativo de la ciudad de Loja en el año 2025 se puede constatar que los requerimientos presentados fueron exitosamente realizados, de esta forma se agiliza el proceso realizado por los agentes de tránsito al momento de registrar un siniestro tanto informativo como oficial y obteniendo una liberación de sprints del 100%.

Gracias al punto de retrospectiva el proyecto pudo pasar por muchas faces de revisiones para retroalimentarse y mejorar su navegabilidad, usabilidad y funcionalidad.

2.7 Fase 4: Pruebas

2.7.1 Informe de pruebas

En esta fase una vez que ya todo está maquetado e implementado, se deberá hacer la revisión del proceso, y además realizar las pruebas necesarias para garantizar el uso y la fiabilidad del sistema desarrollado.

Tabla 39: Módulos y Funciones a evaluar

Release	Función	Categoría
Release 1 - MVP	Autenticación con Gmail/Institucional	Evidente
Release 1 - MVP	Diseño de Interfaz Responsiva	Evidente
Release 1 - MVP	Formularios de siniestros informativos	Evidente
Release 1 - MVP	Sistema de códigos únicos	Oculto
Release 1 - MVP	Autoguardado de formularios	Superflua
Release 2 - Core	Formularios de siniestros con parte oficial	Evidente
Release 2 - Core	Módulo de gestión de usuarios	Oculto
Release 2 - Core	Módulo de roles y permisos	Superflua
Release 2 - Core	Módulo de grupos de control	Evidente
Release 2 - Core	Módulo de agentes de tránsito	Evidente
Release 3 - Avanzado	Geolocalización en tiempo real	Evidente
Release 3 - Avanzado	Upload y gestión de evidencia fotográfica	Evidente
Release 3 - Avanzado	Mapa público de siniestros	Evidente
Release 3 - Avanzado	Sistema de notificaciones	Superflua
Release 4 - IA	Modelo predictivo de puntos negros	Oculto
Release 4 - IA	Dashboard analítico	Superflua
Adicional	Optimización para dispositivos móviles	Evidente

Nota: Tabla de pruebas

Tabla 40: Evaluación de rendimiento

Función	Estado
Tiempo de respuesta en autenticación (<2 segundos)	Aprobado
Carga de formularios de siniestros (<3 segundos)	Aprobado
Procesamiento simultáneo de múltiples registros	Aprobado
Validación de datos en formularios en tiempo real	Aprobado
Generación automática de códigos únicos	Aprobado
Upload de imágenes y compresión automática	Aprobado
Consulta de geolocalización (<5 segundos)	Aprobado
Renderizado de mapas públicos	Aprobado
Procesamiento de modelo predictivo	Aprobado
Sincronización de datos entre usuarios	Aprobado
Notificaciones en tiempo real	Aprobado
Optimización de consultas de base de datos	Aprobado

Nota: Tabla con datos de rendimiento del sistema de siniestros probados con diferentes herramientas tecnologías disponibles en internet como page speed referencia figura 41 y 42

Tabla 41: Datos de restricciones

Restricción	Número
Acceso solo mediante conexión a internet estable	1
Compatibilidad con navegadores Chrome 90+, Firefox 88+, Safari 14+	2
Servidor con capacidad mínima de 100 usuarios concurrentes	3
Ancho de banda mínimo de 5 Mbps para funciones de geolocalización	4
Base de datos PostgreSQL 13+ o MySQL 8.0+	5
Dispositivos con GPS habilitado para geolocalización	6
Almacenamiento mínimo de 2GB para evidencia fotográfica	7
Compatibilidad móvil solo para pantallas 320px+	8
Acceso a APIs de mapas (Google Maps o OpenStreetMap)	9
Permisos de ubicación habilitados en dispositivos	10

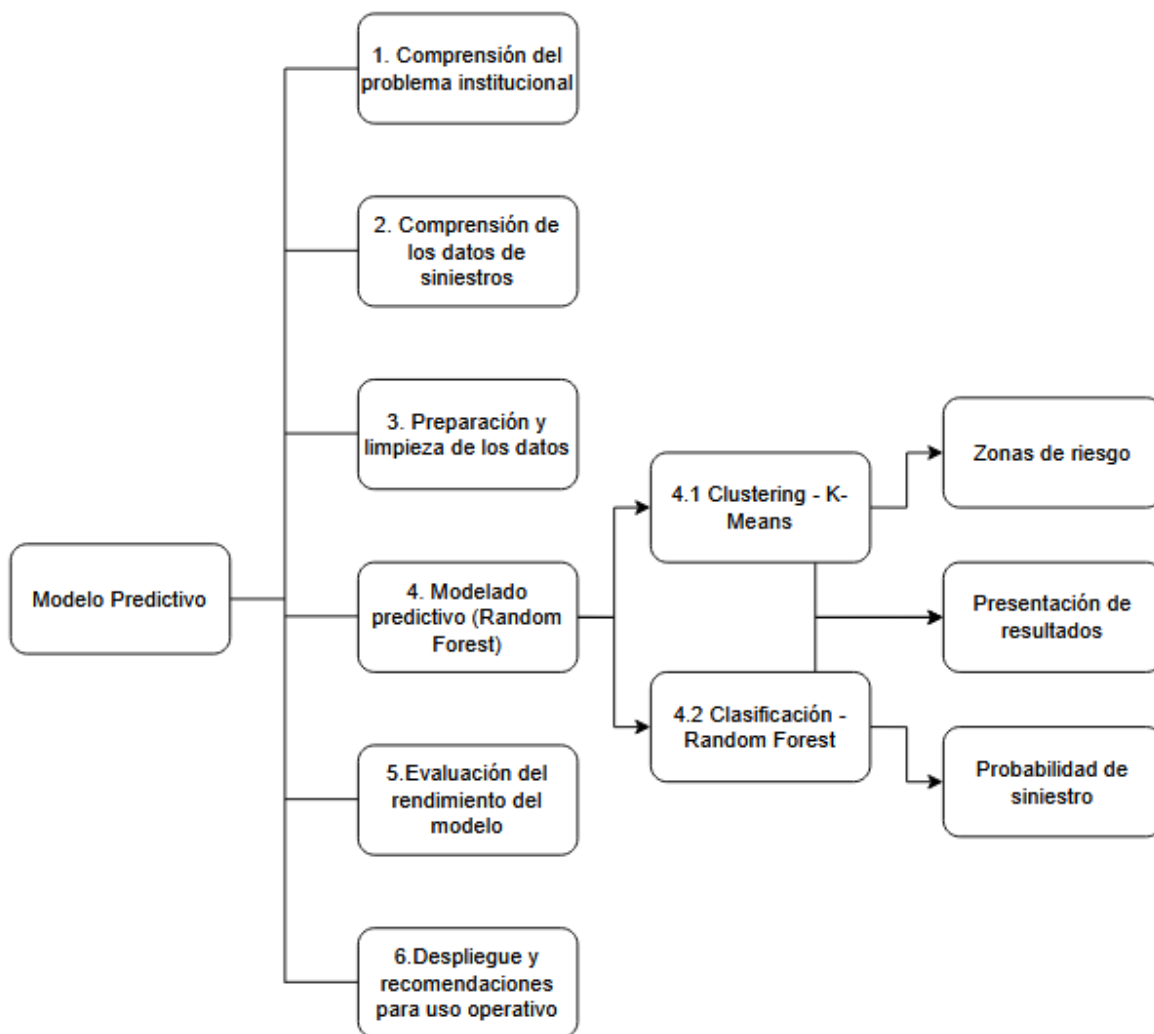
Nota: Tabla que muestra las restricciones para el uso del sistema de siniestros

Tabla 42: Pruebas de Seguridad

Aspecto de Seguridad	Prueba	Estado
Autorización	Acceso restringido según rol de usuario	Aprobado
Datos	Encriptación de información sensible	Aprobado
Comunicación	HTTPS en todas las comunicaciones	Aprobado
Sesiones	Timeout automático de sesiones inactivas	Aprobado
Inputs	Validación y sanitización de formularios	Aprobado
Archivos	Validación de tipos de archivo permitidos	Aprobado
Base de Datos	Protección contra inyección SQL	Aprobado

2.8 Desarrollo del Modelo predictivo

Figura 12: Arquitectura del modelo predictivo

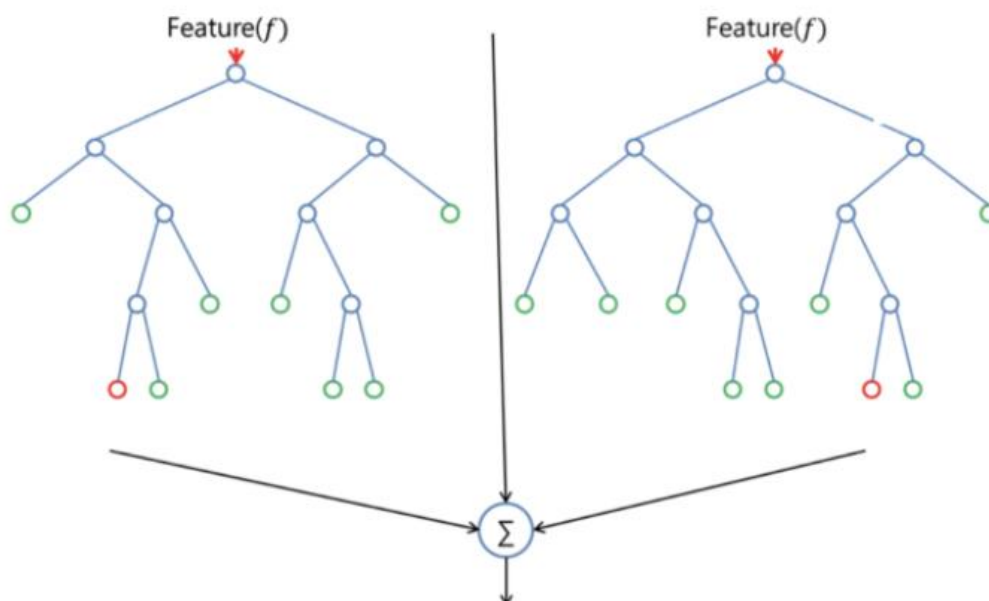


Para el desarrollo de la arquitectura se hizo un análisis desde la comprensión del problema obteniendo seis procesos principales como se muestra en la *figura 2 Arquitectura del modelo predictivo* para entender cada uno de estos procesos se tiene la siguiente explicación:

1. Comprensión del problema institucional. _ Se presenta la necesidad urgente de mejorar el registro, análisis y prevención de siniestros, actualmente gestionados de forma manual por la UCOT, con alto riesgo de pérdida de información y dificultad en la toma de decisiones.
2. Comprensión de los datos de siniestros. _ Se analizaron los datos históricos proporcionados por la UCOT, que incluyen variables como fecha, hora, ubicación, tipo de siniestro, condiciones climáticas y datos del conductor.
3. Preparación y limpieza de los datos. _ Se depuraron, normalizaron y transformaron los datos para poder trabajar con la codificación de variables categóricas y obtener datos limpios óptimos para el trabajo con el modelo de predicción.
4. Modelado predictivo (clustering y clasificación). _ Se implementaron algoritmos de 4.1 *clustering* (K-Means) para identificar zonas críticas “puntos negros” y algoritmos de 4.2 *clasificación* (Random Forest) para predecir riesgos de siniestros.
5. Evaluación del rendimiento del modelo. _ Se aplicaron métricas como matriz de confusión y mapas de calor geográficos para validar la precisión del modelo y su utilidad en la detección de patrones relevantes.
6. Despliegue y recomendaciones para uso operativo. _ Se propone integrar el modelo en una plataforma digital de apoyo para la UCOT, que permita visualizar zonas críticas además de mejorar la planificación de los agentes y las leyes de prevención y control.

2.8.1 Modelo de clasificación random forest

Figura 13: Estructura gráfica de Random Forest



2.9 Arquitectura del modelo

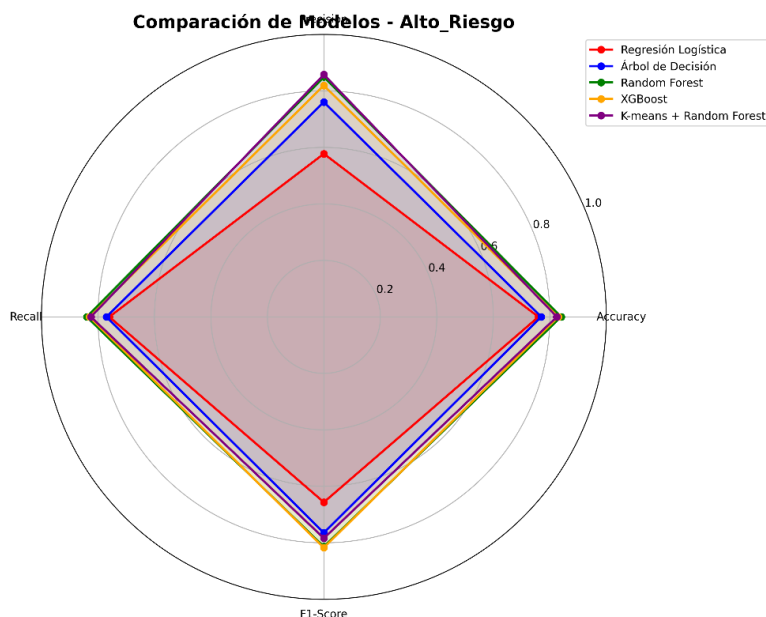
Para usar el algoritmo adecuado que pueda presentar los resultados esperados se ha trabajado con un enfoque híbrido que combina algoritmos de categorización como K-Means para la detección de puntos negros en la provincia y con algoritmos de clasificación como random forest para predecir la probabilidad de ocurrencia bajo ciertas condiciones históricas.

2.10 Técnicas de análisis

De acuerdo al alto análisis y evaluación de comparación entre los diferentes modelos de predicción presentado en la tabla 14 del presente documento se evidenció que el modelo random forest tiene alto desempeño en actividades de predicción con volúmenes de datos históricos para de esta forma obtener con precisión las condiciones bajo las cuales es más probable que ocurra un siniestro de tránsito, mientras que con el algoritmo K- Means de clasificación se agrupa

especialmente los siniestros de acuerdo a su ubicación concentrándonos en los resultados de puntos negros.

Figura 14: Comparación de Modelos en función a target Alto riesgo



2.10.1 Integración de técnicas de agrupamiento

Con una arquitectura híbrida como lo muestra la figura 9 del presente documento permite un análisis más completo del fenómeno de aumento exponencial de los siniestros de tránsito ya que no solo detecta zonas de alto impacto o problemáticas, sino también anticipa condiciones de riesgo como condiciones viales, climáticas o del conductor lo que representa un gran avance frente al registro manual.

Ya que es una solución adaptable, su eficacia podría incrementarse de acuerdo con cómo se vayan incrementando datos históricos, lo que permite una mejora continua en la precisión de los modelos predictivos, permitiendo el uso proactivo de las herramientas tecnológicas para mejorar la calidad de vida de los conductores y sus pasajeros.

CAPÍTULO III: DESARROLLO E IMPLEMENTACIÓN

3.1 Descripción del problema

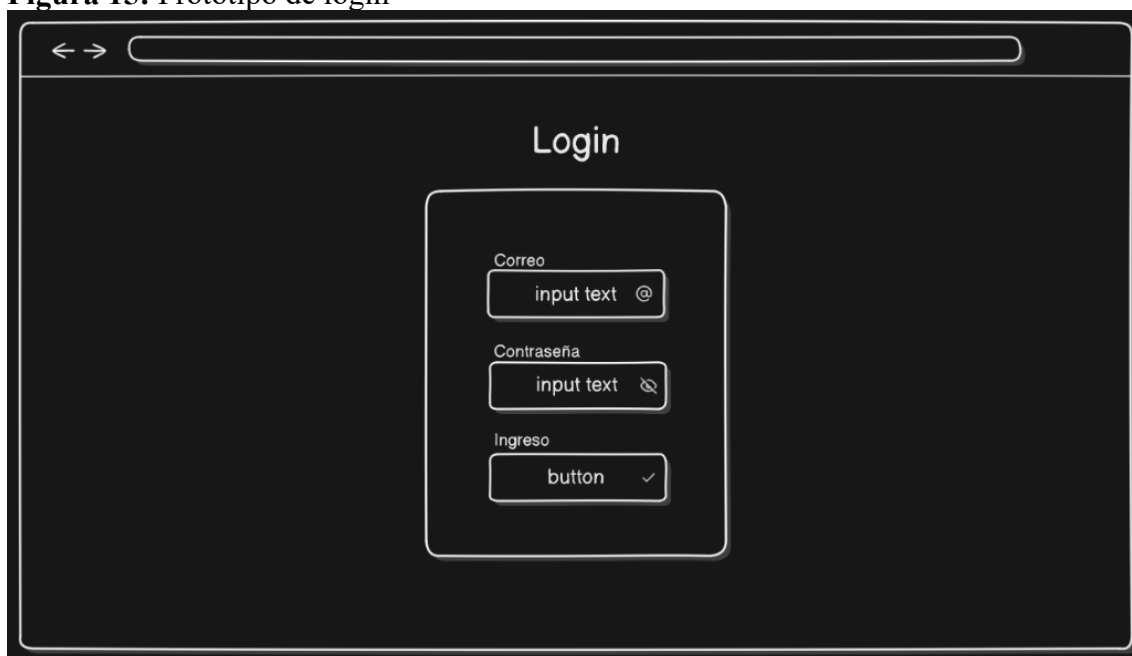
En la actualidad el registro llevado por parte de los agentes de tránsito se realiza de forma manual, por lo que debido a este ineficiente proceso puede existir el riesgo de pérdida, adulteración o daño en estos documentos, y tampoco es un proceso eficaz y eficiente, por este motivo se presenta dificultad para la llegada de los servicios de emergencia como es asistencia médica y asistencia vehicular a los lugares del accidente.

Los siniestros de tránsito se han dado la mayoría de las veces por descuido o por alguna falta cometida por los conductores vehiculares, por lo que representan una de las mayores causas de mortalidad a pesar de los esfuerzos realizados por las autoridades por presentar propuestas que minimicen estos peligros en la vía, una mejora significativa para el problema se puede presentar como una herramienta tecnológica

3.2 Prototipado

3.2.1 Inicio de sesión

Figura 15: Prototipo de login

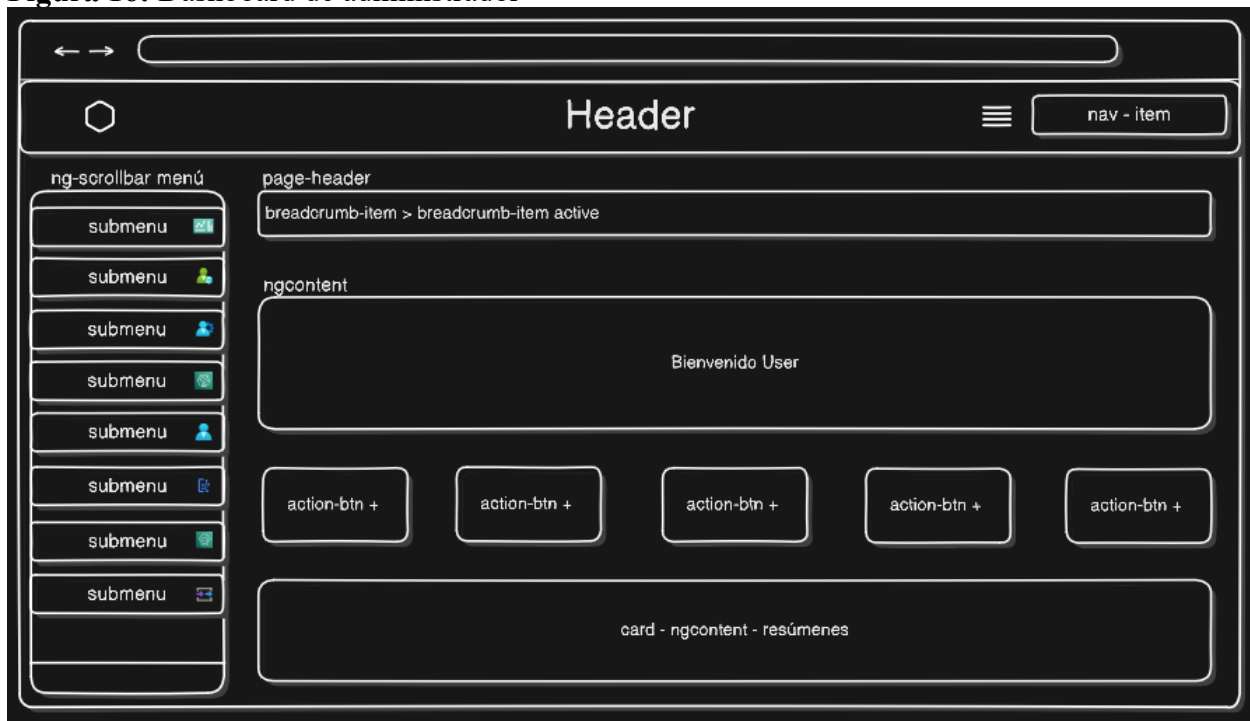
A wireframe prototype of a login page. It features a dark blue background. At the top, there is a header bar with a back arrow, a forward arrow, and a search bar. Below the header, the word "Login" is centered in a large, white, sans-serif font. Underneath "Login", there is a rounded rectangle containing three input fields. The first field is labeled "Correo" and contains the text "input text" followed by an email icon. The second field is labeled "Contraseña" and contains the text "input text" followed by an eye icon. The third field is labeled "Ingreso" and contains the text "button" followed by a checkmark icon.

Nota: Elaboración Propia

- **Header:** Título "Login" centrado.
- **Formulario de autenticación:**
 - Campo "Correo" (input text con ícono de usuario).
 - Campo "Contraseña" (input text con ícono de candado).
 - Botón "Ingreso" (dropdown button).
- **Layout:** Formulario centralizado en fondo oscuro.

3.2.2 DashBoard

Figura 16: Dashboard de administrador



Nota: Elaboración Propia

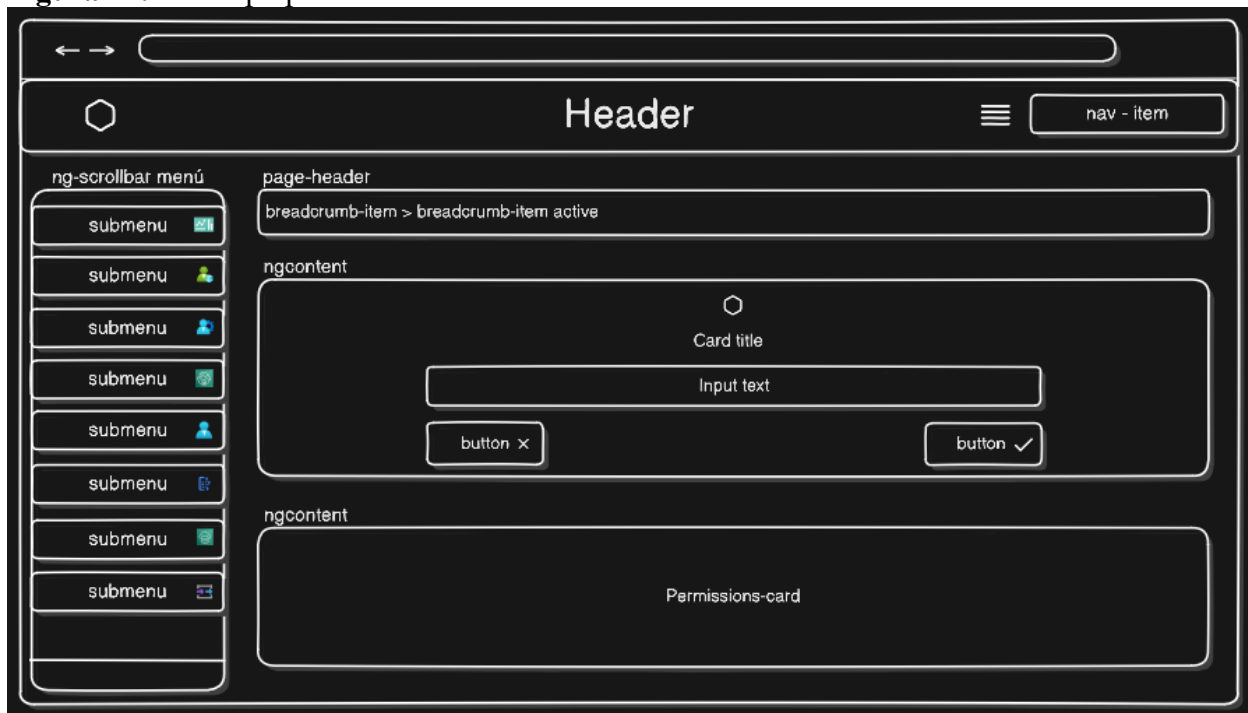
Componentes:

- **Header:** Logo hexagonal + título "Header" + menú hamburguesa + nav-item.
- **Sidebar izquierdo:** 8 submenús con íconos de colores variados.
- **Breadcrumb:** Navegación "breadcrumb-item > breadcrumb-item active".
- **Contenido principal:**
 - Mensaje "Bienvenido User".
 - 5 botones de acción ("action-btn +").

- Card de resúmenes (ngcontent).
- **Layout responsivo:** Sidebar fijo, contenido adaptable.

3.2.3 Crear rol

Figura 17: Prototipo para crear rol



Nota: Elaboración Propia

Componentes:

- **Header:** Igual al dashboard.
- **Sidebar:** Misma estructura de navegación.
- **Contenido principal:**
 - **Card superior:**
 - Título "Card title".
 - Input text.
 - Botones de acción (X y ✓).
 - **Card inferior:** "Permissions-card" (área de permisos).
- **Layout:** Dos cards apiladas verticalmente.

3.2.4 Lista de roles

Figura 18: Prototipo de lista de roles



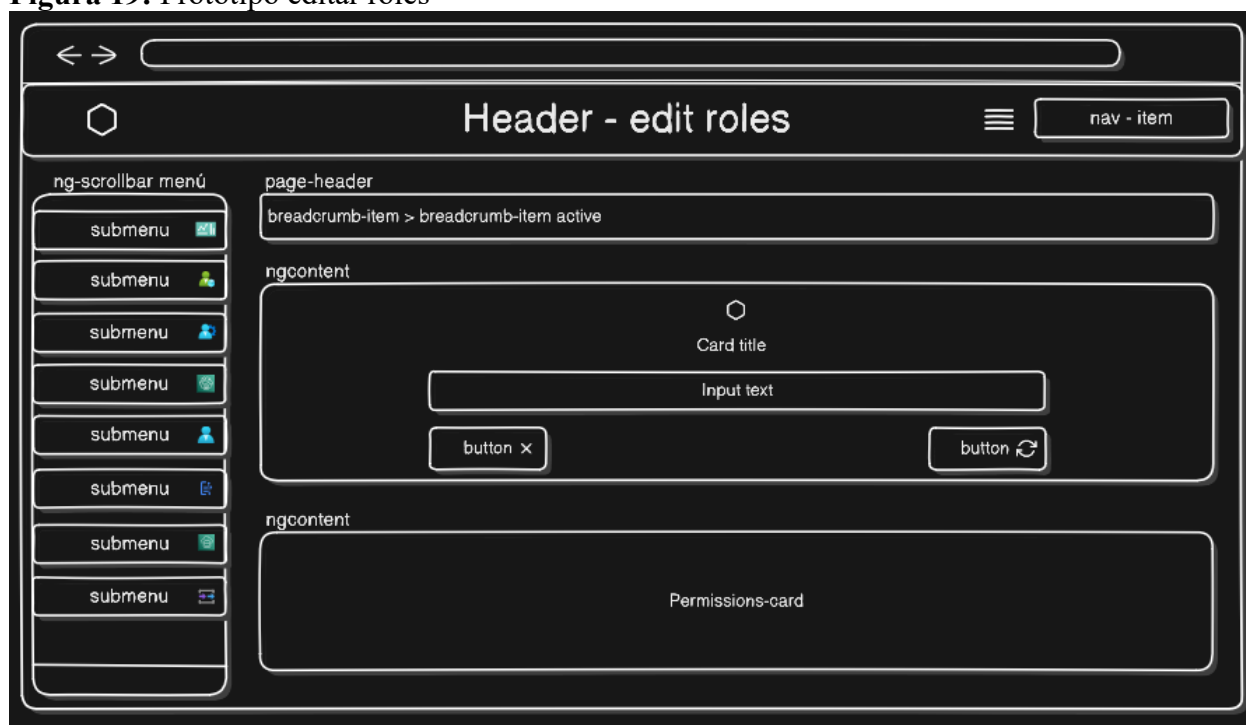
Nota: Elaboración Propia

Componentes:

- **Header:** "Header - list roles".
- **Barra de búsqueda:** Input text + botones "+" y "↺".
- **Tabla de datos:**
 - Headers: "name", "name", "name", "name".
 - Fila de datos: "name item", "data permisos", "data fecha".
 - Botones de acción por fila (editar, eliminar).
- **Layout:** Tabla responsiva con acciones inline.

3.2.5 Editor rol

Figura 19: Prototipo editar roles



Nota: Elaboración Propia

Componentes:

- **Header:** "Header - edit roles".
- **Card de edición:**
 - Título "Card title".
 - Input text para modificación.
 - Botones X (cancelar) y ↺ (actualizar).
- **Permissions-card:** Área de gestión de permisos.
- **Layout:** Estructura similar a crear, pero con datos precargados.

3.2.6 Crear usuario

Figura 20: Prototipo de agregar usuario

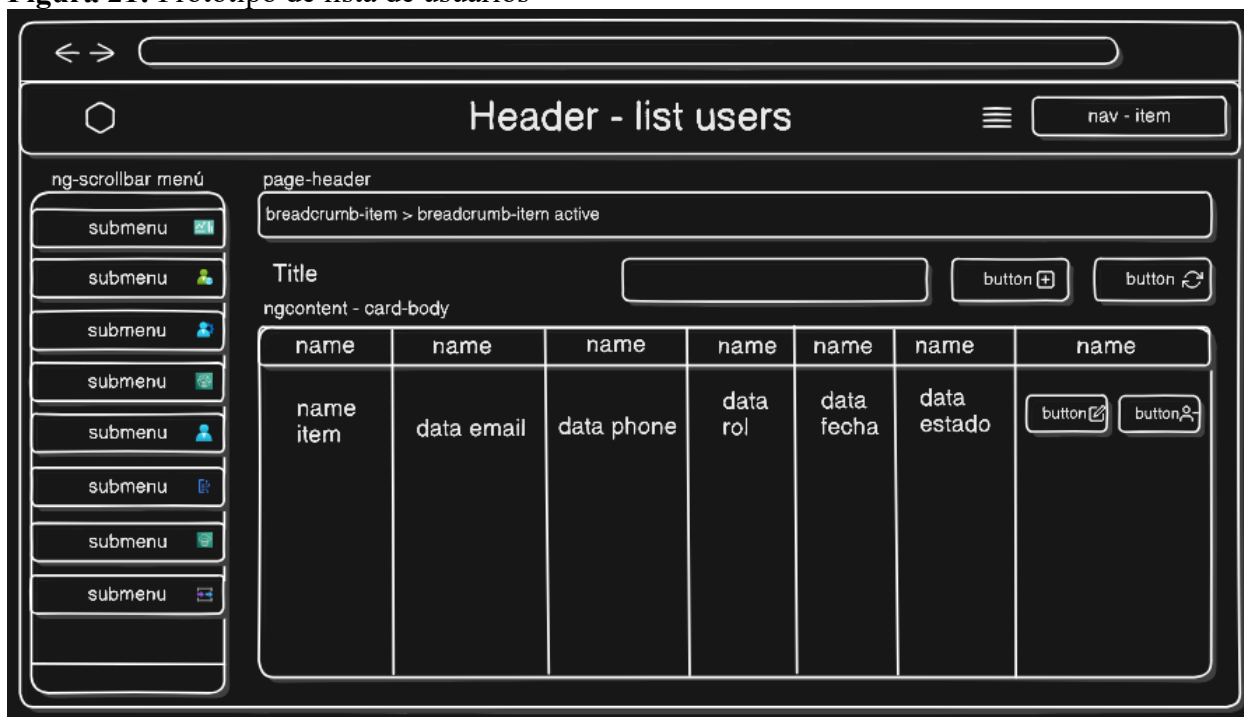
Nota: Elaboración Propia

Componentes:

- **Header:** "Header - add user".
- **Formulario de usuario:**
 - **form-content:** Grid 2x4 de inputs.
 - campos "Input text" organizados en dos columnas.
 - Botón X (cancelar) y ✓ (guardar).
- **Layout:** Formulario estructurado en grid responsive.

3.2.7 Lista de usuarios

Figura 21: Prototipo de lista de usuarios



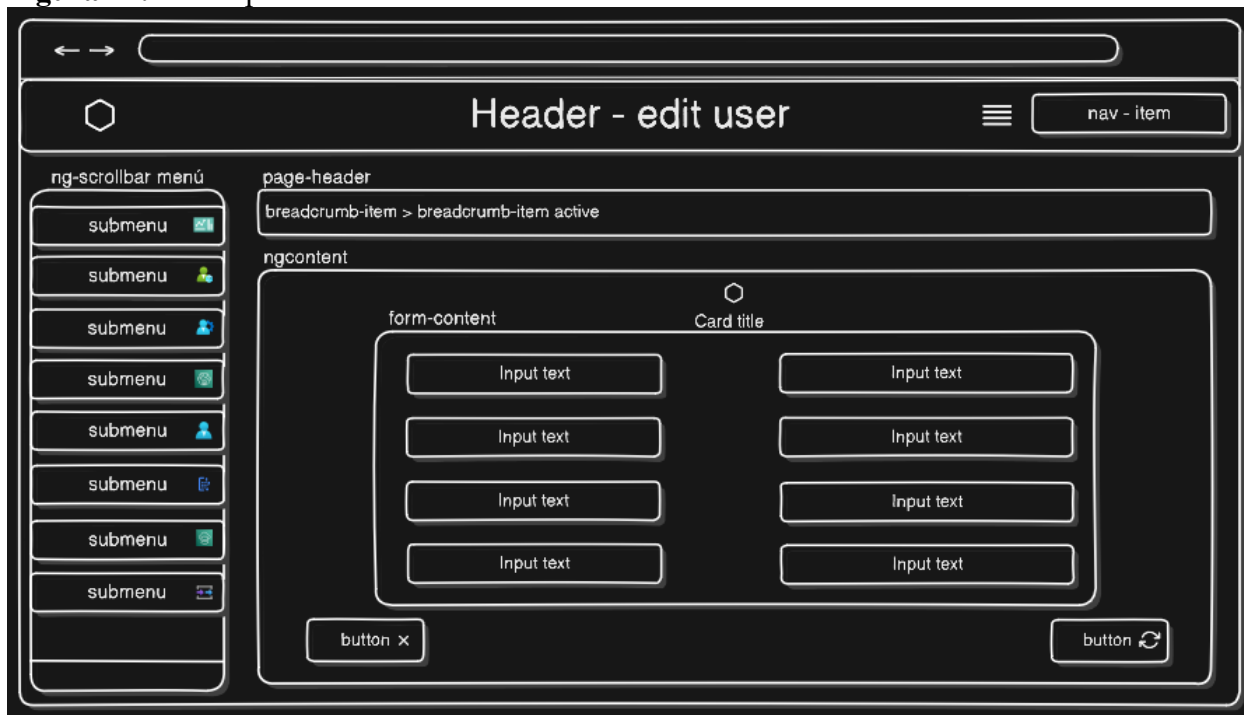
Nota: Elaboración Propia

Componentes:

- **Header:** "Header - list users".
- **Barra de búsqueda:** Input "Title" + botones "+" y "↺".
- **Tabla de usuarios:**
 - Headers: 7 columnas (name x6, name-acciones).
 - Fila ejemplo: "name item", "data email", "data phone", "data rol", "data fecha", "data estado".
 - Botones de acción por fila (editar, eliminar).
- **Layout:** Tabla extendida con más columnas de información.

3.2.8 Editar usuario

Figura 22: Prototipo de editar usuarios



Nota: Elaboración Propia

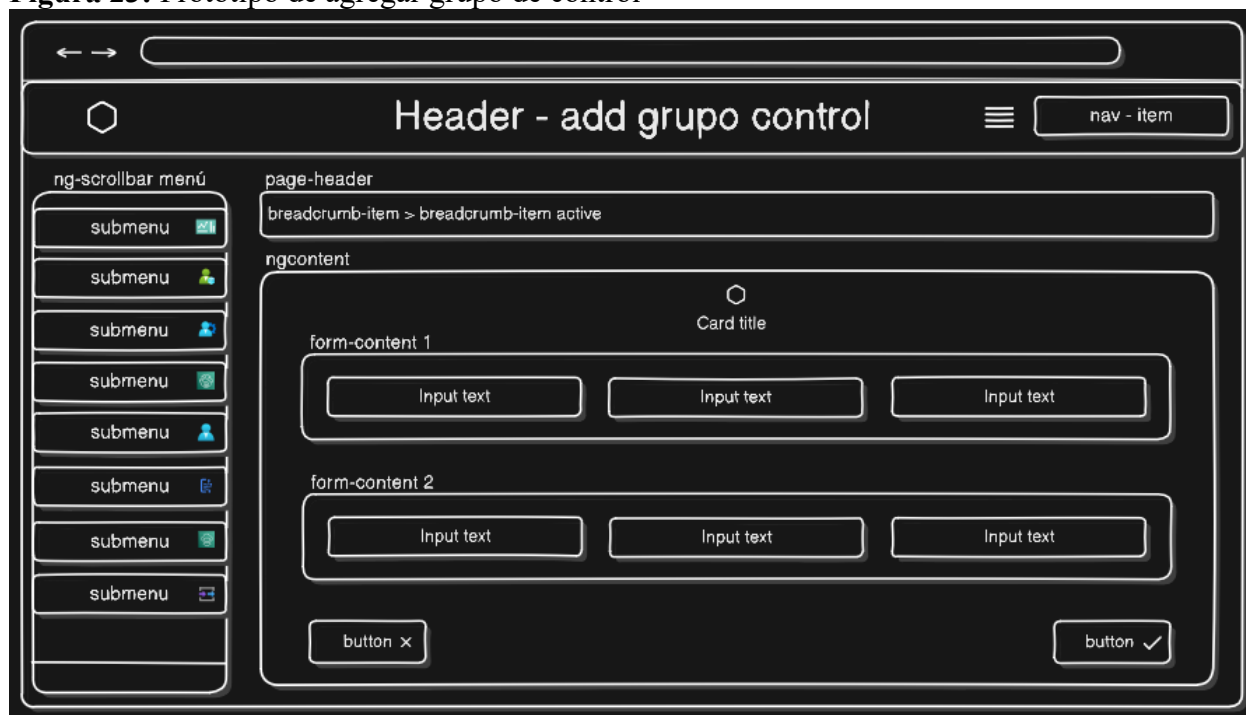
Componentes:

- **Header:** "Header - edit user".
- **Formulario de edición:**
 - **form-content:** Grid 2x4 con datos precargados.
 - campos "Input text" en dos columnas.
 - Botones X (cancelar) y ↻ (actualizar).

Layout: Idéntico a "add user" pero para modificación.

3.2.9 Crear grupo de control

Figura 23: Prototipo de agregar grupo de control



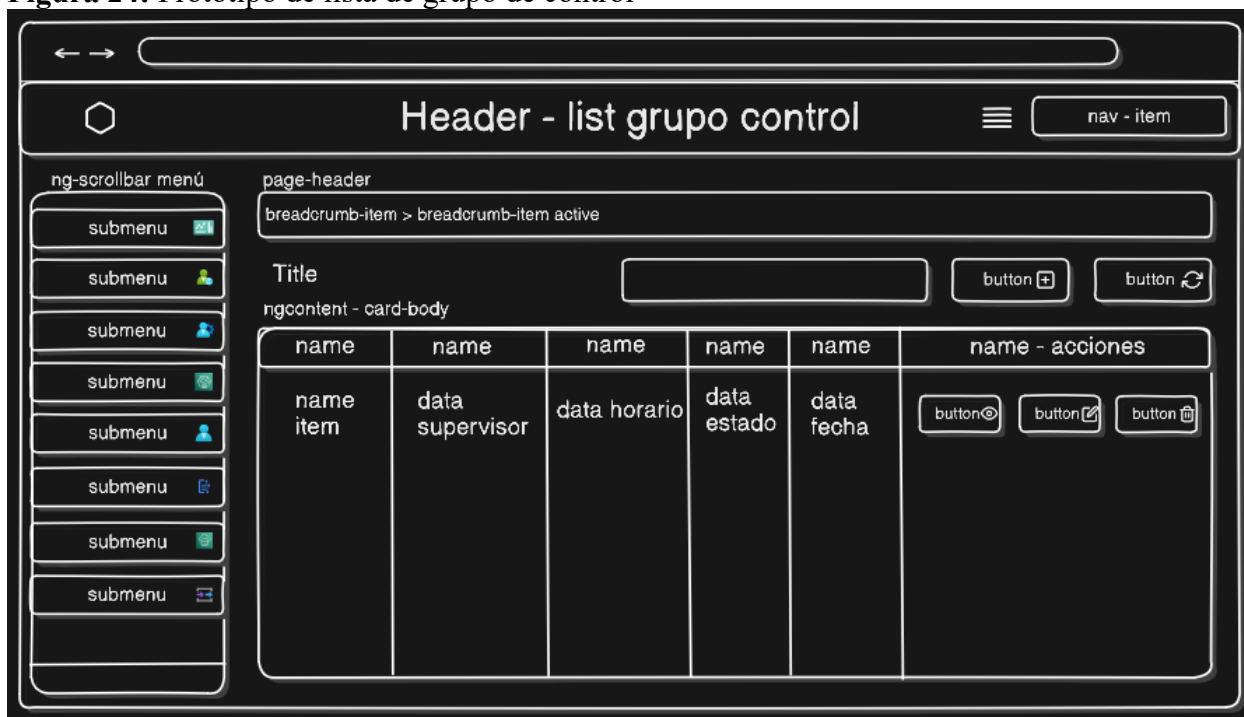
Nota: Elaboración Propia

Componentes:

- **Header:** "Header - add grupo control".
- **Formularios anidados:**
 - **form-content 1:** 3 inputs horizontales.
 - **form-content 2:** 3 inputs horizontales adicionales.
 - Botones X (cancelar) y ✓ (guardar).
- **Layout:** Dos secciones de formulario apiladas verticalmente.

3.2.10 Lista de grupo de control

Figura 24: Prototipo de lista de grupo de control



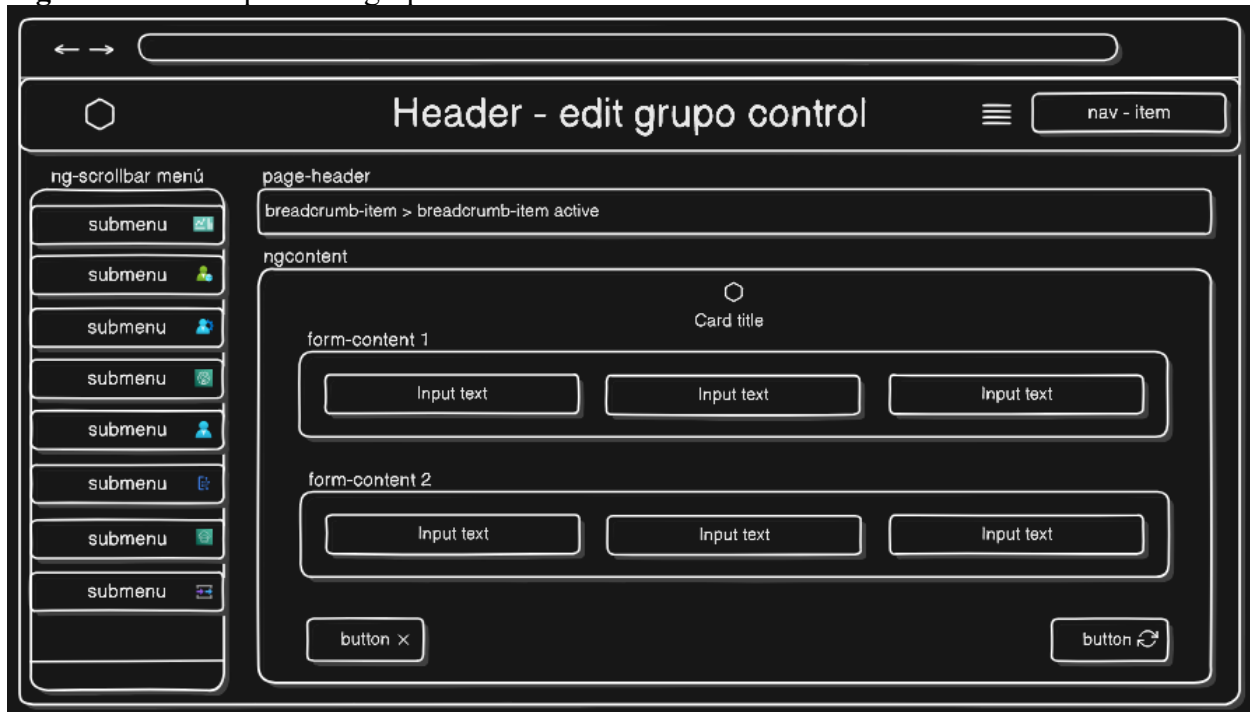
Nota: Elaboración Propia

Componentes:

- **Header:** "Header - list grupo control".
- **Barra de búsqueda:** Input + botones de acción.
- **Tabla de grupos:**
 - Headers: "name" (x5) + "name-acciones".
 - Fila datos: "name item", "data supervisor", "data horario", "data estado", "data fecha".
 - 3 botones de acción por fila.
- **Layout:** Tabla con información específica de grupos de control.

3.2.11 Editar grupo de control

Figura 25: Prototipo editar grupo de control



Nota: Elaboración Propia

Componentes:

- **Header:** "Header - edit grupo control".
- **Formularios de edición:**
 - **form-content 1:** 3 inputs con datos precargados.
 - **form-content 2:** 3 inputs adicionales precargados.
 - Botones X (cancelar) y ↻ (actualizar).
- **Layout:** Estructura idéntica a "add grupo control" pero para edición.

3.2.12 Crear agentes

Figura 26: Prototipo de agregar agentes

The image shows a web application prototype for adding agents. It features a dark-themed interface with a header, a sidebar menu, a breadcrumb trail, and a main form area.

- Header:** Contains a hexagonal logo, the title "Header - add agentes", a hamburger menu icon, and a navigation item "nav - item".
- Sidebar (ng-scrollbar menú):** A vertical list of "submenu" items, each accompanied by a small icon (e.g., a person, a gear, a document).
- Breadcrumb (page-header):** Displays "breadcrumb-item > breadcrumb-item active".
- Main Content (ngcontent):**
 - Card:** A container with a "Card title" and a "form-content" area.
 - Form-content:** A 2x4 grid of "Input text" fields.
 - Buttons:** At the bottom of the form, there are two buttons: "button x" (cancel) and "button ✓" (save).

Nota: Elaboración Propia

Componentes:

- **Header:** "Header - add agentes".
- **Formulario de agente:**
 - **form-content:** Grid 2x4 de inputs.
 - campos "Input text" organizados en columnas.
 - Botones X (cancelar) y ✓ (guardar).
- **Layout:** Formulario estándar similar a usuarios.

3.2.13 Lista de agentes

Figura 27: Prototipo lista de agentes



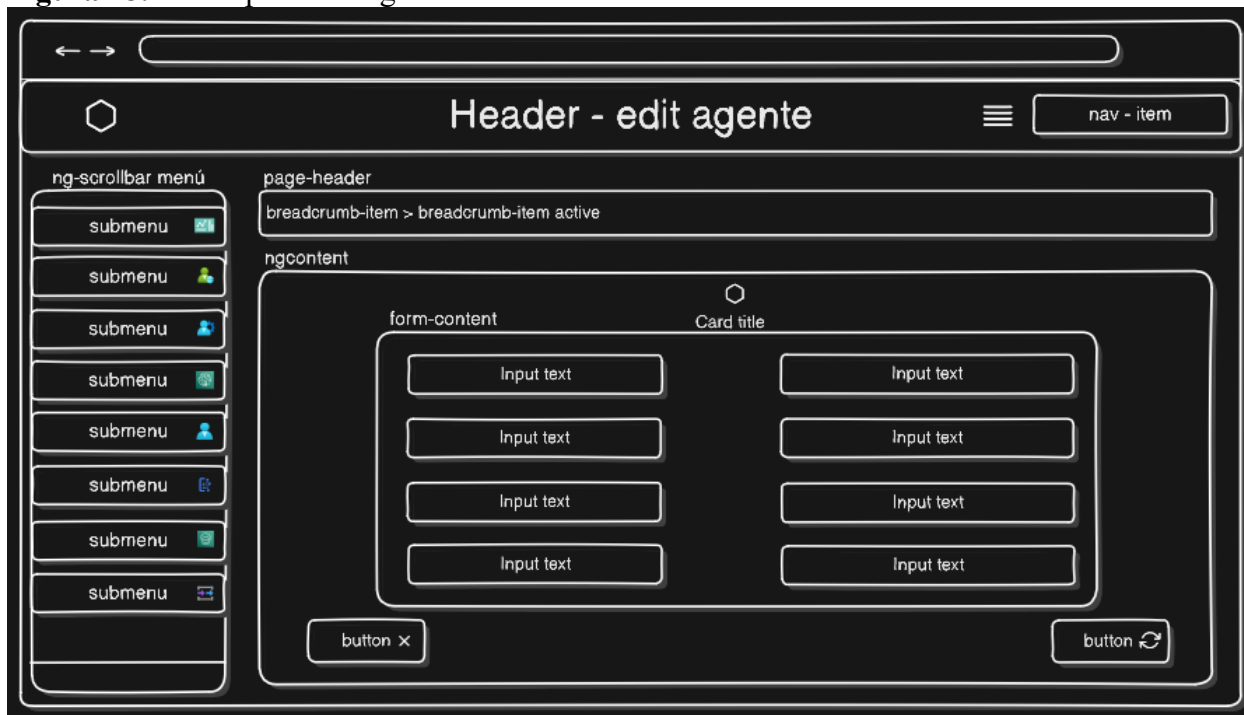
Nota: Elaboración Propia

Componentes:

- **Header:** "Header - list agentes".
- **Tabla de agentes:**
 - Headers: 6 columnas incluyendo "name-acciones".
 - Fila ejemplo: "name item", "data email", "data grupo control", "data estado", "data fecha".
 - 3 botones de acción por fila.
- **Layout:** Tabla con información específica de agentes.

3.2.14 Editar agente

Figura 28: Prototipo editar agente



Nota: Elaboración Propia

Componentes:

- **Header:** "Header - edit agente".
- **Formulario de edición:**
 - **form-content:** Grid 2x4 con datos precargados.
 - campos "Input text" en dos columnas.
 - Botones X (cancelar) y ↺ (actualizar).
- **Layout:** Estructura idéntica a "add agentes" pero para modificación.

3.2.15 Agregar siniestro informativo

Figura 29: Prototipo agregar siniestro informativo

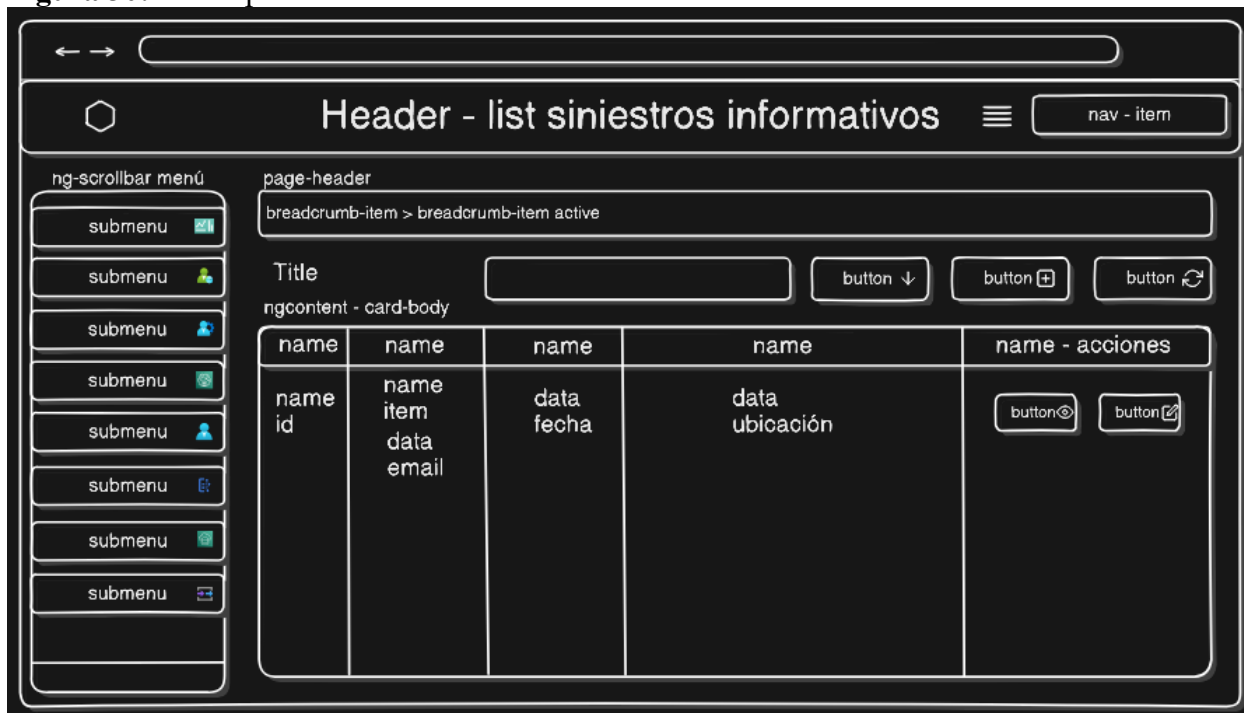
Nota: Elaboración Propia

Componentes:

- **Header:** "Header - add siniestro informativo".
- **Formulario por pasos:**
 - **Indicadores de progreso:** 3 radio buttons (Primer paso de datos, Segundo paso de datos, Tercer paso de datos).
 - **form-content:** Grid 2x3 de inputs (6 campos).
 - Botones X (cancelar) y ✓ (guardar).
- **Layout:** Formulario wizard con navegación por pasos.

3.2.16 Lista de siniestros informativos

Figura 30: Prototipo lista de siniestros informativos



Nota: Elaboración Propia

Componentes:

- **Header:** "Header - list siniestros informativos".
- **Barra de búsqueda:** Input "Title" + botón descarga + botones "+" y "⌂".
- **Tabla de siniestros:**
 - Headers: 5 columnas (4 "name" + "name-acciones").
 - Fila ejemplo: "name id data email", "name item", "data fecha", "data ubicación".
 - 2 botones de acción por fila.

Layout: Tabla específica para información de siniestros públicos.

3.2.17 Editar siniestros informativos

Figura 31: Prototipo editar siniestro informativo

Nota: Elaboración Propia

Componentes:

- **Header:** "Header - edit siniestro informativo".
- **Formulario de edición por pasos:**
 - **Indicadores de progreso:** 3 radio buttons (Primeros datos, Segundos datos, Terceros datos).
 - **form-content:** Grid 2x3 con datos precargados.
 - Botones X (cancelar) y ↻ (actualizar).
- **Layout:** Formulario wizard de 3 pasos para edición de siniestros informativos.

3.2.18 Agregar siniestro oficial con parte

Figura 32: Prototipo agregar siniestro oficial con parte

El prototipo muestra una interfaz de usuario con los siguientes elementos:

- Encabezado:** Título "Header - add siniestro con parte" y un botón de navegación "nav - item".
- Menú lateral:** Un "ng-scrollbar menú" con ocho ítems "submenu", cada uno con un ícono.
- Encabezado de página:** "page-header" con el breadcrumb "breadcrumb-item > breadcrumb-item active".
- Formulario:** Un "ngcontent" que contiene un "form-content" con:
 - Seis indicadores numerados (1-6) con el texto "item" debajo de cada uno.
 - Una grilla de 2x3 de campos de texto "Input text".
 - Un botón "button ×" a la izquierda y un botón "button ✓" a la derecha.

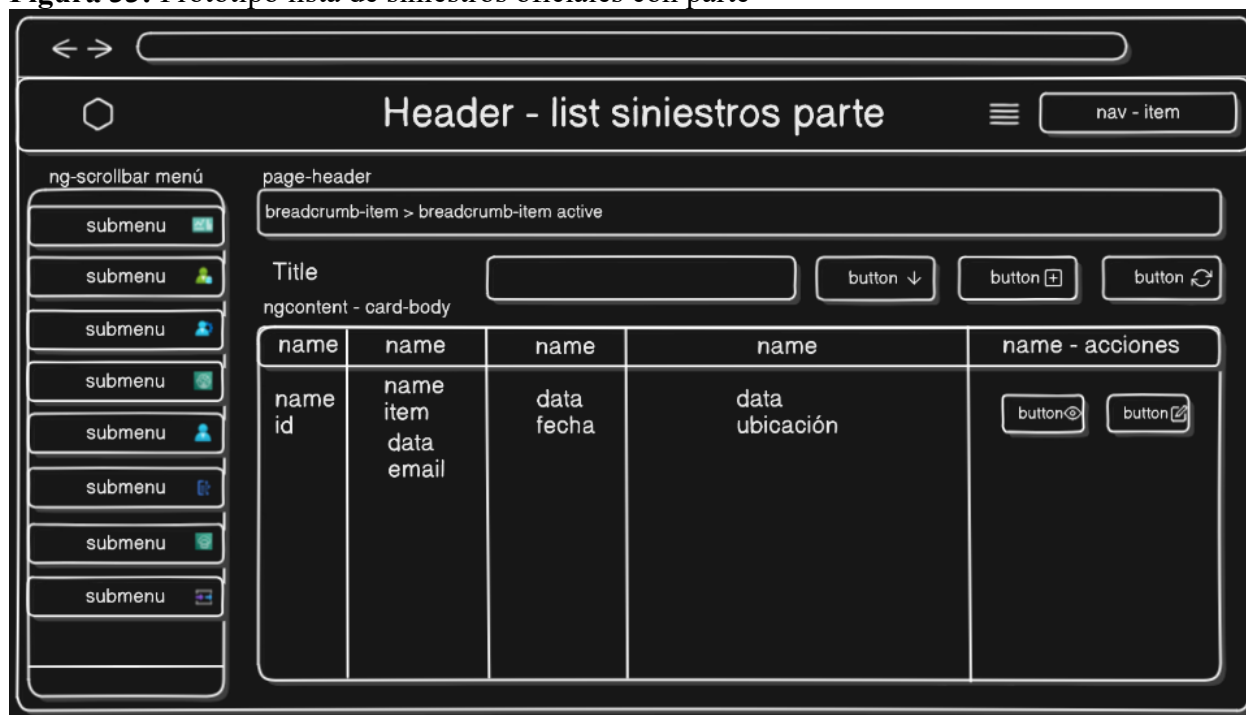
Nota: Elaboración Propia

Componentes:

- **Header:** "Header - add siniestro con parte".
- **Formulario por etapas:**
 - **Indicadores numerados:** 6 círculos numerados (1-6) como stepper.
 - **form-content:** Grid 2x3 de inputs.
 - Botones X (cancelar) y ✓ (guardar).
- **Layout:** Formulario multi-step con más etapas que el informativo.

3.2.19 Lista de siniestro oficial con parte

Figura 33: Prototipo lista de siniestros oficiales con parte



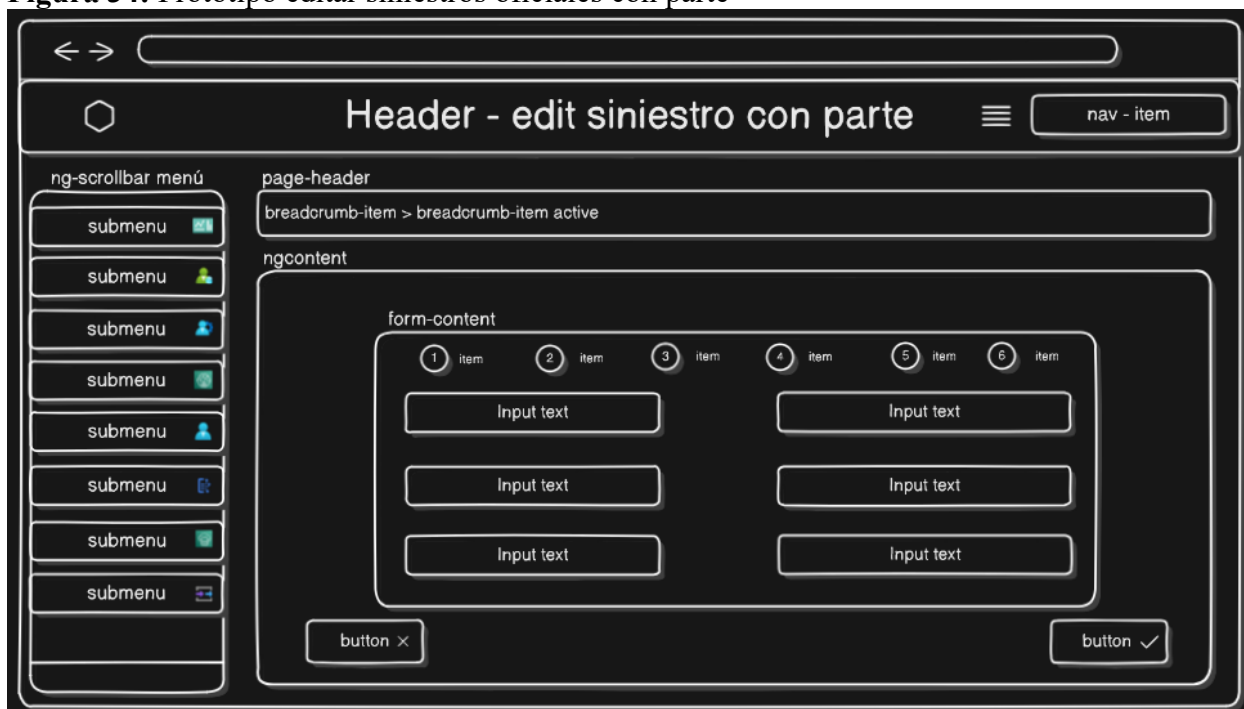
Nota: Elaboración Propia

Componentes:

- **Header:** "Header - list siniestros parte".
- **Barra de herramientas:** Input "Title" + botón descarga + botones "+" y "↺".
- **Tabla oficial:**
 - Headers: 5 columnas (4 "name" + "name-acciones").
 - Fila ejemplo: "name id", "name item data email", "data fecha", "data ubicación".
 - 2 botones de acción por fila.
- **Layout:** Similar a informativos, pero para siniestros oficiales.

3.2.20 Editar siniestro oficial con parte

Figura 34: Prototipo editar siniestros oficiales con parte



Nota: Elaboración Propia

Componentes:

- **Header:** "Header - edit siniestro con parte".
- **Formulario de edición por pasos:**
 - **Stepper numerado:** 6 círculos (1-6) con datos precargados.
 - **form-content:** Grid 2x3 con información existente.
 - Botones X (cancelar) y ✓ (actualizar).
- **Layout:** Formulario multi-step para edición de siniestros oficiales.

3.3 Análisis de código

3.3.1 Arquitectura del Sistema

Sistema de gestión de siniestros con arquitectura separada:

- Backend: Laravel (API REST).

- Frontend: Angular (SPA).
- Base de Datos: MySQL/PostgreSQL.
- IDE: Visual Studio Code.

3.3.2 Roles

- Admin/Supervisor: Gestión completa del sistema.
- Agentes: Gestión de reportes y siniestros.
- Staff: Roles intermedios.

3.3.3 Configuración del Entorno

3.3.3.1 Requisitos Previos

Herramientas necesarias.

- PHP >= 8.1 – Composer.
- Node.js >= 16 - npm o yarn.
- MySQL.
- Visual Studio Code.

Figura 35: Extensiones Recomendadas para VS Code

```

1
2  {
3  "recommendations": [
4    "bmewburn.vscode-intelephense-client",
5    "ms-vscode.vscode-typescript-next",
6    "angular.ng-template",
7    "bradlc.vscode-tailwindcss",
8    "ms-vscode.vscode-json",
9    "formulahendry.auto-rename-tag"
10 ]
11 }

```

Nota: autogenerado

Figura 36: Configuración del Workspace

```

1 |
2 {
3   "php.validate.executablePath": "/usr/bin/php",
4   "typescript.preferences.importModuleSpecifier": "relative",
5   "angular.experimental-ivy": true,
6   "emmet.includeLanguages": {
7     "typescript": "html"
8   }
9 }
10

```

Nota: Para el archivo .vscode/settings.json

3.3.4 Backend – Laravel

Figura 37: Estructura de directorios

```

backend/
├── app/
│   ├── Console/
│   ├── Exceptions/
│   ├── Http/
│   │   ├── Controllers/
│   │   │   ├── Admin/           # Controladores de administración
│   │   │   ├── Agente/         # Controladores de agentes
│   │   │   ├── Rol/            # Gestión de roles
│   │   │   ├── Staff/          # Controladores de staff
│   │   │   ├── Dashboard/      # Dashboard principal
│   │   │   ├── ParteSiniestro/  # Gestión de partes de siniestro
│   │   │   └── Siniestro/       # Gestión de siniestros
│   │   ├── Middleware/
│   │   └── Resources/
│   ├── Models/
│   │   ├── GrupoControl.php
│   │   ├── ParteSiniestro.php
│   │   ├── Siniestro.php
│   │   └── User.php
│   ├── Policies/
│   └── Providers/

```

Nota: estructura que debe seguir el backend para un correcto funcionamiento

3.3.4.1 Configuración Inicial

Tabla 43: Configuración Inicial

Instalación de dependencias	composer install
Configuración del entorno	cp .env.example .env php artisan key:generate
Migraciones y seeders	php artisan migrate php artisan db:seed
Servidor de desarrollo	php artisan serve

Nota: Comandos para empezar con el proyecto

3.3.4.2 Controladores Base

Figura 38: Referencia de controller

```

1  <?php
2  namespace App\Http\Controllers;
3  use App\Http\Controllers\Controller;
4  use Illuminate\Http\Request;
5  use App\Models\User;
6  use Illuminate\Support\Facades\Hash;
7  use Illuminate\Support\Facades\Auth;
8  class AuthController extends Controller
9  {
10 public function login(Request $request)
11     {
12         $credentials = $request->validate([
13             'email' => 'required|email',
14             'password' => 'required'
15         ]);
16         if (Auth::attempt($credentials)) {
17             $user = Auth::user();
18             $token = $user->createToken('authToken')->plainTextToken;
19             return response()->json([
20                 'success' => true,
21                 'user' => $user,
22                 'token' => $token,
23                 'roles' => $user->roles
24             ]);
25         }
26         return response()->json([
27             'success' => false,
28             'message' => 'Credenciales inválidas'
29         ], 401);
30     }
31 public function logout(Request $request)
32     {
33         $request->user()->currentAccessToken()->delete();
34         return response()->json([
35             'success' => true,
36             'message' => 'Sesión cerrada exitosamente'
37         ]);
38     }
39 }
```

Nota: Esta es la estructura básica de un AuthController.php

3.3.4.3 Modelos

Figura 39: Referencia de un model

```

1
2 <?php
3 namespace App\Models;
4 use Illuminate\Foundation\Auth\User as Authenticatable;
5 use Laravel\Sanctum\HasApiTokens;
6 use Illuminate\Database\Eloquent\Relations\BelongsToMany;
7 class User extends Authenticatable
8 {
9     use HasApiTokens;
10    protected $fillable = [
11        'name', 'email', 'password', 'telefono', 'direccion'
12    ];
13    protected $hidden = ['password', 'remember_token'];
14    public function roles(): BelongsToMany
15    {
16        return $this->belongsToMany(Rol::class, 'user_roles');
17    }
18    public function hasRole(string $role): bool
19    {
20        return $this->roles()->where('nombre', $role)->exists();
21    }
22    public function siniestrosComoAgente()
23    {
24        return $this->hasMany(Siniestro::class, 'agente_id');
25    }
26 }

```

Nota: Esta es la estructura básica de un User.php

3.3.5 Frontend – Angular

3.3.5.1 Estructura de Directorios

Figura 40: Estructura de frontend

```

frontend/src/app/
├── authentication/      # Módulo de autenticación
├── common-component/   # Componentes reutilizables
├── config/             # Configuraciones
├── core/               # Servicios centrales
├── error/              # Manejo de errores
├── siniestro/          # Módulo siniestro
├── shared/             # Módulo compartido
├── app-routing.module.ts # Rutas principales
├── app.component.ts    # Componente raíz
└── app.module.ts       # Módulo principal

```

Nota: Autogenerada

3.3.5.2 Configuración Angular

Figura 41: Estructura de app.module.ts

```

1  import { NgModule, LOCALE_ID } from '@angular/core';
2  import { BrowserModule } from '@angular/platform-browser';
3  import { AppRoutingModule } from './app-routing.module';
4  import { AppComponent } from './app.component';
5  import { MatSelectModule } from '@angular/material/select';
6  import { BrowserAnimationsModule } from '@angular/platform-browser/animations';
7  import { SharedModule } from './shared/shared.module';
8  import { HTTP_INTERCEPTORS } from '@angular/common/http';
9  import { PermissionInterceptorInterceptor } from './permission-interceptor.interceptor';
10 import { MatStepperModule } from '@angular/material/stepper';
11 import { ListboxModule } from 'primeng/listbox';
12 import { MatIconModule } from '@angular/material/icon';
13 import { registerLocaleData } from '@angular/common';
14 import localeEs from '@angular/common/locales/es-EC';
15 registerLocaleData(localeEs);
16
17 @NgModule({
18   declarations: [
19     AppComponent,
20   ],
21 },
22 imports: [
23   BrowserModule,
24   AppRoutingModule,
25   BrowserAnimationsModule,
26   SharedModule,
27   MatStepperModule,
28   ListboxModule,
29   MatSelectModule,
30   MatIconModule
31 ],
32 providers: [
33   {
34     provide: HTTP_INTERCEPTORS,
35     useClass: PermissionInterceptorInterceptor,
36     multi: true,
37   },
38   { provide: LOCALE_ID, useValue: 'es-EC' },
39 ],
40 bootstrap: [AppComponent]
41 })
42 export class AppModule { }

```

Nota: referencias de la configuración inicial de angular

3.3.5.3 Servicios

Figura 42: Referencia al archivo auth.service.ts

```

1  import { Injectable } from '@angular/core';
2  import { Router } from '@angular/router';
3  import { routes } from '../routes/routes';
4  import { URL_SERVICIOS } from 'src/app/config/config';
5  import { HttpClient, HttpHeaders, HttpResponse } from '@angular/common/http';
6  import { Observable, throwError } from 'rxjs';
7  import { tap, catchError, map } from 'rxjs/operators';
8  import { switchMap } from 'rxjs/operators';
9
10 @Injectable({
11   providedIn: 'root',
12 })
13 export class AuthService {
14   user: any;
15   token: any;
16
17   constructor(
18     private router: Router,
19     public http: HttpClient,
20   ) {
21     this.getLocalStorage();
22   }
23
24   // ✅ MÉTODO LOGIN CORREGIDO
25   login(email: string, password: string): Observable<any> {
26     const URL = URL_SERVICIOS + "auth/login";
27     return this.http.post(URL, { email, password }).pipe(
28       map((auth: any) => {
29         console.log('📄 Respuesta completa del login:', auth);
30
31         // ✅ VERIFICAR QUE LA RESPUESTA TENGA LOS DATOS NECESARIOS
32         if (auth && auth.access_token && auth.user) {
33           console.log('👤 Usuario recibido del backend:', auth.user);
34
35           // ✅ ASEGURAR QUE EL USUARIO TENGA ID
36           if (!auth.user.id && !auth.user.user_id) {
37             console.error('❌ USUARIO SIN ID RECIBIDO DEL BACKEND');
38             console.error('Datos del usuario:', auth.user);
39           }
40
41           const result = this.saveLocalStorage(auth);
42           console.log('✅ Resultado del guardado:', result);
43           return { success: true, data: auth };

```

Nota: Configuración básica para iniciar un proyecto de angular

3.3.6 Comandos para ejecución

Tabla 44: Comandos para ejecución

Para backend	php artisan serve
Para frontend	ng serve
Contruccion del dist en frontend	ng build --configuration production

Nota: autogenerada

3.4 Eficiencia del proceso

3.4.1 Plan de pruebas e implementación

Para las pruebas se solicitó la ayuda de los agentes de tránsito directamente al municipio de Loja y de esta forma poder trabajar con la UCOT, en el apartado de **Anexos** se encuentra el informe presentado y la respuesta obtenida por parte del municipio.

El plan de pruebas fue realizado con 20 usuarios conectados a la vez el proyecto esta actualmente en producción en un servidor multidominio, con certificados de seguridad ssl <http://karla.nous.ec/>.

3.4.2 Evaluación de usabilidad del sistema.

De acuerdo con page speed el aplicativo tiene un rendimiento del 84% y practicas recomendadas de un 100% las métricas lo muestran, eso es para la respuesta inicial por primera vez del sistema.

Figura 43: Métricas de evaluación de usabilidad

MÉTRICAS

■ First Contentful Paint

1,2 s

● Total Blocking Time

140 ms

■ Speed Index

1,4 s

■ Renderizado del mayor elemento con contenido

1,9 s

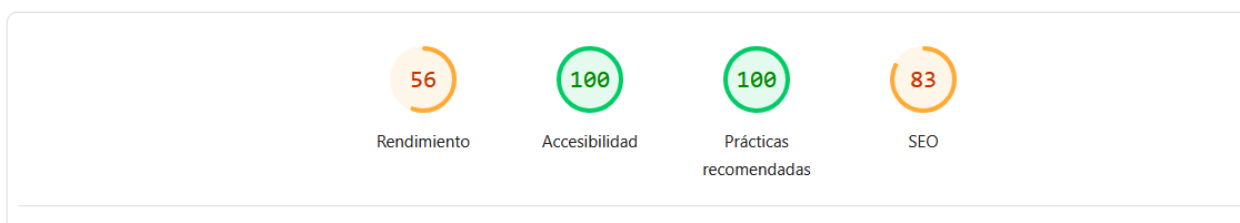
● Cambios de diseño acumulados

0.001

De acuerdo con la encuesta realizada a 30 usuarios diferentes se obtuvo los siguientes resultados en cuando a la usabilidad y su UI/UX.

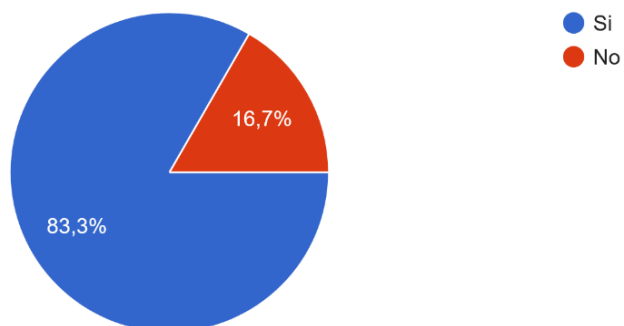
Figura 44: Métricas de rendimiento

🔍 Diagnostica problemas de rendimiento

**Figura 45:** Gráfica de resultados de encuesta realizada

Consideras que el aplicativo web es intuitivo y fácil de entender?

30 respuestas



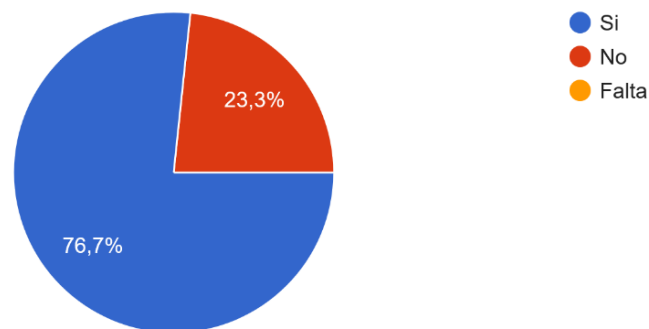
Nota: autogenerada

Con esta data se puede hablar de que el 83,3% de los usuarios que ingresaron les pareció intuitivo con los iconos y las navegaciones que tiene entre páginas, ya que el 76,7% respondió que entendieron la forma de navegar en el aplicativo web.

Figura 46: Prueba de navegación

Crees que la navegación entre paginas es la correcta ?

30 respuestas



Nota: autogenerada

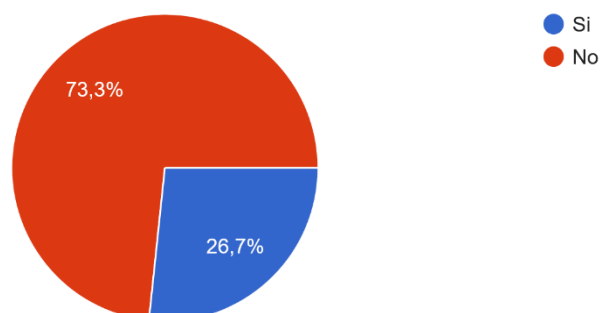
3.4.3 Evaluación de calidad de Respuesta

De acuerdo con las pruebas realizadas y la encuesta que 30 usuarios que usaron la aplicación respondiendo se pudo determinar que el 73,3% de las personas encuestadas tuvieron buena respuesta del servidor y se les cargó los datos rápidos sin presentar algún problema.

Figura 47: prueba de calidad de respuesta

Piensas que se demorar en cargar los datos que el sistema esta lento?

30 respuestas



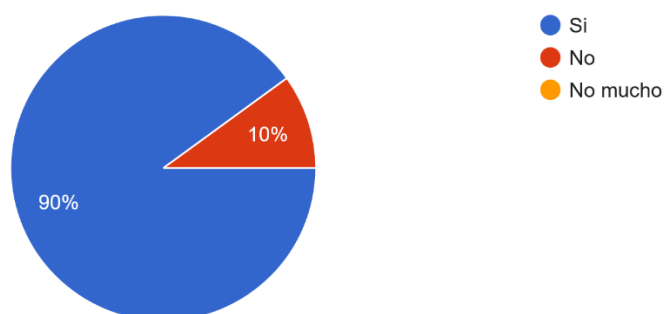
Nota: autogenerada

Además, la manipulación de datos y la carga de información en los formularios de acuerdo a las personas encuestadas el 90% de ellas no tuvieron problemas llenando los formularios cargados para los siniestros.

Figura 48: Prueba de carga de datos

Al momento de ingresar un siniestro, fue fácil de entender como manipular los datos e ingresarlos?

30 respuestas



Nota: autogenerada

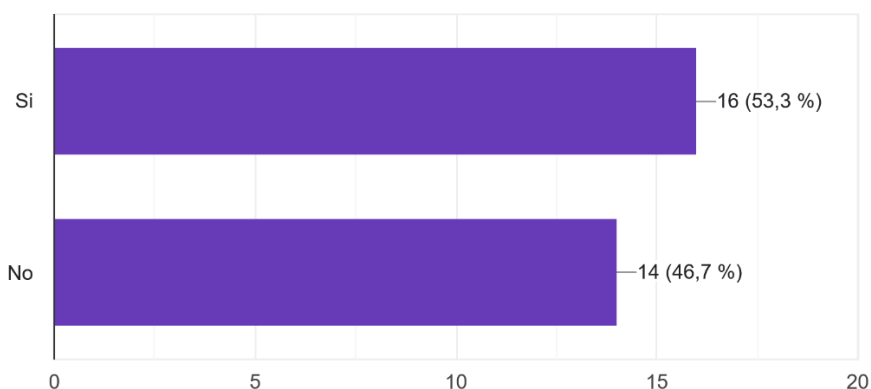
3.4.4 Evaluación de desempeño Técnico.

Para la evaluación técnica se tiene preguntas de seguridad, y aunque no se tiene un margen muy significativo en las respuestas esto puede deberse a que cuando se realizó la encuesta el servidor aún no tenía certificado SSL, un certificado de seguridad, pero a pesar de ellos 53,3% de las respuestas dicen que se tiene la seguridad adecuada en el aplicativo web.

Figura 49: Prueba técnica

Crees que la seguridad del aplicativo web es la correcta?

30 respuestas

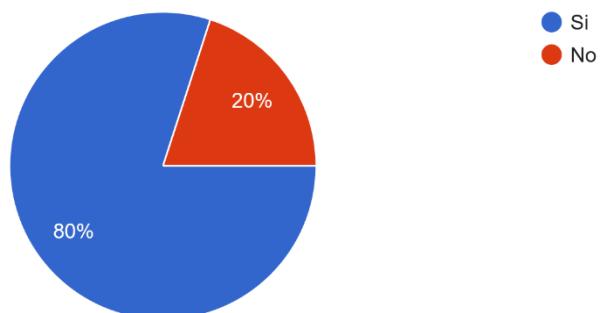


Nota: autogenerada

Para los siniestros oficiales la exportación del documento denominado parte es uno de los factores importantes por lo que a los usuarios se les realizó la pregunta de la facilidad al obtener el pdf, las respuestas fueron satisfactorias con el 80% de los encuestados dando una respuesta positiva.

Figura 50: Evaluación de exportación

Al ingresar un siniestro oficial fue difícil encontrar la forma de encontrar como exportar a pdf?
30 respuestas



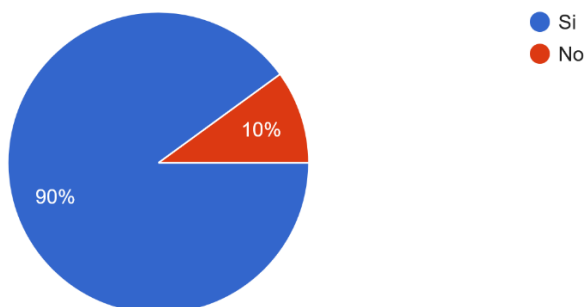
Nota: autogenerada

3.4.5 Evaluación de satisfacción general.

El desarrollo del aplicativo web fue para solucionar el problema que los agentes tenían al momento de hacer el levantamiento de información de un siniestro y el tiempo que les tomaba hacía de esto un proceso poco eficiente, con este antecedente se preguntó a los encuestados si creen que con esta solución tecnológica se soluciona el agente, y las respuestas fueron satisfactorias del 90% a favor de la solución.

Figura 51: Funcionalidad del sistema

La funcionalidad del sistema te parece la adecuada para solucionar un problema para los agentes?
30 respuestas



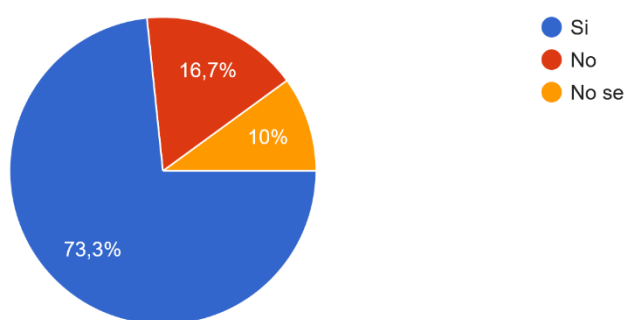
Nota: autogenerada

Otros de los puntos principales es la funcionalidad para los conductores la facilidad de ver en el mapa los siniestros que han ocurrido y tienen cierre de vía para de esta forma poder evitar un embotellamiento y aligerar el trabajo para los agentes de tránsito que además de revisar los siniestros deben dar paso y controlar el tráfico.

Figura 52: Usabilidad para los conductores

Crees que este aplicativo web va a ayudar a los conductores a despejar las calles y no ocasionar cuellos de botella?

30 respuestas



Nota: autogenerada

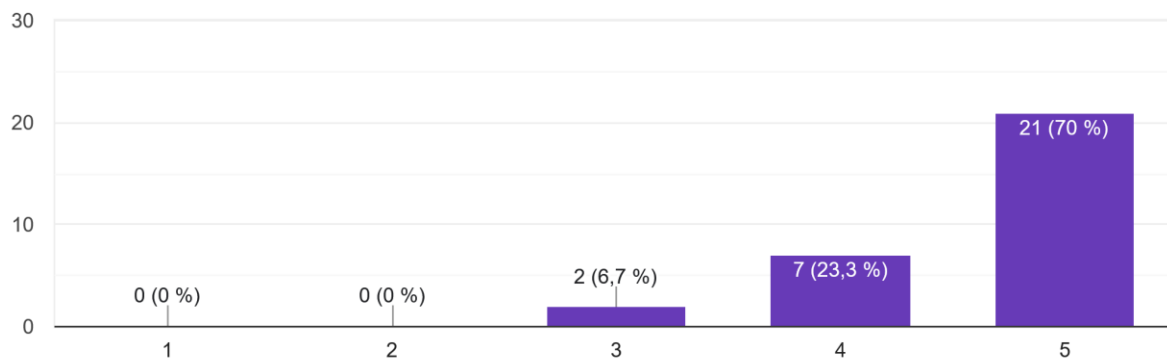
Para la prueba de aceptación de la aplicación se realizó una pregunta específica sobre la experiencia que tuvieron en el sistema, y las respuestas fueron bastante satisfactorias con el 70% de una experiencia excelente.

De acuerdo con estos datos obtenidos se puede concluir que la aplicación web ha cumplido con todos los obtuvo establecidos además de que tiene un UI/UX intuitivo y accesible dado con resultado una aceptación adecuada.

Figura 53: Prueba de aceptación

Del 1 al 5 como calificarías tu experiencia en el sistema siendo 1 pésimo y 5 excelente

30 respuestas



Nota: autogenerada

CAPÍTULO IV: CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones.

4.1.1 Conclusión general.

El desarrollo e implementación de un aplicativo web para la ucot con la gestión completa de siniestros se ha alcanzado satisfactoriamente, ya que los resultados obtenidos por las validaciones han demostrado que se ha optimizado en gran parte el tiempo y agilidad de generación de informes. Además, la integración del modelo predictivo permite identificar zonas de alto riesgo en la ciudad de Loja ayudando a mejorar la movilidad ciudadana.

4.1.2 Conclusiones según objetivos.

- El objetivo de análisis de la situación actual se cumplió realizando un proceso de investigación a los procesos que llevan a cabo los agentes de la UCOT, lo que permitió identificar las limitaciones y problemas que tenían al momento de hacer el levantamiento documentado de los siniestros de tránsito y a falta de información que se tiene para la ciudadanía lo que fue el punto de partida para el diseño de la arquitectura del sistema.
- Con los datos obtenidos gracias al primer objetivo específico se diseñó exitosamente la arquitectura de software y el modelo de inteligencia artificial para el análisis predictivo, de esta forma se integró de manera efectiva los componentes de alto riesgo, las notificaciones y las predicciones.
- Para el desarrollo e implementación de la aplicación web se siguió la arquitectura antes propuesta integrando de manera eficiente el manejo de los datos entre el backend desarrollado en laravel con el uso de APIs y el frontend con angular logrando una fusión estable sin fallas técnicas durante las pruebas.
- El cuarto objetivo se logró mediante la ejecución del plan de pruebas realizados con 20 agentes ingresados a la vez, ingresando siniestros simultáneamente. Confirma que la arquitectura de software es eficiente y no tiene problemas con el manejo de datos

facilitando su uso tanto para agentes como para conductores mediante el mapa de siniestros.

4.2 Recomendaciones.

- Se recomienda expandir la investigación sobre los modelos predictivos para integrar redes neuronales que permitan una mejor precisión al momento de identificar patrones o causas de siniestros, también se podría considerar el consumir datos climáticos de estaciones meteorológicas
- Para mejorar la usabilidad y el funcionamiento se deberían realizar auditorías o testeo completos de la aplicación con los nuevos datos ingresados y como esto ayudar al modelo para las predicciones, además recomendando establecer protocolos de actualización de librerías y una unidad de sistemas enfocada a revisar posibles inconvenientes además de capacitar adecuadamente a todas las personas que van a tener acceso al sistema.
- Se sugiere implementar registros de cambios para obtener en tiempo real quien está activo en ese momento en la aplicación esto ayudara a mejorar el control operativo y laboral de todos los agentes de tránsito para que logren adaptarse a las nuevas tecnologías.

Bibliografía.

- Adeva, R. (27 de Marzo de 2025). *AZ adls zone*. Obtenido de <https://www.adslzone.net/reportajes/software/que-es-android/>
- Android. (13 de Febrero de 2025). *Developer Android*. Obtenido de <https://developer.android.com/about/versions?hl=es-419>
- B.H.B. Honan, D. R. (2017). *Science Direct*. Obtenido de <https://www.sciencedirect.com/topics/computer-science/predictive-model>
- Bridge, K. (13 de Julio de 2023). *Learn Microsoft*. Obtenido de <https://learn.microsoft.com/es-es/windows/apps/performance/introduction>
- Calvo, L. (26 de noviembre de 2024). *GoDaddy*. Obtenido de <https://www.godaddy.com/resources/es/tecnologia/que-es-una-app-y-para-que-se-utiliza>
- Chauhan, N. S. (25 de Septiembre de 2023). *Data Source*. Obtenido de <https://www.datasource.ai/es/data-science-articles/metricas-de-evaluacion-de-modelos-en-el-aprendizaje-automatico>.
- Clientify. (25 de Junio de 2024). *Clientify*. Obtenido de <https://clientify.com/blog/marketing/recoleccion-de-datos-metodos-tecnicas-e-instrumentos>
- Ferrero, R. (Mayo de 2020). *Máxima Formación*. Obtenido de *Máxima Formación*: <https://www.maximaformacion.es/blog-dat/que-son-los-arboles-de-decision-y-para-que-sirven/>
- Gomez, A. (19 de Agosto de 2024). *Sei Blog*. Obtenido de <https://insights.sei.cmu.edu/blog/building-quality-software-4-engineering-centric-techniques/>
- IBM. (28 de febrero de 2021). Obtenido de <https://www.ibm.com/docs/es/iis/11.5.0?topic=types-unified-modeling-language-uml-model>
- IBM. (2024). *IBM*. Obtenido de <https://www.ibm.com/es-es/think/topics/data-security>
- INCIBE. (15 de Enero de 2020). *INCIBE* . Obtenido de <https://www.incibe.es/aprendeciberseguridad/vulnerabilidad>

- INEC. (Julio de 2022). INEC. Obtenido de https://www.ecuadorencifras.gob.ec/documentos/web-inec/Estadisticas_Economicas/Estadistica%20de%20Transporte/2021/2021_SINIESTRO_S_PPT.pdf
- INEC. (23 de noviembre de 2023). Obtenido de El nuevo Ecuador: <https://www.ecuadorencifras.gob.ec/mision-vision-valores/>
- Institute, S. E. (3 de Abril de 2025). Software Engineering Institute. Obtenido de <https://insights.sei.cmu.edu/software-architecture/>
- IONOS. (31 de Enero de 2023). Digital Guide. Obtenido de <https://www.ionos.com/es-us/digitalguide/servidores/know-how/modelo-cliente-servidor/>
- Loja, M. d. (2024). Loja gob. Obtenido de <https://www.loja.gob.ec/contenido/ucot>
- Mert Ozkaya, F. E. (Junio de 2020). Science Direct. Obtenido de https://www-sciencedirect-com.translate.goog/science/article/abs/pii/S259011842030023X?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=tc
- Niklas Donges, B. W. (26 de noviembre de 2024). Builtin. Obtenido de Builtin: <https://builtin.com/data-science/random-forest-algorithm>
- Oviedo, U. d. (s.f.). Unioviedo. Obtenido de https://www.unioviedo.es/compnum/laboratorios_py/new/kmeans.html
- Packard, H. (2025). Hpe. Obtenido de <https://www.hpe.com/lamerica/es/what-is/data-storage.html>
- Pedersen, N. (29 de Agosto de 2021). stibosystems. Obtenido de <https://www.stibosystems.com/es/blog/que-es-la-gestion-de-datos-de-aplicaciones>
- Rikke Friis Dam, T. Y. (2021). Interaction design . Obtenido de https://www-interaction-design-org.translate.goog/literature/article/design-thinking-get-started-with-prototyping?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=tc
- Sarra. (2024). Sarra broker de seguros . Obtenido de <https://sarra.com.ar/diccionario-del-seguro/diferencia-entre-accidente-y-siniestro/>
- Service, A. W. (2024). AWS. Obtenido de <https://aws.amazon.com/es/compare/the-difference-between-frontend-and-backend/>

- *Statista. (Julio de 2021). IBM. Obtenido de IBM: <https://www.ibm.com/mx-es/topics/push-notifications>*
- *Telecomunicaciones, M. (2021). El nuevo Ecuador. Obtenido de <https://www.telecomunicaciones.gob.ec/91-de-ecuatorianos-utiliza-las-redes-sociales-en-su-telefono-inteligente/>*
- *UNIR. (26 de Enero de 2024). Universidad Europea. Obtenido de <https://ecuador.unir.net/actualidad-unir/arquitectura-software/>*
- *Westreicher, G. (01 de mayo de 2020). Economipedia. Obtenido de <https://economipedia.com/definiciones/optimizacion.html>*
- *Abellán, E. (5 de Marzo de 2020). wearemarketing. Obtenido de [wearemarketing.com: https://www.wearemarketing.com/es/blog/metodologia-scrum-que-es-y-como-funciona.html](https://www.wearemarketing.com/es/blog/metodologia-scrum-que-es-y-como-funciona.html)*
- *Asana. (3 de septiembre de 2021). Asana. Obtenido de [Asana.com: https://asana.com/es/resources/scrum-master](https://asana.com/es/resources/scrum-master)*
- *Levy, O. (31 de Julio de 2021). Ittude.com.ar. Obtenido de ttude.com.ar/b/scrum/product-owner/#:~:text=El%20Product%20Owner%20es%20el,quiere%20el%20mercado%20y%20stakeholders.*
- *Instituto Nacional de Estadística y Censos (INEC). (2024). Siniestros de tránsito I trimestre, 2024. https://www.ecuadorencifras.gob.ec/documentos/web-inec/Estadisticas_Economicas/Estadistica%20de%20Transporte/2024/i_trimestre/2024_RESULTADOS_SINIESTROS_IT.pdf*
- *Municipio de Loja. (2023, septiembre). Los accidentes de tránsito no cesan en la ciudad. <https://www.loja.gob.ec/noticia/2023-09/los-accidentes-de-transito-no-cesan-en-la-ciudad>*
- *Diario La Hora. (2023). Loja tiene cifras elevadas en siniestros de tránsito. <https://www.lahora.com.ec/loja/loja-tiene-cifras-elevadas-en-siniestros-de-transito/>*
- *DeTránsito. (s.f.). Aplicación DeTránsito. <https://www.detransito.app/>*
- *Fasecolda. (2024). Conozca Marcus Conductores, la app para solucionar los choques simples en las vías del país. <https://www.fasecolda.com/sala-de-prensa/fasecolda-en->*

linea/noticias/noticias-2024/onzca-marcus-conductores-la-app-para-solucionar-los-choques-simples-en-las-vias-del-pais/

- *ATLAS.ti. (s.f.). ¿Qué es la investigación fenomenológica?*
<https://atlasti.com/es/guias/guia-investigacion-cualitativa-parte-2/investigacion-fenomenologica>
- *EBC. (2024). Método Hermenéutico: Comprende la Interpretación Textual.*
<https://www.ebc.mx/que-es-el-metodo-hermeneutico/>
- *Universidad de la Empresa (UDE). (2022). Visualizando la Metodología Proyectual.*
<https://ude.edu.uy/metodologia-proyectual/>
- *Atlassian. (s.f.). Scrumban: Mastering Two Agile Methodologies.*
<https://www.atlassian.com/agile/project-management/scrumban>
- *Asana. (2025). Scrumban: lo mejor de dos metodologías ágiles.*
<https://asana.com/es/resources/scrumban>

V: ANEXOS

Anexo1: Respuesta del municipio a la solicitud de pruebas

Trámite Nro:
EXT-SA-OS-48795-2025

Departamento Destino:
SECRETARIA DE ALCALDIA

Servicio Solicitado:
OTROS SERVICIOS

Datos del Solicitante:

- Cédula / RUC: **1105105835**
- Nombre / Razón Social: **CONDE PACCHA KARLA PAULINA**
- Correo Principal: **kacondempa@uide.edu.ec**
- Teléfono Principal: **0998567853**
- Correo Secundario:
- Teléfono Secundario:

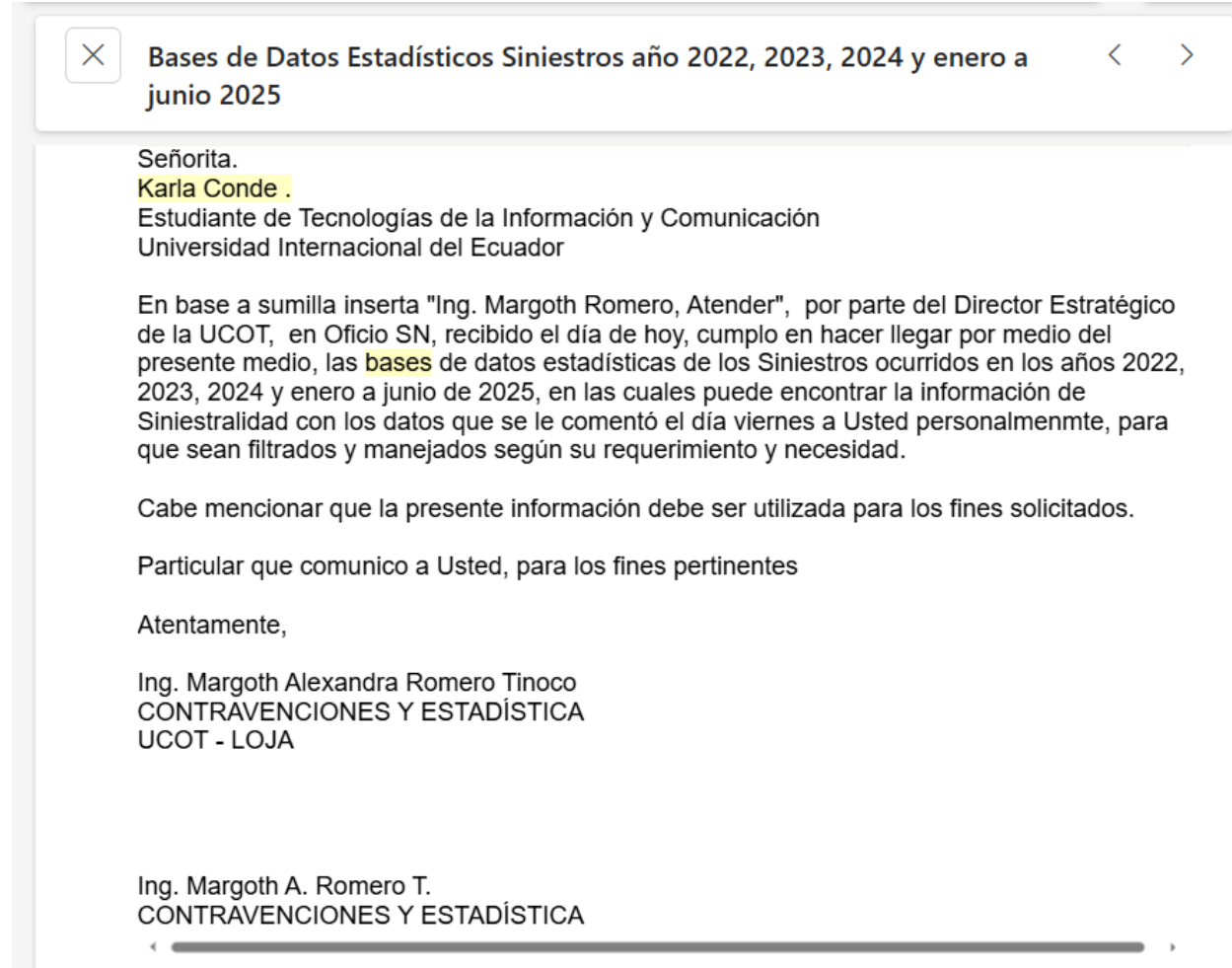
Trámite Ingresado Por:
GUARNIZO VALDIVIEZO ROCIO DEL CISNE

Fecha de Inicio del Trámite:
18/08/2025 18:05:12

Asunto de la Solicitud:
SOLICITA AUTORIZACIÓN PARA EL USO DEL SISTEMA INFORMÁTICO POR PARTE DE LA UCOT, EL MISMO QUE LO HA DESARROLLADO COMO PARTE DEL PROYECTO DE TITULACIÓN. SE REALICE LA VALIDACIÓN Y LAS PRUEBAS DE FUNCIONALIDAD CORRESPONDIENTES.

MUNICIPIO DE LOJA



Anexo 2: Correo de confirmación por parte de la UCOT para el del dataset sobre siniestros

Anexo 3: Preguntas de validación

Evaluación de sistema para la ucot

Escribe tu nombre y elige una talla de camiseta.

[Acceder a Google](#) para guardar el progreso. [Más información](#)

* Indica que la pregunta es obligatoria

Correo *

Tu respuesta

Nombre *

Tu respuesta

Consideras que el aplicativo web es intuitivo y fácil de entender?

☐ Si

☐ No

Crees que la navegación entre paginas es la correcta ?

☐ Si

☐ No

☐ Falta

Crees que la seguridad del aplicativo web es la correcta?

- ☐ Si
- ☐ No

Piensas que se demorar en cargar los datos que el sistema esta lento?

- ☐ Si
- ☐ No

Al momento de ingresar un siniestro, fue fácil de entender como manipular los datos e ingresarlos?

- ☐ Si
- ☐ No
- ☐ No mucho

Al ingresar un siniestro oficial fue difícil encontrar la forma de encontrar como exportar a pdf?

- ☐ Si
- ☐ No

La funcionalidad del sistema te parece la adecuada para solucionar un problema para los agentes?

- ☐ Si
- ☐ No

Crees que este aplicativo web va a ayudar a los conductores a despejar las calles y no ocasionar cuellos de botella?

- ☐ Si
- ☐ No
- ☐ No se

Del 1 al 5 como calificarías tu experiencia en el sistema siendo 1 pésimo y 5 excelente

- | | | | | |
|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| 1 | 2 | 3 | 4 | 5 |
| ☐ | ☐ | ☐ | ☐ | ☐ |

Otros comentarios

Tu respuesta

Enviar

Borrar formulario