



TESIS PROFESIONAL

CONFIGURACIÓN E INTERCONEXIÓN DE LA PLATAFORMA DE
OBSERVACIÓN DE ACCIDENTES VIALES.



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉJICO



VERACRUZ
GOBIERNO
DEL ESTADO



SEV
Secretaría
de Educación

SEMSys
Subsecretaría de Educación
Media Superior y Superior



DET
Departamento de Educación
Tecnológica del Estado
de Veracruz

**CONFIGURACIÓN E INTERCONEXIÓN DE LA
PLATAFORMA DE OBSERVACIÓN DE
ACCIDENTES VIALES.**

INSTITUTO TECNOLÓGICO SUPERIOR DE MARTÍNEZ
DE LA TORRE.

TESIS

**PRESENTADA COMO REQUISITO PARA
OBTENER EL TÍTULO DE:
LICENCIADO (A) EN INGENIERÍA EN SISTEMAS
COMPUTACIONALES**

Presenta:

CLAUDIA LIZETH SEGURA MARTÍNEZ

Asesores:

M.C.A SOFIA ISABEL FERNÁNDEZ GREGORIO

DR. LUIS ALBERTO MORALES ROSALES

Martínez de la Torre, Veracruz, Julio 2021

FICHA TÉCNICA

Estudiante

Apellido paterno

Apellido materno

Nombre(s)

Segura Martínez Claudia Lizeth

No. de control: 16010175

Carrera: Ingeniería en Sistemas Computacionales

Correo: 16010175@tecmartinez.edu.mx

Asesor(es) y/o colaboradores ITSMT

M.C.A Sofia Isabel Fernández Gregorio

Dr. Luis Alberto Morales Rosales

Datos del lugar donde se desarrolló el proyecto

Nombre o razón social

Universidad Michoacana de San Nicolas de Hidalgo

Dirección (calle, número, colonia, ciudad, código postal)

Av. General Francisco J. Múgica s/n Felicitas del Rio Morelia Michoacán

Asesor externo: Dr. Luis Alberto Morales Rosales

Departamento: Maestría de la Infraestructura del Transporte en la Rama de las Vías Terrestres.

Cargo: Catedrático Conacyt

Correo: lamorales@conacyt.mx

Teléfono y extensión: 276-109-26-99

FORMATO DE LIBERACIÓN DEL PROYECTO PARA TITULACIÓN INTEGRAL

	Liberación del Proyecto para Titulación Integral	Página 1 de 1
---	---	---------------

Martínez de la Torre, Ver. a 29 de Junio de 2021

C. DR FROYLAN ROSALES MARTÍNEZ
JEFE(A) DE DEPTO. SERVICIO SOCIAL Y
RESIDENCIAS PROFESIONALES
INSTITUTO TECNOLÓGICO SUPERIOR DE MARTÍNEZ DE LA TORRE
P R E S E N T E

Por este medio informo que ha sido liberado el siguiente proyecto para la Titulación Integral:

Nombre del estudiante:	Claudia Lizeth Segura Martinez		
Carrera:	Ingeniería en Sistemas Computacionales	No. De control:	16010175
Nombre Proyecto:	Configuración e Interconexión de la plataforma de observación de accidentes viales.		
Opción Titulación:	Tesis		

Agradezco su valioso apoyo en esta importante actividad para la formación profesional de nuestros egresados(as).

A T E N T A M E N T E


M.C. JULIO CÉSAR MACÍAS GÓMEZ
 JEFE DE DIVISIÓN DE CARRERA DE
**INGENIERÍA EN SISTEMAS
 COMPUTACIONALES**



**JEFATURA DE CARRERA
 ING. SISTEMAS COMPUTACIONALES**


**M.C.A. Sofía Isabel
 Fernández Gregorio**
ASESOR


**M.R.T. Víctor Guillermo
 Aspi Zetina**
REVISOR


**M.S.C. Angel Salas
 Martínez**
REVISOR

Ccp. Estudiante.
Ccp. Archivo.

F-11-07
Rev. 1

CARTA DE AUTORIZACIÓN DE ENTREGA DE TESIS EN SOPORTE DIGITAL

	Carta de Autorización de Entrega de Tesis en Soporte Digital	Página 1 de 1
---	--	---------------

No. de Oficio: DET/ITSMT/DA/ISC/052/2021

ASUNTO: Autorización de entrega

Martínez de la Torre, Ver., a 14 de Julio de 2021.

C. CLAUDIA LIZETH SEGURA MARTÍNEZ
No DE CONTROL 16010175
EGRESADO (A) DE LA CARRERA
INGENIERÍA EN SISTEMAS COMPUTACIONALES
P R E S E N T E

Por medio de la presente hago constar que ha cumplido satisfactoriamente con lo estipulado por el Lineamiento para la titulación Integral.

Por tal motivo se autoriza la entrega de la Tesis en soporte digital titulada:

Configuración e interconexión de la plataforma de observación de accidentes viales.

Déjense un plazo máximo de 30 días naturales a partir de la fecha de la expedición de la presente para realizar la solicitud del Acta de Recepción para la obtención del Título Profesional.

ATENTAMENTE



M.C.C. y T. Kella Elena Ocaña Drouaillet
Docente de Academia de Ingeniería
en Sistemas Computacionales

C.c.p. División de Estudios Profesionales
C.c.p. Archivo



Ing. Héctor Bautista Ruiz
Jefe de Carrera de Ingeniería
en Sistemas Computacionales



JEFATURA DE CARRERA
ING. SISTEMAS COMPUTACIONALES

F-11-09
Rev. 1

DEDICATORIA

Dedico esta tesis en primer lugar a mis padres, que día con día lucharon para que lograra mi sueño de estudiar una carrera universitaria y culminarla con su ayuda, por motivarme con palabras de aliento cuando sentía que no podía más.

A mi hermana, que siempre está para mí y que a pesar de su corta edad me demuestra día con día que hay que luchar por lo que se quiere conseguir en la vida.

A mis amigos y compañeros que me acompañaron en este proceso y me recordaban que podía con cada uno de los obstáculos establecidos, por creer en mí incluso más de lo que yo lo hacía.

A mis abuelitos que son fuente de amor, por estar tan orgullosos de mí y apoyarme.

Y a mis tres angelitos en el cielo, mi bisabuela Margarita que se cuánto me quería y que estaría muy orgullosa de mí. A mi tío Ernesto que siempre se dedicó a consentirme y quererme, enseñarme y apoyarme para ser de las mejores en la escuela, y a mi tío Fortino, que siempre me aconsejó para culminar mis metas académicas y que también estaría muy orgulloso de mí, pues siempre me decía que quería verme convertirme en una profesionista.

AGRADECIMIENTOS

Primero que nada, quiero darle las gracias a Dios por permitirme llegar hasta este punto en mi formación académica, por darme inteligencia, salud y perseverancia para culminar esta etapa de mi vida, y brindarme la oportunidad de vivir cada día.

A mis padres Claudio y Tere, pues sin ellos no hubiera llegado hasta donde me encuentro ahora, gracias a ellos por no dejarme caer, cuando pensaba que no podía dar más siempre estaban ahí para alentarme a seguir adelante y recordarme lo mucho que me he esforzado por lograr mis metas, por confiar en mí más que cualquier persona, gracias infinitas por todo, son pilar fundamental en mi vida, y sin ellos no sería ni la mitad de la persona que soy hoy en día.

A mi hermana Wendy, por ser mi motivación día con día, por alegrarme la vida con sus ocurrencias y confiar en mí.

Gracias también a mis asesores la M.C.A Sofia Isabel Fernández Gregorio y el Dr. Luis Alberto Morales Rosales por darme la oportunidad de desarrollar el proyecto con ellos, por compartirme sus conocimientos, por su motivación y atención prestada en el tiempo de realización del proyecto y por confiar en mí para lograr los objetivos planteados.

A mis amigos, quienes me apoyaron a lo largo de la carrera, que me brindaron su apoyo y confianza, y se alegraron por cada logro que obtuve durante mi periodo de aprendizaje.

Al Instituto Tecnológico Superior de Martínez de la Torre por abrirme sus puertas y permitirme formarme en ella, por las oportunidades que me brindaron en mi estancia en ella, agradezco por la ayuda brindada a mis maestros y personal administrativo, por confiar en mí y llenarme de su sabiduría y conocimientos.

RESUMEN

Los accidentes viales se han convertido en algo muy común día con día. Cada vez es más común encontrar un accidente en la carretera, ya sea de carros, motos, camionetas o cualquier tipo de vehículo terrestre. Recolectar datos de accidentes viales en una plataforma de observación vial, permite el análisis de los datos recolectados de este tipo de accidentes. La plataforma concentra la información considerando una aplicación web y una móvil, de las cuales los datos son registrados en una base de datos no estructurada. Por lo tanto, es importante tener una correcta configuración del servidor que aloja la base de datos no estructurada, y en donde se establece la comunicación de los datos. Se utilizó el SGBD Cassandra ya que, ya que puede gestionar una gran carga de datos a través de múltiples nodos, evitando fallos, en caso de que un nodo caiga o falle, otro nodo funcionaria en su lugar. Se instalo el sistema operativo Ubuntu 18.04 siendo un SO libre y fácil de usar, es seguro y no se necesitan muchos requisitos para ser instalado. Es por esto que se propone, para la plataforma configurar un servidor de aplicaciones y base de datos no estructurada. De esta forma se obtienen un servidor de aplicaciones que sienta la base de la estructura requerida para la plataforma de observación vial.

Palabras clave: *observación vial, análisis de datos, aplicación móvil, aplicación web, base de datos no estructurada, servidor.*

ABSTRACT

Have a road observation platform, which will allow the analysis of data collected from road accidents. The information of this type of data that is collected from an existing web and mobile application must be registered in a database. Therefore, it is important to have a correct configuration of a site with a previously defined unstructured database to establish data communication. This is why it is proposed, from the applications already developed, to configure a storage and access site from its own server; for which it will be necessary to establish the connection between these applications and the unstructured database.

Keywords: road observation, data analysis, mobile application, web application, unstructured database, server.

CONTENIDO

1. INTRODUCCIÓN	15
2. PLANTEAMIENTO DEL PROBLEMA	16
3. JUSTIFICACIÓN	17
4. OBJETIVOS.....	18
4.1 Objetivo general	18
4.2 Objetivos específicos.....	18
5. HIPÓTESIS	19
6. MARCO TEÓRICO	20
6.1 Marco Conceptual.....	20
Sistemas Gestores de Base de Datos	20
Firebase	20
Cassandra	21
MySQL.....	22
NoSQL	22
SQLite	23
Servidores.....	24
Clasificación de servidores.....	24
Modelo Cliente-Servidor.....	24
Sistemas operativos	25
GNU/Linux.....	25
Ubuntu.....	26
6.2 Marco Referencial.....	27
Drive Smart	27
RNAT (Registro Nacional de Accidentes de Tránsito).....	27
SIRAT (Sistema Integrado de Registro de Accidentes de Tránsito).....	27
7. MATERIALES Y MÉTODOS	29
8. DESARROLLO	31
8.1 Diseño de la arquitectura.	31
Ventajas de la arquitectura cliente/servidor	31

8.2 Configuración del servidor Ubuntu 18.04	34
Instalación de Apache2	34
Instalación de Nodejs	36
Instalación de npm	38
Instalación de cURL.....	39
Instalación de firebase.....	40
Instalación de Cassandra.....	41
Verificación del funcionamiento de Cassandra.....	42
8.3 Configuración de base de datos no estructurada.....	43
Creación de base de datos NoSQL “Accidentes Viales”	43
Creación de tabla Ambiente	43
Creación de tabla Físicas.....	45
Creación de tabla Condición Víctimas.....	46
Creación de tabla Detalle Conductores	46
Creación de tabla Detalles Vehículos.....	47
Creación de tabla Circunstancias	47
Creación de tabla Detalles Lesiones.....	48
9. RESULTADOS	49
10. CONCLUSIONES	54
11. RECOMENDACIONES.....	55
12. FUENTES DE INFORMACIÓN.....	56

LISTA DE FIGURAS

Figura 1. Características que incluye la plataforma Firebase. Fuente: NGuerrero (2020)	21
Figura 2. Logo del SGBD Cassandra. Fuente: Rodríguez N(2020)	22
Figura 3. Logo del SGBD MySQL. Fuente: Ramírez F(2018)	22
Figura 4. Logo del SGBD SQLite. Fuente: Kreibich (2010)	23
Figura 5. Descripción del Modelo Cliente-Servidor. Fuente. Schiaffarino A (2020)	25
Figura 6. Modelo Cliente-Servidor y como funciona. Fuente: Elaboración propia	32
Figura 7. Instalación de apache. Fuente: Elaboración propia	35
Figura 8. Instalación de Nodejs. Fuente: Elaboración propia	36
Figura 9. Finalización de instalación de Nodejs. Fuente: Elaboración propia	37
Figura 10. Instalación de npm. Fuente: Elaboración propia	38
Figura 11. Final de Instalación de npm. Fuente: Elaboración propia	38
Figura 12. Instalación de cURL. Fuente: Elaboración propia	39
Figura 13. Instalación de Firebase. Fuente: Elaboración propia	40
Figura 14. Instalación de Cassandra. Fuente: Elaboración propia	41
Figura 15. Funcionamiento de Cassandra. Fuente: Elaboración propia	42
Figura 16. Creación Base de Datos NoSQL. Fuente: Elaboración propia	43
Figura 17. Creación tabla ambiente. Fuente: Elaboración propia	44
Figura 18. Datos tabla ambiente. Fuente: elaboración propia	44
Figura 19. Tabla Físicas en la base de datos “Accidentes Viales” Fuente: Elaboración propia	45
Figura 22. Creación tabla Detalles Vehículos. Fuente: Elaboración propia	47
Figura 23. Creación tabla Circunstancias. Fuente: Elaboración propia	47
Figura 24. Creación tabla Detalles Lesiones. Elaboración: Fuente propia	48
Figura 25. Tablas existentes en la base de datos. Fuente: Elaboración propia	49
Figura 26. Datos tabla ambiente. Fuente: Elaboración propia	50

Figura 28. Datos Condición Víctimas. Fuente: Elaboración propia.....	50
Figura 29. Datos de tabla Detalles Conductores. Fuente: Elaboración propia.	51
Figura 30. Datos de tabla Detalles Lesiones. Fuente: Elaboración propia.....	51
Figura 31. Datos de tabla Detalles Vehículos. Fuente: Elaboración propia.	52
Figura 32. Datos de tabla Circunstancia. Fuente: Elaboración propia.	52

LISTA DE TABLAS

Tabla 1. Tecnologías instaladas en el servidor. Fuente: Elaboración propia. 30

1. INTRODUCCIÓN

El uso de un vehículo para desplazarse se ha convertido en algo necesario y una práctica habitual para millones de personas, lo realizan sin pensar en las causas que se pueden desencadenar. Esta práctica produce cada día un número muy elevado de lesiones.

Es por eso, que se pensó en la idea de realizar una plataforma de seguridad vial, la cual estará conectada a una aplicación móvil y una aplicación web, en las cuales se puede capturar su ubicación, llevar registro de los hechos ocurridos, almacenar fotografías del accidente y posteriormente hacer su respectivo registro en la base de datos no estructurada con toda la información de los implicados en los accidentes, como datos del conductor, vehículos, personas afectadas, ubicación, lesiones causadas y observaciones, para que los accidentes sean reportados de manera más eficaz y llevar un control de las causas de los accidentes y como se provocan.

El proyecto busca colaborar con soluciones para el análisis de las causas de los accidentes de tránsito, así como establecer medidas de seguridad para la prevención vial, y así poder reducir el número de accidentes, muertos y lesiones que se han presentado en los últimos años.

2. PLANTEAMIENTO DEL PROBLEMA

Día a día en las vialidades urbanas transitan diferentes tipos de vehículos. En la vialidad podemos encontrar situaciones diversas como: alcoholismo, drogadicción, exceso de velocidad, vialidad en mal estado, entre otras.

Actualmente los reportes de accidentes viales se registran llamando a las autoridades para reportar los detalles del accidente, lesiones o reclamos, también se intercambia información como la del seguro, contacto y vehículo con los demás afectados, es importante tomar fotos, y escribir los nombres y números de contacto de cualquier testigo (especialmente si la otra persona tuvo la culpa).

Es por eso que haciendo uso de un servidor que tenga almacenada una base de datos NoSQL que pueda obtener información de diferentes bases de datos para poder obtener datos en otros formatos que nos ayude a comparar sus tablas y alimentar las ya existentes con nueva información. Por ejemplo, saber cuáles son las principales causas y consecuencias de los accidentes viales, llenando en la base de datos las tablas correspondientes para obtener un informe completo, y realizando un registro en tiempo real, así como la información que se irá subiendo conforme se vayan registrando por medio de aplicación web o aplicación móvil.

3. JUSTIFICACIÓN

El presente proyecto pretende contribuir con la recolección de información de diversos factores de riesgos y tener una herramienta que nos permita evaluar y monitorear el avance en la solución de los problemas por causa de los accidentes de tránsito.

Por ese motivo, la configuración e interconexión de la plataforma de observación de accidentes viales es un gran paso para tener monitoreados las principales causas, tomando en cuenta cuales son los principales factores.

Por lo anterior mencionado, con este proyecto se busca mejorar las condiciones de movilidad de los vehículos y ayudar mediante herramientas tecnológicas reducir los índices de accidentalidad. A pesar de que existen bases de datos para reportes de accidentes la información de estas no es completa o solo admite un tipo de formato, la plataforma de observación vial está configurada para admitir cualquier formato, y poder comparar información de diferentes fuentes, eliminando el conjunto de datos duplicados y haciendo más completo el proceso de reporte.

4. OBJETIVOS

4.1 Objetivo general

Configurar un servidor de aplicaciones y bases de datos para almacenar información referente a accidentes viales con el fin de sentar las bases de la creación de una plataforma de observación vial.

4.2 Objetivos específicos

- Configurar y definir un sitio de almacenamiento para acceder a los datos de las aplicaciones.
- Configurar la base de datos no estructurada en el sitio de almacenamiento para conceder el acceso a las aplicaciones.
- Dar alojamiento a las aplicaciones web y móvil en el sitio de la plataforma para permitir su manipulación como administrador.

5. HIPÓTESIS

La configuración correcta de un servicio web y un gestor de base de datos no relacional permitirá que la plataforma de observación vial brinde la información sobre diversas causas que ocasionan accidentes viales en tiempo real.

6. MARCO TEÓRICO

A continuación, se muestran algunos conceptos de información científica que fueron investigados para tener conocimiento nuevo, que nos sirven para no cometer errores en nuestro proyecto desarrollado y comparar las diferentes tecnologías que se pueden usar, evitando usar las que no ayudarían al enfoque del proyecto. En el desarrollo del mismo, se utilizó: Sistema Gestor de Bases de Datos, Servidor, Arquitectura del Servidor, Sistema Operativo y también se analizaron aplicaciones ya existentes en el tema de accidentes viales.

6.1 Marco Conceptual

Sistemas Gestores de Base de Datos.

Firestore

Firestore es una plataforma de desarrollo propiedad de Google. Firestore está disponible para las diferentes plataformas como son: Web, Android y iOS, es utilizada en la nube lo que permite acceder a la información desde cualquier dispositivo. Tiene como objetivo crear aplicaciones de alta calidad, escalable para los usuarios.

Las principales herramientas que provee Firestore a los desarrolladores son bases de datos en tiempo real, Autenticación de Usuarios, Almacenamiento, Laboratorio de Test, Informes sobre fallos, Monitoreo de Rendimiento y Notificaciones. Una de sus principales ventajas es la seguridad (respaldo de Google), integración de infraestructura, sincronización de datos y medición y cuantificación de la información de la aplicación.

Firestore Realtime Database es una base de datos basada en la nube NoSQL que sincroniza datos entre todos los clientes en tiempo real y proporciona funcionalidad sin conexión. Los datos se almacenan en la base de datos en tiempo real como JSON, y todos los clientes conectados comparten una

instancia, y reciben automáticamente actualizaciones con los datos más recientes, Moroney L. (2017).

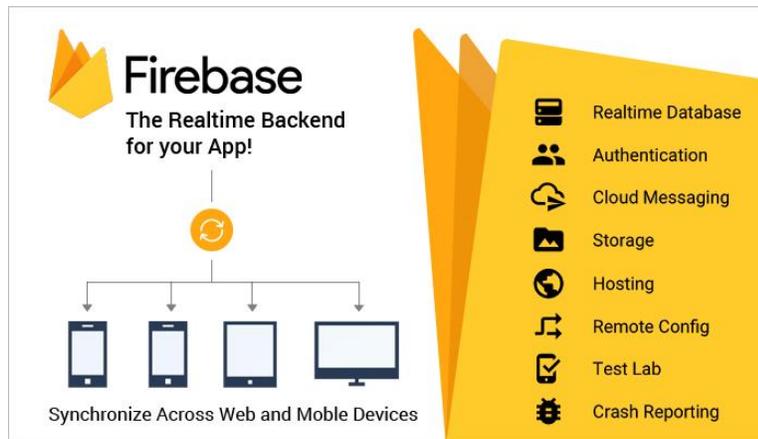


Figura 1. Características que incluye la plataforma Firebase. Fuente: NGuerrero (2020)

Cassandra

Cassandra es un sistema de almacenamiento distribuido para administrar grandes cantidades de datos estructurados distribuidos en muchos servidores básicos, al tiempo que brinda un servicio de alta disponibilidad sin un solo punto de falla. Cassandra tiene como objetivo ejecutarse sobre una infraestructura de cientos de nodos (posiblemente distribuidos en diferentes centros de datos). A esta escala, los componentes pequeños y grandes fallan continuamente. La forma en que Cassandra gestiona el estado persistente frente a estas fallas impulsa la confiabilidad y escalabilidad de los sistemas de software que dependen de este servicio. Si bien en muchos sentidos Cassandra se parece a una base de datos y comparte muchas estrategias de diseño e implementación con ella, Cassandra no admite un modelo de datos relacionales completo; en cambio, proporciona a los clientes un modelo de datos simple que admite el control dinámico sobre el diseño y el formato de los datos, (Lakshman, A. y Malik, P. (2010).



Figura 2. Logo del SGBD Cassandra. Fuente: Rodríguez N(2020)

MySQL

MySQL es un sistema gestor de bases de datos (SGBD, DBMS por sus siglas en inglés) muy conocido y ampliamente usado por su simplicidad y notable rendimiento.

MySQL está disponible para múltiples plataformas, la seleccionada para este proyecto es GNU/Linux. Sin embargo, las diferencias con cualquier otra plataforma son prácticamente nulas, ya que la herramienta utilizada en este caso es el cliente `mysql-client`, que permite interactuar con un servidor MySQL (local o remoto) en modo texto. De este modo es posible realizar todos los ejercicios sobre un servidor instalado localmente o, a través de Internet, sobre un servidor remoto, (Santillán, L. A. C., Ginestà, M. G., & Mora, Ó. P. (2014)



Figura 3. Logo del SGBD MySQL. Fuente: Ramírez F(2018)

NoSQL

Las bases de datos NoSQL son sistemas de almacenamiento de información que no cumplen con el esquema entidad-relación. Mientras que las

tradicionales bases de datos relacionales basan su funcionamiento en tablas, joins y transacciones. Las bases de datos NoSQL no imponen una estructura de datos en forma de tablas y relaciones entre ellas, sino que proveen un esquema mucho más flexible.

Las bases NoSQL son adecuadas para una escalabilidad realmente enorme, y tienden a utilizar modelos de consistencia relajados, no garantizando la consistencia de los datos, con el fin de lograr una mayor performance y disponibilidad. A esto se agrega el inconveniente de que no tienen un lenguaje de consulta declarativo, por lo que requiere de mayor programación para la manipulación de los datos, (June, 2013).

SQLite

SQLite es una base de datos de código abierto que está integrada en Android. SQLite admite estándar características de bases de datos relacionales como sintaxis SQL, transacciones y declaraciones preparadas. Además, se requiere solo poca memoria en tiempo de ejecución (aprox. 250 KByte).

SQLite admite los tipos de datos TEXT (similar a String en Java), INTEGER (similar a long en Java) y REAL (similar al doble en Java). Todos los demás tipos deben convertirse en uno de estos campos antes guardándolos en la base de datos. SQLite en sí no valida si los tipos escritos en las columnas son en realidad, del tipo definido, p. puede escribir un número entero en una columna de cadena y viceversa, (Vogel, L. (2010).



Figura 4. Logo del SGBD SQLite. Fuente: Kreibich (2010)

Servidores

Un servidor es un tipo de software que realiza ciertas tareas en nombre de los usuarios. El termino servidor también se utiliza para referirse al ordenador físico en el cual funciona ese software, cuyo propósito es proveer datos de modo que otras máquinas puedan utilizar esos datos.

Clasificación de servidores

- Servidor Multihebra. Considerando el entorno multithread (multihilo), cada thread (hilo, flujo de control del programa) representa un proceso individual ejecutándose en un sistema. A veces se les llama procesos ligeros o contextos de ejecución.
- Servidor Multiproceso. El multiproceso se refiere a dos programas que se ejecutan a la vez, bajo el control del Sistema Operativo. Los programas no necesitan tener relación unos con otros, simplemente el hecho de que el usuario desee que se ejecuten a la vez.

Modelo Cliente-Servidor

El modelo Cliente/Servidor es un modelo de aplicación distribuida en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. Las aplicaciones Clientes realizan peticiones a una o varias aplicaciones Servidores, que deben encontrarse en ejecución para atender dichas demandas.

El modelo Cliente/Servidor permite diversificar el trabajo que realiza cada aplicación, de forma que los Clientes no se sobrecarguen, cosa que ocurriría si ellos mismos desempeñan las funciones que le son proporcionadas de forma directa y transparente. En esta arquitectura la capacidad de proceso está repartida entre los

clientes y los servidores, aunque son más importantes las ventajas de tipo organizativo debidas a la centralización de la gestión de la información y la separación de responsabilidades, lo que facilita y clarifica el diseño del sistema. Tanto el Cliente como el Servidor son entidades abstractas que pueden residir en la misma máquina o en máquinas diferentes.

Una disposición muy común son los sistemas multicapa en los que el servidor se descompone en diferentes programas que pueden ser ejecutados por diferentes computadoras aumentando así el grado de distribución del sistema. Marini, E. (2012)

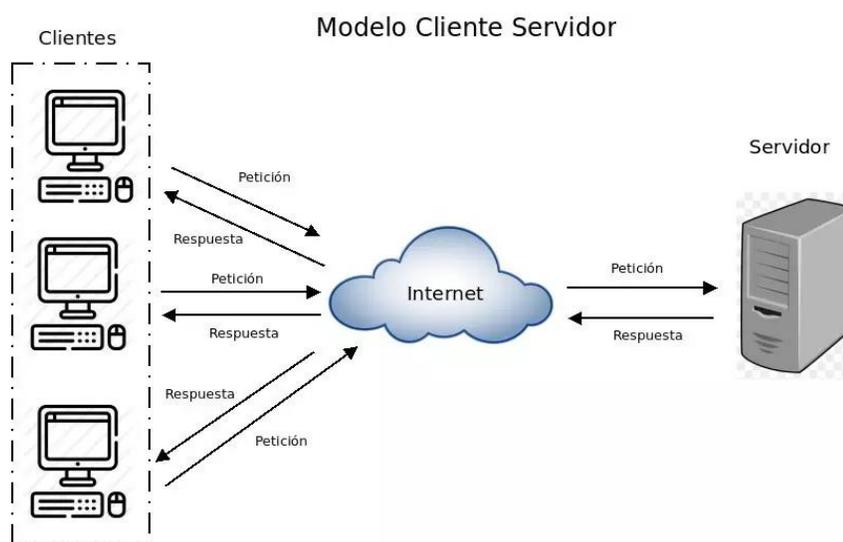


Figura 5. Descripción del Modelo Cliente-Servidor. Fuente. Schiaffarino A (2020)

Sistemas operativos

GNU/Linux

Actualmente, GNU/Linux es uno de los sistemas operativos más fiables y eficientes que podemos encontrar. Aunque su naturaleza de software libre creó inicialmente ciertas reticencias por parte de usuarios y empresas, GNU/Linux ha demostrado estar a la altura de cualquier otro sistema operativo existente.

GNU/Linux es un sistema operativo de la familia Unix, gratuito, creado mediante la política de “código abierto”. Estas características implican un gran ahorro en los costes de instalación de los equipos, pero también una mayor especialización por parte del personal informático.

En todo sistema Unix existe un usuario administrador (root), que controla el funcionamiento completo del sistema, tiene acceso universal y puede realizar cualquier operación con los datos y los dispositivos de la máquina, Viñas, R. B., & Llinàs, F. A. (2003).

Ubuntu

Una distribución GNU/Linux (también abreviado como “distro”) consiste en una recopilación de aplicaciones y herramientas junto al núcleo Linux. Se encuentran empaquetadas de una determinada manera y con utilidades extras para facilitar la configuración del sistema.

Su filosofía está basada en el Manifiesto Ubuntu, el cual promueve que el software debe estar disponible sin coste alguno y con la posibilidad de adaptarlo a las necesidades de cada usuario.

Actualmente Ubuntu soporta las arquitecturas: Intel x86 (IBM-compatible PC), AMD64 (Hammer) y PowerPC (Apple iBook y Powerbook, G4 y G5).

En la distribución se incluyen más de 1000 paquetes que van desde el núcleo Linux hasta GNOME 2.12, cubriendo todas las aplicaciones necesarias para el escritorio, acceso a Internet, programación y servicios. Cuaresma, S. B. Manual básico Ubuntu GNU/Linux

6.2 Marco Referencial

Drive Smart

Drive Smart es una app gratuita que ha sido diseñada para analizar y mejorar el comportamiento de los conductores al volante.

Acompaña al conductor para monitorizar, interpretar y detectar conductas de riesgo en la conducción. El sistema evalúa y analiza las maniobras al volante: desde la reacción ante señales de tráfico y la circulación en rotondas hasta la toma de curvas urbanas, la frenada o la velocidad, entre otras. A través de una serie de datos y variables registradas por la app, Drive Smart aporta al conductor información específica sobre su conducción, de forma que progresivamente le ayuda a mejorar su conducta al volante. Cárcel, J. I. L., Bertolín, A. M. B., Seguí, P. B., Osma, C. L., Robla, C. E., & Villarreal, R. P. (2016, June)

RNAT (Registro Nacional de Accidentes de Tránsito)

Aplicativo desarrollado sobre plataforma Web hace ya algunos años, permite alimentar una base de datos nacional sobre los accidentes registrados en cada organismo de tránsito, esta información es un poco ambiguo, puesto que no ayuda realmente al organismo a optar medidas para la prevención de los mismo. Ortiz Rivera, J. A., & Buelvas Dumar, W. J. (2018)

SIRAT (Sistema Integrado de Registro de Accidentes de Tránsito)

Aplicación móvil para el registro de accidentes de tránsito en el Municipio de Montelíbano, permite almacenar la información recopilada en una Base de Datos durante el registro de los Accidentes de tránsito ocurridos en el municipio. Cualquier ciudadano puede reportar accidentes usando su dispositivo móvil, activando su GPS puede capturar su ubicación, relatar los hechos de lo ocurrido y enviar fotografía del accidente.

Entre algunas investigaciones que orientan esta investigación se encuentra:
Adaptación de la Metodología Dream 3.0 para el análisis e Investigación de
Accidentes de Tránsito en la ciudad de Bogotá. Ortiz Rivera, J. A., & Buevas
Dumar, W. J. (2018).

7. MATERIALES Y MÉTODOS

Actividad 1. Recolectar información sobre servidores, arquitectura de servidores, bases de datos no estructuradas, y acerca del sistema operativo Linux.

Actividad 2. Se analizaron los diseños de arquitectura de servidores, eligiendo cliente-servidor.

Actividad 3. Se realizó un análisis para la configuración del servidor y se instalaron los servicios necesarios para su funcionamiento, como apache, http, node js, npm y cURL en Ubuntu 18.04

Actividad 4. Recabar información acerca de los Sistemas Gestores de Bases de Datos para realizar una base de datos no estructura, comparando ventajas y desventaja, se llegó a la conclusión de usar Cassandra, ya que es una base de datos distribuida en donde la información almacenada se reparte en todos los nodos que componen el clúster, además de proporcionar alta disponibilidad sin ningún punto único de fallo, comparando estas ventajas con firebase que usa JSON y casi no hay funciones SQL, por lo que no podría migrar fácilmente desde la base de datos existente se eligió Cassandra. Se realizó la instalación en el servidor de Cassandra por medio de comandos y se incorporó el uso de cURL, que nos ayuda a verificar la conexión a las URL y transferir datos.

Actividad 5. Se creó una base de datos NoSQL en Cassandra ya que son más eficientes en el procesamiento de datos y todos los cambios realizados en los datos serán replicados a todos los nodos del sistema con 7 tablas diferentes las cuales son: circunstancias, detalles lesiones, detalles conductores, ambiente, condición víctimas, detalles vehículos, y físicas para almacenar información acerca de los accidentes viales, los detalles que se tomaron en cuenta para su creación fueron el tipo de ambiente de la carretera, datos personales de los conductores y autoridades, el ambiente que provoca los accidentes viales, vehículos, circunstancias e información que se consideró relevante para llenar de información las tablas.

Sistema Operativo:	Ubuntu 18.04
Servidores Instalados:	Apache2 Nodejs npm cURL JavaScript
Sistema Gestor de Base de Datos:	Firebase Cassandra

Tabla 1. Tecnologías instaladas en el servidor. Fuente: Elaboración propia.

En esta tabla se muestran algunas tecnologías que se utilizaron para llevar a cabo el correcto funcionamiento del servidor (Ver Tabla 1).

8. DESARROLLO

8.1 Diseño de la arquitectura.

En este proyecto se utilizó la arquitectura cliente-servidor, algunas de sus características es que permite balancear la potencia de cálculo aplicada hacia el lado servidor o hacia el lado cliente, se usan para las páginas web, alojando allí todos los datos, elementos e información, seguridad de acceso al sistema, a nivel de objetos de datos, a nivel de datos, y protección de los almacenamientos físicos de los datos. Se usó un servidor de base de datos para poder proveer servicios de base de datos a otros programas o computadoras.

Ventajas de la arquitectura cliente/servidor

Las principales ventajas que ofrece son:

- **Administración centrada en el servidor.** Es decir que se tienen controlados a los usuarios y se minimiza al máximo los registros y accesos distintos, llenando las tablas con la información requerida en la base de datos.
- **Centralización de los recursos.** Los recursos comunes a todos los usuarios se administran en el servidor.
- **Mejora de la seguridad.** Esto quiere decir que evita la inconsistencia de información, o repetición de datos.
- **Escalabilidad de la instalación.** Se pueden añadir o suprimir clientes sin que el funcionamiento de la red se vea afectado.

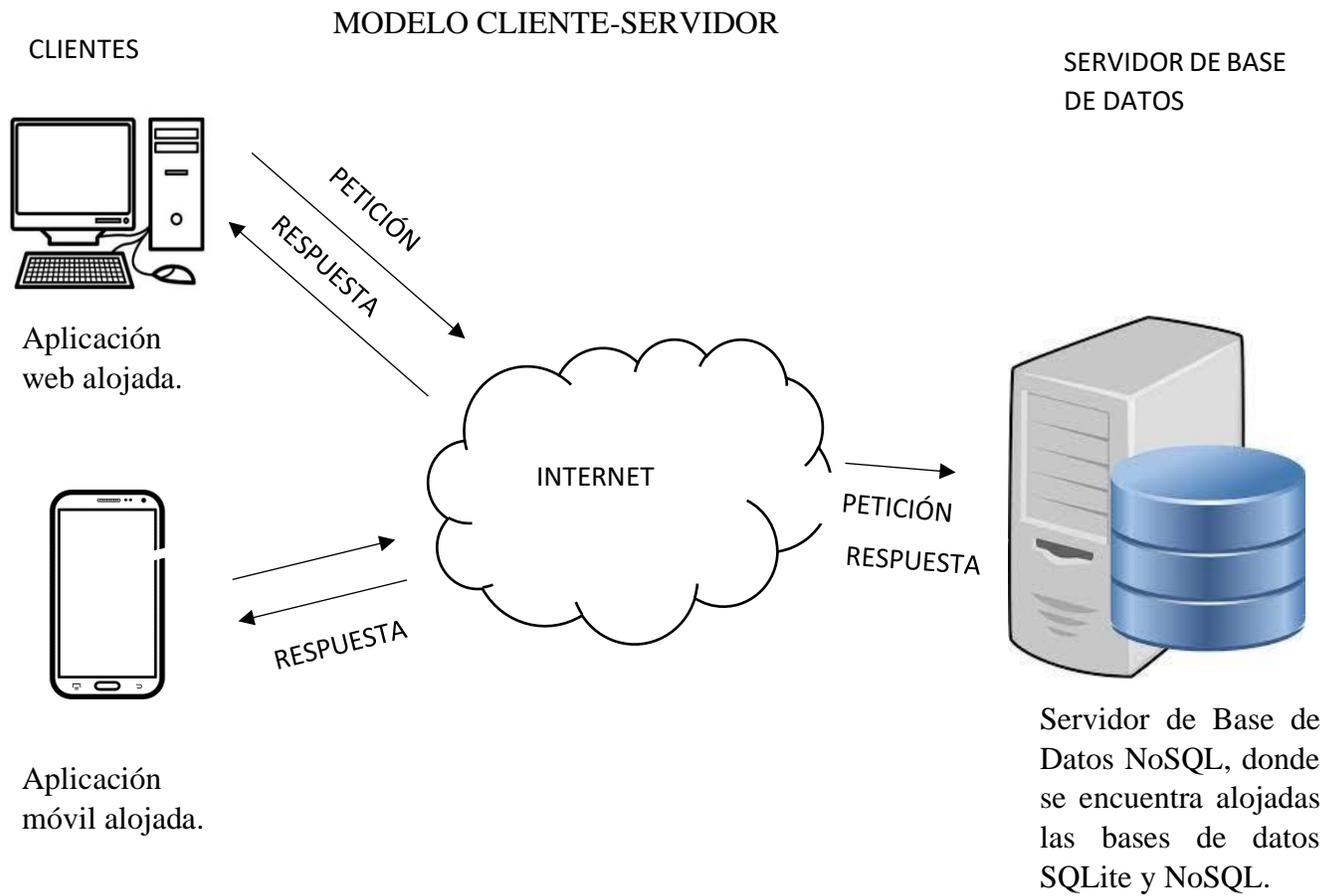


Figura 6. Modelo Cliente-Servidor y como funciona. Fuente: Elaboración propia

En la figura 6 se muestra el modelo de cliente servidor, en la cual se representa un servidor de aplicaciones en donde esta almacenadas las bases de datos no estructuradas, así como dos aplicaciones, una aplicación web que ayuda con la recolección de datos guardando imágenes y videos y una aplicación móvil que unas de sus principales funciones es guardar coordenadas, mostrar ubicación y demás herramientas que alimentan las bases de datos que se guardan en el servidor.

El servidor que se utiliza tiene servicios instalados como son: apache2, http, npm, node js, y cassandra el sistema gestor de base de datos no estructurada, en él se creó la base de datos “Accidentes Viales”, que es donde se compara la información con la de las demás aplicaciones.

8.2 Configuración del servidor Ubuntu 18.04

Para llevar a cabo la instalación de Ubuntu 18.04 se necesitan los siguientes requerimientos mínimos:

- 2GB de RAM.
- Procesador de doble núcleo (2Ghz).
- 25 GB de espacio libre en el disco duro.
- Medios de instalación (DVD o USB).
- Recomendada la conectividad a Internet si desea instalar software de terceros y actualizaciones durante la instalación.

Instalación de Apache2

Como primer paso se hizo la instalación de apache en el servidor de Ubuntu Server 18.04 como se muestra en la imagen 3.

Apache HTTP Server es un software de servidor web gratuito, que le permite a los propietarios de sitios web servir contenido en la web.

Por este motivo, se realizó la instalación de este servicio para que pueda funcionar el sitio web de la plataforma de observación de accidentes viales.

```
luis@dell-enc:
File Edit View Search Terminal Help
Selecting previously unselected package apache2.
(Reading database ... 313659 files and directories currently installed.)
Preparing to unpack ../apache2_2.4.29-1ubuntu4.14_amd64.deb ...
Unpacking apache2 (2.4.29-1ubuntu4.14) ...
Setting up apache2 (2.4.29-1ubuntu4.14) ...
Enabling module mpm_event.
Enabling module authz_core.
Enabling module authz_host.
Enabling module authn_core.
Enabling module auth_basic.
Enabling module access_compat.
Enabling module authn_file.
Enabling module authz_user.
Enabling module alias.
Enabling module dir.
Enabling module autoindex.
Enabling module env.
Enabling module mime.
Enabling module negotiation.
Enabling module setenvif.
Enabling module filter.
Enabling module deflate.
Progress: [ 67%] [#####.....]
```

Figura 7. Instalación de apache. Fuente: Elaboración propia.

Como se puede observar en la figura 7, la instalación de apache es un proceso fácil, pero requiere de atención para que se realice correctamente su instalación, pues si no es así el servicio no funcionara de manera adecuada.

Instalación de Nodejs

Nodejs es un entorno de tiempo de ejecución de JavaScript, utiliza un modelo de entrada y salida sin bloqueo controlado por eventos que lo hace ligero y eficiente. Puede leer o escribir archivos de cualquier tipo hasta hacer una solicitud HTTP.

A terminal window titled 'luis@dell-emc: ~' showing the command 'sudo apt install nodejs' and its output. The output includes package lists, dependency tree building, and a confirmation prompt 'Do you want to continue? [Y/n]'.

```
luis@dell-emc:~$ sudo apt install nodejs
[sudo] password for luis:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libc-ares2 libhttp-parser2.7.1 nodejs-doc
The following NEW packages will be installed:
  libc-ares2 libhttp-parser2.7.1 nodejs nodejs-doc
0 upgraded, 4 newly installed, 0 to remove and 32 not upgraded.
Need to get 5606 kB of archives.
After this operation, 24.7 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Figura 8. Instalación de Nodejs. Fuente: Elaboración propia.

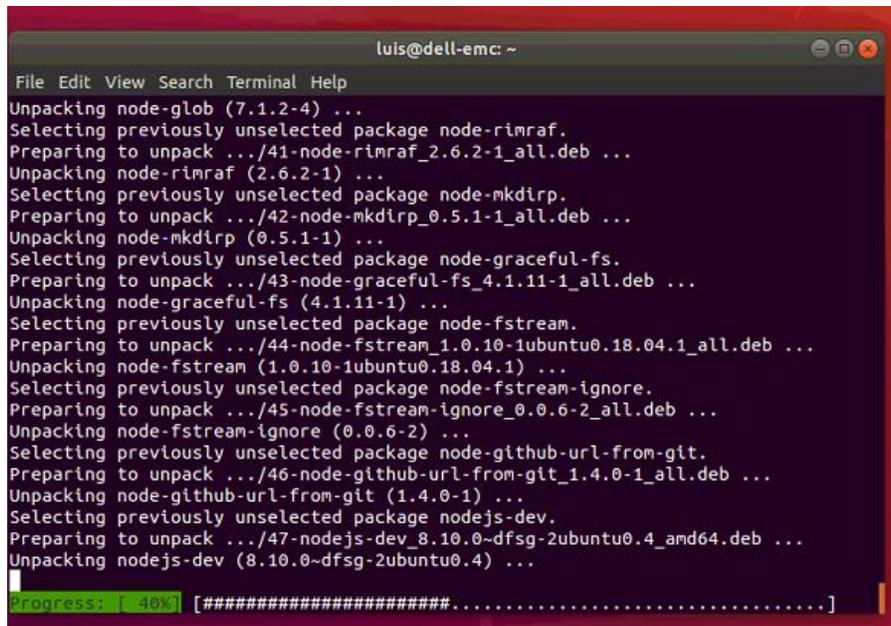
Su instalación es de manera fácil, se realiza a través del comando de ejecución **sudo apt install node js**.

```
luis@dell-emc: ~  
File Edit View Search Terminal Help  
Selecting previously unselected package nodejs-doc.  
(Reading database ... 254342 files and directories currently installed.)  
Preparing to unpack .../nodejs-doc_8.10.0-dfsg-2ubuntu0.4_all.deb ...  
Unpacking nodejs-doc (8.10.0-dfsg-2ubuntu0.4) ...  
Selecting previously unselected package libc-ares2:amd64.  
Preparing to unpack .../libc-ares2_1.14.0-1_amd64.deb ...  
Unpacking libc-ares2:amd64 (1.14.0-1) ...  
Selecting previously unselected package libhttp-parser2.7.1:amd64.  
Preparing to unpack .../libhttp-parser2.7.1_2.7.1-2_amd64.deb ...  
Unpacking libhttp-parser2.7.1:amd64 (2.7.1-2) ...  
Selecting previously unselected package nodejs.  
Preparing to unpack .../nodejs_8.10.0-dfsg-2ubuntu0.4_amd64.deb ...  
Unpacking nodejs (8.10.0-dfsg-2ubuntu0.4) ...  
Setting up nodejs-doc (8.10.0-dfsg-2ubuntu0.4) ...  
Setting up libhttp-parser2.7.1:amd64 (2.7.1-2) ...  
Setting up libc-ares2:amd64 (1.14.0-1) ...  
Setting up nodejs (8.10.0-dfsg-2ubuntu0.4) ...  
update-alternatives: using /usr/bin/nodejs to provide /usr/bin/js (js) in auto mode  
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...  
Processing triggers for libc-bin (2.27-3ubuntu1.2) ...  
luis@dell-emc:~$
```

Figura 9. Finalización de instalación de Nodejs. Fuente: Elaboración propia.

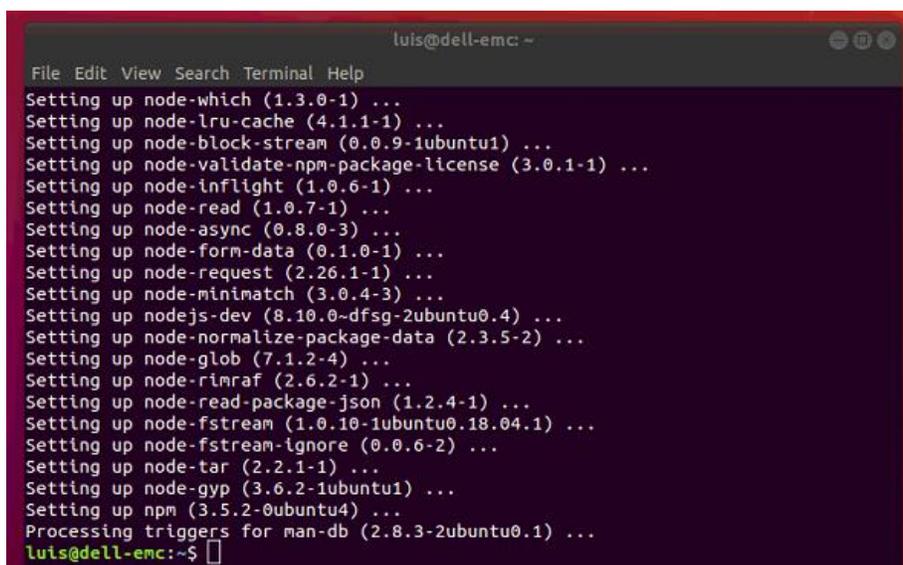
Instalación de npm

En las imágenes 10 y 11 se muestra cómo se llevó a cabo la instalación de npm, este servicio es necesario para desarrollar el proyecto de manera correcta ya que es gestor de paquetes que ayuda a obtener cualquier librería y permite agregar dependencias de forma simple.



```
luis@dell-enc: ~
File Edit View Search Terminal Help
Unpacking node-glob (7.1.2-4) ...
Selecting previously unselected package node-rimraf.
Preparing to unpack ../41-node-rimraf_2.6.2-1_all.deb ...
Unpacking node-rimraf (2.6.2-1) ...
Selecting previously unselected package node-mkdirp.
Preparing to unpack ../42-node-mkdirp_0.5.1-1_all.deb ...
Unpacking node-mkdirp (0.5.1-1) ...
Selecting previously unselected package node-graceful-fs.
Preparing to unpack ../43-node-graceful-fs_4.1.11-1_all.deb ...
Unpacking node-graceful-fs (4.1.11-1) ...
Selecting previously unselected package node-fstream.
Preparing to unpack ../44-node-fstream_1.0.10-1ubuntu0.18.04.1_all.deb ...
Unpacking node-fstream (1.0.10-1ubuntu0.18.04.1) ...
Selecting previously unselected package node-fstream-ignore.
Preparing to unpack ../45-node-fstream-ignore_0.0.6-2_all.deb ...
Unpacking node-fstream-ignore (0.0.6-2) ...
Selecting previously unselected package node-github-url-from-git.
Preparing to unpack ../46-node-github-url-from-git_1.4.0-1_all.deb ...
Unpacking node-github-url-from-git (1.4.0-1) ...
Selecting previously unselected package nodejs-dev.
Preparing to unpack ../47-nodejs-dev_8.10.0-dfsg-2ubuntu0.4_amd64.deb ...
Unpacking nodejs-dev (8.10.0-dfsg-2ubuntu0.4) ...
Progress: [ 40%] [#####.....]
```

Figura 10. Instalación de npm. Fuente: Elaboración propia.

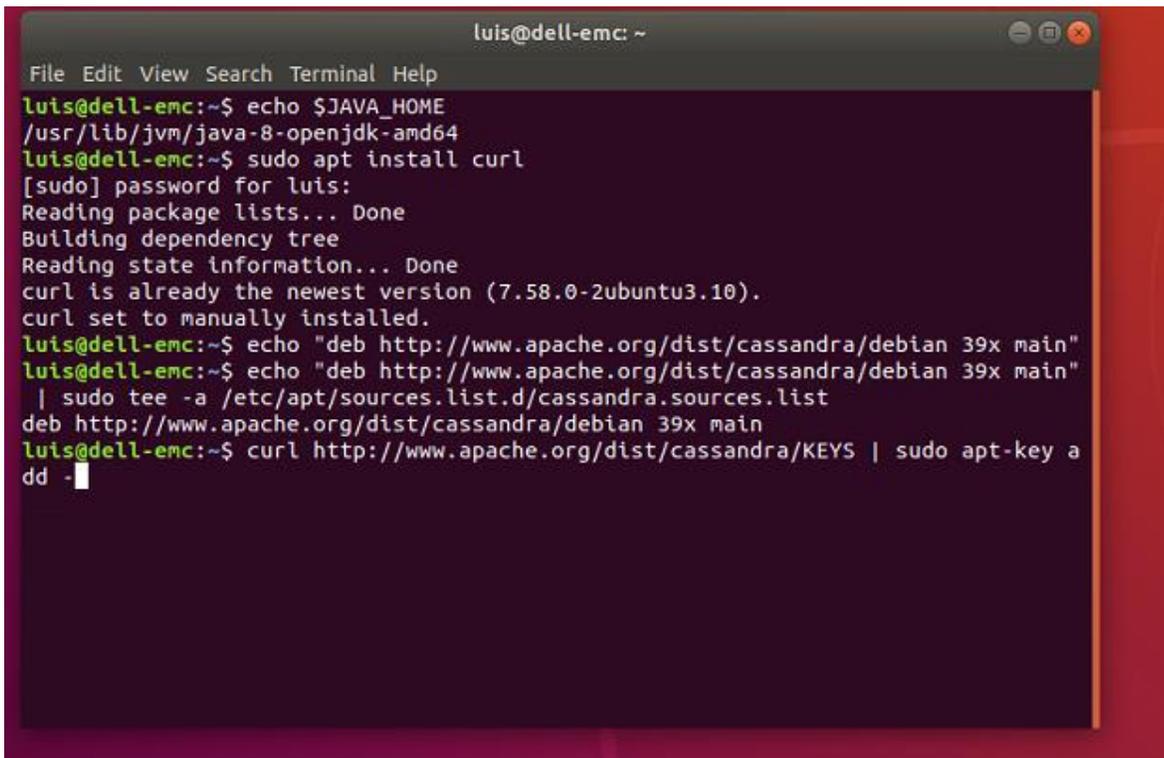


```
luis@dell-enc: ~
File Edit View Search Terminal Help
Setting up node-which (1.3.0-1) ...
Setting up node-lru-cache (4.1.1-1) ...
Setting up node-block-stream (0.0.9-1ubuntu1) ...
Setting up node-validate-npm-package-license (3.0.1-1) ...
Setting up node-inflight (1.0.6-1) ...
Setting up node-read (1.0.7-1) ...
Setting up node-async (0.8.0-3) ...
Setting up node-form-data (0.1.0-1) ...
Setting up node-request (2.26.1-1) ...
Setting up node-minimatch (3.0.4-3) ...
Setting up nodejs-dev (8.10.0-dfsg-2ubuntu0.4) ...
Setting up node-normalize-package-data (2.3.5-2) ...
Setting up node-glob (7.1.2-4) ...
Setting up node-rimraf (2.6.2-1) ...
Setting up node-read-package-json (1.2.4-1) ...
Setting up node-fstream (1.0.10-1ubuntu0.18.04.1) ...
Setting up node-fstream-ignore (0.0.6-2) ...
Setting up node-tar (2.2.1-1) ...
Setting up node-gyp (3.6.2-1ubuntu1) ...
Setting up npm (3.5.2-0ubuntu4) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
luis@dell-enc:~$
```

Figura 11. Final de Instalación de npm. Fuente: Elaboración propia.

Instalación de cURL

Es una librería de funciones para conectar con servidores y trabajar con ellos. Se realiza en formato URL, sirve para usar archivos que hay en URLs de Internet, soportando los protocolos HTTP, FTP HTTPS, etc. Su instalación fue necesaria porque el SGBD no funcionaba sin él, puesto que no podía conectarse los protocolos de internet como se menciona al principio.



```
luis@dell-emc: ~  
File Edit View Search Terminal Help  
luis@dell-emc:~$ echo $JAVA_HOME  
/usr/lib/jvm/java-8-openjdk-amd64  
luis@dell-emc:~$ sudo apt install curl  
[sudo] password for luis:  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
curl is already the newest version (7.58.0-2ubuntu3.10).  
curl set to manually installed.  
luis@dell-emc:~$ echo "deb http://www.apache.org/dist/cassandra/debian 39x main"  
luis@dell-emc:~$ echo "deb http://www.apache.org/dist/cassandra/debian 39x main"  
| sudo tee -a /etc/apt/sources.list.d/cassandra.sources.list  
deb http://www.apache.org/dist/cassandra/debian 39x main  
luis@dell-emc:~$ curl http://www.apache.org/dist/cassandra/KEYS | sudo apt-key a  
dd -
```

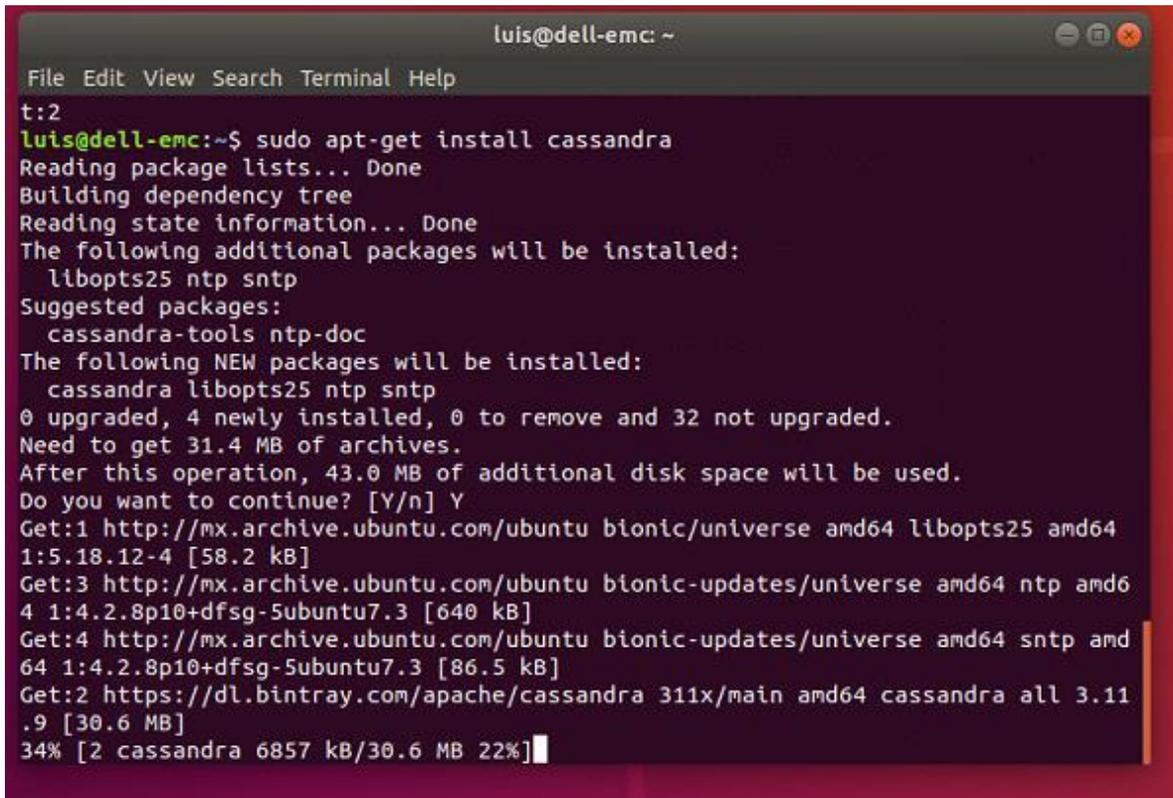
Figura 12. Instalación de cURL. Fuente: Elaboración propia.

En la figura 12 se describe la instalación de cURL, un proceso de fácil instalación ocupando los comandos específicos para poder realizarla, es importante su instalación para el desarrollo de este proyecto ya que contiene las funciones para obtener conexión con los servidores.

Instalación de Cassandra

Cassandra es un software NoSQL distribuido de código abierto para bases de datos muy grandes, en ella se realizó la base de datos no estructurada.

Se eligió este SGBD porque cuenta con un enfoque redundante, lo que reduce la probabilidad de fallo. En la imagen 14 se puede observar el proceso de instalación que se realizó, haciendo uso de comando en la terminal del servidor, así como sus dependencias agregadas para su correcto funcionamiento.

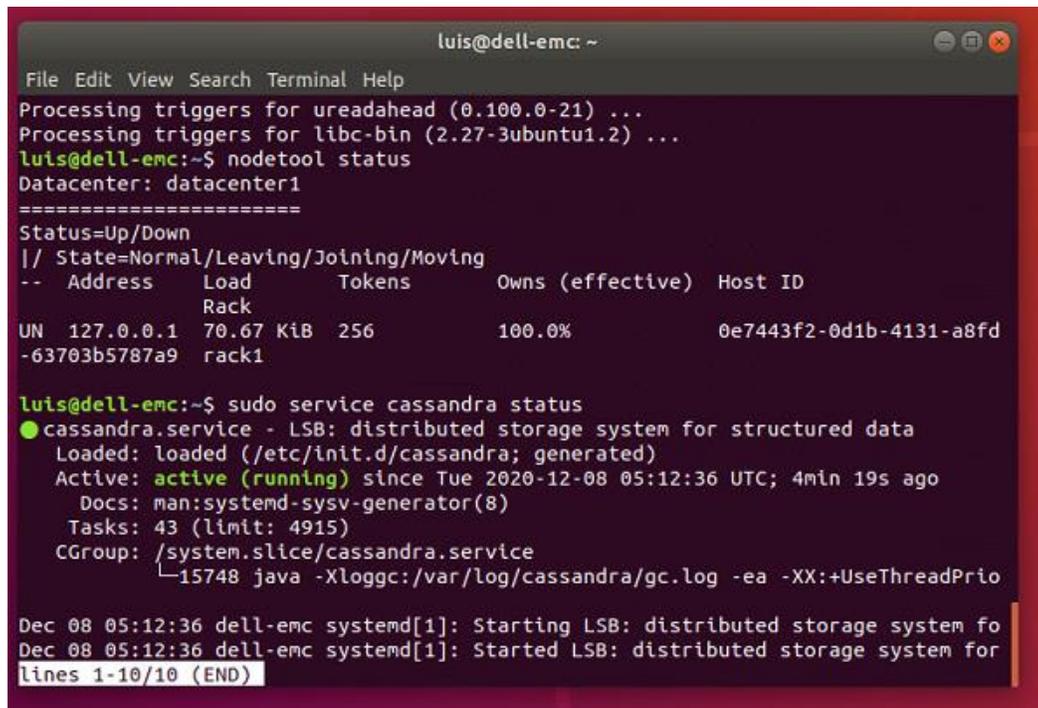


```
luis@dell-emc: ~  
File Edit View Search Terminal Help  
t:2  
luis@dell-emc:~$ sudo apt-get install cassandra  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following additional packages will be installed:  
  libopts25 ntp sntp  
Suggested packages:  
  cassandra-tools ntp-doc  
The following NEW packages will be installed:  
  cassandra libopts25 ntp sntp  
0 upgraded, 4 newly installed, 0 to remove and 32 not upgraded.  
Need to get 31.4 MB of archives.  
After this operation, 43.0 MB of additional disk space will be used.  
Do you want to continue? [Y/n] Y  
Get:1 http://mx.archive.ubuntu.com/ubuntu bionic/universe amd64 libopts25 amd64  
1:5.18.12-4 [58.2 kB]  
Get:3 http://mx.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 ntp amd64  
1:4.2.8p10+dfsg-5ubuntu7.3 [640 kB]  
Get:4 http://mx.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 sntp amd64  
1:4.2.8p10+dfsg-5ubuntu7.3 [86.5 kB]  
Get:2 https://dl.bintray.com/apache/cassandra 311x/main amd64 cassandra all 3.11  
.9 [30.6 MB]  
34% [2 cassandra 6857 kB/30.6 MB 22%]
```

Figura 14. Instalación de Cassandra. Fuente: Elaboración propia.

Verificación del funcionamiento de Cassandra

Como se muestra en la Figura 15, Cassandra fue instalado, es importante verificar que su instalación haya sido correctamente ya que, en caso de tener una equivocación, al momento de ingresar el comando para realizar una base de datos mandará error, una vez completado este procedimiento se procede a realizar la Base de Datos.



```
luis@dell-emc: ~
File Edit View Search Terminal Help
Processing triggers for ureadahead (0.100.0-21) ...
Processing triggers for libc-bin (2.27-3ubuntu1.2) ...
luis@dell-emc:~$ nodetool status
Datacenter: datacenter1
=====
Status=Up/Down
// State=Normal/Leaving/Joining/Moving
-- Address      Load          Tokens         Owns (effective)  Host ID
   Rack
UN 127.0.0.1    70.67 KiB     256            100.0%            0e7443f2-0d1b-4131-a8fd
-63703b5787a9 rack1

luis@dell-emc:~$ sudo service cassandra status
● cassandra.service - LSB: distributed storage system for structured data
   Loaded: loaded (/etc/init.d/cassandra; generated)
   Active: active (running) since Tue 2020-12-08 05:12:36 UTC; 4min 19s ago
     Docs: man:systemd-sysv-generator(8)
    Tasks: 43 (limit: 4915)
   CGroup: /system.slice/cassandra.service
           └─15748 java -Xloggc:/var/log/cassandra/gc.log -ea -XX:+UseThreadPrio

Dec 08 05:12:36 dell-emc systemd[1]: Starting LSB: distributed storage system fo
Dec 08 05:12:36 dell-emc systemd[1]: Started LSB: distributed storage system for
lines 1-10/10 (END)
```

Figura 15. Funcionamiento de Cassandra. Fuente: Elaboración propia.


```
luis@dell-emc: ~
File Edit View Search Terminal Help
cqlsh:accidentesviales> CREATE TABLE ambiente (condicion varchar, geometria varchar, iluminacionVia varchar, zonasObras varchar, PRIMARY KEY (condicion));
cqlsh:accidentesviales> SELECT * FROM ambiente;

condicion | geometria | iluminacionvia | zonasobras
-----+-----+-----+-----
(0 rows)
cqlsh:accidentesviales>
```

Figura 17. Creación tabla ambiente. Fuente: Elaboración propia.

Posterior a ello, se agregaron datos a cada apartado de la tabla, como se muestra en la imagen 18 en donde la tabla ambiente tiene registrados los datos correspondientes como son: condición, geometría, iluminación de la vía y zonas de obras que son de gran ayuda para saber qué factores influyeron en el accidente.

```
cqlsh:accidentesviales> INSERT INTO ambiente (condicion, geometria, iluminacionvia, zonasobras) VALUES ('humeda', 'curva', 'si', 'no');
cqlsh:accidentesviales> SELECT * FROM ambiente;

condicion | geometria | iluminacionvia | zonasobras
-----+-----+-----+-----
humeda | curva | si | no
(1 rows)
cqlsh:accidentesviales>
```

Figura 18. Datos tabla ambiente. Fuente: elaboración propia.

Creación de tabla Físicas

La tabla física (figura 19) nos muestra los apartados de jurisdicción, tipo de carretera, tipo de construcción, y vía. En ella se agregarán los datos correspondientes a cada apartado para tener idea de alguna causa o consecuencia que pueda ocasionar un accidente vial.

```
cqlsh:accidentesViales> CREATE TABLE fisicas (jurisdiccion varchar, tipoCarretera varchar, tipoConstruccion varchar, via varchar, PRIMARY KEY (jurisdiccion));
cqlsh:accidentesViales> SELECT * FROM fisicas;

jurisdiccion | tipocarretera | tipoconstruccion | via
-----+-----+-----+-----
(0 rows)
```

Figura 19. Tabla Físicas en la base de datos “Accidentes Viales”

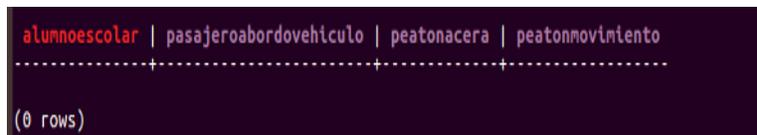
Fuente: Elaboración propia.

Creación de tabla Condición Víctimas

La tabla condición víctimas nos permitirá saber el tipo de pasajero que iba a bordo, es por eso que es importante tener la información de esta tabla.

Consulta:

```
Cqlsh:accidentesviales> CREATE TABLE condicionVictimas (alumnoEscolar
varchar, pasajeroAbordoVehiculo varchar, peatonAcera varchar,
peatonMovimiento varchar, PRIMARY KEY (alumnoEscolar));
Cqlsh:accidentesViales> SELECT * FROM condicionVictimas;
```



alumnoescolar	pasajeroabordovehiculo	peatonacera	peatonmovimiento
(0 rows)			

Figura 20. Creación de tabla Condición Víctimas. Fuente: Elaboración propia.

En la figura 20 se muestran creada la tabla **Condición Víctimas** para obtener mejor registro de datos.

Creación de tabla Detalle Conductores

En esta tabla se maneja información acerca del conductor del vehículo, por ejemplo, su nombre, si va alcoholizado, drogado, o llevaba las medidas de seguridad como lo son el cinturón o en el caso de las motocicletas el casco.

Consulta:

```
Cqlsh:accidentesviales> CREATE TABLE DetalleConductores (alcohol varchar,
casco varchar, cinturón varchar, drogas varchar, edad int, licencia varchar,
nombreConductor varchar, PRIMARY KEY (nombreConductor));
Cqlsh:accidentes viales> SELECT * FROM DetallesConductores;
```

```

nombreconductor | alcohol | casco | cinturón | drogas | edad | licencia
-----+-----+-----+-----+-----+-----+-----
(0 rows)

```

Figura 21. Creación tabla Detalles Conductores. Fuente: Elaboración propia.

Creación de tabla Detalles Vehículos

En la siguiente tabla se muestra la realización de la tabla detalles vehículos en donde se podrá registrar el tipo de vehículo que sufrió el accidente, el año, la descripción de los daños, largo de huellas y el modelo. Información importante para registrar un accidente.

```

cqlsh:accidentesviales> CREATE TABLE DetallesVehiculos (largohuellas varchar, año int, descripciondatossufridos varchar, modelo varchar, t
tipoVehiculo varchar, PRIMARY KEY (tipoVehiculo));
cqlsh:accidentesviales> SELECT * FROM DetallesVehiculos;

tipovehiculo | año | descripciondatossufridos | largohuellas | modelo
-----+-----+-----+-----+-----
(0 rows)

```

Figura 22. Creación tabla Detalles Vehículos. Fuente: Elaboración propia.

Creación de tabla Circunstancias

En esta tabla se guardará información relevante para averiguar más acerca de los accidentes viales, como ejemplo, cuantos vehículos hay involucrados, el día, la hora, lugar, placas, nombre del policía que asista al caso, entre otros apartados como se muestra en la figura 23.

```

cqlsh:accidentesviales> CREATE TABLE circunstancias (año int, calles varchar, cantidadvehiculos int, día int, gradoSeveridad varchar, hora time, lugar varchar, mes var
char, nombrePolicia varchar, placa varchar, puntoKilometrico varchar, PRIMARY KEY (nombrePolicia));
cqlsh:accidentesviales> SELECT * FROM circunstancias;

nombrepolicia | año | calles | cantidadvehiculos | día | gradoseveridad | hora | lugar | mes | placa | puntokilometrico
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----
(0 rows)

```

Figura 23. Creación tabla Circunstancias. Fuente: Elaboración propia.

Creación de tabla Detalles Lesiones

Esta tabla se llenará con información de las personas involucradas en el accidente vial, como el nombre de las víctimas, cuantas víctimas hay, sexo, tipo de lesión en caso de resultar heridos y la severidad de la lesión. (Figura 24).

```
cqlsh:accidentesviales> CREATE TABLE DetallesLesiones (nombreVictima varchar, numero_victimas int, severidadLesiones varchar, sexo varchar, tipoLesion varchar, PRIMARY KEY (nombreVictima));
```

Figura 24. Creación tabla Detalles Lesiones. Elaboración: Fuente propia.

9. RESULTADOS

El uso del servidor es de gran importancia en este proyecto, ya que es la base del mismo, en él se llevó a cabo la instalación de los servicios que ayudaron a las aplicaciones que complementan este proyecto como lo es la aplicación móvil y sistema web.

Una vez realizada la Base de Datos NoSQL “Accidentes Viales” se llevó a cabo la inserción de datos en cada tabla, para mostrar que estuviera funcionando correctamente y que podía realizarse la inserción de datos.

Es importante mostrar la creación de todas las tablas en la Base de Datos para comprobar que estén todas, como se muestra a continuación en la figura 25.

```
(1 rows)
cqlsh:accidentesviales> DESCRIBE TABLES;

circunstancias      detalleslesiones    detallesconductores  ambiente
condicionvictimas  detallesvehiculos   fisicas
cqlsh:accidentesviales> █
```

Figura 25. Tablas existentes en la base de datos. Fuente: Elaboración propia.

Cuando se concluyó con la creación de tablas, se continuó agregando los datos en cada campo que se mencionaron para poder obtener la información requerida.

Una vez creadas todas las tablas se procedió a la inserción de datos en cada una de ellas para hacer manejo de la información requerida como se muestra a continuación en las siguientes imágenes.

Tabla Ambiente

```
cqlsh:accidentesviales> INSERT INTO ambiente (condicion, geometria, iluminacionvia, zonasobras) VALUES ('humeda', 'curva', 'si', 'no');
cqlsh:accidentesviales> SELECT * FROM ambiente;
```

condicion	geometria	iluminacionvia	zonasobras
humeda	curva	si	no

(1 rows)
cqlsh:accidentesviales>

Figura 26. Datos tabla ambiente. Fuente: Elaboración propia.

En esta tabla se llevó a cabo el registro de los datos que ayudaron a saber el tipo de ambiente que hay en el camino, calle, carretera, etc.

Tabla Físicas

```
cqlsh:accidentesviales> SELECT * FROM fisicas;
```

jurisdiccion	tipocarretera	tipoconstruccion	via
federal	asfalto	publica	libre

(1 rows)

Figura 27. Datos tabla Físicas. Fuente: Elaboración propia.

Para la tabla **físicas** se tomó en cuenta el tipo de lugar que fue donde se reportó el accidente, esta información es de vital importancia porque ayudo a llevar un registro de donde se ocasionan más los accidentes.

Tabla Condición Víctimas

```
cqlsh:accidentesviales> INSERT INTO condicionvictimas (alumnoescolar, pasajeroabordovehiculo, peatonacera, peatonmovimiento) VALUES ('si', 'no', 'no', 'si');
cqlsh:accidentesviales> SELECT * FROM condicionvictimas;
```

alumnoescolar	pasajeroabordovehiculo	peatonacera	peatonmovimiento
si	no	no	si

(1 rows)
cqlsh:accidentesviales>

Figura 28. Datos Condición Víctimas. Fuente: Elaboración propia.

Para la tabla de **condición víctimas** se tomaron en cuenta datos acerca de las personas que fueron afectadas, para llevar un mejor registro de quienes son los más propensos a sufrir accidentes viales.

Tabla Detalle Conductores

```
cqlsh:accidentesviales> INSERT INTO detallesconductores (nombreConductor, alcohol, casco, cinturón, drogas, edad, licencia) VALUES ('Santiago Romero Sanchez', 'no', 'no', 'si', 'no', 25, 'si');
cqlsh:accidentesviales> SELECT * FROM detallesconductores;
```

nombreconductor	alcohol	casco	cinturon	drogas	edad	licencia
Santiago Romero Sanchez	no	no	si	no	25	si

(1 rows)
cqlsh:accidentesviales>

Figura 29. Datos de tabla Detalles Conductores. Fuente: Elaboración propia.

En la tabla Detalles Conductores se tomaron en cuenta registros acerca del estado del conductor que son muy importantes para llevar mejor control de cómo se ocasionan los accidentes viales, ya que en ocasiones los conductores van alcoholizados o no toman las medidas de seguridad necesarias para evitar accidentes viales.

Tabla Detalles Lesiones

```
cqlsh:accidentesviales> INSERT INTO detalleslesiones (nombreVictima, numero_victimas, severidadLesiones, sexo, tipoLesion) VALUES ('Daniel Salgado Morales', 2, 'grave', 'masculino', 'fractura de hueso');
cqlsh:accidentesviales> SELECT * FROM detalleslesiones;
```

nombrevictima	numero_victimas	severidadlesiones	sexo	tipolesion
Daniel Salgado Morales	2	grave	masculino	fractura de hueso

(1 rows)
cqlsh:accidentesviales>

Figura 30. Datos de tabla Detalles Lesiones. Fuente: Elaboración propia

En los accidentes es muy común sufrir lesiones ya sean ligeras o graves, esta información se registró también en una tabla de la base de datos para llevar un registro de que tanto impactan en la salud cada accidente vial como se puede observar en la figura 30.

Tabla Detalles Vehículos

```
(0 rows)
cqlsh:accidentesviales> INSERT INTO DetallesVehiculos (tipovehiculo, anio, descripciondatossufridos, largohuellas, modelo) VALUES ('automovil', 2016, 'leves laterales',
'5m', 'tsuru');
cqlsh:accidentesviales> SELECT * FROM DetallesVehiculos;

tipovehiculo | anio | descripciondatossufridos | largohuellas | modelo
-----
automovil | 2016 | leves laterales | 5m | tsuru
(1 rows)
cqlsh:accidentesviales> |
```

Figura 31. Datos de tabla Detalles Vehículos. Fuente: Elaboración propia.

En cada reporte que se maneja por accidente vial se debe tomar en cuenta todos los factores involucrados, por ese motivo se realizó una tabla para llevar el control del tipo de vehículo que fue afectado: año, los daños sufridos, largo de huellas y el modelo, con esa información se pudo comparar que tipo de automóviles sufren más accidentes.

Tabla Circunstancias

Consulta:

```
Cqlsh:accidentesViales> INSERT INTO circunstancias (nombrePolicia, anio, calles,
cantidadvehiculos, dia, gradoseveridad,hora, lugar, mes, placa, puntokilometrico) VALUES
('Pablo Hernández Aguilar', 2016, 'carretera federal 18', 2, 1, 'leve', '14:30:33', 'monterrey',
'diciembre, 'WU-88-66', '31km');
```

```
nombrepolicia | anio | calles | cantidadvehiculos | dia | gradoseveridad | hora | lugar | mes | placa | puntokilometrico
-----
Pablo Hernandez Aguilar | 2016 | carretera Federal 18 | 2 | 1 | leve | 14:30:33.000000000 | monterrey | diciembre | WU-80-66 | 31km
```

Figura 32. Datos de tabla Circunstancia. Fuente: Elaboración propia.

Explicar las circunstancias es de gran relevancia para un informe de accidentes viales, por eso se llevó a cabo el registro de los datos en esa tabla para saber el nombre del policía que llevo a cabo el reporte, el año, calles, cantidad de vehículos, día, gravedad, hora, lugar, mes, placa y el punto de kilometraje que

llevaba el automóvil, pues esos datos se archivan y se lleva un control de cuantos accidentes hubo en ese año, en que calle fue y tomando en cuenta los demás datos se puede tener un informe mejor detallado para comparar datos en la plataforma. (Figura 32).

10. CONCLUSIONES

Como resultado, el proyecto denominado Plataforma de Observación Vial, llevará a cabo el registro de los accidentes viales y sus principales causas, de tal forma que ayudará a evitar incrementar el número de accidentes una vez que los datos sean analizados.

Su principal función es llevar un registro de los accidentes viales que hay, uniendo información de una aplicación móvil y una aplicación web ya existentes para tomar en cuenta las consecuencias que provocan un accidente, teniendo estos datos se crearan tablas de información que junten lo antes mencionado y establezca la conexión entre dichas aplicaciones.

Es importante mencionar que el proyecto permite la escalabilidad de la inserción de información referente a accidentes viales. Se establece la base de conocimientos a través de una base de datos no estructurada que permite la visualización y exploración de la información de la base de datos del INEGI. No se generó una aplicación de consultas ya que el propósito de esta tesis era demostrar la carga de fuentes heterogéneas referentes a accidentes viales.

11. RECOMENDACIONES

Una recomendación para este proyecto es realizar una interfaz gráfica para que sea más rápido subir bases de datos y comparar la información, así como buscar información por medio de filtros.

También es importante mantener en constante vigilancia las versiones de software instalados en el servidor, para garantizar el mejor funcionamiento de las configuraciones realizadas.

12. FUENTES DE INFORMACIÓN

- Ortiz Rivera, J. A., & Buelvas Dumar, W. J. (2018). Desarrollo e implementación de una aplicación móvil para el registro de accidentes de tránsito en el municipio de Montelíbano–Córdoba.
- Lakshman, A. y Malik, P. (2010). Cassandra: un sistema de almacenamiento estructurado descentralizado. *Revisión de sistemas operativos ACM SIGOPS*, 44 (2), 35-40.
- Lijarcio Cárcel, J., Martí-Belda Bertolín, A., Boso Segui, P., López Osma, C., Escamilla Robla, C., & Puchades Villarreal, R. (2016, June). Contribución de las aplicaciones móviles a la Seguridad vial. Drive Smart como ejemplo de aplicación. In *XII Congreso de ingeniería del transporte. 7, 8 y 9 de junio, Valencia (España)* (pp. 1528-1535). Editorial Universitat Politècnica de València.
- Marchionni, E. A. (2011). *Administrador de servidores* (Vol. 210). USERSHOP.
- Silberschatz, A., Korth, H. F., Sudarshan, S., Pérez, F. S., Santiago, A. I., & Sánchez, A. V. (2002). Fundamentos de bases de datos.
- Santillán, L. A. C., Ginestà, M. G., & Mora, Ó. P. (2014). Bases de datos en MySQL. *Universitat oberta de Catalunya*.
- Romero, A. C., Sanabria, J. S. G., & Cuervo, M. C. (2012). Utilidad y funcionamiento de las bases de datos NoSQL. *Facultad de Ingeniería*, 21(33), 21-32.
- Martín, A. E., Chávez, S. B., Rodríguez, N. R., Valenzuela, A., & Murazzo, M. A. (2013, June). Bases de datos NoSQL en cloud computing. In *XV Workshop de Investigadores en Ciencias de la Computación*.
- Montesdeoca Chacha, A. M. (2012). *Estudio Estadístico de Tipos de Servidores para la Comunicación con Plataformas de Rastreo Móvil en la Implementación de una Puerta de Enlace para Equipos Enfora* (Bachelor's thesis).

Vogel, L. (2010). Base de datos de Android SQLite y tutorial de contenido del proveedor. *Tutoriales de programación Java, Eclipse, Android y Web*, 8.

Labrador, R. M. G. (2014). 14127 ADMINISTRACIÓN DE SERVIDORES LINUX (UBUNTU/FEDORA/CENTOS).

Cuaresma, S. B. Manual básico Ubuntu GNU/Linux. *Obtenido de <http://www.uls.edu.sv/pdf/ubuntu>. Pdf.*

NGuerrero (2020). ¿Qué es Firebase? (plataforma de desarrollo de Google). Obtenido de: <https://cutt.ly/vmGE0MB>

Schiaffarino A (2020). Modelo cliente servidor. Infranetworking. Obtenido de: <https://cutt.ly/3mGR5EV>